ALTOS
COMPUTER SYSTEMS

Computer System ACS8600 User Manual

Revision B March 22, 1982

Part Number: 690-11392

FEDERAL COMMUNICATIONS COMMISSION NOTICE

W A R N I N G

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 on expense, will be required to take whatever measures may be required to correct the interference.

P R E L I M I N A R Y

ACS 8600 COMPUTER SYSTEM

USER MANUAL

REVISION B

MARCH 22, 1982

DISCLAIMER

THIS IS A PRELIMINARY PUBLICATION OF THE ALTOS ACS 8600 COMPUTER SYSTEM DOCUMENTATION AND IS PRESENTED FOR INFORMATION PURPOSES ONLY; IT WILL BE CHANGED WITHOUT NOTICE.

THE INFORMATION CONTAINED HEREIN REPRESENTS THE MOST CURRENT INFORMATION AVAILABLE AT THE TIME OF PRINTING, HOWEVER, THE INFORMATION HAS NOT BEEN THOROUGHLY REVIEWED NOR HAS IT BEEN APPROVED IN ITS PRESENT FORM. ALTOS COMPUTER SYSTEMS ASSUMES NO LIABILITY WHATEVER RELATED TO USE OF INFORMATION PRESENTED IN THIS PUBLICATION.

Manual Part Number: 690-11392

CONTENTS

SECTION 1. INTRODUCTION

CONTENTS

CONTENTS

NOTE

The ACS System Specification has been completely reprinted as Section 7 of this manual. The Specification contains its own Table of Contents, List of Figures and List of Tables (see Section 7)

APPENDICES

FIGURES

TABLES

ALTOS ACS 8600 COMPUTER SYSTEM

USER MANUAL

SECTION 1.

INTRODUCTION

GENERAL INFORMATION

This section describes the structure of the basic User Manual and its supplements (see Figure 1-1), instructions for using the basic manual, and a general description of the system.

ACS 8600 COMPUTER SYSTEM PUBLICATION ORGANIZATION

Altos Computer Systems has designed a publication structure that differs from most user manuals. The purpose of this structure is fourfold:

(1) To provide accurate system information to the user as quickly as possible.

(2) To permit updating of more dynamic areas of information as design improvements and system enhancements are achieved.

(3) To isolate and identify information blocks so that referencing time is minimized.

(4) To present information in a general-to-specific manner, thus meeting the needs of beginning users as well as more experienced users.

The publications you receive with the computer are divided into three groups (see Figure 1-1):

1. Column 1 of Figure 1-1 illustrates the forms that accompany the manual, and the Basic User Manual sections and appendices which contain relatively unchanging information pertaining to all models in the series.

2. Column 2 illustrates Altos Supplements 3, 4, 5 and 6, which contain more dynamic information pertaining to executing diagnostic programs, installing operating systems, using Altos utilitiy programs, and installing upgrade and add kits. The supplements are designed to be inserted into the User Manual after the shipping container is unpacked.

3. Column 3 illustrates Supporting Manuals, which include pre-printed manuals produced by developers of related software products that operate on the ACS 8600 Computer System. This group may also include more informal technical notes related to such systems, thus insuring that you have the most current information possible.

Forms Accompanying this Manual

The following forms accompany this manual. They are valuable records and are provided separately so that they may be taken from the manual and stored in a secure place where they will be available if repairs happen to be needed.

COVER LETTER
(info to come)

CHECKOFF SHEET
(info to come)

WARRANTY CARD
(info to come)

QUALITY CONTROL REPORTS
(Info to come)

System Checkoff Sheet
(info to come)

System Checkout Sheet
(info to come)

Burn-In Summary Sheet
(info to come)

Quality Control Inspection Sheet
(info to come)

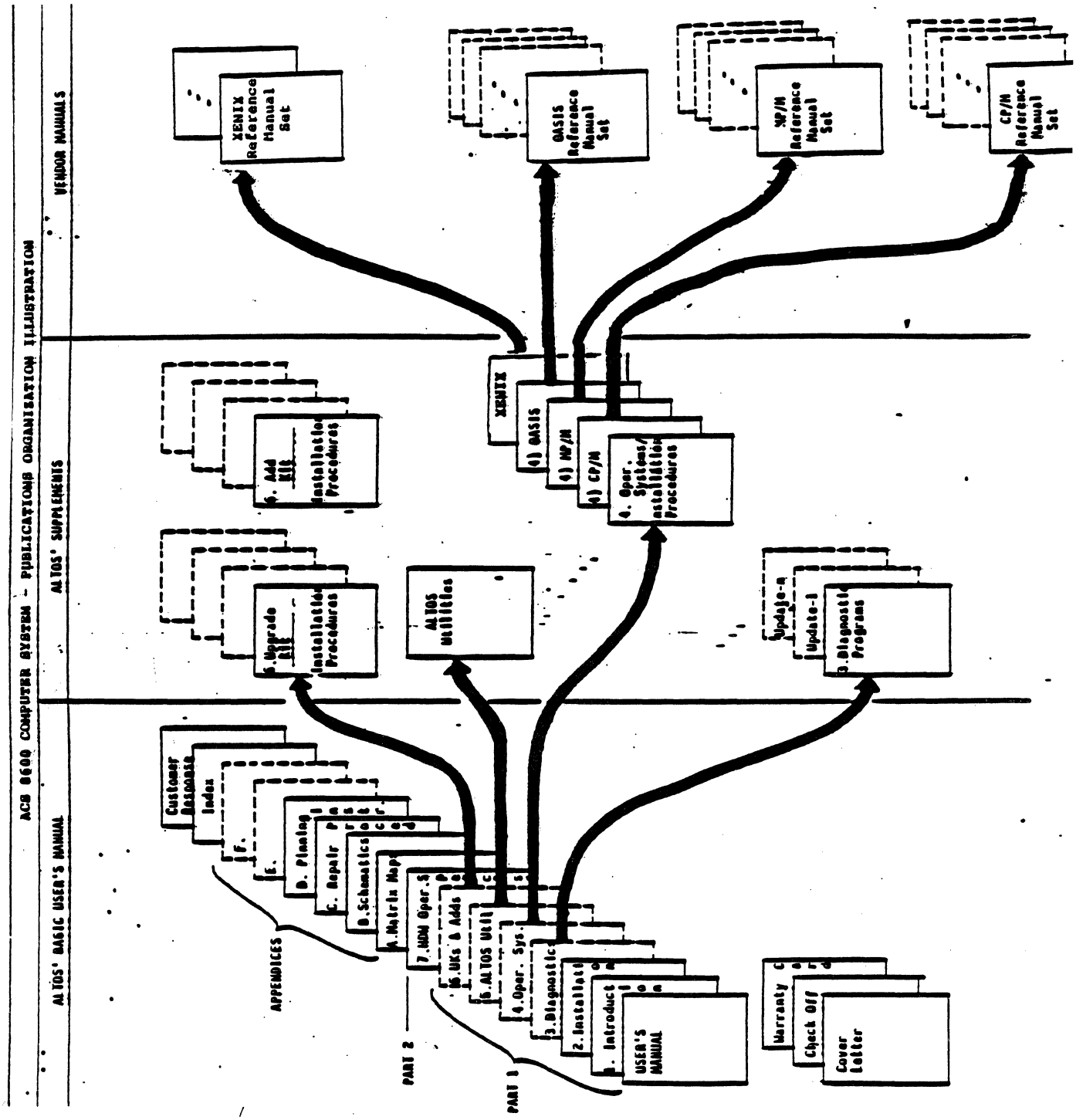


Figure 1-1. ACS 8600 Computer System - Publications Organization Illustration

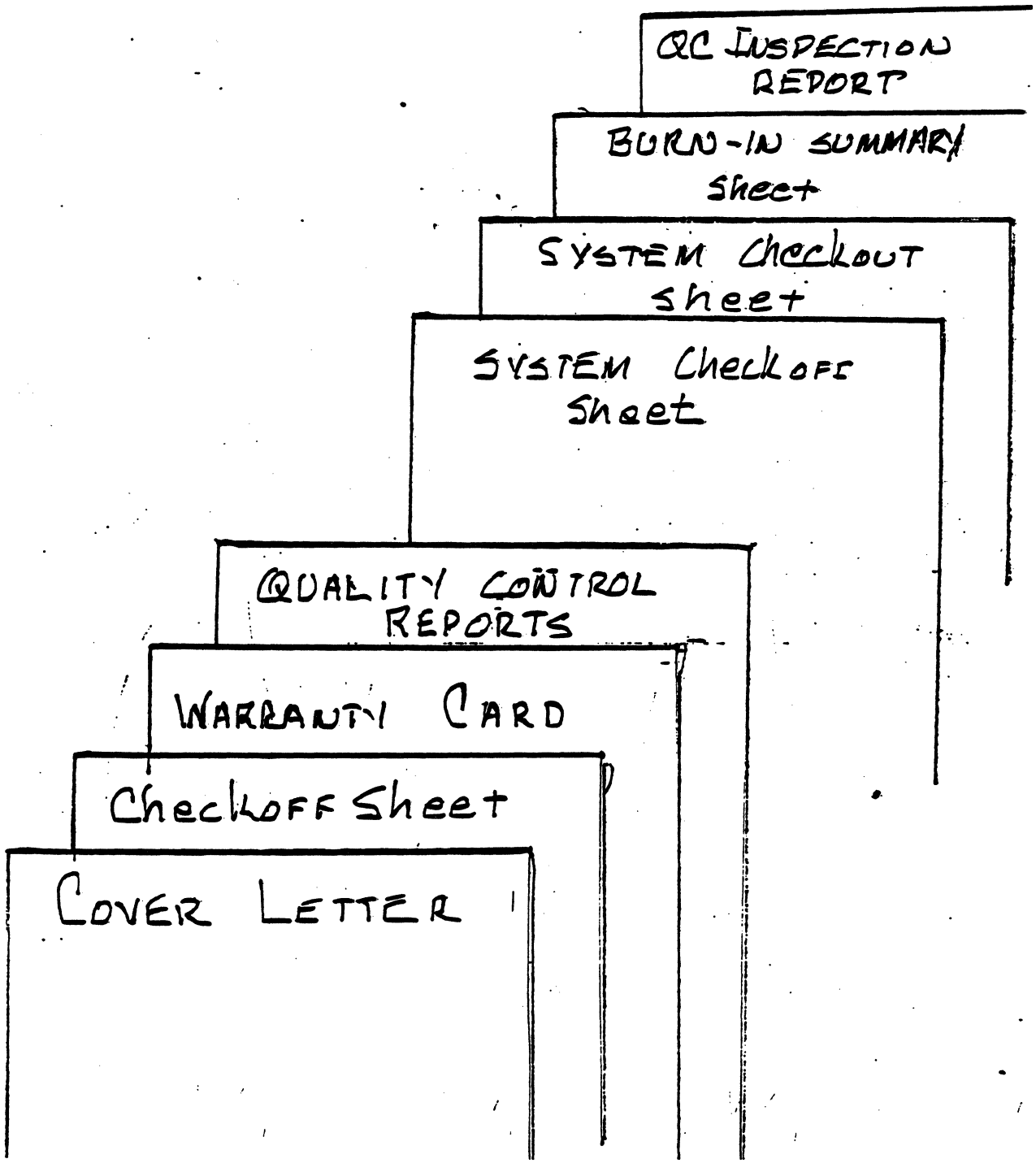


Figure 1-2. Forms Accompanying This Manual

Basic User Manual Description

SCOPE AND PURPOSE

This basic User Manual presents sufficient information so that any user, whether familiar with microcomputers or not, can become familiar with the ACS 8600 Computer System, its component parts, features, capabilities, architecture, and system operation. The manual either presents the information in itself, or provides easily-found references to other publications in which such information can be found.

The manual instructs the user in all required operations from unpacking the shipping container through installing the Operating System and preparing the system to perform the jobs for which the system was purchased.

ORGANIZATION

This User Manual is divided into seven sections, seven appendices, a glossary, an index, and a Customer Response Sheet.

This section, INTRODUCTION introduces you to the basic User Manual itself, and instructs you in its use. It also introduces the manual supplements and other related publications, and introduces the ACS 8600 Computer System, including its features, capabilities, configurations, and component parts (both hardware and software).

Section 2. SETTING UP THE COMPUTER SYSTEM provides instructions for removing the computer and its components from the shipping container, identifying the components, storing related important papers, and physically assembling the system by following step-by-step instructions. This section instructs you in preparing the system for executing the Altos Diagnostic Executive (ADX) Program.

Section 3. THE ALTOS DIAGNOSTIC EXECUTIVE (ADX) PROGRAM introduces Supplement 3. EXECUTING THE ALTOS DIAGNOSTIC EXECUTIVE (ADX) PROGRAM which, in turn, provides step-by-step instructions for executing diagnostic programs under supervision of the diagnostic executive to verify operation of the system and its components. ADX prepares the computer system for installation of the selected Operating System.

Section 4. OPERATING SYSTEMS provides information related to operating systems in general, and introduces one or more versions of Supplement 4, which, in turn, provide specific, step-by-step installation procedures for particular operating systems. One or more of the Operating System Supplements will be provided with the system. Each Supplement 4. contains all the information necessary to install the selected operating system after the diagnostic programs have been successfully executed.

Only the Operating System Supplement related to the Operating System(s) purchased (CP/M-86, MP/M-86, OASIS-16, or XENIX) is provided with the system.

Section 4. also introduces you to the supporting manuals and other publications related to the subject operating system.

Section 5 . ALTOS UTILITY PROGRAMS provides general information related to utility programs and introduces you to SUPPLEMENT 5. ALTOS UTILITY PROGRAMS - USER INSTRUCTIONS which describes Altos Utility Programs available for use with the system, and instructs you in their use.

Section 6. ADD KITS AND UPGRADE KITS provides general information regarding kits available for increasing system capabilities. This section introduces Supplement 6. UPGRADE KITS AND ADD KITS - INSTALLATION PROCEDURES. Purchasers of Add Kits and Upgrade Kits receive simply written installation procedures written specifically for those kits.

Section 7. HARDWARE OPERATING SPECIFICATIONS specifically and extensively describes system architecture and functional operation, including port assignments, controllers, interfaces, buffer use, and memory management.

Appendix A. ALTOS WARRANTY DESCRIPTION AND SHIPPING DAMAGE REPAIR PROCEDURES describe the Altos Warranty and provide instructions for filing the End-User Registration Card and the Dealer Registration Card. Instructions are also provided for inspecting for shipping damage, and obtaining repairs and/or service if such damage occurs. This appendix also briefly describes Altos factory repair and shipping procedures.

Appendix B. TROUBLE SHOOTING PROCEDURES provides basic instructions for isolating and correcting troublesome conditions and determining whether or not a true malfunction actually exists.

Appendix C. COMMON CRT TERMINAL AND PRINTER INTERFACE CONFIGURATIONS provides instructions for connecting and setting most-commonly-used terminals and printers.

Appendix D. INSTRUCTIONS FOR PINNING PRINTED CIRCUIT BOARDS (PCB) provides instructions for setting pinning blocks on the various circuit boards to achieve the operating configuration desired.

Appendix F. MATRIX MAPS includes all matrix maps in the form current at time the current revision of the manual is printed. (Early revisions may not have this appendix.)

Appendix G. SCHEMATIC DIAGRAMS includes all schematic diagrams in the form current at the time the current revision of the manual is printed. (Early revisions may not have this appendix.)

GLOSSARY contains definitions of terms used in the manual that may be unfamiliar to the user. (Early revisions may not have a Glossary.)

INDEX is designed to assist you in referencing desired information. (Early revisions may not have an Index.)

CUSTOMER RESPONSE FORM is a pre-printed form provided to facilitate forwarding of comments and requests to Altos Computer Systems.

INSTRUCTIONS FOR USING THIS MANUAL

This manual (and its supplements) guides you through the complete installation process for the computer system from unpacking the shipping container through calling in and executing application programs.

To use this manual most effectively, perform the following steps:

1. Read Section 1 to become familiar with the publications structure and the system design.

2. Refer to Section 2 and follow the instructions for unpacking the system and physically connecting the system components.

3. Assemble the publications and verify that all are present. When the shipping container is unpacked, insert the supplements into their proper place in the manual.

3. Refer to Section 3 for general information regarding Altos Diagnostic Executive (ADX) program.

   Follow the instructions in Supplement 3 to configure the system, execute the diagnostic programs, and initialize the system. The diagnostic programs verify correct operation of the system and its components. Supplement 3 also instructs you in making backup copies of the diagnostic program and operating system diskettes as required.

5. Refer to Section 4 for information regarding operating systems in general and to the applicable Supplement 4.

   Refer to Supplement 4 for information regarding the specific operating system to be used. This section tells you how to install that operating system and how to load and execute application programs.

   One of four Operating Systems will be installed. Each of these systems has its own loading procedures and operating characteristics. Some operating systems also

have their own set of diagnostic programs.

6. Refer to Section 5. for information regarding available Altos Utility Programs developed for the system.

   Supplement 5 provides specific information regarding individual utility programs.

7. Refer to Section 6. for information regarding increasing system capabilities through the use of Add Kits and Upgrade Kits.

   Supplement 6 provides specific procedures for installing individual Upgrade Kits or Add Kits.

8. Refer to Section 7 and the appendices for more specific information regarding the specifics of system operation.

The set up procedure (described in Section 2, and diagnostic procedures (described in Section 3) are generally similar for all ACS 8600 computer models. They will vary somewhat, depending upon the configuration of the model and the peripherals included with the system.

Manual Supplement Descriptions

A description of SUPPLEMENT 3. EXECUTING THE DIGNOSTIC EXECUTIVE (ADX) PROGRAM is provided in Section 3.

A description of four Operating System Supplements (Control Program Monitor - CP/M-86, Multi-Program Monitor - MP/M-86, XENIX, and OASIS-16 Operating Systems is provided in Section 4. OPERATING SYSTEMS. This section also provides a description of supporting publications.

A description of SUPPLEMENT 5. ALTOS UTILITY PROGRAMS - USER INSTRUCTIONS is provided in Section 5.

A description of SUPPLEMENT 6. UPGRADE KITS AND ADD KITS - INSTALLATION PROCEDURES is provided in Section 6.

GENERAL SYSTEM DESCRIPTION

The Altos ACS 8600 Computer System is a multi-user system designed to fit many commercial, technical, industrial, scientific, and educational applications, as well as many personal, home-computer applications.

This system is designed to perform larger, more complex tasks for more users than other microprocessors. Larger word size allows the use of more powerful instructions and direct addressing of more memory. The system requires fewer instructions and fewer computer cycles to accomplish the same job. It can handle complex mathematical problems, larger data

base searches and other demanding applications in far less time.

More powerful processing capability, larger memory, and special memory management techniques extend multi-user capabilities and allows multi-tasking without long waits for service.

Although the ACS 8600 Computer System is particularly designed to meet multi-user and multi-tasking requirements, it may also be configured as a powerful single-user system. The enhanced computing power of the system is achieved not only through the use of the Intel 8086 sixteen-bit microprocessor chip technology, but also through the use of a separate Intel 8089 Direct Memory Addressing (DMA) I/O processsor, and a unique hardware memory management scheme that greatly increases efficient use of main memory.

The Central Processing Unit (CPU) is the proven 8086 16-bit HMOS microprocessor. It operates at a fast 5 Mhz and can address 1-Mbyte of memory. The CPU uses assembly language compatible with the 8080 and 8085 processors, and performs 8- and 16-bit signed and unsigned arithmetic functions in binary or decimal, including multiplication and division operations.

The ACS 8600 family uses more than one processor. While the 8086 serves as the master CPU, other processors are dedicated to special tasks. For example, a second processor, one 8089 DMA, handles direct memory accessing for I/O processing. The optional mathematical processor (8087 Floating Point Processor) handles mathematical functions. Although each processor is optimized for its own special task, they all work together to share the workload for faster execution and decreased response time.

While many mid-sized minicomputer memories lack Error Correction Code (ECC), the ACS 8600 improves operational reliability with its own ECC. The memory has full error detection for single-bit and double-bit errors, and automatic correction of single-bit errors, thus reducing system errors and increasing the already-high system dependability.

The design of the ACS 8600 family of computers includes 500,000 bytes (500-Kbytes) of random access main memory. Main memory is expandable to one- million bytes (1-Mbyte) with full error detection and correction. It also includes I/O controllers for supporting a variety of peripherals such as an additional hard disk, CRT terminals, printers, magnetic tape units and floppy disk units. Full communications support, including asynchronous, bisynchronous and networking capabilities are also provided.

ACS 8600 proprietary memory management system assigns a logical-to-physical address translation and various attributes to each 4-Kbyte block of memory so that the non-contiguous physical memory can be assigned to any given task. This capability reduces the need for swapping users or tasks out to disk. The result is

that data base sorting, and other tasks requiring large memory are performed faster. The system also provides write and access protection to increase reliability and performance.

Altos Computer Systems supports and distributes four of the most popular operating systems, Control Program Monitor (CP/M-86), Multiple Program Monitor (MP/M-86), OASIS-16, and XENIX (UNIX, Version 7). More than 300,000 current users of CP/M, MP/M and OASIS can easily upgrade their systems to the ACS 8600 Computer System knowing that their application software is compatible with the ACS 8600 computer.

BASIC, COBOL, PASCAL, and FORTRAN languages are also distributed and supported by Altos Computer Systems.

Although Altos does not generally distribute application program packages, it does provide a list of such compatible program packages, developed by leading software houses, on request.

System Functional Description

The ACS 8600 Computer System is an advanced-design, 16-bit, general-purpose computer manufactured by Altos Computer Systems. System design supports up to 1-Mbyte of Random Access Memory (RAM) with full error detection and correction, and a wide variety of mass storage peripherals. Unique memory management hardware increases efficiency of main memory use. The system can be easily extended to meet the requirements of a wide variety of applications.

The ACS 8600 System is based on the Intel 8086 microprocessor family. While the system is primarily designed to meet multi-user and multi-tasking requirements it may easily be configured for large single-user applications. The following list summarizes the main features and capabilities:

PRIMARY SYSTEM ENCLOSURE - PHYSICAL DETAILS

Size

19 inches wide, 23 inches deep, 7 inches high

Weight

Free-standing weight - 60 pounds
Shipping weight - 80 pounds

Front Panel Controls

AC ON/OFF Switch (upper right hand corner)
RESET Switch (to left of ON/OFF switch)

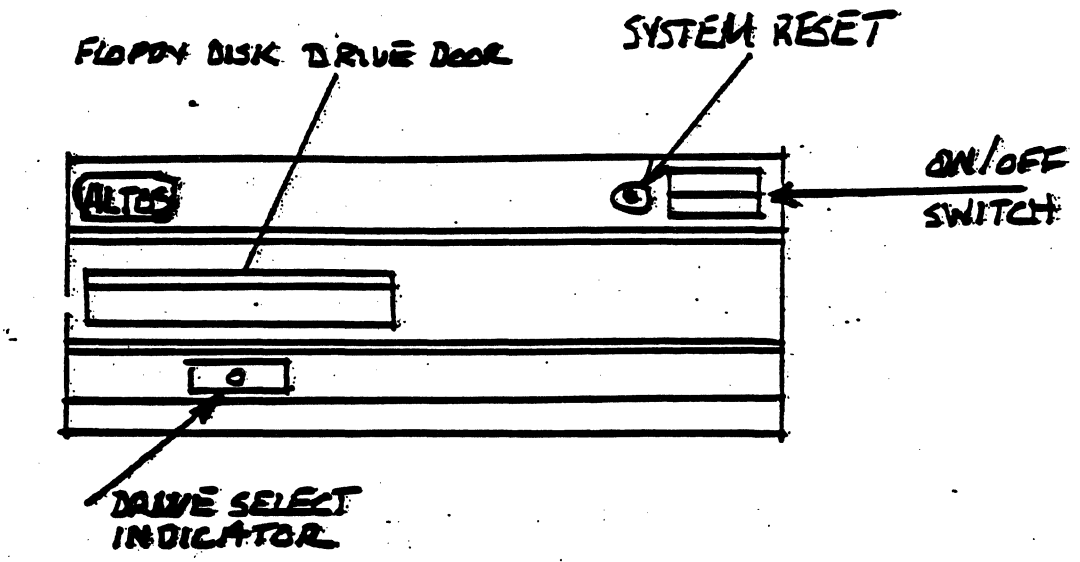


Figure 1-3. ACS 8600 Computer System - Hard Disk/Floppy Disk Version

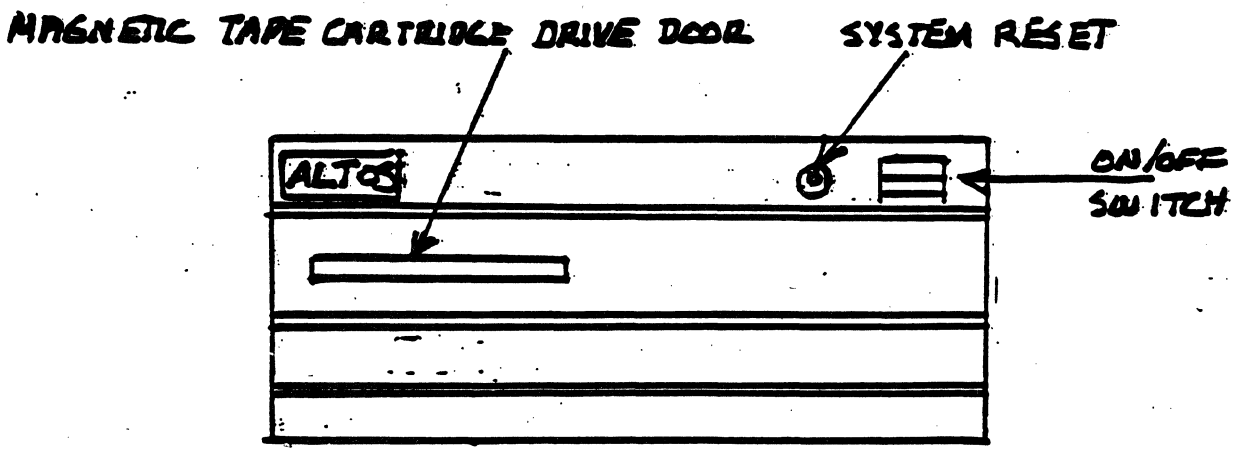


Figure 1-4. ACS 8600 Computer System - Hard Disk/Magnetic Tape Unit Version

Rear Panel Connectors (see Figure 2-3)

AC Power Receptacle
Fuse Holder
Parallel Port Interface Connector (DB 37)
Hard Disk Expansion Connectors (20 pin and 50 pin)
Console Terminal RS-232-C Connector (DB 25)
Multibus Expansion Interface Connectors (72 pins)
Magnetic Tape Unit Expansion Connector (50 pin)
Serial Port Connectors (1-through-8)

(Refer to Section 7 for specifications.)

SECONDARY SYSTEM ENCLOSURE - PHYSICAL DETAILS

Size

19 inches wide, 23 inches deep, 7 inches high

Weight

Free-standing weight - 55 pounds
Shipping weight - 75 pounds

Front Panel Controls

AC ON/OFF Switch (upper right hand corner)
RESET Switch (to left of ON/OFF switch)

Rear Panel Connectors (see Figure 2-3)

AC Power Receptacle
Fuse Holder
Parallel Port Interface Connector (DB 37)
Hard Disk Expansion Connectors (20 pin and 50 pin)
Console Terminal RS-232-C Connector (DB 25)
Multibus Expansion Interface Connectors (72 pins)
Magnetic Tape Unit Expansion Connector (50 pin)
Serial Port Connectors (1-through-8)

(Refer to Section 7 for specifications.)

FEATURES AND CAPABILITIES

Sixteen-bit CPU section consisting of an 8086 microprocessor, an 8089 Direct Memory Access (DMA) - I/O Processor and an (optional) 8087 Numeric Data Processor.

Sixteen-bit main memory, consisting of 512-Kbytes which can be expanded to one megabyte.

Error correction and detection on the main memory providing single- and double-bit error detection with single-bit correction.

Memory management on the main memory providing address translation, write protection and access control for each 4-Kbyte block of main memory.

Bootstrap and configuration control in Read Only Memory (ROM) permitting system initialization from any online mass storage device.

Flexible diskette controller capable of controlling up to four 8-inch, floppy-diskette drives with support for single- and double-density recording as well as single- and double-sided diskette recording.

Rigid disk controller for one or two 8-inch, Winchester-type, rigid, fixed-disk drives.

One programmable, parallel interface capable of controlling a high speed, parallel input line printers.

Ten serial interfaces designed to support asynchronous terminals and printers, synchronous interfaces, and local networks. (Eight of the serial ports have rear panel connectors.)

NOTE

The Magnetic Tape Unit mentioned throughout this manual is presently scheduled for release in the third quarter of 1982.

Magnetic tape controller for cartridge-type, magnetic tape transports.

Fully vectored interrupt capability.

Virtually unlimited expansion through a Multibus-compatible expansion interface.

SYSTEM CONFIGURATIONS

The ACS 8600 Computer System is packaged in a desktop enclosure which may optionally be rack mounted. This enclosure will contain a hard disk drive and either a floppy disk drive or a Magnetic Tape Unit (MTU) shown in Table 1-1

Table 1-1. ACS 8600 - Model/Storage Media Reference Table

ACS 8600 Model	Primary Media	Secondary Media
8600-10	10-Mbyte Hard Disk	500-Kbyte Floppy Disk
8600-12	20-Mbyte Hard Disk	500-Kbyte Floppy Disk
8600-14	40-Mbyte Hard Disk	500-Kbyte Floppy Disk
8600-10 MTU	10-Mbyte Hard Disk	17-Mbyte MTU
8600-12 MTU	20-Mbyte Hard Disk	17-Mbyte MTU
8600-14 MTU	40-Mbyte Hard Disk	17-Mbyte MTU

The primary enclosure contains the CPU with up to 1-Mbyte of RAM, up to eight serial channels, all peripheral controllers, power supplies and expansion interfaces, a 10-, 20-, or 40-Mbyte Winchester-type, hard-disk drive, and either an eight-inch, 500-Kbyte, single-sided, floppy-diskette drive, or a cartridge-type, 17-Mbyte, Magnetic Tape Unit (MTU)

One additional secondary enclosure containing power supplies, a second hard disk and an MTU can be added to assemble a wide variety of system configurations (see Table 1-2).

Table 1-2. Add Kit/ Storage Media Reference Table

Kit Number	Primary Media	Secondary Media
Add-10	10-Mbyte Hard Disk	None
Add-12	20-Mbyte Hard Disk	None
Add-14	40-Mbyte Hard Disk	None
Add-10 MTU	10-Mbyte Hard Disk	17-Mbyte MTU
Add-12 MTU	20-Mbyte Hard Disk	17-Mbyte MTU
Add-14 MTU	40-Mbyte Hard Disk	17-Mbyte MTU
Add RAM	(Adds 500-Kbytes of Random Access Memory)	
Add MTU-3	None	17-Mbyte MTU
Add ???	(Adds 8087 Floating Point Chip Hardware and FORTRAN Software)	

Hardware Functional Description

The ACS 8600 Computer System is capable of supporting from one to eight users. The system is designed around a single printed circuit board which contains most of the processing resources. This single board computer also contains interfaces

permitting it to be expanded to handle very large computing tasks. The system architecture is shown in Figure 1-5.

CENTRAL PROCESSING UNIT

The Central Processing Unit (CPU) of the system is based on three members of the Intel 8086 microprocessor family. The primary processor, the 8086 16-bit HMOS microprocessor, is responsible for executing nearly all code. An 8089 Dual Channel DMA I/O Processor is arranged with the 8086 in a local configuration. The primary purpose of the 8089 is to provide 16-bit-wide DMA I/O transfer capability. An optional 8087 Numeric Data Processor (NDP) can perform high-speed floating- point calculations for the 8086. The 8087 NDP co-decodes instructions fetched by the 8086 and will bus-request the 8086 as required to perform the arithmetic operations.

SYSTEM MEMORY

System Memory is implemented with 64-Kbyte dynamic RAM devices and 2732-type Erasable-Programmable- Read-Only-Memory (EPROM) devices. The system permits up to 1-Mbyte of RAM which maybe overlayed by the EPROM. RAM memory is organized as 16-bit words with six extra bits used to support error correction. RAM memory is divided into eight banks of twenty-two devices which provide 128-Kbytes of memory per bank.

Four lower banks of the memory (containing 512-Kbytes of RAM) are located on the CPU Board; four upper banks (containing an additional 512-Kbytes of RAM) are located on the optional expansion memory board. RAM memory is supported by three other blocks of logic; the memory controller, the error correction circuitry and the memory manager.

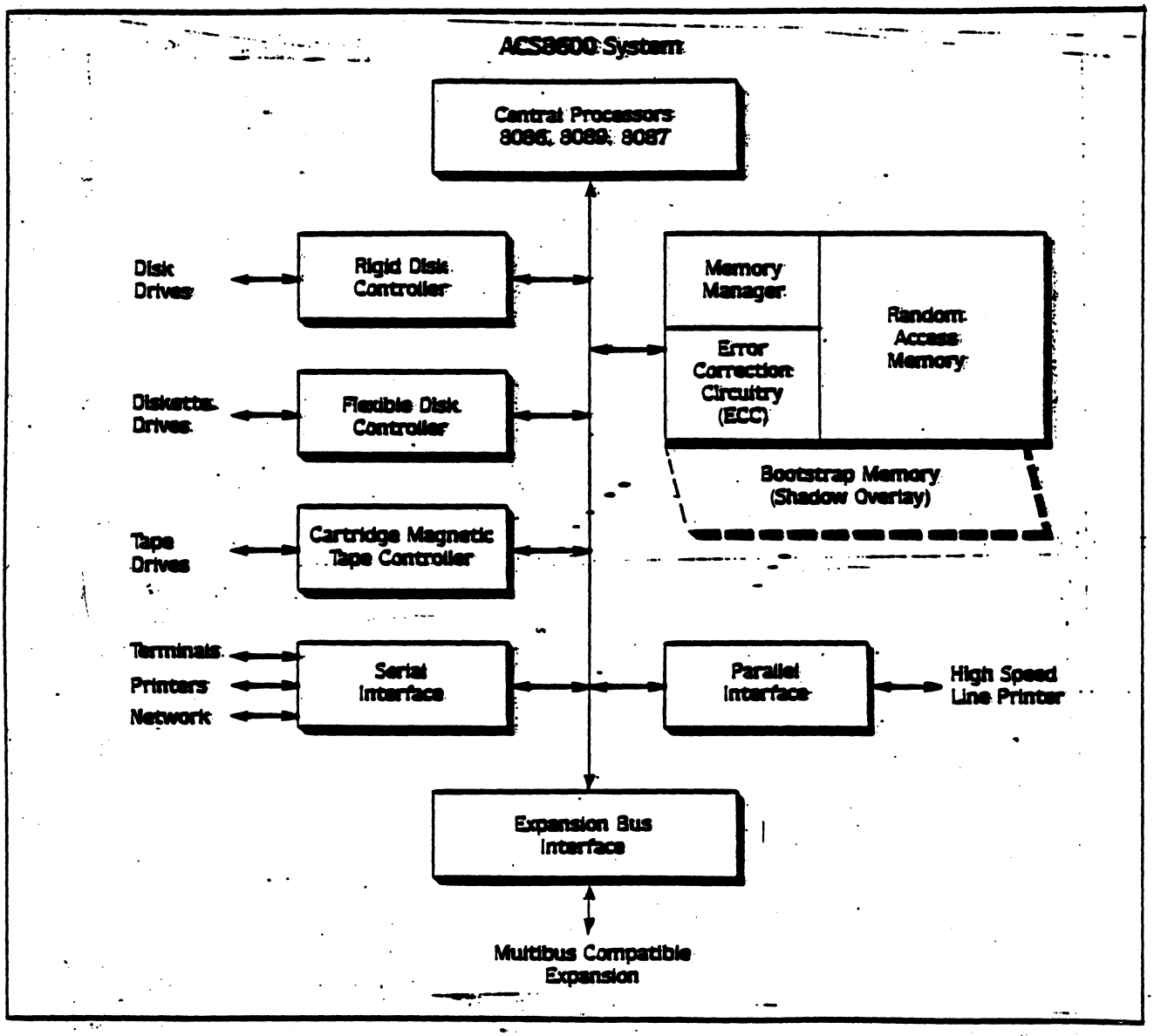


Figure 1-5. ACS 8600 Computer System Architecture - Block Diagram

MEMORY CONTROLLER

The memory controller allows word-organized memory to be accessed either as words or as bytes, assures proper memory refreshing, provides timing for the error correction circuitry, and notifies the CPU when memory data is ready. Nearly all of this logic is implemented in Large Scale Integrated (LSI) devices. An LSI dynamic RAM controller device and programmable logic arrays provide memory control.

RANDOM ACCESS MEMORY (RAM)

The RAM memory of the ACS 8600 is equipped with an LSI device that can detect all single-bit and double-bit errors when data is read from the RAM array. When a word is read back from the memory, check bits are also read and checked. If the check bits compare then no error has occurred but if the check bits do not compare then one or more data or check bits are in error. The error correction code is chosen so that the syndrome generated by the checking process can be used to indicate which single bit is in error.

If a data bit is in error it is corrected by inverting but, if a check bit is in error, it is simply reported. This method of error correction can only correct single-bit errors. However, single-bit errors are the most frequently occurring errors. When an error is detected the syndrome and the address associated with the error are latched and the CPU is interrupted.

When the CPU services the interrupt, it can perform a scrubbing operation to attempt to remove the error. The CPU can also log the error for diagnostic and maintenance purposes and indicate if the error was successfully scrubbed.

MEMORY MANAGEMENT

The purpose of memory management is to provide both position independence and memory protection. The 8600 System uses an Intel 8086 microprocessor as the computing and controlling element. This processor provides position independence through the use of segment registers.

The 8086 generates addresses which allow it to access one million (1,048,576) bytes of memory, however this memory must be accessed in blocks of 64K (65,536) bytes. Limited protection is provided by the fact that the memory beyond the current four 64K segments cannot be accessed without altering the segment register. No limit checking or segment overlap protection is provided by the processor. Furthermore, any program may access the segment registers at any time.

EXTENDED MEMORY MANAGEMENT

The 8600 system extends the memory management facilities of the 8086 by supporting write protection, access protection, limit

checking and non-contiguous memory allocation. Finally the memory manager implements the concept of privileged instructions.

The purpose of privileged instructions is to prevent application programs from altering the system environment. By definition an application program must run with interrupts enabled and must make all I/O access requests through the operating system. Therefore hardware inhibits all I/O operations while applications programs are executing and the memory manager will generate an interrupt if it detects either the execution of a HALT instruction or the resetting of the interrupt-enable flip-flop while the system is running an application program.

PAGES IN MEMORY

The one million bytes of main memory are managed as 256 pages of 4K (4,096) bytes each. Each page is assigned a relocation value, and thirteen flag and control bits. This memory relocation and attribute data is stored in a static RAM.

When memory is accessed, the relocation value is used to map the access onto the desired physical memory location. At the same time attribute bits are examined to test the validity of the access.

The violation port is the first of the three error reporting registers supported by the memory manager. This 16- bit register indicates the type of memory access violation that has occurred. Two other I/O ports are used to report the address at which the violation occurred.

ERASABLE-PROGRAMMABLE-READ-ONLY-MEMORY (EPROM)

The system contains 8-Kbytes of Read Only Memory (ROM) primarily intended for bootstrapping purposes. This memory is currently implemented with two 2732 type Erasable Programmable Read Only Memory (EPROM) devices. However the hardware can be strapped to accommodate 2716-type devices if less EPROM memory is required. This memory overlays the uppermost 8-Kbytes of the 1-Mbyte RAM memory space when 2732-type devices are used.

When 2716-type devices are used the uppermost 4-Kbytes of RAM are overlayed. This memory can be switched out of memory to permit access to a full Mbyte of RAM. Boot memory contains a unique system serial number that permits software to be "locked" to a specific hardware system. In addition to the serial number, boot memory contains system configuration information.

FLOPPY DISK CONTROLLER

The main board of the ACS 8600 system contains a floppy diskette controller. This controller is implemented with LSI devices and can support up to four 8-inch floppy diskette drives operating in either single-density or double-density recording mode.

HARD DISK CONTROLLER

The system provides additional high-speed mass storage through a generalized parallel interface to a family of rigid disk controller boards that mount directly to the CPU board. This interface can support rigid disk controllers with an 8-bit data path. This type of rigid disk controller and interface provides a future growth path that permits the support of 5-inch Winchester disks as well as other disk drive technologies.

The initial member of the family of daughter boards provide a controller for the 8-inch Winchester-type hard disks. The controller supports one or two drives, giving mass storage capability ranging from 10- to 80-Mbytes.

PARALLEL INTERFACE

A 24-bit programmable parallel interface is provided on the main system board. This interface is primarily for driving high-speed line printers but may be reconfigured (by software) to support a wide variety of applications. It is implemented with an 8255 Programmable Peripheral Interface (PPI) device which contains three 8-bit ports. Ports A and B of the 8255 are buffered with bi-directional buffers that permit these ports to be used for either input or output. Port C is buffered with a unidirectional buffer that provides three bits of input and five bits of output. The buffering on Port C supports the handshake and interrupt signals used by port B in mode one.

SERIAL CHANNELS

The system provides a minimum of two, and a maximum of eight serial channels. Two channels are contained on the main CPU board and are primarily intended for standalone operation in manufacturing test. The other eight are contained on an Intelligent Serial Concentrator Board. One channel on the CPU board can be strapped for either synchronous or asynchronous operation. This multipurpose channels can be strapped to interface to a variety of networks operating at data rates up to 800 kilobits per second. All serial channels on the CPU board are supported with fully vectored interrupts.

(Some system configurations may be offered which will not include the Intelligent Serial Concentrator Board.)

INTELLIGENT SERIAL CONCENTRATOR

The Intelligent Serial Concentrator Board consists of a Z80 processor with eight serial channels under its control. This board interfaces to the main board through the Multibus interface. The Z80 processor communicates with the 8086 via main memory and a pair of interrupt lines. The 8086 may interrupt the Z80 and the Z80 may interrupt the 8086. The eight serial channels are capable of operating in asynchronous mode at all

standard baud rates between 110 baud and 38.4 kilobaud.

In addition, one channel has provisions for external clocking on both transmit and receive clocks to support bi-synchronous serial devices. Each serial channel has a corresponding programmable timer for baud rate generation to allow for independent baud rates. This board is intended to decrease activity in the host processor so that the majority of character interrupt processing may be handled by the Intelligent I/O. This should allow the host processor to devote more of its power to higher level tasks and handle users in a more timely fashion.

MAGNETIC TAPE UNIT DRIVE CONTROLLER

A magnetic tape drive controller, capable of controlling up to eight tape transports, can be mounted on the CPU board. This controller controls a DEI Funnel Tape Transport with a the codec and controller options installed. This combination provides complete archival and backup capability for the rigid disks on the system.

MULTIBUS CARD CAGE

An additional enclosure, containing a standard Multibus card cage, can be connected through rear-panel connections on the primary enclorure. This expansion capability allows system users to add custom interfaces to the system. The interface supports both 8-bit and 16-bit bus masters that use the daisy chain technique for resolving the priority of bus requests. Multibus interface design includes socketed termination resistors for all Multibus signals that must be terminated on one of the bus masters. The signals in the interface meet the electrical and driver-receiver requirements of the Multibus Specification published by Intel.

CONVENTIONS

Altos Computer Systems manufactures equipment for both domestic and foreign markets. Domestic equipment operates on 115-volt AC power; foreign equipment operates on 230-volt AC power. Domestic equipment is identified by an "A" after the model designator on the back panel identfication plate; foreign equipment is identified by an "E". This convention is sometimes reflected in the part numbering scheme, wherein the final three digits "001" usually designate domestic-use parts and "002" usually designate foreign-use parts. For brevity, the final three digits of part numbers may be omitted in this manual.

TERMS AND ABBREVIATIONS

For brevity, the following abbreviations are used in this manual:

MTU	-	Magnetic Tape Unit or Transport
MTC	-	Magnetic Tape Controller (board)
CPU	-	Central Processing Unit (board)
HDU	-	Hard (Rigid) Disk (Drive) Unit
HDC	-	Hard Disk Controller (board)
PCB	-	Printed Circuit Board
Kbyte	-	Kilobyte (one thousand bytes)
Mbyte	-	Megabyte (one million bytes)

RELATED PUBLICATIONS

The following publications are related to or used wih the ACS 8600 Computer System:

(LIST TO COME)

SECTION 2.

UNPACKING AND SETTING UP THE COMPUTER SYSTEM

GENERAL INFORMATION

This section explains how to:

Select a computer site.

Unpack the shipping container.

Store important papers for future reference.

Configure the system (if necessary).

Connect major system components (computer,CRT terminal, and printer).

Turn on major components.

SELECTING A COMPUTER SITE

The ACS 8600 Computer System is rugged and dependable under any reasonable conditions, however, it operates most satisfactorily when conditions are as close to ideal as possible. Consider the human needs of the operator; light, comfort, quiet, etc.

Power cords and interconnect cables should be routed to avoid walkways.

Power cords should be plugged into a single power strip of adequate capacity. A dedicated power line is recommended to minimize the liklihood of power surges or overloads.

NOTE

Be sure that the AC input voltage conforms to that specified on the 8600 computer rear-panel Identification Tag.

Positiion the terminal so that it faces away from windows in order to reduce screen glare.

Either select an area with minimal static electricity, or use anti-static mats and other anti-static materials as necessary.

Ensure that air flow through the top intake vents and the rear exhaust vent is not obstructed.

Remote multiple-user terminals may be located anywhere that can be reached by a communications cable of up to 50 feet in length.

UNPACKING

Remove the computer system from the shipping container as follows:

Turn the shipping container right side up and carefully cut the binding tape and open the container.

CAUTION

Use reasonable care in opening the shipping container; sharp or pointed instruments may pierce the protective cover and scratch the finish of the computer housing.

Note the method of packing, and remove the computer components. Store pertinent papers in a safe place for future reference.

Gently tip the container on its side and slide out the computer chassis and the foam packing pieces. Place the chassis on a suitable work surface and replace the molded foam pieces in the container.

Store the shipping container and the packing materials in a safe place (in case the computer must be shipped again).

The following items are included in the shipping container. Check off each item as it is unpacked.

One Altos ACS 8600 Computer

One three pronged power cord

Either one 4-ampere fuse (for 115 vac systems), or one 2-ampere fuse (for 230 vac systems)

One Supplement 3. EXECUTING THE ALTOS DIAGNOSTIC EXECUTIVE (ADX) PROGRAMS and one floppy diskette containing the ADX programs used to test your computer system

One Supplement 4. which provides installation and execution instructions for the operating system to be installed (CP/M-86, MP/M-86, OASIS-16, or XENIX), and one set of operating system diskettes.

This ACS 8600 Computer System User Manual, accompanied by the following:

One Cover Letter

One Checkoff Sheet

One Warranty Card

Four Quality Control Reports

System Checkoff Sheet

System Checkout Sheet

Burn-In Summary Sheet

Quality Control Inspection Sheet

UNLOCKING THE HARD DISK DRIVES

Removing the Bottom Cover Plate

Follow this procedure to remove the bottom cover:

1. Carefully turn the computer enclosure upside down on a non-abrasive work surface.
2. Remove the four retaining screws (Figure 2-1) securing the rubber feet. Save the rubber feet and the four screws.
3. Remove the Bottom Cover Plate (Figure 2-1) and set it aside.

Unlocking the 10-Mbyte Hard Disk Drive

Follow this procedure to unlock the 10-Mbyte Hard Disk Drive:

1. Remove the screw and locking bracket from the hard disk spindle motor (Figure 2-2). Store the screw and bracket in a safe place.

CAUTION

The bracket and screw must be reinstalled any time the computer is moved any further than it is convenient to hand carry it.

2. Replace the bottom cover on the enclosure and secure the four rubber feet previously removed.

3. Carefully return the unit to its upright position and proceed with the installation.

Unlocking the 20-Mbyte or 40-Mbyte Hard Disk Drive

Follow this procedure to unlock either the 20-, or 40-Mbyte Hard Disk Drive:

1. Loosen the drive motor nut (with a hex socket or hex nut driver) from the hard disk spindle motor (Figure 2-3). Rotate the locking clip away from the pulley. **Do not rotate the pulley.** Re-tighten the nut. **Do not overtighten.**

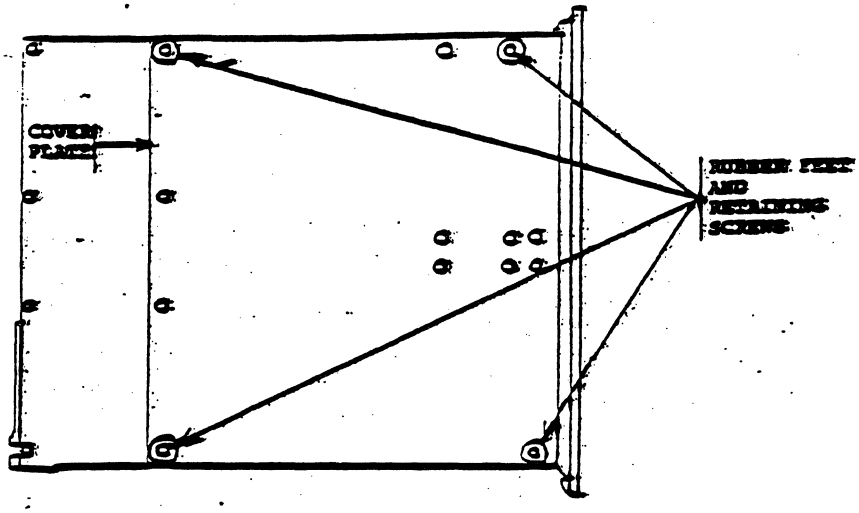
CAUTION

The locking clip must be reinstalled any time the computer enclosure is moved any further than it is convenient to hand carry it.

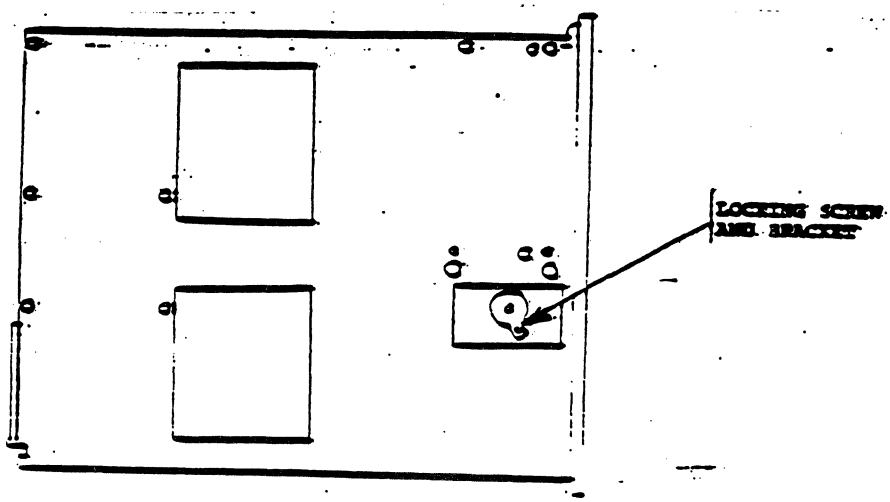
2. Locate the Actuator Lock Access cutout (Figure 2-3). Using a straight-bladed screwdriver, rotate the actuator lock counter-clockwise 1/4 turn from the "LOCK" position to the "RUN" position.

3. Replace the Bottom Cover Plate on the computer enclosure and secure the four rubber feet with the four retaining screws previously removed.

4. Carefully return the unit to its upright position and proceed with the installation.

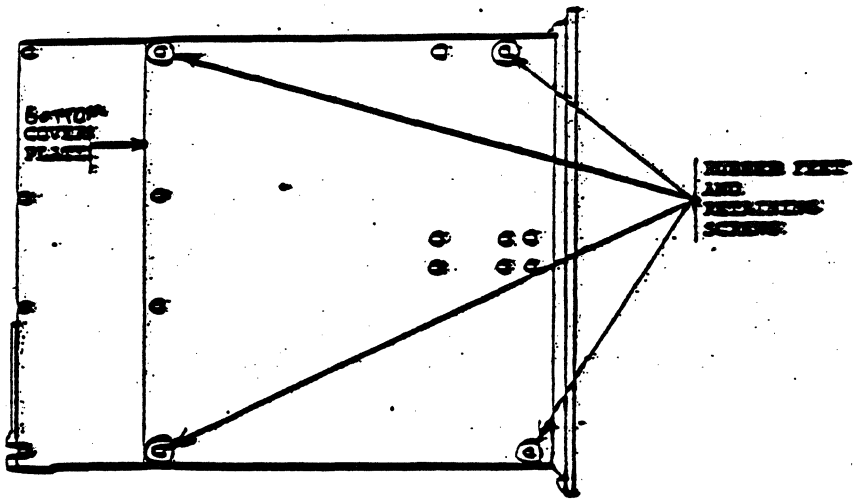


Computer Chassis with Bottom Plate Mounted

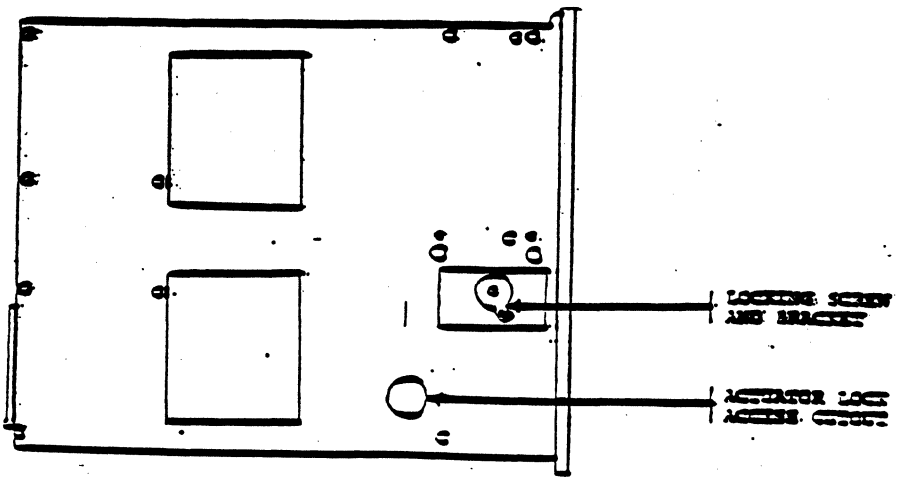


Computer Chassis with Bottom Plate Removed and Locking Devices Exposed

Figure 2-1. Unlocking the 10-Mbyte Hard Disk



Computer Chassis with Bottom Plate Mounted



Computer Chassis with Bottom Plate Removed and Locking Devices Exposed

Figure 2-2. Unlocking the 20-, or 40-Mbyte Hard Disk

SYSTEM CONFIGURATION

(Refer to Appendix C. COMMON CRT AND PRINTER INTERFACE CONFIGURATIONS, and Appendix D. INSTRUCTIONS FOR PINNING CIRCUIT BOARDS.)

A complete ACS 8600 Computer System includes:

An 8600 computer (complete with power cord, fuse and fuse cap.)

Model ACS 8600-10, -10 MTU, -12, -12 MTU, -14, or -14 MTU.

A user-supplied keyboard CRT terminal.

Serial data transmission rate of 9600 Baud required. RS-232-C interface.

Optionally, a user-supplied printer (dot matrix or impact quality print).

RS-232-C serial or Centronics parallel interface, one or two (one serial and one parallelinterface) can be supported, depending on the operating system used.

User-supplied interface cables for the terminal and printer.

To maintain Class A compliance and limit possible radio/TV interference, all interface cables should have a grounded shield and be electrically and mechanically secure to the units to which they are connected.

Altos Diagnostic Executive (ADX) program diskette and operating instructions (contained in Supplement 3).

Provided with the system.

Operating system diskette(s), installation instructions (provided in Supplement 4) and supporting publications.

At least one operating system is required. The diskettes, instructions, required pblications must be purchased separately from the system.

NOTE

All operating systems are registered with the licensed seller and distributed under license to the end user to operate only on a single computer. CP/M-86, MP/M-86, OASIS-18, and XENIX must be purchased from Altos Computer Systems and the diskette label should so identify it, with a version number and serial number. No other copies of these systems are authorized except by special agreement.

CONNECTING PERIPHERAL EQUIPMENT

Install the interface cables and power cords for the terminal(s) and printer(s).

Console Terminal

The console terminal communications cable must be connected from the RS-232-C port from the terminal to serial port 1 of the 8600 computer (reference Figure 2-3). The terminal must be programmed to communicate with serial port 1 of the 8600 computer according to the parameters specified below:

NOTE

Most RS-232-C interface cables can be easily made using male shielded RS-232-C 25-pin type connectors and connecting the pins in a one-to-one correspondence. All cables should be fastened with the connector retaining screws to provide a proper shield ground path.

RS-232-C serial communications

9600 Baud

Asynchronous Mode with Data-Terminal-Ready (DTR) handshaking. Request-to-send (RTS) and clear-to-send (CTS) handshaking is disabled

One start-and-stop bit

No parity bit

Eight-bit word length

The pinout signals for the console terminal port (port 1) are as follows:

PIN 1	Chassis Ground
PIN 2	Transmitted Data (from the terminal)
PIN 3	Received Data (to the terminal)
PIN 4	Request-to-Send (optional--requires jumper change)
PIN 5	Clear-to-Send
PIN 6	Data set Ready
PIN 7	Signal Ground
PIN 8	+12 vdc
PIN 9	N/C
PIN 10	N/C
PIN 11	N/C
PIN 12	N/C
PIN 13	N/C
PIN 14	N/C
PIN 15	N/C

PIN 16 N/C
PIN 17 N/C
PIN 18 N/C
PIN 19 N/C
PIN 20 Data-Terminal-Ready (DTR)

CONNECTING THE POWER SOURCE TO THE COMPUTER

To provide power to the computer:

Verify that the power specifications on the rear panel identification plate match the voltage available for the equipment (see Figure 2-3).

Domestic equipment uses 115 volts AC and a 2-ampere fuse, and is designated by an "A" after the model number; foreign equipment uses 230 volts AC and a 4-ampere fuse, and is designated by an "E" after the model number;

Insert the fuse and fuse cap (provided wih the system) into the fuse receptacle in the rear panel.

Install the three-pronged power cord in the rear panel AC receptacle and plug it into a power source (wall plug, etc.).

Be certain that the floppy disk transport (or the magnetic tape cartridge transport) is unloaded and the door is open.

Turn the terminal on and allow sufficient warmup time so computer messages can be observed when that unit is turned on.

Press the top half of the power switch located at the upper right hand corner of the front panel. The power indicator will light and the following message will appear on the screen.

ALTOS COMPUTER SYSTEMS

MONITOR VERSION n.nn

PRESS ANY KEY TI INTERRUPT BOOT

Immediately press any key on the keyboard to interrupt the boot process.

NOTE

You must interrupt the boot process within two seconds.

If the boot process is not interrupted, the following display appears:

BOOTING FROM HARD DISK

Because the operating system has not yet been installed, the computer will not boot successfully. Press the round RESET button (to the left of the POWER switch on the computer front panel) to restart the boot process, then (within two seconds) press any key.

The system is ready for execution of dignostic programs as described in Section 3.

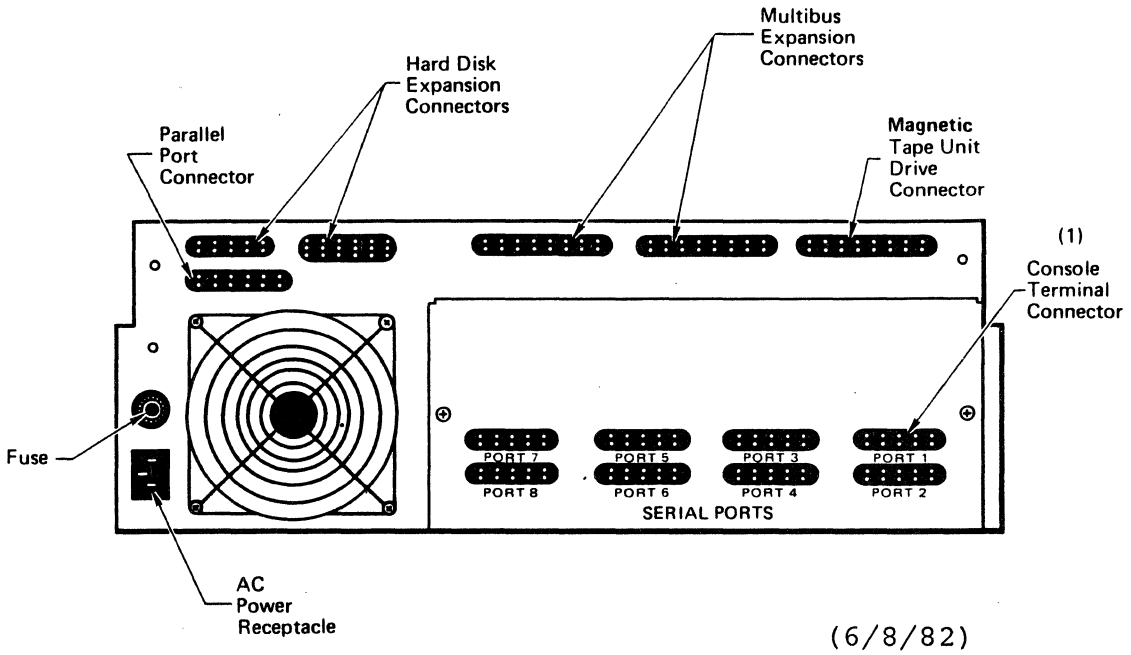


Figure 2-3. ACS Computer System - Rear Panel

SECTION 3.

DIAGNOSTICS

GENERAL INFORMATION

This section introduces Supplement 3. EXECUTING THE ALTOS DIAGNOSTIC EXECUTIVE (ADX) PROGRAM which should be inserted in this part of the manual when the shipping container is unpacked.

Supplement 3 provides step-by-step instructions for executing diagnostic programs under supervision of the diagnostic executive to verify operation of the system and its components. ADX prepares the computer system for installation of the selected Operating System.

Follow the instructions in Supplement 3 to configure the system, execute the diagnostic programs, and initialize the system. The diagnostic programs verify correct operation of the system and its components. Supplement 3 also instructs you in making backup copies of the diagnostic program and operating system diskettes as required.

SECTION 4.

OPERATING SYSTEMS

GENERAL INFORMATION

This section introduces the following versions of Supplement 4, one or more of which should be inserted in this part of the User Manual when they are unpacked from the shipping container.

Supplement 4. CONTROL PROGRAM MONITOR (CP/M-86) OPERATING SYSTEM - INSTALLATION INSTRUCTIONS

Supplement 4. MULTI PROGRAM MONITOR (MP/M-86) OPERATING SYSTEM - INSTALLATION INSTRUCTIONS

Supplement 4. OASIS (OASIS-16) OPERATING SYSTEM - INSTALLATION INSTRUCTIONS

Supplement 4. XENIX (UNIX-VERSION 7) OPERATING SYSTEM - INSTALLATION INSTRUCTIONS

(Only the supplement(s) related to the Operating System(s) purchased is provided with the system.)

Each supplement provides specific, step-by-step installation procedures for one particular operating system, and also introduces you to the supporting manuals and other publications related to that operating system. One of four Operating Systems will be installed. Each of these systems has its own loading procedures and operating characteristics and may have their own set of diagnostic programs.

Altos Computer Systems supports and distributes four of the most popular operating systems, CP/M-86, MP/M-86, OASIS-16, and XENIX. More than 300,000 current users of CP/M, MP/M and OASIS can easily upgrade their systems to the ACS 8600 Computer System knowing that their application software is compatible with the ACS 8600 computer.

CP/M-86 SINGLE-USER OPERATING SYSTEM

CP/M-86 is the 16-bit version of the popular CP/M operating system. CP/M-86 is a gernal-purpose operating system desig specially for microprocessors that use the 8086 or 8088 microprocessor.

CP/M-86 is efficient and powerful with time-tested modular design. CP/M-86 system can support application programs that range from small to comlex. Managing up to a 1-Mbyte (1,048,576 bytes) of main memory, CP/M-86 gives the application programs full advantage of the 8086 address space. Complete file compatibility simplifies conversion from the 8-bit CP/M system to the 16-bit CP/M-86 system.

Requirements for this system are:

An 8086 or 8088 microprocessor.

Minimum of 56Kbytes of RAM.

One to sixteen disk drives of up to 8 Mbytes each.

An ASCII console device such as a CRT terminal.

MP/M-86 MULTI-USER OPERATING SYSTEM

MP/M is a multi-user, 16-bit, multi-tasking operating system designed for microcomputers that use the 8086/8088 microprocessor. Upward compatible wih CP/M, MPM-II and CP/M-86 operating systems, MP/M-86 provides record and file locking with password protection for use in business settings where data base integrity is essential.

MP/M-86 supports up to sixteen logical drives managing up to 512-Mbytes each. MP/M-86 features extensive error handling and reporting, real-time capabilities, and date and time stamps on files.

Requirements for this operating system are:

A compatible 8086/8088 microprocessor.

Minimum of 64-Kbytes of RAM

One-to-sixteen drives of up to 512-Mbytes each.

One or more ASCII terminals.

Real-time clock.

CP/M-86 must be implemented on the target machine.

128-Kbytes of RAM for implementation.

OASIS OPERATING SYSTEM

(THIS INFORMATION WAS NOT READY WHEN THIS MANUAL WAS PRINTED)

XENIX (UNIX - VERSION 7) OPERATING SYSTEM

(THIS INFORMATION WAS NOT READY WHEN THIS MANUAL WAS PRINTED)

PROGRAMMING LANGUAGES

BASIC, COBOL, PASCAL, and FORTRAN languages are also distributed and supported by Altos Computer Systems.

APPLICATION PROGRAMS

Although Altos does not generally distribute application program packages, it does provide a list of such compatible program packages, developed by leading software houses, on request.

SECTION 5.

ALTOS UTILITY PROGRAMS

GENERAL INFORMATION

This section introduces Supplement 5. ALTOS UTILITY PROGRAMS - USER INSTRUCTIONS which should be inserted in this part of the manual when the shipping container is unpacked.

Supplement 5 describes Altos Utility Programs available for use with the system, and instructs you in their use.

(No Altos Utility Programs have been released at the time this manual was printed.)

SECTION 6.

UPGRADE KITS AND ADD KITS

GENERAL INFORMATION

This section introduces the following versions of Supplement 6, one or more of which should be inserted in this part of the User Manual when they are unpacked from the shipping container.

(You will receive kit installation instructions only for the add kits or upgrade kits you purchase)

The primary ACS 8600 Computer System enclosure contains the CPU with up to 1-Mbyte of RAM, up to ten serial channels, all peripheral controllers, power supplies and expansion interfaces, a 10-, 20-, or 40-Mbyte Winchester-type, hard-disk drive, and either an eight-inch, 500-Kbyte, single-sided, floppy-diskette drive, or a cartridge-type, 17-Mbyte, Magnetic Tape Unit (MTU) as shown in Table 6-1.

NOTE

The Magnetic Tape Unit mentioned throughout this manual is presently scheduled for release in the third quarter of 1982.

Table 6-1. ACS 8600 - Model/Storage Media Reference Table

ACS 8600 Model	Primary Media	Secondary Media
8600-10	10-Mbyte Hard Disk	500-Kbyte Floppy Disk
8600-12	20-Mbyte Hard Disk	500-Kbyte Floppy Disk
8600-14	40-Mbyte Hard Disk	500-Kbyte Floppy Disk
8600-10 MTU	10-Mbyte Hard Disk	17-Mbyte MTU
8600-12 MTU	20-Mbyte Hard Disk	17-Mbyte MTU
8600-14 MTU	40-Mbyte Hard Disk	17-Mbyte MTU

One additional secondary enclosure containing power supplies, a second hard disk and an MTU can be added to assemble a wide variety of system configurations (see Table 1-2).

Table 6-2. Add Kit/ Storage Media Reference Table

Kit Number	Primary Media	Secondary Media
Add-10	10-Mbyte Hard Disk	None
Add-12	20-Mbyte Hard Disk	None
Add-14	40-Mbyte Hard Disk	None
Add-10 MTU	10-Mbyte Hard Disk	17-Mbyte MTU
Add-12 MTU	20-Mbyte Hard Disk	17-Mbyte MTU
Add-14 MTU	40-Mbyte Hard Disk	17-Mbyte MTU
Add RAM	(Adds 500-Kbytes of Random Access Memory)	
Add MTU-3	None	17-Mbyte MTU
Add	(Adds 8087 Floating Point Chip Hardware and FORTRAN Software)	

SECTION 7.

SYSTEM SPECIFICATIONS

GENERAL INFORMATION

The ACS 8600 System Specifications have been included in this preliminary User Manual so that the information can be distributed as soon as possible. This information will be incorporated into the manual in later revisions.

ACS 8600 SYSTEM SPECIFICATION

Revision 4.2

May 27, 1982

Any questions or comments concerning this document should be directed to:

Altos Computer Systems
Attn. Jim Willott
2360 Bering Drive
San Jose, CA 95131
Phone (408) 946-6700

CHANGE NOTICE NUMBER ONE

This Change Notice adds one Note and three Warnings and changes Table 4-26. PARALLEL (CENTRONICS) INTERFACE CONNECTOR on Page 65 as noted below.

The new Notes and Warnings are contained on Pages ii and iii.

Line 1 of Table 4-26 on Page 65 previously read:

| J6-15 | 6 | 10 | ACKNOWLEDGE- from LP |

It has been changed to read:

| J6-15 | 8 | 10 | ACKNOWLEDGE- from LP |

Additional related changes:

This Change Notice has been added as Page i.

Pages ii and iii have been added to the specification.

Page i in the previous revision has been changed to iv.

Page ii in the previous revision has been changed to v.

Page iii in the previous revision has been changed to vi.

No other changes have been made to the specification.

NOTE

The Etch Revision Letter referenced in the following Warnings is etched in the copper on the solder side of the CPU Board along one of the edges.

WARNING

All ACS 8600 Computer Systems have internal terminations on the Multibus interface "BCLK" and "CCLK" signals. These terminations must be disabled when using the Multibus Expansion Port. On Etch Revisions H and earlier (Revisions C, E, and H) of the CPU Board, the jumpers 18 and 19 that perform this function are located at device location 22R as listed below:

Jumper 18 - 22R Pin 10 to Pin 11
Jumper 19 - 22R Pin 13 to Pin 14

On Etch Revision J, jumpers 18 and 19 are located near device location 1U.

WARNING

On Etch Revisions H and earlier (Revisions C, E, and H) of the CPU Board, the terminations used on the "BCLK" and "CCLK" signals of the Multibus Interface are physically located on RP6 which is also used to terminate some of the parallel interface signals. Those users who are planning a non-standard use for the parallel interface which requires removal or replacement of RP6 must also provide 220-230 Ohm terminations for these two signals to assure reliable operation of the Intelligent I/O Interface board which handles of all terminals.

WARNING

ACS 8600 Computer Systems using Etch Revision C CPU Boards do not have Jumper 17 in the parallel interface. If the user plans to use the parallel interface 8255 such that Port B is programmed in Mode 0, the trace between 28S Pin 16 and 29T Pin 5 must be cut to prevent damage to the output driver of the 8255.

TABLE OF CONTENTS

L I S T O F F I G U R E S

L I S T O F T A B L E S

ACS 8600 SYSTEM SPECIFICATION

Revision 4.2

May 27, 1982

1. ABSTRACT

This specification describes the ACS 8600 System - an advanced sixteen bit general purpose computer manufactured by Altos Computer Systems. The system design supports up to one megabyte of random access memory with full error detection and correction and a wide variety of mass storage peripherals. The system memory is provided with unique memory management hardware that improves the efficient use of main memory. The ACS 8600 can be easily extended to meet the requirements of a wide variety of applications. The various system configurations are listed with electrical and environmental requirements defined. All hardware elements are described in detail with specific attention being given to the hardware-software interfaces.

2. SYSTEM FUNCTIONAL SPECIFICATION

2.1. Features and Capabilities

The ACS 8600 System is a sixteen bit general purpose computer based on the Intel 8086 microprocessor family. While the system is designed to meet multiuser and multitasking requirements it may be configured for large single user applications. The following list summarizes the main features:

- Sixteen bit CPU section consisting of an 8086 microprocessor, an 8089 Direct Memory Access (DMA) - I/O Processor and an optional 8087 Numeric Data Processor.
- Sixteen bit main memory consisting of 512 kilobytes which can be expanded to one megabyte.
- Error correction and detection on the main memory providing single and double bit error detection with single bit correction.
- Memory management on the main memory providing address translation, write protection and access control for each four kilobyte block of main memory.
- Bootstrap and configuration control read only memory permitting system initialization from any online mass storage device.
- Flexible diskette controller capable of controlling up to four eight-inch diskette transports with support for single and double density recording as well as single and double sided diskette drives.
- Rigid disk controller for up to two drives of the eight inch Winchester type rigid, fixed disk technology.
- Configurable parallel interface capable of controlling one high speed, parallel input line printers.
- Serial interfaces designed to support asynchronous terminals and printers, synchronous interfaces and local networks.
- Magnetic tape controller for cartridge type magnetic tape transports.
- Fully vectored interrupt capability.
- Virtually unlimited expansion through a Multibus compatible expansion interface.

2.2. System Configurations

The ACS 8600 System is packaged in one or more desktop enclosures which may optionally be rack mounted. The main package contains the CPU with up to one megabyte of RAM, up to ten serial channels, all peripheral controllers, power supplies and expansion interfaces, and any two peripherals from the list of peripherals given in Table 2-1 below. Additional enclosures containing power supplies and two additional peripherals can be added to assemble a wide variety of system configurations some of which are listed below in Table 2-2.

- A ten, twenty or forty megabyte Winchester type rigid disk.
- An eight inch flexible diskette drive (one-half megabyte, single sided or one megabyte, double sided).
- A cartridge type magnetic tape transport (twelve megabyte on an American National Standards Institute X3.55-1977, 0.250 inch magnetic tape cartridge).

Table 2-1: LIST OF SUPPORTED PERIPHERALS

- Main enclosure -- ten, twenty or forty megabyte rigid disk and eight inch flexible diskette.
- Main enclosure -- ten, twenty or forty megabyte rigid disk and cartridge tape.
- Peripheral enclosure -- ten, twenty or forty megabyte rigid disk and cartridge tape.

Table 2-2: STANDARD ENCLOSURE CONFIGURATIONS

3. HARDWARE FUNCTIONAL SPECIFICATION

The ACS 8600 is an advanced sixteen bit general purpose computing system capable of supporting one or more users. The system is designed around a single printed circuit board which contains most of the resources to support up to eight users. This single board computer also contains interfaces that permit it to be expanded to handle very large computing tasks. The system architecture is shown in Figure 3-1 on the next page.

The central processing unit (CPU) of the ACS 8600 System is based on three of the processors in the Intel 8086 microprocessor family. The block diagram for the CPU section is shown in Figure 3-2, page 6. The primary processor is the 8086 which is responsible for executing nearly all of the code. An 8089 Dual Channel Direct Memory Access-I/O Processor is arranged with the 8086 in a local configuration. The primary purpose of the 8089 is to provide sixteen bit wide DMA transfer capability. It can be used to process some of the I/O transfers. Provisions have been made for an optional 8087 Numeric Data Processor which can perform high speed floating point calculations for the 8086. The 8087 co-decodes instructions fetched by the 8086 and will bus request the 8086 as required to perform the arithmetic.

The block diagram for the ACS 8600 System Memory is shown in Figure 3-3, page 7. The System Memory is implemented with 64K dynamic random access memory (RAM) devices and 2732 type erasable, programmable read only memory (EPROM) devices. The system is designed to permit up to one megabyte of RAM which maybe overlayed by the EPROM. The RAM memory is organized as sixteen bit words with six extra bits used to support error correction. The RAM memory is divided into eight banks of twenty-two devices which provides 128 kilobytes of memory per bank. The four lower banks of the memory are located on the CPU Board for 512 kilobytes of memory and the four upper banks are located on the Expansion Memory Board for the remainder of the one megabyte of system RAM. The RAM memory is supported by three other blocks of logic: the memory controller, the error correction circuitry and the memory manager.

The Memory Controller allows the word organized memory to be accessed either as words or as bytes, assures proper memory refresh, provides timing for the Error Correction Circuitry and notifies the CPU when memory data is ready. Nearly all of this logic is implemented in large scale integrated (LSI) devices. An LSI dynamic RAM controller device is used to provide much of the control of the memory while programmable logic arrays provide the remainder.

Figure 3-1: ACS 8600 SYSTEM BLOCK DIAGRAM

Figure 3-2: CPU SECTION BLOCK DIAGRAM

Figure 3-3: MEMORY SECTION BLOCK DIAGRAM

The RAM memory of the ACS 8600 is equipped with an LSI device that can detect all single bit and all double bit errors on data read from the RAM array. When a word is read back from the memory, check bits are also read and checked. If the check bits compare then no error has occurred but if the check bits do not compare then one or more data or check bits are in error. The error correction code is chosen so that the syndrome generated by the checking process can be used to indicate which single bit is in error. If a data bit is in error it is corrected by inverting while if a check bit is in error is is simply reported. This method of error correction can only correct single bit errors. However, single bit errors are the most frequently occurring errors. When ever an error is detected the syndrome and the address associated with the error are latched and the CPU is interrupted. When the CPU services the interrupt it can perform a scrubbing operation to attempt to remove the error. The CPU can also log the error for diagnostic and maintenance purposes and indicate if the error was successfully scrubbed.

The purpose of memory management is to provide both position independence and protection for the contents of memory. The Altos ACS 8600 uses an Intel 8086 microprocessor as the computing and controlling element. This processor provides position independence through the use of segment registers. The 8086 generates addresses which allow it to access one million (1,048,576) bytes of memory, however this memory must be accessed in blocks of 64K (65,536) bytes. Limited protection is provided by the fact that the memory beyond the current four 64K segments cannot be accessed without altering the segment register. No limit checking or segment overlap protection is provided by the processor. Further more, any program may access the segment registers at any time.

The ACS 8600 extends the memory management facilities of the 8086 by supporting write protection, access protection, limit checking and non-contiguous memory allocation. Finally the memory manager implements the concept of privileged instructions. The purpose of privileged instructions is to prevent application programs from altering the system environment. By definition an application program must run with interrupts enabled and must make all I/O access requests through the operating system. Therefore hardware inhibits all I/O operations while applications programs are executing and the memory manager will generate an interrupt if it detects the execution of an instruction which resets the interrupt enable flip-flop while the system is running an application program.

The one million bytes of main memory are managed as 256 pages of 4K (4,096) bytes. Each page is assigned a relocation value, and thirteen flag and control bits. This memory relocation and attribute data is stored in a static RAM. When a memory access is made the relocation value is used to map the access onto the desired physical memory location. At the same time the attribute bits are examined to test the validity of the access.

The violation port is the first of the three error reporting registers supported by the memory manager. This sixteen bit register indicates the type of memory access violation that has occurred. Two other I/O ports are used to report the address at which the violation occurred.

The ACS 8600 system contains eight kilobytes of read only memory primarily intended for bootstrapping purposes. This memory is currently implemented with two 2732 type erasable programmable read only memory (EPROM) devices. However the hardware can be strapped to accommodate 2716 type devices if less EPROM memory is required. This memory overlays the upper most eight kilobytes of the one megabyte RAM memory space when 2732 type devices are used. When 2716 type devices are used the upper most four kilobytes of RAM are overlayed. This memory can be switched out of memory to permit access to a full megabyte of RAM. The boot memory has space reserved for the customer to program the PROM with a unique system serial number that permits software to be "locked" to a specific hardware system. In addition to the serial number, the boot memory contains system configuration information.

The main board of the ACS 8600 system contains a flexible diskette controller whose block diagram is shown in Figure 3-4 on the next page. This controller is implemented with large scale integrated devices and can support up to four eight-inch flexible diskette drives operating in either single density or double density recording mode.

The ACS 8600 system provides additional high speed mass storage through a generalized parallel interface to a family of rigid disk controller boards that mount directly to the CPU board. This interface can support rigid disk controllers with an eight bit data path. This type of rigid disk controller and interface provides a future growth path that permits the support of five inch Winchester disks as well as other disk drive technologies. The initial member of the family of daughter boards provide a controller for the eight inch Winchester type rigid disks. The controller supports up to two drives giving mass storage capability of from ten megabytes to eighty megabytes.

A twenty-four bit programmable parallel interface is provided on the main board of the ACS 8600 system. This interface is primarily intended for driving a high speed line printer but may be reconfigured by software to support a wide variety of applications. It is implemented with an Intel 8255 Programmable Peripheral Interface device which contains three eight bit ports. Ports A and B of the 8255 are buffered with bi-directional buffers that permit these ports to be used for either input or output. Port C is buffered with a uni-directional buffer that provides one bits of input and seven bits of output. The buffering on Port C supports the handshake and interrupt signals used by port B in mode one.

Figure 3-4: FLEXIBLE DISKETTE CONTROLLER BLOCK DIAGRAM

The ACS 8600 system provides a minimum of two serial channels and a maximum of ten serial channels. Two channels are contained on the main CPU board and are primarily intended for stand alone operation in manufacturing test. The other eight are contained on an Intelligent Serial Concentrator Board. One channel on the CPU board can be strapped for either synchronous or asynchronous operation. This multipurpose channels can be strapped to interface to a variety of networks operating at data rates up to 800 kilobits per second. All serial channels on the CPU board are supported with fully vectored interrupts. Some system configurations may be offered which will not include the Intelligent Serial Concentrator Board.

The Intelligent Serial Concentrator Board consists of a Z80 processor with eight serial channels under its control. This board interfaces to the main board through the Multibus interface. The Z80 processor communicates with the 8086 via main memory and a pair of interrupt lines. The 8086 may interrupt the Z80 and the Z80 may interrupt the 8086. The eight serial channels are capable of operating in asynchronous mode at all standard baud rates between 110 baud and 38.4 kilobaud. In addition one channel has provisions for external clocking on both transmit and receive clocks to support bi-synchronous serial devices. Each serial channel has a corresponding programmable timer for baud rate generation to allow for independent baud rates. This board is intended to off-load the host processor such that the majority of character interrupt processing may be handled by the Intelligent I/O. This allows the host processor to devote more of its power to higher level tasks and handle users in a more timely fashion.

A magnetic tape drive controller which is capable of controlling up to eight tape transports can be mounted on the CPU board. This controller is intended to control a DEI Funnel Tape Transport with a the codec and controller options installed. This combination provides complete archival and backup capability for the rigid disks on the system.

The ACS 8600 system provides a rear panel connector that permits connection to an additional enclosure containing a standard Multibus card cage. This expansion capability allowes system users to add custom interfaces to the system. The interface supports both eight bit and sixteen bit bus masters that use the daisy chain technique for resolving the priority of bus requests. The design of the Multibus interface includes socketed termination resistors for all Multibus signals that must be terminated on one of the bus masters. The signals in the interface meet the electrical and driver-receiver requirements of the Multibus Specification published by Intel.

4. HARDWARE - DETAIL SPECIFICATION

This section of the ACS 8600 System Specification presents a detailed description of the hardware of the system. Specific attention is given to the hardware-software interfaces. In addition some hardware operational details are presented. Each module or section of the system is discussed separately.

4.1. Central Processor

The central processing unit of the ACS 8600 System is based on three of the processors in the Intel 8086 microprocessor family. The block diagram for this section is shown in Figure 3-1 (see Section 3.1, page 5) and the logic diagram is contained on sheets one and two of the ACS 8600 CPU Board schematic. The primary processor is the 8086 which is responsible for executing nearly all of the code. An 8089 Dual Channel DMA-I/O Processor is arranged with the 8086 in a "local" configuration. The primary purpose of the 8089 is to provide sixteen bit wide DMA transfer capability. It can be used to process some of the I/O transfers but can contribute little to that process because it lacks interrupt processing capability and must request the system bus away from the 8086. Provisions have been made for an optional 8087 Numeric Data Processor which can perform high speed floating point calculations for the 8086. The 8087 co-decodes instructions fetched by the 8086 and will bus request the 8086 as required to perform the arithmetic.

Since the three processors of the central processing unit are arranged in a "local" configuration, they must be supplied with a common clock. The clock used by the CPU section is generated by an 8284 Clock Generator. Since all three processors must operate from the same clock, the slowest available processor determines the system speed. Currently only five megahertz 8087 and 8089 processors are available. For this reason the system must operate at five megahertz.

The multiplexed local CPU bus is buffered and supplied to the rest of the system as three demultiplexed system buses -- Address Bus, Data Bus and Control Bus. These buses are modeled after the Multibus specification but differ in several ways -- the Address Bus and Data Bus are both positive true rather than negative true; the Control Bus contains additional signals that indicate a sixteen bit transfer is being made by the current bus master, the main CPU has control of the buses and a memory write will be performed by the main CPU during the current bus cycle. The Control Bus is generated by an 8288 Bus Controller which provides negative true (active low) signals for memory read, memory write, early memory write, I/O read, I/O write and early I/O write. An additional gate is used to generate the signal 16 BIT TRANSFER which is positive true. The 8288 also controls the direction and enabling of the data bus drivers. The Address Bus and the Data Bus are both positive true (active high), three-state buses. Since a Multibus compatible expansion port is

supported, an 8289 Bus Arbiter is used to resolve system bus access requests and to control all of the CPU bus buffers.

As part of the support for automated testing of the CPU board a gate has been added to the Data Bus enable signal. The Data Bus can be disabled by grounding pin 8R-4 to allow an automatic test system to supply instructions to the CPU section and verify proper operation of all of the CPU elements. In addition this technique can be used to supply test diagnostic software in place of the ROM.

4.2. I/O Port Decoder

The 8086 supports both eight bit and sixteen bit I/O and a total of 65,536 ports. In the case of eight bit I/O operations, the 8086 uses the fact that a given I/O address is even or odd to determine which half of the sixteen bit data bus to use--even addresses move data over Data Bus bits 0 through 7. The block diagram for the ACS 8600 I/O port decoder is shown in Figure 4-1, on the next page. The decoder can be thought of as having three parts: the memory manager port decoder, the tape controller port decoder and the general I/O port decoder. Physically the tape controller port decoder is located on the Tape Controller daughter board while the remainder of the decoder is on the CPU board. The logic of the decoder on the Tape Controller board is shown on sheet one of the Tape Controller Board Schematic while the logic contained on the CPU board is shown on sheet eighteen of the ACS 8600 CPU Board Schematic.

The I/O port decoder on the Tape Controller board decode ports 0D8 hexadecimal (hex) through 0DF hex in groups of four even addressed ports and four odd addressed ports. The specific port assignment is detailed in Table 4-1 on pages 15 through 19. The specific details concerning the significants of the various bits in the various ports is covered in Section 4.8.

The I/O decoder on the CPU board is a two level decoder. The first level decodes the chip enables for reading and writing the 512 sixteen bit I/O ports used to control the memory manager as well as the enable to the second level of the decoder. The second level decodes ports 000 hex through 03F hex as eight groups of eight sixteen-bit ports as well as decoding ports 040 hex through 07F hex as eight groups of four even addressed ports and eight groups of four odd addressed ports. It should be noted that while the eight groups of eight sixteen-bit ports are decoded, each group selects only one port or caries at most two states of significants. The detailed port assignment is given in Table 4-1 which is found on pages 15 through 19. The specific detailes of the meanings of the bits in the various ports is discussed in later sections dealing with the hardware modules.

Figure 4-1: I/O PORT DECODER BLOCK DIAGRAM

ADDR HEX	PORT SIZE	FUNCTION AND COMMENTS
0000	16	VIOLATION PORT - READ ONLY
0001		ACCESSES PORT 0000
0002		ACCESSES PORT 0000
0003		ACCESSES PORT 0000
0004		ACCESSES PORT 0000
0005		ACCESSES PORT 0000
0006		ACCESSES PORT 0000
0007		ACCESSES PORT 0000
0008	0	CLEAR VIOLATION PORT
0009		ACCESSES PORT 0008
000A		ACCESSES PORT 0008
000B		ACCESSES PORT 0008
000C		ACCESSES PORT 0008
000D		ACCESSES PORT 0008
000E		ACCESSES PORT 0008
000F		ACCESSES PORT 0008
0010	16	ERROR ADDRESS 1 - READ ONLY
0011		ACCESSES PORT 0010
0012		ACCESSES PORT 0010
0013		ACCESSES PORT 0010
0014		ACCESSES PORT 0010
0015		ACCESSES PORT 0010
0016		ACCESSES PORT 0010
0017		ACCESSES PORT 0010
0018	16	ERROR ADDRESS 2 - READ ONLY
0019		ACCESSES PORT 0018
001A		ACCESSES PORT 0018
001B		ACCESSES PORT 0018
001C		ACCESSES PORT 0018
001D		ACCESSES PORT 0018
001E		ACCESSES PORT 0018
001F		ACCESSES PORT 0018

Port Size Definitions:

- 0 Canbe either 8 or 16 bit access,does not use Data Bus.
- 8 Requires 8 bit access, uses Data Bus 8 - 15.
- 8/16 Can be either 8 or 16 bit access, uses Data Bus 0 - 7 and uses 0 - 15 for 16 bit access. A 16 bit access will access the addressed port and port+1 simultaneously.
- 16 Requires 16 bit access, uses Data Bus 0 - 15.

Table 4-1: I/O PORT ADDRESS DEFINITIONS

ADDR HEX	PORT SIZE	FUNCTION AND COMMENTS
0020	8/16	HARD DISK PORT 1 - DRV & HEAD, LS 8 BIT
0021		DECODED BUT NOT USED
0022	8/16	HARD DISK PORT 1 - DATA / OLD CYL, LS 8
0023		DECODED BUT NOT USED
0024	8/16	HARD DISK PORT 1 - NEW CYL, 2 BYT, LS 8
0025		DECODED BUT NOT USED
0026	8/16	HARD DISK PORT 1 - STATUS / COMD, LS 8
0027		DECODED BUT NOT USED
0028	8/16	HARD DISK PORT 2 - NOT USED
0029		ACCESSES PORT 0028
002A		ACCESSES PORT 0028
002B		ACCESSES PORT 0028
002C		ACCESSES PORT 0028
002D		ACCESSES PORT 0028
002E		ACCESSES PORT 0028
002F		ACCESSES PORT 0028
0030	8/16	USER MODE FLAG, ROM ENABLE, COLD START
0031		ACCESSES PORT 0030
0032		ACCESSES PORT 0030
0033		ACCESSES PORT 0030
0034		ACCESSES PORT 0030
0035		ACCESSES PORT 0030
0036		ACCESSES PORT 0030
0037		ACCESSES PORT 0030
0038	0	I/O CHANNEL ATTEN CH 1 - MASTER
0039	0	I/O CHANNEL ATTEN CH 2 - SLAVE
003A		ACCESSES PORT 0038
003B		ACCESSES PORT 0039
003C		ACCESSES PORT 0038
003D		ACCESSES PORT 0039
003E		ACCESSES PORT 0038
003F		ACCESSES PORT 0039

Port Size Definitions:

- 0 Canbe either 8 or 16 bit access,does not use Data Bus.
- 8 Requires 8 bit access, uses Data Bus 8 - 15.
- 8/16 Can be either 8 or 16 bit access, uses Data Bus 0 - 7 and uses 0 - 15 for 16 bit access. A 16 bit access will access the addressed port and port+1 simultaneously.
- 16 Requires 16 bit access, uses Data Bus 0 - 15.

Table 4-1: I/O PORT ADDRESS DEFINITIONS (CONTINUED)

ADDR HEX	PORT SIZE	FUNCTION AND COMMENTS
0040	8/16	PARALLEL INTERFACE - PORT A DATA R/W
0041	8	FLOPPY CONTROLLER - STATUS / COMD. R/W
0042	8/16	PARALLEL INTERFACE - PORT B DATA R/W
0043	8	FLOPPY CONTROLLER - TRACK REGISTER R/W
0044	8/16	PARALLEL INTERFACE - PORT C DATA R/W
0045	8	FLOPPY CONTROLLER - SECTOR REGISTER R/W
0046	8/16	PARALLEL INTERFACE - CONTROL, WRITE
0047	8	FLOPPY CONTROLLER - DATA REGISTER R/W
0048	8/16	SERIAL CH A DATA R/W - PRINTER #1
0049	8	INTERVAL TIMER - SERIAL CH A BAUD RATE
004A	8/16	SERIAL CH B DATA R/W - CONSOLE #1
004B	8	INTERVAL TIMER - SERIAL CH B BAUD RATE
004C	8/16	SERIAL CH A CONTROL R/W - PRINTER #1
004D	8	INTERVAL TIMER - TIME SLICE TIMER
004E	8/16	SERIAL CH B CONTROL R/W - CONSOLE #1
004F	8	INTERVAL TIMER - CONTROL, WRITE ONLY
0050		DECODED BUT NOT USED
0051	8	ROM PORT HI - STRAPPING OPTIONS, INPUT
0052		DECODED BUT NOT USED
0053	8	ROM PORT HI - FLOPPY SELECT, NMI INHIB.
0054		DECODED BUT NOT USED
0055	8	ROM PORT HI - MISCEL. CONTROL BITS
0056		DECODED BUT NOT USED
0057	8	ROM PORT HI - 8255 CONTROL, WRITE ONLY
0058	8/16	INTERRUPT MASTR. CNTLR. - REQ/SRV REG
0059		CLEAR SYSTEM CALL REQUEST
005A	8/16	INTERRUPT MASTR. CNTLR. - MASK REG
005B		DUPLICATE CLEAR SYSTEM CALL
005C	8/16	DUPLICATE INT. MAST. CNTLR. - REQ/SRV
005D		DUPLICATE CLEAR SYSTEM CALL
005E	8/16	DUPLICATE INT. MAST. CNTLR. - MASK REG
005F		DUPLICATE CLEAR SYSTEM CALL

Port Size Definitions:

- 0 Canbe either 8 or 16 bit access,does not use Data Bus.
- 8 Requires 8 bit access, uses Data Bus 8 - 15.
- 8/16 Can be either 8 or 16 bit access, uses Data Bus 0 - 7 and uses 0 - 15 for 16 bit access. A 16 bit access will access the addressed port and port+1 simultaneously.
- 16 Requires 16 bit access, uses Data Bus 0 - 15.

Table 4-1: I/O PORT ADDRESS DEFINITIONS (CONTINUED)

ADDR HEX	PORT SIZE	FUNCTION AND COMMENTS
0060	8/16	INTERRUPT SLAVE 1 - REQ/SRV REG & CNTL
0061		DECODED BUT NOT USED
0062	8/16	INTERRUPT SLAVE 1 - MASK REG & CNTL
0063		DECODED BUT NOT USED
0064		ACCESSES PORT 0060
0065		DECODED BUT NOT USED
0066		ACCESSES PORT 0062
0067		DECODED BUT NOT USED
0068	8/16	INTERRUPT SLAVE 2 - REQ/SRV REG & CNTL
0069		DECODED BUT NOT USED
006A	8/16	INTERRUPT SLAVE 2 - MASK REG & CNTL
006B		DECODED BUT NOT USED
006C		ACCESSES PORT 0068
006D		DECODED BUT NOT USED
006E		ACCESSES PORT 006A
006F		DECODED BUT NOT USED
0070	16	16 BIT ECC DIAGNOSTIC LATCH
0071		DECODED BUT NOT USED
0072		ACCESSES PORT 0070
0073		ACCESSES PORT 0071
0074		ACCESSES PORT 0070
0075		ACCESSES PORT 0071
0076		ACCESSES PORT 0070
0077		ACCESSES PORT 0071
0078	0	INTELLIGENT I/O CHANNEL ATTN. 0
0079	0	INTELLIGENT I/O CHANNEL ATTN. 1
007A		ACCESSES PORT 0078
007B		ACCESSES PORT 0079
007C		ACCESSES PORT 0078
007D		ACCESSES PORT 0079
007E		ACCESSES PORT 0078
007F		ACCESSES PORT 0079

Port Size Definitions:

- 0 Canbe either 8 or 16 bit access,does not use Data Bus.
- 8 Requires 8 bit access, uses Data Bus 8 - 15.
- 8/16 Can be either 8 or 16 bit access, uses Data Bus 0 - 7 and uses 0 - 15 for 16 bit access. A 16 bit access will access the addressed port and port+1 simultaneously.
- 16 Requires 16 bit access, uses Data Bus 0 - 15.

Table 4-1: I/O PORT ADDRESS DEFINITIONS (CONTINUED)

ADDR HEX	PORT SIZE	FUNCTION AND COMMENTS
0080 TO 00D7		RESERVED FOR EXPANSION
00D8	8/16	TAPE SERIAL DATA - R/W
00D9	8	TAPE CONTROL REGISTER - TRK & UNIT R/W
00DA	8/16	UNUSED SERIAL PORT - DATA - R/W
00DB	8	TAPE CONTROL REGISTER - STATUS READ
00DC	8/16	TAPE SERIAL CONTROL - R/W
00DD	8	TAPE CONTROL REGISTER - DIR. SEL. WRT.
00DE	8/16	UNUSED SERIAL PORT - CONTROL - R/W
00DF	8	TAPE CONTROL REGISTER - CONTROL WRITE
00E0 TO 00FF		RESERVED FOR EXPANSION
0100 TO 01FF		RESERVED FOR MULTIBUS I/O
0200 TO 03FF	16	MEM. MAGR. PORT 1 - 16 BIT R/W USE ONLY EVEN NUMBERED PORTS (EX. 0200, 0202, 0204, ...) (SUB. BLOCK NUMBER WITH LSB 0 FOR LS 9 BITS OF ADDRESS)
0400 TO 05FF	8/16	MEM. MAGR. PORT 2 - 16 BIT R/W USE ONLY EVEN NUMBERED PORTS (EX. 0400, 0402, 0404, ...) (SUB. BLOCK NUMBER WITH LSB 0 FOR LS 9 BITS OF ADDRESS)
0600 TO FFFF		RESERVED FOR MULTIBUS I/O

Port Size Definitions:

- 0 Canbe either 8 or 16 bit access,does not use Data Bus.
- 8 Requires 8 bit access, uses Data Bus 8 - 15.
- 8/16 Can be either 8 or 16 bit access, uses Data Bus 0 - 7 and uses 0 - 15 for 16 bit access. A 16 bit access will access the addressed port and port+1 simultaneously.
- 16 Requires 16 bit access, uses Data Bus 0 - 15.

Table 4-1: I/O PORT ADDRESS DEFINITIONS (CONTINUED)

4.3. System Memory

The block diagram for the System Memory is shown in Figure 3-3 (see Section 3.2, page 7). The logic diagrams for this section are shown on sheets three through ten of the ACS 8600 CPU Board schematic and on sheets one and two of the Expansion Memory Board schematic. The ACS 8600 System Memory is implemented with 64K dynamic random access memory (RAM) devices and 2732 type erasable, programmable read only memory (EPROM) devices. The system is designed to permit up to one megabyte of RAM which may be overlayed by eight kilobytes of EPROM. The RAM memory is organized as sixteen bit words with six extra bits used to support error correction. The RAM memory is divided into eight banks of twenty two devices which provides 128 kilobytes of memory per bank. The four lower banks of the memory are located on the CPU Board for 512 kilobytes of memory and the four upper banks are located on the Expansion Memory Board for the remainder of the one megabyte of system RAM. The RAM memory is supported by three other blocks of logic: the memory controller, the error correction circuitry and the memory manager.

4.3.1. Memory Controller

The Memory Controller allows the word organized memory to be accessed either as words or as bytes, assures proper memory refresh, provides timing for the Error Correction Circuitry and notifies the CPU when memory data is ready. The Memory Controller Block Diagram is shown on the next page and the logic is shown on sheet six of the ACS 8600 CPU Board schematic. Nearly all of this logic is implemented in large scale integrated (LSI) devices. Six programmable logic arrays are used to generate the row address strobe (RAS), the column address strobe (CAS), the refresh timer and the memory sequence control logic. An LSI dynamic RAM controller device is used to multiplex the memory address for the RAM array and to distribute the RAS and CAS to the buffer-drivers used to drive the address and strobe information into the RAM array.

4.3.2. Error Correction Circuitry

The RAM memory of the ACS 8600 is equipped with an LSI device that can detect all single bit and all double bit errors on data read from the RAM array. This device uses a modified Hamming code to generate six parity or check bits for each sixteen bit word written into the memory. When a word is read back from the memory, the associated check bits are also read and compared with a new set of check bits generated from the data read. If both sets of check bits compare then no error has occurred but if the check bits do not compare then one or more bits are in error. The Hamming code is chosen so that the syndrome that is generated by comparing the two sets of check bits can be used to indicate which bit of the sixteen data bits or six check bit is in error. If a data bit is in error it is

Figure 4-2: MEMORY CONTROLLER BLOCK DIAGRAM

corrected by inverting while if a check bit is in error is is simply reported. The syndrome does not carry enough information to permit correction of more than single bit errors however those are the most frequently occurring errors. When ever an error is detected the syndrome and the address associated with the error are latched and a non-maskable interrupt is sent to the CPU. When the CPU services the non-maskable interrupt it can perform a scrubbing operation to attempt to remove the error by rereading and rewriting the reported memory location. The CPU can also log the error for diagnostic and maintenance purposes and indicate if the error was successfully scrubbed. Table 4-2 can be used to decode the syndrome and determine which bit is in error. All syndromes not listed in Table 4-2 are produced by multiple bit errors.

SYNDROME 5 4 3 2 1 0	BIT IN ERROR	PORT 018 HEX BITS 0 - 7
0 0 0 0 0 0	NONE	0C0 HEX
0 0 0 0 0 1	CHECK BIT 0	081 HEX
0 0 0 0 1 0	CHECK BIT 1	082 HEX
0 0 0 1 0 0	CHECK BIT 2	084 HEX
0 0 1 0 0 0	CHECK BIT 3	088 HEX
0 0 1 0 1 1	DATA BIT 1	08B HEX
0 0 1 1 1 0	DATA BIT 0	08E HEX
0 1 0 0 0 0	CHECK BIT 4	090 HEX
0 1 0 0 1 1	DATA BIT 2	093 HEX
0 1 0 1 0 1	DATA BIT 3	095 HEX
0 1 0 1 1 0	DATA BIT 4	096 HEX
0 1 1 0 0 1	DATA BIT 5	099 HEX
0 1 1 0 1 0	DATA BIT 6	09A HEX
0 1 1 1 0 0	DATA BIT 7	09C HEX
1 0 0 0 0 0	CHECK BIT 5	0A0 HEX
1 0 0 0 1 1	DATA BIT 8	0A3 HEX
1 0 0 1 0 1	DATA BIT 9	0A5 HEX
1 0 0 1 1 0	DATA BIT 10	0A6 HEX
1 0 1 0 0 1	DATA BIT 11	0A9 HEX
1 0 1 0 1 0	DATA BIT 12	0AA HEX
1 0 1 1 0 0	DATA BIT 13	0AC HEX
1 1 0 0 0 1	DATA BIT 14	0B1 HEX
1 1 0 1 0 0	DATA BIT 15	0B4 HEX

Table 4-2: ECC SYNDROME DECODED TO BIT IN ERROR
Syndromes not listed are produced by multiple bit errors.

When the memory system is first powered up the check bit memory is not properly initialized for the data memory contents. If memory read operations are performed at this time they will produce invalid memory errors. To properly initialize the error correction circuitry the entire memory must be written. A simple method to initialize the memory is to write zeros to each word location.

For diagnostic and maintenance purposes the error correction circuitry can be placed in a diagnostic mode. This mode is set by setting a one in bit zero of I/O port 055 hex (see Table 4-22, page 62). At power on the state of this bit is a one. This bit is set to zero as part of the I/O port initialization performed by the bootstrap ROM. When bit zero of I/O port 055 hex is a one the contents of the diagnostic latch contained on the error detection and correction chip override the hardwired mode control pins of the chip. The meanings of the various bits of the diagnostic latch are given below in Table 4-3 while the various modes of operation are given on page 25 in Table 4-4. Memory diagnostic software can take advantage of these capabilities to validate the functionality of the error detection and correction chip.

BIT	BIT NAME	TRUE
0	DIAGNOSTIC CHECK BIT 0	1
1	DIAGNOSTIC CHECK BIT 1	1
2	DIAGNOSTIC CHECK BIT 2	1
3	DIAGNOSTIC CHECK BIT 3	1
4	DIAGNOSTIC CHECK BIT 4	1
5	DIAGNOSTIC CHECK BIT 5	1
6	DON'T CARE (bit not used)	
7	DON'T CARE (bit not used)	
8	CODE ID BIT 0	1
9	CODE ID BIT 1	1
10	CODE ID BIT 2	1
11	DIAGNOSTIC MODE CONTROL BIT 0	1
12	DIAGNOSTIC MODE CONTROL BIT 1	1
13	CORRECT MODE CONTROL BIT	1
14	PASS THRU MODE CONTROL BIT	1
15	DON'T CARE (bit not used)	

Table 4-3: ECC DIAGNOSTIC LATCH BIT DEFINITIONS

When the error detection and correction chip (EDC) is in diagnostic mode, the operation of the chip can be controlled by the Diagnostic Mode Control bits (diagnostic latch bits eleven and twelve). The state of these bits is tabulated in Table 4-4, on the next page, by the column labeled "D M". The "GEN PIN" referenced in Table 4-4 is the Generate Pin of the error detection and correction chip. This pin is driven low by the hardware during memory writes and causes the chip to generate check bits which are sent to the memory. When this pin is high the chip checks the data and check bits read from the memory. The column labeled "COR" refers to the state of the Correct Mode Control bit (diagnostic latch bit thirteen) listed in Table 4-3. This bit determines if errors are corrected and reported, or only reported. The three Code ID bits (diagnostic latch bits eight through ten) should always be zero since this indicates that the EDC should operate in sixteen bit mode. When the Pass Thru bit (diagnostic latch bit fourteen) is high the current contents of the check bit latch are output to the memory and the current contents of the input data latch are placed in the data output latch. This mode does not appear to have much practical use in this application.

The error detection and correction chip (EDC) can be placed in any one of four diagnostic modes. Mode zero causes the chip to operate under software control in the same manner as it would if it were not in diagnostic mode. When the Correct Mode Control bit is zero the EDC generates check bits on memory writes and detects errors on memory reads but does not correct the errors. When the Correct Mode bit is one the EDC generates check bits on memory writes and detects and corrects errors on memory reads. In mode one the diagnostic latch check bits are substituted for the check bits normally generated (to be written into memory) while either normal detection or normal correction is performed on memory reads. Correction or detection depends on the state of the correct mode bit in the same manner as in mode zero. In mode two the diagnostic latch check bits are substituted for the check bits read from memory but normal generation is performed during memory writes. In mode two as in modes zero and one, the Correct Mode bit determines if the errors that may be detected by comparing the data read from memory and the check bits from the diagnostic latch are corrected and reported, or just reported. Mode three is the initialize mode. The initialize mode forces the data outputs of the error detection and correction controller to be zero and the check bit outputs to be correct for zero data. This mode may not be used to initialize the memory array because during write operations the error detection and correction controller is not permited to output any data other than the check bits.

D M 1	D M 0	COR	GEN PIN	FUNCTION
0	0	0	0	GENERATE AS IF NOT IN DIAG. MODE
0	0	0	1	DETECT AS IF NOT IN DIAG. MODE
0	0	1	0	GENERATE AS IF NOT IN DIAG. MODE
0	0	1	1	CORRECT AS IF NOT IN DIAG. MODE
0	1	0	0	GENERATE USING DIAGNOSTIC LATCH
0	1	0	1	DETECT AS IF NOT IN DIAG. MODE
0	1	1	0	GENERATE USING DIAGNOSTIC LATCH
0	1	1	1	CORRECT AS IF NOT IN DIAG. MODE
1	0	0	0	GENERATE AS IF NOT IN DIAG. MODE
1	0	0	1	DETECT USING DIAGNOSTIC LATCH
1	0	1	0	GENERATE AS IF NOT IN DIAG. MODE
1	0	1	1	CORRECT USING DIAGNOSTIC LATCH
1	1	X	X	INITIALIZE

Table 4-4: DIAGNOSTIC LATCH OPERATING MODES

4.3.3. Memory Manager Section

The purpose of memory management is to provide both position independence and protection for the contents of memory. The Altos ACS 8600 uses an Intel 8086 microprocessor as the computing and controlling element. This processor provides position independence through the use of segment registers which allows code that is designed to run in one segment to be relocated into other segments and still run correctly. Four types of memory are defined (CODE, DATA, STACK and EXTRA DATA) and a segment register is provided for each memory type. The 8086 generates addresses which allow it to access one million (1,048,576) bytes of memory, however this memory must be accessed in blocks of up to 64K (65,536) bytes. The address is generated by taking the sixteen bit segment number, appending four bits of zero to the least significant bit position and adding the resulting twenty bit quantity to the sixteen bit offset specified in the instruction. With this approach memory blocks may be established on any sixteen byte boundry but must consist of contiguous memory.

Limited protection is provided by the fact that the memory beyond the current four 64K segments cannot be accessed without altering the segment register. No limit checking or segment overlap protection is provided by the processor. Further more, any program may access the segment registers at any time.

The ACS 8600 extends the memory management facilities of the 8086 by supporting write protection, access protection, limit checking and non-contiguous memory allocation. The one million bytes of main memory are managed as 256 pages of 4K (4,096) bytes. Each page is given a relocation value, and thirteen flag and control bits. As shown in Figure 4-3 on the next page,

Figure 4-3: MEMORY ADDRESS GENERATION IN THE ACS 8600

this memory relocation and attribute data is stored in a 256 by 21 bit RAM. This RAM is addressed by the most significant eight bits of the address generated by the 8086 CPU. When a memory access is made the eight bit relocation value is substituted for the eight most significant bits of the address generated by the 8086 to form a new 20 bit physical address. At the same time the twelve attribute bits are examined to test the validity of the access. The remaining bit provide status information to the operating system software. This bit is a read-only bit that indicates that the memory page has had a memory write performed upon it since the memory manager was last updated. The operating system can use this bit to decide if the associated memory page should be written out to disk to update the disk copy.

In addition to the four types of memory implemented by the 8086, the ACS 8600 provides identification of four memory users -- Operating System, Application, Direct Memory Access (DMA) and Expansion. Each page of memory can be protected from access (either read or write) through any combination of 8086 segment registers, from access (either read or write) by each of the four memory users and from write access by each of the four memory users. For example, the Memory Manager has provisions for write protecting each memory page against Operating System Mode memory writes thus making the page read only. The same page can be marked not to allow any access by an Application Mode program. All writes performed through the stack segment register are limit checked to detect the presents of impending stack overflow. If the page being accessed is marked as a Stack Boundary Segment and if a write is performed in the lowest 128 bytes of the 4,096 byte block then an overflow is flagged and a non-maskable interrupt is generated.

Finally the memory manager implements the concept of privileged instructions. The purpose of privileged instructions is to prevent application programs from altering the system environment. The ACS 8600 treats all I/O instructions and interrupt disabling instructions as privileged instructions. By definition an application program must run with interrupts enabled and must make all I/O access requests through the operating system. Therefore the memory manager will generate a non-maskable interrupt if it detects the execution of an instruction which resets the interrupt enable flip-flop while an application program is running. In addition all I/O writes which are not performed by the 8086 running in System Mode or by the 8089 are inhibited. Any I/O instruction executed by programs run in Application Mode is interpreted as a System Call. Programs executed by bus masters located on the Expansion bus may not access I/O ports internal to the ACS 8600 system. The 8086 CPU cannot abort an instruction and restore the previous machine state, therefore it has difficulty dealing with run-time errors such as memory faults, access violations and invalid operations. Since there is no way to abort the instruction, the instruction is executed and the CPU is then interrupted by the non-maskable interrupt. Due to the timing requirements of the 8086, one additional instruction will be executed after the instruction

that generates the violation before the non-maskable interrupt can be processed. However, all I/O accesses other than that required to generate a System Call are inhibited as well as all memory writes attempted on write protected memory pages.

Operating System Mode and Application Mode are controlled by a flip-flop which is accessed as bit zero of I/O ports 030 hex. Setting this bit to a one requests that Application Mode be set as soon as interrupts are enabled. If this request is generated while the interrupts are enabled, Application Mode is set immediately. The intended application is for the operating system to request application mode just prior to restoring the registers of the application program and returning to the execution of the application. The Application Mode flip-flop is reset to Operating System Mode by reset and by the occurrence of a System Call, a non-maskable interrupt or an interrupt acknowledge. These events indicate that the application is turning control over to the operating system and therefore the Application Mode must be cleared. DMA and Expansion Modes are set by the bus arbitration logic whenever the bus is turned over to another master. The system can be in only one mode at any one time and the memory manager automaticly changes the attribute checking logic to make certain the memory is protected.

The 256 by 21 bit RAM portion of the memory management hardware is accessed through 512 sixteen-bit read-write I/O ports while the error and status registers of the memory management hardware are accessed through three sixteen-bit read-only I/O ports. The 512 I/O ports are the even numbered ports from 0200 hex through 05FF hex and must be accessed by I/O instructions using register indirect addressing (through register DX). The memory manager hardware uses I/O port address bits one through eight (inclusive) in the same manner as it uses memory address bits twelve through nineteen (inclusive). As an example, an I/O access to port 0268 hex will read or write the attributes of memory page thirty-four hex (memory locations 034000 hex through 34FFF hex). Table 4-5 on pages 29 through 36 give the relationship between main memory pages and the I/O ports of the memory manager. The three status ports may be accessed by I/O instructions that use either register indirect addressing or immediate addressing. Table 4-6 through Table 4-10 on the following pages provide detailed information about the function of each bit in each of the memory manager I/O ports.

Table 4-6 on page 38, describe the thirteen attribute bits associated with each memory page. These bits are accessed though the even numbered I/O ports from 0200 hex to 03FE hex. Bits zero, one, two and four are ignored on a data write. On I/O read bit zero will be a one if the system is currently in Application (User) Mode. Bit one of ports 0200 hex through 03FE hex can be read to determine if the read only memory (ROM) is mapped into the memory or disabled. A one in this bit position indicates that the ROM is mapped into memory and may be used. Bit four is always forced to zero on I/O read. Bit two is ignored on a data write and is set to a one if the addressed memory page has had a

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
00	00000-00FFF	0200	0400
01	01000-01FFF	0202	0402
02	02000-02FFF	0204	0404
03	03000-03FFF	0206	0406
04	04000-04FFF	0208	0408
05	05000-05FFF	020A	040A
06	06000-06FFF	020C	040C
07	07000-07FFF	020E	040E
08	08000-08FFF	0210	0410
09	09000-09FFF	0212	0412
0A	0A000-0AFFF	0214	0414
0B	0B000-0BFFF	0216	0416
0C	0C000-0CFFF	0218	0418
0D	0D000-0DFFF	021A	041A
0E	0E000-0EFFF	021C	041C
0F	0F000-0FFFF	021E	041E
10	10000-10FFF	0220	0420
11	11000-11FFF	0222	0422
12	12000-12FFF	0224	0424
13	13000-13FFF	0226	0426
14	14000-14FFF	0228	0428
15	15000-15FFF	022A	042A
16	16000-16FFF	022C	042C
17	17000-17FFF	022E	042E
18	18000-18FFF	0230	0430
19	19000-19FFF	0232	0432
1A	1A000-1AFFF	0234	0434
1B	1B000-1BFFF	0236	0436
1C	1C000-1CFFF	0238	0438
1D	1D000-1DFFF	023A	043A
1E	1E000-1EFFF	023C	043C
1F	1F000-1FFFF	023E	043E

Table 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
20	20000-20FFF	0240	0440
21	21000-21FFF	0242	0442
22	22000-22FFF	0244	0444
23	23000-23FFF	0246	0446
24	24000-24FFF	0248	0448
25	25000-25FFF	024A	044A
26	26000-26FFF	024C	044C
27	27000-27FFF	024E	044E
28	28000-28FFF	0250	0450
29	29000-29FFF	0252	0452
2A	2A000-2AFFF	0254	0454
2B	2B000-2BFFF	0256	0456
2C	2C000-2CFFF	0258	0458
2D	2D000-2DFFF	025A	045A
2E	2E000-2EFFF	025C	045C
2F	2F000-2FFFF	025E	045E
30	30000-30FFF	0260	0460
31	31000-31FFF	0262	0462
32	32000-32FFF	0264	0464
33	33000-33FFF	0266	0466
34	34000-34FFF	0268	0468
35	35000-35FFF	026A	046A
36	36000-36FFF	026C	046C
37	37000-37FFF	026E	046E
38	38000-38FFF	0270	0470
39	39000-39FFF	0272	0472
3A	3A000-3AFFF	0274	0474
3B	3B000-3BFFF	0276	0476
3C	3C000-3CFFF	0278	0478
3D	3D000-3DFFF	027A	047A
3E	3E000-3EFFF	027C	047C
3F	3F000-3FFFF	027E	047E

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
40	40000-40FFF	0280	0480
41	41000-41FFF	0282	0482
42	42000-42FFF	0284	0484
43	43000-43FFF	0286	0486
44	44000-44FFF	0288	0488
45	45000-45FFF	028A	048A
46	46000-46FFF	028C	048C
47	47000-47FFF	028E	048E
48	48000-48FFF	0290	0490
49	49000-49FFF	0292	0492
4A	4A000-4AFFF	0294	0494
4B	4B000-4BFFF	0296	0496
4C	4C000-4CFFF	0298	0498
4D	4D000-4DFFF	029A	049A
4E	4E000-4EFFF	029C	049C
4F	4F000-4FFFF	029E	049E
50	50000-50FFF	02A0	04A0
51	51000-51FFF	02A2	04A2
52	52000-52FFF	02A4	04A4
53	53000-53FFF	02A6	04A6
54	54000-54FFF	02A8	04A8
55	55000-55FFF	02AA	04AA
56	56000-56FFF	02AC	04AC
57	57000-57FFF	02AE	04AE
58	58000-58FFF	02B0	04B0
59	59000-59FFF	02B2	04B2
5A	5A000-5AFFF	02B4	04B4
5B	5B000-5BFFF	02B6	04B6
5C	5C000-5CFFF	02B8	04B8
5D	5D000-5DFFF	02BA	04BA
5E	5E000-5EFFF	02BC	04BC
5F	5F000-5FFFF	02BE	04BE

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
60	60000-60FFF	02C0	04C0
61	61000-61FFF	02C2	04C2
62	62000-62FFF	02C4	04C4
63	63000-63FFF	02C6	04C6
64	64000-64FFF	02C8	04C8
65	65000-65FFF	02CA	04CA
66	66000-66FFF	02CC	04CC
67	67000-67FFF	02CE	04CE
68	68000-68FFF	02D0	04D0
69	69000-69FFF	02D2	04D2
6A	6A000-6AFFF	02D4	04D4
6B	6B000-6BFFF	02D6	04D6
6C	6C000-6CFFF	02D8	04D8
6D	6D000-6DFFF	02DA	04DA
6E	6E000-6EFFF	02DC	04DC
6F	6F000-6FFFF	02DE	04DE
70	70000-70FFF	02E0	04E0
71	71000-71FFF	02E2	04E2
72	72000-72FFF	02E4	04E4
73	73000-73FFF	02E6	04E6
74	74000-74FFF	02E8	04E8
75	75000-75FFF	02EA	04EA
76	76000-76FFF	02EC	04EC
77	77000-77FFF	02EE	04EE
78	78000-78FFF	02F0	04F0
79	79000-79FFF	02F2	04F2
7A	7A000-7AFFF	02F4	04F4
7B	7B000-7BFFF	02F6	04F6
7C	7C000-7CFFF	02F8	04F8
7D	7D000-7DFFF	02FA	04FA
7E	7E000-7EFFF	02FC	04FC
7F	7F000-7FFFF	02FE	04FE

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
80	80000-80FFF	0300	0500
81	81000-81FFF	0302	0502
82	82000-82FFF	0304	0504
83	83000-83FFF	0306	0506
84	84000-84FFF	0308	0508
85	85000-85FFF	030A	050A
86	86000-86FFF	030C	050C
87	87000-87FFF	030E	050E
88	88000-88FFF	0310	0510
89	89000-89FFF	0312	0512
8A	8A000-8AFFF	0314	0514
8B	8B000-8BFFF	0316	0516
8C	8C000-8CFFF	0318	0518
8D	8D000-8DFFF	031A	051A
8E	8E000-8EFFF	031C	051C
8F	8F000-8FFFF	031E	051E
90	90000-90FFF	0320	0520
91	91000-91FFF	0322	0522
92	92000-92FFF	0324	0524
93	93000-93FFF	0326	0526
94	94000-94FFF	0328	0528
95	95000-95FFF	032A	052A
96	96000-96FFF	032C	052C
97	97000-97FFF	032E	052E
98	98000-98FFF	0330	0530
99	99000-99FFF	0332	0532
9A	9A000-9AFFF	0334	0534
9B	9B000-9BFFF	0336	0536
9C	9C000-9CFFF	0338	0538
9D	9D000-9DFFF	033A	053A
9E	9E000-9EFFF	033C	053C
9F	9F000-9FFFF	033E	053E

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
A0	A0000-A0FFF	0340	0540
A1	A1000-A1FFF	0342	0542
A2	A2000-A2FFF	0344	0544
A3	A3000-A3FFF	0346	0546
A4	A4000-A4FFF	0348	0548
A5	A5000-A5FFF	034A	054A
A6	A6000-A6FFF	034C	054C
A7	A7000-A7FFF	034E	054E
A8	A8000-A8FFF	0350	0550
A9	A9000-A9FFF	0352	0552
AA	AA000-AAFFF	0354	0554
AB	AB000-ABFFF	0356	0556
AC	AC000-ACFFF	0358	0558
AD	AD000-ADFFF	035A	055A
AE	AE000-AEFFF	035C	055C
AF	AF000-AFFFF	035E	055E
B0	B0000-B0FFF	0360	0560
B1	B1000-B1FFF	0362	0562
B2	B2000-B2FFF	0364	0564
B3	B3000-B3FFF	0366	0566
B4	B4000-B4FFF	0368	0568
B5	B5000-B5FFF	036A	056A
B6	B6000-B6FFF	036C	056C
B7	B7000-B7FFF	036E	056E
B8	B8000-B8FFF	0370	0570
B9	B9000-B9FFF	0372	0572
BA	BA000-BAFFF	0374	0574
BB	BB000-BBFFF	0376	0576
BC	BC000-BCFFF	0378	0578
BD	BD000-BDFFF	037A	057A
BE	BE000-BEFFF	037C	057C
BF	BF000-BFFFF	037E	057E

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
C0	C0000-C0FFF	0380	0580
C1	C1000-C1FFF	0382	0582
C2	C2000-C2FFF	0384	0584
C3	C3000-C3FFF	0386	0586
C4	C4000-C4FFF	0388	0588
C5	C5000-C5FFF	038A	058A
C6	C6000-C6FFF	038C	058C
C7	C7000-C7FFF	038E	058E
C8	C8000-C8FFF	0390	0590
C9	C9000-C9FFF	0392	0592
CA	CA000-CAFFF	0394	0594
CB	CB000-CBFFF	0396	0596
CC	CC000-CCFFF	0398	0598
CD	CD000-CDFFF	039A	059A
CE	CE000-CEFFF	039C	059C
CF	CF000-CFFFF	039E	059E
D0	D0000-D0FFF	03A0	05A0
D1	D1000-D1FFF	03A2	05A2
D2	D2000-D2FFF	03A4	05A4
D3	D3000-D3FFF	03A6	05A6
D4	D4000-D4FFF	03A8	05A8
D5	D5000-D5FFF	03AA	05AA
D6	D6000-D6FFF	03AC	05AC
D7	D7000-D7FFF	03AE	05AE
D8	D8000-D8FFF	03B0	05B0
D9	D9000-D9FFF	03B2	05B2
DA	DA000-DAFFF	03B4	05B4
DB	DB000-DBFFF	03B6	05B6
DC	DC000-DCFFF	03B8	05B8
DD	DD000-DDFFF	03BA	05BA
DE	DE000-DEFFF	03BC	05BC
DF	DF000-DFFFF	03BE	05BE

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

BLOCK NUMBER	ADDRESS RANGE	ATTRIBUTE PORT	PHYSICAL ADDR. PORT
E0	E0000-E0FFF	03C0	05C0
E1	E1000-E1FFF	03C2	05C2
E2	E2000-E2FFF	03C4	05C4
E3	E3000-E3FFF	03C6	05C6
E4	E4000-E4FFF	03C8	05C8
E5	E5000-E5FFF	03CA	05CA
E6	E6000-E6FFF	03CC	05CC
E7	E7000-E7FFF	03CE	05CE
E8	E8000-E8FFF	03D0	05D0
E9	E9000-E9FFF	03D2	05D2
EA	EA000-EAFFF	03D4	05D4
EB	EB000-EBFFF	03D6	05D6
EC	EC000-ECFFF	03D8	05D8
ED	ED000-EDFFF	03DA	05DA
EE	EE000-EEFFF	03DC	05DC
EF	EF000-EFFFF	03DE	05DE
F0	F0000-F0FFF	03E0	05E0
F1	F1000-F1FFF	03E2	05E2
F2	F2000-F2FFF	03E4	05E4
F3	F3000-F3FFF	03E6	05E6
F4	F4000-F4FFF	03E8	05E8
F5	F5000-F5FFF	03EA	05EA
F6	F6000-F6FFF	03EC	05EC
F7	F7000-F7FFF	03EE	05EE
F8	F8000-F8FFF	03F0	05F0
F9	F9000-F9FFF	03F2	05F2
FA	FA000-FAFFF	03F4	05F4
FB	FB000-FBFFF	03F6	05F6
FC	FC000-FCFFF	03F8	05F8
FD	FD000-FDFFF	03FA	05FA
FE	FE000-FEFFF	03FC	05FC
FF	FF000-FFFFF	03FE	05FE

TABLE 4-5: MEMORY MANAGER PORTS AND MEMORY ADDRESSES (CONT.)

memory write performed on it since the attribute port was last written. This bit can be useful to software which wishes to save back to disk only those memory pages which have been changed. Bit three is set by the software to indicate that the addressed memory page is the lowest four kilobytes of the application stack space. When the hardware detects a memory write to the lowest 128 bytes of a memory page that is marked as a stack bound segment it generates a non-maskable interrupt to warn the operating system that the application is about to overflow the stack.

Bits five through eight of I/O ports 0200 hex through 03FE hex control the ability of various "processors" to write into the addressed memory page. When bit five is set to a one, bus masters which are on the Multibus interface are allowed to write into the associated memory page. If bit five is a zero the attempt to write is inhibited by the hardware and a non-maskable interrupt is generated. Bit six is similar to bit five in that it must be set to a one to permit the direct memory access controller and I/O processor (8089) to write into the indicated memory page. If a write is attempted while the bit is zero it will be inhibited and a non-maskable interrupt will be generated. Bit seven controls the ability of the 8086 CPU to write into the indicated memory page while running an application program. The numeric data processor (8087) is always considered to be part of the 8086 running in application mode. Therefore bit seven also controls the ability of the 8087 to write into the indicated memory page regardless of the state of the Application (User) Mode flip-flop. This bit operates in the same manner as bits five and six. Bit eight of I/O ports 0200 hex through 03FE hex controls the ability of the 8086 CPU to write into the indicated memory page when Operating System Mode is set and functions in the same manner as bits five, six and seven.

Bits nine, ten and eleven control the ability of the various processors to read the selected page of memory in a manner similar to bits five six, seven and eight. For example, if bit nine is a zero for the selected memory page then a non-maskable interrupt will be generated each time a bus master on the Multibus interface attempts to read from that memory. The memory access is not inhibited as is the case with memory writes. The 8086 CPU running in Operating System Mode is allowed read access to any page of memory at all times.

Bits twelve, thirteen, fourteen and fifteen control the ability of the 8086 CPU to read or write the selected page of memory through the various segment registers while running in application mode. Here the status bits generated by the 8086 are decoded and compared against these bits to determine if the access is allowed. Unless writes are inhibited by bit seven the access is made and a non-maskable interrupt is generated if the attributes are violated. For example, a non-maskable interrupt will be generated if the 8086 attempted to fetch opcodes from the selected memory page and bit fifteen of the associated memory manager attribute register is zero. These bits permit the

BIT	BIT NAME	TRUE
0	User Mode (read only)	1
1	ROM Enable (read only)	1
2	BLOCK WRITTEN (read only)	1
3	USER STACK BOUND SEGMENT	1
4	not used, forced to 0 on read	
5	ALLOW EXP. PROC. WRITE	1
6	ALLOW DMA - MATH PROC. WRITE	1
7	ALLOW USER WRITE	1
8	ALLOW SYSTEM WRITE	1
9	ALLOW EXP. PROC. ACCESS	1
10	ALLOW DMA - MATH PROC. ACCESS	1
11	ALLOW USER ACCESS	1
12	ALLOW EXTRA DATA ACCESS	1
13	ALLOW STACK ACCESS	1
14	ALLOW DATA ACCESS	1
15	ALLOW CODE ACCESS	1

Table 4-6: EVEN I/O PORTS 0200 HEX THROUGH 03FE HEX

operating system to define pages of memory as execute-only memory or as data memory.

The function of the memory manager attribute checking logic can be disabled for memory initialization purposes by writing 0FFF0 hex into all of the even numbered I/O ports from 0200 hex to 03FF hex. This allows all processors and all processor modes to read and write any location in memory. The value 0FFF0 hex does not define any segment as a stack boundary segment.

Table 4-7, on the next page, describe the use of the bits in the even I/O ports from 0400 hex to 05FF hex. These ports are used to set the eight bit relocation value for each page of memory. Since only the eight least significant bits are used these ports may be accessed as either eight or sixteen bit ports. When they are read as sixteen bit ports, the most significant eight bits of the port are undefined and may take on any value. The memory relocation capability can be disabled during memory initialization by writing the memory page number into each relocation I/O port. For example, writing 034 hex into I/O port 0468 hex will cause access to logical memory page 34 (034000 hex through 034FFF) to be relocated to physical memory page 34 (034000 hex through 034FFF).

The violation port is the first of the three error reporting registers supported by the memory manager and is described in Table 4-8 on page 42. This sixteen bit register is read through I/O port 00 hex and indicates the reason for the non-maskable interrupt. After processing the non-maskable interrupt the software should output to or input from I/O port 08 hex to clear the violation port. It is the act of accessing port 08 hex that clears the violation port, all data transfers with port 08 hex are ignored.

Bit zero of the violation port, port 00 hex, will be a zero and a non-maskable interrupt will be generated if the status signals generated by the 8086 while it is in application mode indicate that interrupts have become disabled.

Violation port bit one will be set to a zero in the event that the memory error correction circuitry detects a memory error. In addition, bit two will be set to a zero if the detected memory error was an uncorrectable multi-bit memory error. Bit three of I/O port 00 hex is set to a zero when the memory manager detects a memory write to the lowest 128 bytes of a user stack boundary memory segment page.

Bits four through seven of I/O port 00 hex are set to zero when the memory manager detects a violation of the one of the four write protection attributes. Bit four is a zero if the 8086 CPU violates the write protection attribute while in Operating System Mode. Bit five is a zero if a processor on the Multibus interface has attempted a write on a protected page. Bit six is a zero if the write violation was attempted by the 8089 (DMA - I/O Processor). And bit seven is a zero if the 8086 while

BIT	BIT NAME	TRUE
0	PHYSICAL ADDRESS 12	1
1	PHYSICAL ADDRESS 13	1
2	PHYSICAL ADDRESS 14	1
3	PHYSICAL ADDRESS 15	1
4	PHYSICAL ADDRESS 16	1
5	PHYSICAL ADDRESS 17	1
6	PHYSICAL ADDRESS 18	1
7	PHYSICAL ADDRESS 19	1
8	not used, undefined	
9	not used, undefined	
10	not used, undefined	
11	not used, undefined	
12	not used, undefined	
13	not used, undefined	
14	not used, undefined	
15	not used, undefined	

Table 4-7: EVEN I/O PORTS 0400 HEX THROUGH 05FF HEX

running in application mode or the 8087 (Numeric Data Processor) have attempted to violate the memory write protection attribute.

Bit eight of the violation port is set to a zero when there has been no bus activity for approximately ten microseconds. This condition can occur if an I/O access is made to a nonexistent port and no ready is generated back to the 8086 CPU. If a time out occures a ready is generated to allow the processor to continue running and violation port bit eight is set to a zero to cause a non-maskable interrupt.

Bits nine through fifteen are set to zero in the event the memory manager detects various processor or memory type access violations. Bits nine through eleven are processor dependent while bits eleven through fifteen are memory type dependent. For example, bit ten will be set to zero if the 8089 (DMA - I/O processor) attempts to access memory it is not supposed to access (bit ten of the attribute register for the selected memory page - ports 0200 hex through 03FE hex not set to a one). And bit fourteen of I/O port 00 hex is set to zero if a memory access was attempted on a memory page which has attributes that do not allow data access and the data segment register of the 8086 CPU was used while the 8086 CPU is in application mode.

Tables 4-9 and 4-10, on the next two pages detail the significance of the bits in the two I/O ports used to report the address involved in the bus cycle which generated the non-maskable interrupt. The specific error is reported through the violation port described above. I/O port 010 hex contains the least significant sixteen bits of the last address strobed onto the CPU address bus prior to a condition which generated a non-maskable interrupt. I/O port 018 hex contains the most significant four bits of the error address in bits twelve through fifteen. Bits zero through five of I/O port 018 hex contain the syndrome bits generated by the memory error correction circuitry. The type of memory error is reported on bits six and seven of I/O port 018 hex and is a duplicate of the information reported on bits one and two of the violation port (I/O port 00 hex). Bit eight indicates the current state of the Application (User) Mode circuitry and is a duplicate of bit zero of ports 0200 hex through 03FE hex. Bit nine is a one when ever the read only memory (ROM) is enabled. Bit ten is forced to a zero at all times. And bit eleven reflects the state of the warm start latch. The functioning of the warm start latch is described as part of I/O port 030 hex (see page 45).

Normally the violation port will show only one error and the error address ports will contain the address at which the error occurred. However, the error address ports are simple registers that contain the address generated by the current bus master and are updated on each address latch enable generated by the bus controller supporting the current bus master. The updating process is stopped when an error is detected. The corresponding bit of the violation port is set to indicate the error and a non-maskable interrupt is generated. If another error should occur

BIT	BIT NAME	TRUE
0	INVALID OP-	0
1	LATCHED MEMORY ERROR	0
2	LATCHED MULTI-BIT MEMORY ERROR	0
3	END OF STACK-	0
4	SYSTEM WRITE VIOLATION-	0
5	EXP. PROC. WRITE VIOLATION-	0
6	DMA PROC. WRITE VIOLATION-	0
7	USER WRITE VIOLATION-	0
8	LATCHED BUS ACCESS TIME OUT ERROR-	0
9	EXP. PROC. ACCESS VIOLATION-	0
10	DMA PROC. ACCESS VIOLATION-	0
11	USER ACCESS VIOLATION-	0
12	EXTRA DATA ACCESS VIOLATION-	0
13	STACK ACCESS VIOLATION-	0
14	DATA ACCESS VIOLATION-	0
15	CODE ACCESS VIOLATION-	0

Table 4-8: READ FROM I/O PORT 00 HEX - VIOLATION PORT

BIT	BIT NAME	TRUE
0	MEMORY ERROR ADDRESS 0	1
1	MEMORY ERROR ADDRESS 1	1
2	MEMORY ERROR ADDRESS 2	1
3	MEMORY ERROR ADDRESS 3	1
4	MEMORY ERROR ADDRESS 4	1
5	MEMORY ERROR ADDRESS 5	1
6	MEMORY ERROR ADDRESS 6	1
7	MEMORY ERROR ADDRESS 7	1
8	MEMORY ERROR ADDRESS 8	1
9	MEMORY ERROR ADDRESS 9	1
10	MEMORY ERROR ADDRESS 10	1
11	MEMORY ERROR ADDRESS 11	1
12	MEMORY ERROR ADDRESS 12	1
13	MEMORY ERROR ADDRESS 13	1
14	MEMORY ERROR ADDRESS 14	1
15	MEMORY ERROR ADDRESS 15	1

Table 4-9: READ FORM I/O PORT 010 HEX - ERROR ADDRESS 1

BIT	BIT NAME	TRUE
0	SYNDROME BIT 0	1
1	SYNDROME BIT 1	1
2	SYNDROME BIT 2	1
3	SYNDROME BIT 3	1
4	SYNDROME BIT 4	1
5	SYNDROME BIT 5	1
6	MEMORY ERROR-	0
7	MULTI-BIT MEMORY ERROR-	0
8	APPLICATION (USER) MODE	1
9	ROM ENABLE	1
10	not used, forced to 0	
11	WARM START FLAG (0=COLD, 1=WARM)	1
12	MEMORY ERROR ADDRESS 16	1
13	MEMORY ERROR ADDRESS 17	1
14	MEMORY ERROR ADDRESS 18	1
15	MEMORY ERROR ADDRESS 19	1

Table 4-10: READ FROM I/O PORT 018 HEX - ERROR ADDRESS 2

before the updating process can be restarted by the clearing of the violation port by the non-maskable interrupt routine, the address of the second error will be lost and the violation port will show that multiple errors have occurred. The most likely cases for multiple errors are catastrophic memory failure yielding a memory error on every access and bus masters other than the 8086 CPU (the 8087, 8089 or Multibus expansion master) which make access that generate multiple errors before turning the bus back over to the 8086 CPU. It should be noted that since bus masters on the Multibus expansion do not generate address latch enable pulses, the address of violations caused by expansion bus masters will not be captured.

Port 030 hex is a write only port used to control three special latches in the ACS 8600. Bit zero is used to access the Application (User) Mode circuitry. Setting this bit to a one requests that Application Mode be set as soon as interrupts are enabled. If interrupts are enabled when this request is generated, Application Mode is set immediately. The intended application is for the operating system to request application mode just prior to returning to the execution of an application program. The Application Mode flip-flop is reset to Operating System Mode by a reset, a System Call, a non-maskable interrupt or an interrupt acknowledge. These events indicate that control is being turned over to the operating system and therefore the Application Mode must be cleared. Bit one is used to access the read only memory (ROM) enable latch. When this bit is set the ROM is mapped over the top most 8K (8,192) bytes of memory (from 0FE000 hex to 0FFFFF hex). Writing a zero into bit one of port 030 hex will cause the ROM to be inaccessible. Any random access memory (RAM) in the address range of 0FE000 hex to 0FFFFF hex will be accessed in place of the ROM. Bit eight is used to access the warm start latch. This latch is reset by power-on reset and not by the front panel reset. This allows the ROM software to distinguish between power-on reset and manual reset. At power-on the error correction memory must be initialized but at manual reset no memory initialization is performed so that error recovery memory dumps can be taken. Bit eight may be written to a zero before jumping to the start of the ROM to force a software controlled reinitialization of main memory.

BIT	BIT NAME	TRUE
0	APPLICATION (USER) MODE - WRITE ONLY	1
1	ROM ENABLE - WRITE ONLY	1
2	(not used - ignored)	
3	(not used - ignored)	
4	(not used - ignored)	
5	(not used - ignored)	
6	(not used - ignored)	
7	(not used - ignored)	
8	WARM START FLAG - WRITE ONLY	1
9	(not used - ignored)	
10	(not used - ignored)	
11	(not used - ignored)	
12	(not used - ignored)	
13	(not used - ignored)	
14	(not used - ignored)	
15	(not used - ignored)	

Table 4-11: WRITE TO I/O PORT 030 HEX - SPECIAL LATCHES
Write only -- see ports 18 and 200 through 3FE for reading.

4.3.4. Expansion Memory

The expansion memory is a daughter board that can be connected to the ACS 8600 CPU board to provide up to 512 kilobytes of additional memory. The logic diagrams for the memory expansion board are shown on sheets one and two of the Expansion Memory Board schematic. The memory is implemented with 64K dynamic random access memory (RAM) devices. This daughter board contains 88 dynamic memory devices organized as four banks of 22 devices. This provides a sixteen bit word with six check bits for error correction and detection. All timing, address and data signals are generated on the CPU board and passed to the daughter board where four buffer-driver devices drive the expansion memory array. Error detection and correction is performed on the CPU board.

4.3.5. Boot Memory

The ACS 8600 system contains eight kilobytes of read only memory primarily intended for bootstrapping purposes. This memory is currently implemented with two 2732 type erasable programmable read only memory (EPROM) devices. However the hardware can be strapped to accommodate 2716 type devices if less read-only memory is required. The circuitry involved is depicted on sheets nine and ten of the ACS 8600 CPU Board schematic. This memory overlays the upper most eight kilobytes of the one megabyte RAM memory space when 2732 type devices are used. When 2716 type devices are used the upper most four kilobytes of RAM are overlayed. This read-only memory can be switched out of the memory space to permit access to a full megabyte of RAM. The presents or absence of the read-only memory in the system memory space is controlled by bit one of I/O port 30 hex which was described on page 45. It is planned for the boot memory to contain the following modules of code which are listed in priority order:

1. Initial program loader
2. Configuration control table and serial number
3. Limited diagnostics
4. Primitive driver for all serial channels
5. Primitive driver for flexible diskette drive
6. Primitive driver for rigid disk drive
7. Primitive driver for magnetic tape drive

The initial program loader (IPL) will first initialize the memory and I/O ports to meet the requirements of the primitive drivers contained in the EPROM. After hardware initialization the program loader will interrogate the configuration control table and system strapping port 051 hex to determine the baud rate and address of the master console. The EPROM will then run

limited power on diagnostic and report the results to the master console. Next the IPL will re-examine port 051 hex and the configuration control table to determine which mass storage device is the primary bootstrap device. It will then load the bootstrapping routine from the indicated device into memory and start executing that routine. The bootstrapping routine will permit the selective execution of additional diagnostic tests which are stored one of the mass storage devices before loading and starting the operating system.

The configuration control table reserves space for the customer to program a unique system serial number that will permit software to be "locked" to a specific system in addition to configuration information.

BIT	BIT NAME	TRUE
0		
1		
2		
3		
4		
5		
6		
7	ENABLE RAM BASED ROM TESTING	0

Table 4-12: I/O PORT 051 HEX - SYSTEM STRAPPING

4.4. Flexible Diskette Controller

The flexible diskette controller is implemented with a large scale integrated (LSI) device and random TTL logic. The controller block diagram is shown in Figure 3-4 (see Section 3.4, page 10) and the logic diagram for the controller is found on sheet eleven of the ACS 8600 CPU Board schematic while the logic diagram for the data separator and phase locked loop is found on sheet twelve. The flexible diskette controller can support up to four eight-inch flexible diskette drives operating in either single density or double density recording mode.

Drive selection is performed through I/O port 053 hex. The drive select logic is shown on sheet nine of the ACS 8600 CPU Board schematic. To select one of the four drives, the corresponding bit of port 053 hex is set to a zero. For example, setting bit zero of port 053 hex to a zero will select drive zero. This data is summarized in Table 4-13 on page 50. At power on the hardware guarantees that no drives are selected. After the boot memory has completed I/O port initialization, drive zero will be selected.

Double density operation is controlled by bit two of I/O port 055 hex (see Table 4-23, page 63). When bit two is a one the controller operates in double density mode. And when bit two of I/O port 055 hex is a zero the controller operates in single density mode. Double density mode is the default mode at power on and after I/O port initialization.

Bit three of I/O port 055 hex (see Table 4-23, page 63) is provided to permit the software to selectively reset the flexible diskette controller. The controller is held reset when ever bit three is a one. This bit is a one at power on and set to zero during I/O port initialization.

The flexible diskette controller device is assigned to I/O ports 041, 043, 045 and 047 hex. These ports can only be accessed as eight bit ports. Port 041 hex is the status port for the controller. The significances of the various bits of this port is detailed in Table 4-14 on page 50. Port 043 hex is the flexible diskette controller track register. The sector register of the controller is accessed through port 045. And the data port of the controller is accessed through port 047 hex. For details on the programming of the LSI flexible diskette controller used refer to Western Digital documentation on the 1797 Floppy Disk Formatter / Controller. One suitable document is the application guide entitled **FD 179X-01 Floppy Disk Formatter / Controller Family**.

To facilitate the transfer of data to and from the flexible diskette, a direct memory access (DMA) capability is provided by the 8089 I/O Processor and DMA Controller. Channel two of this two channel processor may be used for DMA transfers through the

BIT	BIT NAME	TRUE
0	FLOPPY DRIVE 0 SELECT-	0
1	FLOPPY DRIVE 1 SELECT-	0
2	FLOPPY DRIVE 2 SELECT-	0
3	FLOPPY DRIVE 3 SELECT-	0
4	INHIBIT NMI	1
5	(not used)	
6	(not used)	
7	(not used)	

Table 4-13: I/O PORT 053 HEX - FLOPPY DRIVE SELECT

BIT	BIT NAME	TRUE
0	BUSY	1
1	INDEX (SEEK), DMA REQ. (READ / WRITE)	1
2	TRK 0 (SEEK), LOST DATA (RD / WRT)	1
3	CRC ERROR	1
4	SEEK ERR. (SEEK), REC. NOT FND. (R/W)	1
5	HD LOADED (S), REC. TYP. (R), W FLT.	1
6	DISKETTE IS WRITE PROTECTED	1
7	NOT READY	1

Table 4-14: I/O PORT 041 HEX - FLOPPY DISK STATUS

flexible diskette controller. The channel attention of channel 2 is I/O address 039 hex. Channel 2 is also used for DMA transfers with either the magnetic tape controller or the high speed serial channel. A multiplexer is used to select the desired DMA request line before performing the data transfer. This multiplexer is controlled by bits four and five in I/O port 055 hex (see Tables 4-23 and 4-24, page 63). Bits four and five of port 055 hex must both be set to zero to allow the flexible diskette controller DMA request to reach the 8089. These bits must be set before the transfer begins. Table 4-24 on page 63 summarizes the other states of these bits. For further details on the programming and use of the 8089, consult **The 8086 Family User's Manual** published by Intel (document number 9800722-03).

Bit four of I/O port 53 hex provides a hardware mask for the non-maskable interrupt. When bit four is set to a one, all non-maskable interrupts are disabled and therefore no violation or error reporting can occur. This bit is intended for the exclusive use of the I/O port initialization software contained in the read-only memory. At power-on this bit is set to a one so that no non-maskable interrupts will occur before the vector table in low memory is built. After system initialization is complete this bit is set to a zero to permit full error reporting. This bit should never be set to a one by any operating system software or any application software.

4.5. Rigid Disk Controllers and Interface

The ACS 8600 system provides a generalized parallel interface to a family of rigid disk controller boards that mount directly to the CPU board. This interface can support rigid disk controllers with eight bit data paths. This type of rigid disks controller and interface provides a future growth path that permits the support of five inch Winchester disks as well as other disk drive technologies. The current members of the family of daughter boards provide controllers for both eight inch and five inch Winchester type rigid disks. The logic diagram for the rigid disk controller interface is found on sheet fourteen of the ACS 8600 CPU Board schematic. A complete copy of the **USER INFORMATION: EIGHT INCH SOFT SECTORED DRIVES AND THE ALTOS CONTROLLER** is contained in Appendix A of this specification.

Some of the rigid disk controller boards require a signal that indicates that the disk transfer has been completed and that control is changing from the DMA controller to the CPU. Bit one in I/O port 055 hex provides software control over this signal. Current rigid disk controllers look for a zero-to-one transition on this bit to signal the end of the disk transfer. The rigid disk driver should set bit one of I/O port 055 hex to a zero and then back to a one at the end of each data transfer to or from the rigid disk.

The ACS 8600 System will be offered with one of the rigid disk drives listed below.

Model Number	Disks	Heads	Cylinders	Total Capacity
Shugart				
SA1004	2	4	256	10
Quantum				
Q2010	1	2	512	10
Q2020	2	4	512	20
Q2030	3	6	512	30
Q2040	4	8	512	40

Table 4-15: RIGID DISK DRIVES SUPPORTED ON ACS 8600

Aside from capacity, there are two significant differences between these drives: the Quantum drives have twice the number of cylinders that the Shugart drives have and the Quantum drives have a REZERO capability that is neither provided for nor needed on the Shugart drives.

On the controller there is a jumper at S1 that needs to be set to coincide with the type of hard disk in the system. This must be set to S for the Shugart drives and to Q for the Quantum drives. This jumper sets the cylinder number at which write precompensation takes place (at cylinder 128 for the Shugart drives and 256 for the Quantum).

Data is stored on the disks in blocks or sectors of either 256 or 512 bytes (not programmable) identified by cylinder, head and sector numbers. The address for each sector is written previously (by a FORMAT command) the beginning of the data field. When reading from or writing to the disk, this address is read and verified before the data field is read or written. A correct match must be found for the address and its check bytes before the sector can be written or read.

The general procedure to read or write data is to first position the heads over the desired cylinder via a SEEK command, load the head and sector numbers into the appropriate ports and finally issue a READ or WRITE command. When the correct sector is found, the data is transferred between disk and system memory under control of the 8089 DMA controller by way of a FIFO buffer.

The controller is addressed by way of four I/O ports -- 020, 022, 024 and 026 hex. The following table summarizes the port usage.

Port	Read	Write
020H	X	Drive and head numbers
022H	Data	Old cylinder number (in two bytes), sector number and data
024H	X	New cylinder number (in two bytes)
026H	Status	Commands

Table 4-16: RIGID DISK CONTROLLER PORT USAGE

Port 026 hex is the command port when writing and the status port when reading. There are seven commands that may properly be issued:

RESET - initializes the controller circuit. A reset is done automatically at power on and is not normally required thereafter.

RECAL - commands the heads to be positioned over track zero. This must be done for each drive following power on before any disk access is attempted and thereafter as part of the error recovery routine. A recal requires about 15ms (Quantum) to 18ms (Shugart) per cylinder for a maximum time of about 8 seconds for the Quantum drives and about 4 seconds for the Shugart drives.

SEEK - causes the heads to move from the programmed "old cylinder" to the programmed "new cylinder".

READ WITH HEADER - is the command to read the addressed sector off of the disk, transferring the three byte header followed by the data sector. There are also two bytes of CRC that follow the data may be optionally transferred. This command is useful for validating operation of the disk controller.

READ WITHOUT HEADER - is the command to read the addressed sector off of the disk. Only the data from the sector is transferred, the three byte header is striped off by the controller.

WRITE - is the command to write to the addressed sector on the disk.

FORMAT - is the command to format an entire track. This consists of writing out the address or header for each sector on the specified (by cylinder and head) track.

REZERO - This command applies only to the Quantum Drive and causes the drive to do an internal recalibration of the head positioning circuitry. This takes about 2 seconds to complete.

Bit number	7	6	5	4	3	2	1	0	Hex
NULL	0	0	0	0	0	0	0	0	00
READ W/HEADER	0	0	0	0	0	0	0	1	01
READ WO/HEADER	0	0	0	0	1	0	0	1	09
WRITE	0	0	0	0	0	0	1	0	02
FORMAT	0	0	0	0	0	1	0	0	04
SEEK	0	0	0	1	0	0	0	0	10
RECAL	0	0	1	0	0	0	0	0	20
REZERO*	0	1	0	0	0	0	0	0	40
RESET	1	0	0	0	0	0	0	0	80

Table 4-17: RIGID DISK CONTROLLER COMMANDS (Port 026 hex)

No more than one command may be issued at one time. The controller will issue an interrupt on completion of all the commands except NULL and RESET. Following this interrupt, the status port must be read to clear the status register and permit subsequent interrupts. If more than one command is outstanding (a second issued before the first interrupts back), both commands will be aborted. The status register reports on the status of the drive and controller. The meaning of the various bits are detailed on the next page and in table 4-18 on page 56.

READY - indicates that the selected drive is powered and up to speed. No operation will take place until a drive is selected and ready is true. Normally being READY indicates that the drive is ready to be read from or written to but a special case exists with the Shugart drives: upon being powered on these drives require a two minute stabilization time before they should be read from or written to. Unfortunately, READY will go true before these two minutes are up (in about five seconds) and does not allow for this warm-up time.

WRITE FAULT - indicates a disk error during writing. To clear a write fault, the selected drive must be de-selected then reselected.

CRC ERROR - indicates that either the header or data field read off the disk was in error. If the operation was a write, the error was incurred in reading the header and the sector was not written. If the operation was a read, then the error could have occurred in either the header or data field. If it occurred in the header, the operation was aborted and the data field was not read. If it occurred in the data field, the data was already transferred before the error was detected.

RECORD NOT FOUND - indicates that during a read or write operation the specified header was not found in four revolutions of the disk. This may be caused by a seek error or incorrectly specified head or sector. RNF during a read may also indicate that the data sync field (zeros, "A1" or "F8") was not found even after the header was located. This is most likely caused by trying to read a sector after it has been formatted, but before it was written.

BAD SECTOR - indicates that the sector to be read or written was previously specified as a bad sector. A sector is flagged as bad during formatting with a one at bit 3 of the header byte. Good sectors have a Ø in this bit position.

TASK COMPLETE - indicates that the requested command has been carried out or aborted. Analysis of the status byte will determine whether the operation was aborted.

BUSY - indicates that the controller is in the process of carrying out an operation. This signal will be a Ø at the completion of an operation.

Bit	mnemonic	Meaning
7	RDY	Selected drive is ready
6	WR FLT	Write fault (from drive)
5	(none)	
4	CRC ERR	CRC error encountered
3	RNF	Record not found
2	BD SECT	Bad sector encountered
1	TC	Task complete
0	BUSY	Busy

Table 4-18: RIGID DISK CONTROLLER STATUS REG. (Port 026 hex)

Port 024 hex is used to specify the new cylinder number. This is entered in two consecutive bytes, the low order eight bits first then the high order three bits. The following table detailes the bit assignment in the port.

	Bits of Port 024 hex							
Cylinder bytes:	7	6	5	4	3	2	1	0
Least significant	7	6	5	4	3	2	1	0
Most significant	X	X	X	X	X	A	9	8

Table 4-19: RIGID DISK CONTROLLER CYL. ADDR. (Port 024 hex)

Port 022 hex has multiple functions. Before doing a SEEK the old cylinder is entered into this port (in the same manner as the new cylinder). Before doing a READ or WRITE the sector number is entered into this port, and when data is being transferred between disk and memory it is done through this port. Care should be exercised to insure that the correct data is present in port 022 hex before issuing any commands. Specifically this port should contain the old cylinder number before issuing a SEEK command and the sector number before issuing a READ or WRITE command.

Port 020 hex is used to select the drive and head to be used. The heads are selectable in a binary sequence (limited by the number of heads on the drive). Only one of the drives should be selected at any time. The table on the next page gives the details of the bit assignments for this port.

Bit	Name	Function
7	HS3	Head select bit 3
6	HS2	Head select bit 2
5	HS1	Head select bit 1
4	HS0	Head select bit 0
3		(none)
2		(none)
1	DS2	Drive select 2
0	DS1	Drive select 1

Table 4-20: RIGID DISK CONTROLLER DRV/HD REG. (Port 020 hex)

Since these drives are soft-sectored, an entire track must be formatted to keep the proper timing relationship among the sectors. Formatting of a track destroys any data on the track, so if the header for a sector ever needs to be changed (such as marking a sector as a BAD SECTOR) all the data on that track must be backed off of the disk before formatting. After the track has been formatted the data may be restored.

The formatting procedure is as follows: first the heads must be positioned to the desired cylinder using the SEEK command. Then a header image consisting of four bytes (an "0FE hex", cylinder number, head number, and sector number) must be placed in memory and written out to the disk by the 8089 DMA controller as each sector arrives under the head during one rotation of the disk. The cylinder number in the header must agree with the cylinder that the heads are on and the head number must agree with the selected head. However, the sector number can be any unique number between 0 and 255. This permits the staggering the sectors if required. The 8089 DMA controller must be programmed for a four byte continuous transfer in bus locked mode.

Byte name	Description
Sync byte	"0FE hex"
Cylinder byte	Low order bits (7-0) of the cylinder number
Head byte	Head bits (3-0) in the top four bits, a bad sector flag at bit 3, and high order cylinder bits (A-8) at the bottom three bits
Sector byte	Sector number in hex

Table 4-21: RIGID DISK CONTROLLER -- HEADER IMAGE

To write the header, the FORMAT command must be issued. When that header has been written out, an interrupt is generated. Then the header image must be updated to the next sector number (only the sector numbers change down the track) and a new FORMAT command issued before the next sector arrives at the head (464 microseconds for a 256 byte sector). The program must issue one FORMAT command for each sector on the track. It is the responsibility of the program to issue the correct number of FORMAT commands. The controller will generate an interrupt after accepting the header image data for each sector. When the program stops issuing FORMAT commands, the controller will automatically start writing a gap pattern from the end of the last sector to the physical index mark and will issue another interrupt when the index is detected. It is the responsibility of the program to read the status register and expect the additional interrupt at the end of each track. This can best be done by issuing a NULL command after the interrupt from the last FORMAT command has been received. This process must be repeated on each track and each cylinder to format the entire disk.

Two sector sizes are provided for: 256 bytes and 512 bytes. One track will accommodate 31 sectors at 256 bytes and 17 at 512. The sector size is determined by jumpers on the controller at S2. Two jumpers are used and both must be in either the 256 or the 512 position.

At the time of formatting, the controller fills in every byte of the data field with "Ø4E hex". Following the formatting of a track, any attempt to read a sector on that track that has not been specifically written to will result in a record not found (RNF) as the sync bytes will be missing for these sectors. Each sector must subsequently be written to with some other data (i.e, "ØE5 hex" to initialize the sector for CP/M, or MP/M). Any data stored on a track gets destroyed when that track is formatted.

The sequence of sectors on a track is under control of the formatting program. Physically adjacent sectors may be used if desired. It has been determined that a program can be generated for the 8Ø89 which can transfer consecutive sectors on a single track. The memory control circuitry will maintain memory refresh during the data transfers performed by the 8Ø89 DMA controller. The main disadvantage of long multisector transfers is that the 8Ø86 is locked out due to the need to lock the bus to the 8Ø89 during the data transfer to support the data transfer rate.

The soft sectoring of the drive precludes the ability to perform a specific READ ID command. This is because the location of the sector on a track is unknown. When READING or WRITING, a sector is located by reading all the headers encountered along the track until the correct one is found. In performing a READ WITH HEADER, when the correct header is found, the header bytes are read off the disk and transferred into memory. A diagnostic can used to verify that the data read was from the specified sector. The READ WITHOUT HEADER command is provided to for

normal disk reads performed by the operating system so that multiple sector reads may be performed directly into the final memory locations.

WARNING: Upon being powered on, the Shugart drive requires a two minute stabilization time before reading or writing. The READY signal from the drive will go true about five seconds after power-on, it is up to the disk driver software to wait these two minutes.

Once the drive has been properly initialized following power on by selecting and issuing a RECAL command to the drive, a seek can be performed. To seek a specific cylinder, the current cylinder must be loaded into the old cylinder address register port (Port 022 hex -- following a RECAL the current cylinder would be 000 hex), the desired new cylinder address must be loaded into port 024 hex, and a SEEK command must be issued. An interrupt will occur when the head has arrived at the new cylinder and settled down (i.e. no additional head settle time is required).

To write or read a sector, the sector must first be found. The controller does this by reading and comparing the header images (which are written on the disk during formatting) as they pass under the head to the data stored in ports 020 hex-024 hex (020 hex has the head number, 022 hex has the sector number and 024 hex has the new cylinder number). Since the sector register is overwritten by data during a write to disk, the sector number must be rewritten into port 022 hex before the next read or write command can be issued. When switching between drives the new cylinder number must be reloaded into port 024 hex, even if a seek is not necessary.

To read a sector, prepare a buffer area in main memory into which the data from disk can be placed (either 256 or 512 bytes for WRITE and READ WITHOUT HEADER or 259 or 515 for READ WITH HEADER) and request the primitive rigid disk driver contained in the bootstrap ROM to make the access or transfer control to your own 8089 control program to make the data transfer. For detailes on interfacing with the primitive disk driver consult the **ALTOS 8600 SYSTEM MONITOR SPECIFICATION.**

In general the 8089 program that transfers data from the rigid disk will first load the address (cylinder, head/drive and sector) of the sector to be read. The cylinder number should already be correct from having previously done a SEEK. The 8089 should load the head number into port 020 hex without changing the drive number (they share the same byte). Then the 8089 should load the sector number into port 022 hex. (A 20 microsecond delay is required between selecting a drive or a head and issuing a read or write command, but the inherent programming delays normally take care of this). Finally the 8089 will execute the transfer instruction which activates the 8089 DMA controller and then issue the READ command to the disk controller. At this

point the 8089 will return the bus to the 8086 until the disk controller issues a DMA request. When the drive/controller locates the desired sector, the controller raises the DMA request line. The active DMA request causes the 8089 to request the bus from the 8086 and lock it to the 8089 for the duration of the data transfer. The data transfer is carried out by the 8089 DMA controller directly into memory. When the transfer is complete, the controller generates another interrupt and clears the READ command. The status register must then be read to clear the interrupt. If an error condition is detected by the disk controller, it will generate an interrupt which will terminate the 8089 DMA transfer command. This allows the 8089 program to resume and determine the error and perhaps retry the command. If a RNF is indicated, the data was not transferred to memory. If a CRC ERR or BAD SECTOR is indicated, the data was transferred but its accuracy is questionable.

To WRITE a sector, a similar procedure is used with the buffer area first being filled with the data to be written to the disk. The header and CRC bytes are provided by the hardware, so only the data bytes need be transferred by the 8089. Again, the status register must be read at the end of the operation to clear the interrupt. If a RNF, CRC ERR or BAD SECTOR is indicated the data was not written out to disk. When writing, all the sync bytes are provided by the controller. However, when FORMATTING the drive, a sync byte of "0FE hex" must precede the header bytes.

The drive should always be selected before issuing any command. When a command is issued without a drive being ready, the controller will wait until the selected drive becomes ready before the operation is carried out. Then at the completion of the operation an interrupt is generated. However issuing a SEEK or RECAL command to a drive that is not ready will cause the controller to hang in a state that does not return an interrupt. The system must be reset in order to clear this condition. RESET does not generate an interrupt neither does selecting or deselecting a drive.

To prevent extraneous writes to the disk when the power goes off, the controller has a sensing circuit that deselects the drives when the +5 volt supply falls below +4.5 volts. This circuit does not guarantee, however, that data being written to a sector will not be lost if power goes off. To insure against such a loss, the system should be powered down only when no writing is being done to the disk.

As noted earlier, this controller has the capability of flagging bad sectors. A sector may be flagged as being bad by entering the appropriate bit (bit 3 in the header byte - see the header image information in Table 4-21 on page 57) in the header during the format operation. Thereafter, when such a sector is accessed, writes and reads will be aborted and a BAD SECTOR status reported. Note that a BAD SECTOR status is not reported just because the accessed sector is bad, but because it has been

flagged as being bad. The usual status returned when reading a sector that has a defect but is not flagged is a CRC error.

The reason for flagging a sector as bad is that the recording media occasionally contains flaws that do not permit reliable writing and reading in certain areas. By marking those sectors containing known flaws as BAD SECTORS the diagnostic that tests the disk drive can distinguish between defective media and a failure of the drive or controller to operate properly.

Each drive manufacturer does an analog (and therefore very sensitive) test of each drive that produces a "media scan" record indicating where the defective areas are. This map can be used to identify the sectors that should be marked as BAD SECTORS so that the operating system will not attempt to use them.

Shugart provides an SA1004 MEDIA ERROR MAP with each drive which gives the track number (TRK), head number (HD), beginning location with respect to the index pulse (BYTE COUNT) and length of the defect in bits (LENGTH [BITS]). To convert these numbers to nomenclature that the controller recognizes, only the sector number needs to be deciphered. The cylinder number is the same as Shugart's track number and the head number needs no translation. The number of the bad sector(s) is determined from the table on the next page. This gives the range in bytes for each sector relative to the index pulse. The defective area given by the media scan spans from the BYTE COUNT to the BYTE COUNT plus the result of LENGTH divided by 8. Any sector whose range falls within the defective area is bad. Defective areas may extend across more than one sector. In addition, the possible speed variation of the disk during formatting (+/-3%) may result in an ambiguity as to which sector in which a defect will fall. This is reflected in the table where the sector ranges overlap. In such a case, all potentially defective sectors need to be identified. This table assumes no skewing and 512 byte sectors.

To avoid the delay in seeking to a sector only to find that it is bad, and to get around the problem that may occur if a defect occurs in the header of a sector (in which case the BAD SECTOR flag may not be recoverable but rather a RNF status would be returned), it has been proposed that cylinder 0 (guaranteed to be defect-free) be used to store the bad sector table which the operating system would check before writing or reading a sector. (See ALTOS Format for Hard Disk Systems 11/19/80). Another approach is to assign dummy files to the sectors that are bad so that the operating system would not try to access them.

BYTE COUNT	PHYSICAL SECT NO	LOGICAL SECT NO
15 - 596	0	0
588 - 1204	1	6
1162 - 1813	2	12
1735 - 2422	3	1
2308 - 3031	4	7
2881 - 3639	5	13
3455 - 4248	6	2
4028 - 4857	7	8
4601 - 5466	8	14
5175 - 6074	9	3
5748 - 6683	10	9
6321 - 7292	11	15
6894 - 7901	12	4
7468 - 8509	13	10
8041 - 9118	14	16
8614 - 9727	15	5
9187 - 10335	16	11

Table 4-22: CONVERSION TABLE - BYTE COUNT TO SECTOR NUMBER

Channel one of the 8089 I/O Processor - DMA Controller is intended to provide the direct memory access control necessary to support the data transfer rates of the rigid disks. This channel is totally dedicated to supporting the rigid disk controller and does not have a multiplexer like channel 2 of the 8089 (See Section 4.4). The channel attention of channel 1 is I/O address 038 hex. For further details on the programming and use of the 8089, consult **The 8086 Family User's Manual** published by Intel (document number 9800722-03).

4.6. Parallel Interface

The ACS 8600 provides a 24 bit programmable parallel interface. The logic involved in this interface is shown on sheet fifteen of the ACS 8600 CPU board. This interface is implemented with an 8255 Programmable Peripheral Interface device which contains three eight bit ports. Ports A and B of the 8255 are buffered with bi-directional buffers that permit these ports to be used for either input or output. Port C is buffered with a uni-directional buffer that provides one bit of input and seven bits of output. The buffering on Port C supports the handshake and interrupt signals used by Port B in mode one. Modes one and two for Port A are not supported. Note that if the lower half of Port C is used for output in mode zero the shunt near the 8255A-5 (Location 28S) should be removed to prevent damage to the 8255. If the shunt is not removed, the output driver of Port C bit 2 will be fighting the output driver of the buffer. Tables 4-25

and 4-26 on pages 64 and 65 gives a summary of the capabilities and signal assignment for the parallel port. The standard parallel port driver software (Centronics compatible) uses the lines as defined in Table 4-26.

BIT	BIT NAME	TRUE
0	MEMORY ERROR CORRECTION ENABLE	0
1	END HARD DISK OPERATION	1
2	DOUBLE DENSITY FLOPPY ENABLE	1
3	FLOPPY DISK CONTROLLER RESET	1
4	DIRECT MEMORY ACCESS SELECT 0 (DMA 0)	1
5	DIRECT MEMORY ACCESS SELECT 1 (DMA 1)	1
6	PARALLEL INTF. CH.A INPUT ONLY-	0
7	PARALLEL INTF. CH.B INPUT ONLY-	0

Table 4-23: I/O PORT 055 HEX - MISCELLANEOUS CONTROL BITS

DMA 1	DMA 0	DMA REQUEST SELECTED
0	0	FLEXIBLE DISKETTE CONTROLLER
0	1	MAGNETIC TAPE CONTROLLER
1	0	HIGH SPEED SERIAL CHANNEL
1	1	NONE

Table 4-24: I/O PORT 055 HEX, BITS 4 AND 5 - DMA REQUEST SELECT

CPU BOARD HEADER	REAR PANEL CONNECTOR	PORT AND BIT	CAPABILITY
J6-5	3	P 40, B 0	MODE 0, IN or OUT
J6-8	23	P 40, B 1	MODE 0, IN or OUT
J6-10	24	P 40, B 2	MODE 0, IN or OUT
J6-12	25	P 40, B 3	MODE 0, IN or OUT
J6-11	6	P 40, B 4	MODE 0, IN or OUT
J6-9	5	P 40, B 5	MODE 0, IN or OUT
J6-7	4	P 40, B 6	MODE 0, IN or OUT
J6-6	22	P 40, B 7	MODE 0, IN or OUT
J6-29	15	P 42, B 0	MODE 0 or 1, IN or OUT
J6-24	31	P 42, B 1	MODE 0 or 1, IN or OUT
J6-26	32	P 42, B 2	MODE 0 or 1, IN or OUT
J6-25	13	P 42, B 3	MODE 0 or 1, IN or OUT
J6-23	12	P 42, B 4	MODE 0 or 1, IN or OUT
J6-30	34	P 42, B 5	MODE 0 or 1, IN or OUT
J6-27	14	P 42, B 6	MODE 0 or 1, IN or OUT
J6-28	33	P 42, B 7	MODE 0 or 1, IN or OUT
J6-13	7	P 44, B 0	MODE 0 OUT; P 42 INT
J6-16	27	P 44, B 1	MODE 0 OUT; P 42 BUF FULL
J6-15	8	P 44, B 2	MODE 0 IN; P 42 STRB-/ACK-
J6-18	28	P 44, B 3	MODE 0 OUT
J6-14	26	P 44, B 4	MODE 0 OUT
J6-19	10	P 44, B 5	MODE 0 OUT
J6-20	29	P 44, B 6	MODE 0 OUT
J6-17	9	P 44, B 7	MODE 0 OUT
J6-1	1		GROUND
J6-2	20		GROUND
J6-3	2		GROUND
J6-4	21		GROUND
J6-21	11		GROUND
J6-22	30		GROUND
J6-31	16		GROUND
J6-32	35		GROUND
J6-33	17		GROUND
J6-34	36		GROUND
	18		NO CONNECTION
	19		NO CONNECTION
	37		NO CONNECTION

Table 4-25: PARALLEL INTERFACE PINS AND CAPABILITIES

CPU BOARD HEADER	REAR PANEL CONNECTOR	CENTRONICS CONNECTOR	FUNCTION
J6-15	8	10	ACKNOWLEDGE- from LP
J6-5	3	11	BUSY from LP
J6-8	23	32	FAULT- from LP
J6-10	24	12	PAPER OUT from LP
J6-12	25	13	SELECTED from LP
J6-29	15	2	DATA 1 to LP
J6-24	31	3	DATA 2 to LP
J6-26	32	4	DATA 3 to LP
J6-25	13	5	DATA 4 to LP
J6-23	12	6	DATA 5 to LP
J6-30	34	7	DATA 6 to LP
J6-27	14	8	DATA 7 to LP
J6-28	33	9	DATA 8 to LP
J6-14	26	1	DATA STROBE to LP
J6-20	29	31	INPUT PRIME to LP
J6-1	1	28	GROUND
J6-2	20	29	GROUND
J6-3	2	20	GROUND
J6-4	21	21	GROUND
J6-21	11	22	GROUND
J6-22	30	23	GROUND
J6-31	16	24	GROUND
J6-32	35	25	GROUND
J6-33	17	26	GROUND
J6-34	36	27	GROUND

Table 4-26: PARALLEL (CENTRONICS) INTERFACE CONNECTOR

Port A of the parallel interface is accessed as an eight bit port at I/O port address 040 hex. Port B is accessed as an eight bit port through port address 042 hex. And port C is accessed as an eight bit port through address 044 hex. The control port of the 8255 is accessed as an eight bit port through I/O port 046 hex. Extreme care should be exercised when accessing these ports. Attempting to access them as sixteen bit ports will result in the simultaneous access of both the parallel interface and the LSI flexible diskette controller device.

The direction of the buffers on ports A and B of the 8255 are controlled by bits six and seven in I/O port 055 hex (see Table 4-23, page 63). Bit six controls the buffers for port A while bit seven controls the buffers for port B. When the bits are set to one the buffers are turned toward the 8255. The buffers are turned away from the 8255 by setting the appropriate bit to zero. At power on and after I/O port initialization bits six and seven of I/O port 055 are set to one, the buffers are set for input.

4.7. Serial Interfaces

The ACS 8600 system provides a minimum of two serial channels and a maximum of ten serial channels. Two channels are contained on the CPU board while the other eight are contained on an Intelligent Serial Concentrator Board. One channel on the CPU board can be strapped for either synchronous or asynchronous operation. The multipurpose channel on the CPU board can be strapped to interface to a verity of serial networks operating at data rates up to 800 kilobits per second. The two channels on the CPU board are intended for use in manufacturing test and stand alone applications. These serial channels are initialized to operate in asynchronous mode and the hardware is strapped to operate with the request-to-send line internally tied to the clear-to-send line. Simple mechanical shunts are provided to permit field alteration of the strapping of all serial channels.

The Intelligent Serial Concentrator Board consists of a Z80A CPU processor with eight serial channels under its control. The processor communicates with the CPU Board via main memory and a pair of interrupt lines. The interface used for this communication is the Multibus. The CPU Board may interrupt the Z80 and the Z80 may interrupt the CPU Board. The Intelligent Serial Concentrator Board is intended to offload the CPU Board processor so that the majority of character interrupt processing may be handled by the Intelligent Concentrator. This allows the 8600 CPU processor to devote more of its power to higher level tasks and handle users in a more timely fashion. A complete copy of the hardware specification for this board is included in Appendix B.

4.7.1. Primary Serial Channels

The primary serial channels are located on the CPU board and are intended to provide serial interfaces for manufacturing test and stand alone applications of the CPU Board. These channels are implemented with an Intel 8274 Multi-Protocol Serial Controller. This device contains two independent serial channels which can be programmed to operate in a verity of modes. For detailed information on the programming of this device consult the Intel documentation on the 8274.

Channel A of the 8274 is identified as ACS 8600 serial printer channel zero and is accessed through I/O ports 048 hex and 04C hex. Port 048 hex is the data port and port 04C hex is the control and status port. This channel is the multipurpose channel referred to above. It is intended for use either as a serial printer channel operating in asynchronous mode or as an interface to a network. As originally manufactured channel A is strapped for operation as an RS-232C port without handshaking (request-to-send is internally tied to clear-to-send).

For network operations printer channel zero can be operated under direct memory access (DMA) control at speeds up to 800 kilobits per second. A set of EIA Standard RS-422 drivers and receivers can be strapped to this channel to interface to a high speed network implemented with two twisted pairs of wires; a single twisted pair of wires or a co-axial cable; or a special network transceiver. All network interface control, timing and power signals are available on a 34 pin header located at the rear edge of the CPU Board.

Printer channel zero may also be strapped to operate in synchronous mode with EIA Standard RS-232C drivers and receivers. In this mode the transmit and receive clocks must be supplied from an external source. There are no provisions for the CPU board to supply these clocks.

Channel B of the 8274 is identified as ACS 8600 serial console channel zero and is intended as a console interface channel operating in asynchronous mode with RS-232C drivers and receivers. This channel is accessed through I/O ports 04A hex and 04E hex. Port 04A hex is the data port and port 04E hex is the control and status port. As originally manufactured channel one is strapped for operation without request-to-send -- clear-to-send handshaking.

Both serial channels of the CPU board are supported by independent baud rate generators. These generators and the system time slice clock are implemented with an Intel 8253 Interval Timer. All timing generated by the 8253 are derived by dividing down a 1.2288 megahertz clock. The 8253 timer is accessed through I/O ports 049, 04B, 04D and 04F hex. Timers zero and one (I/O ports 049 hex and 04B hex) are programmed by

the I/O initialization routine to mode three - square wave generator for baud rate generation. I/O port 049 hex determines the baud rate clock supplied to channel A of the 8274 (I/O ports 048 and 04C hex). While I/O port 04B hex is used to set the baud rate for 8274 channel B (I/O ports 04A and 04E hex). Table 4-27 lists the divisors that must be loaded into these channels to produce various baud rates. All rates listed in Table 4-27 assume that the serial interface device (8274) is operating in sixteen times (X16 CLOCK) mode.

BAUD RATE	DIVISOR
50	1536
75	1024
110	698
134.5	571
150	512
300	256
600	128
900	85
1200	64
1800	43
2400	32
3600	21
4800	16
7200	11
9600	8
19200	4
38400	2

Table 4-27: BAUD RATES AND DIVISORS FOR X16 CLOCK MODE

I/O port 04D hex access the time slice timer. After I/O port initialization this timer is set to mode zero with a divisor of 12,288. This programming will cause this timer to generate an interrupt every ten milliseconds.

4.7.2. Additional Serial Channels

An Intelligent Serial Concentrator Board is mounted on the rear panel of the system enclosure to provide eight additional serial channels. A separate CPU is provided on this board along with dedicated read only memory (ROM) and random access memory (RAM) to service the needs of the eight serial channels. This board accepts serial I/O channel commands and moves blocks of characters between main memory and the eight terminals.

Communication between the CPU Board and the Intelligent Serial Concentrator Board is accomplished by means of memory registers. Initially, one 24-bit register (called the "Initialization Register") exist at location 01FFFC hex. This register contains a pointer to a group of memory registers. The layout of this group of registers which may exist anywhere in memory is shown in Figure 4-4 on pages 70 and 71. All of the memory registers occupy a total of 182 (decimal) bytes of memory. When the Concentrator is first interrupted, the initialization register is read. Initialization is accomplished according to the parameters in the register group pointed to by the Initialization Register. The initialization register will be read only after the first interrupt received by the controller, allowing this reserved location to be reused by the host until reinitialization is required.

The first six bytes of the 182 bytes of memory form the five system registers. These are the Firmware Version Register, the System Command Register, the System Status Register, the Interrupt Vector register and the New Command Register. The remaining 176 bytes of memory comprise eight sets of Channel Registers. There are twenty-two bytes in each set of registers. Each set contains the twelve register listed below. Note that when the TTY Receive Register is used it replaces the second byte of the Receive Buffer Output Pointer Register.

Parameter Register
Channel Status Register
Channel Command Register
Transmit Buffer Address Register
Transmit Buffer Length Register
Receive Buffer Address Register
Receive Buffer Length Register
Receive Buffer Input Pointer Register
Receive Buffer Output Pointer Register
TTY Receive Register
Selectable Rate Register
Expansion Register

IR + 000H	Firmware Version Register
IR + 001H	System Command Register
IR + 002H	System Status Register
IR + 003H	Interrupt Vector Register (bits 0 - 7)
IR + 004H	Interrupt Vector Register (bits 8 - 15)
IR + 005H	New Command Register
IR + 006H	Channel 0 Parameter Register (bits 0 - 7)
IR + 007H	Channel 0 Parameter Register (bits 8 - 15)
IR + 008H	Channel 0 Status Register (bits 0 - 7)
IR + 009H	Channel 0 Status Register (bits 8 - 15)
IR + 00AH	Channel 0 Command Register
IR + 00BH	Ch 0 Transmit Buffer Address (bits 0 - 7)
IR + 00CH	Ch 0 Transmit Buffer Address (bits 8 - 15)
IR + 00DH	Ch 0 Transmit Buffer Address (bits 16 - 23)
IR + 00EH	Ch 0 Transmit Buffer Length (bits 0 - 7)
IR + 00FH	Ch 0 Transmit Buffer Length (bits 8 - 15)
IR + 010H	Ch 0 Receive Buffer Address (bits 0 - 7)
IR + 011H	Ch 0 Receive Buffer Address (bits 8 - 15)
IR + 012H	Ch 0 Receive Buffer Address (bits 16 - 23)
IR + 013H	Ch 0 Receive Buffer Length (bits 0 - 7)
IR + 014H	Ch 0 Receive Buffer Length (bits 8 - 15)
IR + 015H	Ch 0 Receive Buffer Input Pointer (bits 0 - 7)
IR + 016H	Ch 0 Receive Buffer Input Pointer (bits 8 - 15)
IR + 017H	Ch 0 Receive Buffer Output Pointer (bits 0 - 7)
IR + 018H	Ch 0 Receive Buffer Output Pointer (bits 8 - 15)
IR + 018H	Ch 0 TTY Receive Register (Replaces above byte)

Figure4-4: SERIAL CONCENTRATOR MEMORY REGISTERS

IR + 019H	Ch 0 Selectable Rate Reg (bits 0 - 7)
IR + 01AH	Ch 0 Selectable Rate Reg (bits 8 - 15)
IR + 01BH	Channel 0 Expansion Register
IR + 01CH	Channel 1 Registers
IR + 032H	Channel 2 Registers
IR + 048H	Channel 3 Registers
IR + 05EH	Channel 4 Registers
IR + 074H	Channel 5 Registers
IR + 08AH	Channel 6 Registers
IR + 0A0H	Channel 7 Registers
IR + 0B6H	

Figure4-4: SERIAL CONCENTRATOR MEMORY REGISTERS (Continued)

The Firmware Version Register is a one byte memory register where the Controller writes the level of the firmware in the Controller. The bootstrap software will initialize this location with zero at system initialization. When Controller 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). Version numbers are stored in a two part format. The most significant five bits of the byte contain the version number in the range of one to sixty-three. The least significant bits of the byte contain the sub-version number in the range of zero to seven. This permits the representation of a version number such as 2.0 where 2 is stored in bits three through seven and 0 is stored in bits zero through three.

The System Command Register is an eight bit register used to pass commands to all channels simultaneously. The command is passed in bits zero through six while handshaking is accomplished by setting bit seven to a one to indicate the issuing of a command. The 8086 should insure that the last command has been executed by testing bit seven of this register for zero before issuing the next command. The commands are given in the following table.

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

Table 4-28: SERIAL CONCENTRATOR COMMANDS

The System Status Register is an eight bit register used to pass status to the 8086 concerning channel-independent status information. The status bits are defined in the table 4-29 below. Bit zero enables and disables the entire controller. Bit one enables the interrupt interface from the controller back to the 8086. Bit two allows the 8086 to poll the controller for an interrupt condition without enabling interrupts. Bit three is used by the controller to report bus errors back to the 8087. A bus error is generated by the main memory when a multi-bit memory error is detected by the memory error detection and correction circuitry.

BIT	BIT NAME	TRUE
0	CONTROLLER ENABLED	1
1	INTERRUPTS ENABLED	1
2	INTERRUPT PENDING (NOT QUAL. BY ENA)	1
3	BUS ERROR DETECTED	0
4	(not used)	0
5	(not used)	0
6	(not used)	0
7	(not used)	0

Table 4-29: SERIAL CONCENTRATOR STATUS REGISTER

The Interrupt Vector Register is a sixteen bit register containing data to allow quick response to an interrupting condition. The register is logically divided into four fields of four bits. The bit definitions are given in the Table 4-30 on the next page.

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 eight channel command registers.

BIT	BIT NAME	TRUE
0	MODEM INTERRUPT CHANNEL # - LSB	1
1	MODEM INTERRUPT CHANNEL #	1
2	MODEM INTERRUPT CHANNEL # - MSB	1
3	MODEM INTERRUPT PENDING	1
4	RECEIVE INTERRUPT CHANNEL # - LSB	1
5	RECEIVE INTERRUPT CHANNEL #	1
6	RECEIVE INTERRUPT CHANNEL # - MSB	1
7	RECEIVE INTERRUPT PENDING	1
8	TRANSMIT INTERRUPT CHANNEL # - LSB	1
9	TRANSMIT INTERRUPT CHANNEL #	1
10	TRANSMIT INTERRUPT channel # - MSB	1
11	TRANSMIT INTERRUPT PENDING	1
12	(not used)	1
13	(not used)	1
14	(not used)	1
15	(not used)	1

Table 4-30: SERIAL CONCENTRATOR INTERRUPT VECTOR REGISTER

The remaining registers (176 bytes) form eight sets of Channel Registers -- one set of registers for each of the channels on the Intelligent Concentrator Board. The first of the Channel Registers is the Channel Parameter Register. The Channel Parameter Register is a sixteen bit register used to establish the bit level characteristics of the channel. The definition of the bits is given in Table 4-31 on the next page. Most of these bits deal with the programming of the receiver-transmitter device and the baud rate generator device. One exceptions is bit seven.

Bit seven selects between buffered receive mode and TTY mode. In TTY mode each character is passed through the one byte TTY Receive Register which replaces the second byte of the Receive Buffer Output Pointer Register. In this mode an interrupt is generated each time the buffer is fulled. In buffered receive mode characters are loaded into the receive ring buffers whose base addresses are contained in Receive Buffer Address Register. Again an interrupt is generated each time one or more characters are transferred to the buffer.

Bit 0	=1		Parity enable
Bit 1	=1		Parity even
Bits 2-3	=0		Illegal
	=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	=1		Ring buffer receiver enable
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 or undefined
Bit 12			Unused
Bit 13			Unused
Bit 14	=1		CTS active
Bit 15	=1		DSR active

Table 4-31: SERIAL CONCENTRATOR CHANNEL PARAMETER REGISTER

The Channel Status Register is a sixteen bit register used to report the current status of the respective serial channel. This register is updated before the Controller generates an interrupt. In general an interrupt is generated when a status bit is set to a one. The specific definition of bits in the Channel Status Register and the field of the Interrupt Vector Register associated with each status are detailed in Table 4-32 below.

bit	definition	interrupt
bit 0	unused - always 0	none
bit 1	Data Terminal Ready	modem
bit 2	unused - always 0	
bit 3	Request To Send	modem
bit 4	Parity Error	receive
bit 5	Receive Data Lost (SIO overrun)	receive
bit 6	Framing Error	receive
bit 7	Parity OR Overrun OR Framing Error	receive
bit 8	Receive Character(s) Ready	receive
bit 9	unused - always 0	none
bit 10	unused - always 0	none
bit 11	unused - always 0	none
bit 12	Transmitter Ready	transmit
bit 13	unused - always 0	none
bit 14	unused - always 0	none
bit 15	unused - always 0	none

Table 4-32: SERIAL CONCENTRATOR STATUS REGISTER

The channel command register is an eight bit register used to pass channel specific commands to the controller. The command is encoded in bit zero through three. Bits four through six are used to indicate the logic level corresponding to the interrupt enable flags. Bit seven is used to indicate the existence of a valid command. The 8086 should test bit seven to insure that the last command has been executed before loading a new command. The controller will clear bit seven and bits zero through three after reading the command. Bits four through six will remain unchanged to continuously indicate the interrupt enable status. The commands available are summarized in the following table.

Bits 0 - 3	
0	No operationel
1	Initialize channel
2	Start transmitter
3	Acknowledge receiver
4	Abort transmitter
5	No operationel
6	No operationel
7	No operationel
8	Change parameters
9	Reset error conditions
10	Reset modem interrupt request
Bit 4	Modem interrupt enable
Bit 5	Receive interrupt enable
Bit 6	Transmit interrupt enable
Bit 7	Command valid

Table 4-33: SERIAL CONCENTRATOR CHANNEL COMMAND REGISTER

The Transmit Data Buffer Address Register is a 24-bit register containing the address of the data to be transmitted. After transmit termination, this register will contain the address of the last character transmitted plus one. The number of characters to be transmitted is contained in the sixteen bit Transmit Data Buffer Length Register. If an abort transmitter is issued, the Transmit Data Buffer Address Register will be updated to point to the first byte NOT transmitted and the Transmit Data Buffer Length Register will be update to contain the number of bytes NOT transmitted.

The Receive Data Buffer Address Register is a 24-bit register containing the base addresses of the receive buffer to be used in Ring Buffer Mode. The Receive Data Buffer Length Register is a sixteen bit register containing the length of the receive data buffer. The receive buffer length may be in the range of 2 to 32,768 bytes. Also associated with the Receive Data Buffer are the Receive Buffer Input Pointer Register and the Receive Buffer Output Pointer Register. These pointers are both sixteen bit registers. The Input Pointer always points to the first empty location to be filled by the Controller. As the Controller updates the Receive Data Buffer This pointer is incremented modulo the Receive Buffer Length. Bit 15 is set to a one when the contents are are being updated and therefore are invalid. Bit 15 may be ignored if the buffer length is specified to be less than or equal to 256. The Receive Buffer Output Pointer points to the start of the Receive Data Buffer. As the 8086 removes data from the Receive Data Buffer, it must increment this pointer modulo the Receive Buffer Length. Again bit 15 must be set to a one to indicate that the contents of the Output Pointer Register are invalid because an update is in process unless the Buffer Length is specified to be less than or equal to 256.

The TTY receive register is an eight 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. Note that this register replaces the second byte of the Receive Buffer Output Pointer Register which is not used.

The selectable rate register is a sixteen bit register used to transfer a sixteen bit value directly to the bit rate generator. When the rate is specified through this register, the SIO controller is programmed to further divide the clock supplied by the rate generator by sixteen. The value to be passed through this register may be calculated using the following equation:

$$\text{Value} = \frac{115{,}200}{\text{Desired Bit Rate}}$$

Finally the Expansion Register is defined to permit future expansion of the Channel Register Array. This register is also defined to bring the total number of bytes required for the register array to an even number. This can simplify the generation and handling of the array under some languages.

Software interface to the Altos Intelligent Serial I/O Concentrator can best be divided into eight sections:

Controller Activation
8086/Controller Communication
Controller Initialization
Channel Initialization
Channel Transmit Operation
Channel TTY Receive Operation
Channel Ring Buffered Receive Operation
Modem Control/Status Operation

After a power-up or controller hardware reset, the controller first initializes all hardware devices, then clears internal RAM, and finally waits in a tight loop. To activate the controller, the 8086 computer must first build and initialize the system and channel register arrays. The system register array followed by the eight channel register arrays form a single contiguous block referred to as the Controller Control Block or CCB. After the CCB is built, the address of the Controller Control Block is stored at location 1FFFC hex, lowest byte first, highest byte last. The last step is for the 8086 to generate a "Channel Attention" signal to the controller by writing to I/O

port 078 hex. 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. After activation, the controller is in an idle state waiting for the system command to enable the controller. When this command is issued, controller interrupts may also be enabled.

Command information for the controller is passed through ten registers within the CCB. The process that reads the command registers is generally in a quiet but not asleep state. When the process runs 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 when no commands are pending, but does demand the proper sequencing of events. Any time the 8086 issues a command, it must alter the value in the "New Command Register". The recommended procedure is to increment the "New Command Register" after setting up the other registers for 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, it is essential that bit 7 be tested before issuing a new command. The programmer must note that the channel status is not valid while a command is pending on that channel. 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.

After defining the desired operating parameters (i.e., parity, word length, etc.), which must be stored into the channel parameter register, the channel may now be initialized with the "Initialize Channel" command. Once this initialization command has been issued the channel is ready to transmit or receive. If any of the interrupt enable bits are set with the command, and their associatie conditions become true, an interrupt will be generated. A No-Op command may be issued at any time to alter only the interrupt enable bits. On the following pages are sample pseudo-code procedures that may be used as models for generating Intelligent I/O Controller driver routines.

Use the following procedure to transmit a block of data:

```
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
```

To receive a character in TTY mode, use the following procedure:

```
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
```

The following procedure should be used to receive a character in Ring Buffered Receive mode:

```
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
     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) <- 0;
     }
```

Use the following procedure to check the state of the Modem status lines:

```
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
```

4.8. Magnetic Tape Controller

An optional magnetic tape drive controller which is capable of controlling up to eight tape transports may be mounted on the CPU board. This controller is intended to control a DEI Funnel Tape Transport with a the codec and controller options installed. This controller is implemented with an 8274 and an 8255. The 8274 serializes and deserializes the data as well as controlling and monitoring the tape drives readiness to send or receive data. The tape drive supplies both a write and a read clock so the 8274

must be programmed to operate in synchronous mode. The data channel is assigned to I/O port ØD8 hex while the control channel for the 8274 is assigned to I/O port ØDC hex.

Tape motion, drive select and track select is performed through the 8255. Port A of the 8255 is assigned to I/O address ØD9 hex and is used to select the drive and track. Table 4-34 describes the meanings of the bits in this port. Notice that the bits have been assigned to the port to permit each four bit nibble to contain the true binary value for the drive or track to be selected. The DEI tape transport numbers drives from one to eight and tracks from one to four. Port B is assigned to I/O port address ØDB hex and is used to read the status of the tape transport. These bits are summarized in Table 4-35 on the next page. Port C is assigned to I/O port address ØDD hex and is used to control tape motion. These bits are listed in Table 4-36 also on the next page. The I/O port initialization routine initializes the 8255 by writing Ø82 hex to the 8255 control port which is assigned I/O address ØDF hex.

BIT	BIT NAME	TRUE
Ø	LSB OF TACK NUMBER	1
1	BIT OF TACK NUMBER	1
2	MSB OF TACK NUMBER	1
3	not used	
4	LSB OF DRIVE SELECT NUMBER	1
5	BIT OF DRIVE SELECT NUMBER	1
6	BIT OF DRIVE SELECT NUMBER	1
7	MSB OF DRIVE SELECT NUMBER	1

Table 4-34: I/O PORT ØD9 HEX - TAPE DRIVE AND TRACK SELECT

BIT	BIT NAME	TRUE
0	TAPE UNIT SELECTED	0
1	TAPE UNIT BUSY	0
2	TAPE UNIT WRITE ENABLED	0
3	TAPE CARTRIDGE UNPROTECTED	0
4	TAPE AT BEGIN OF TAPE MARKER	0
5	TAPE PASSED EARLY END OF TAPE MARKER	0
6	TAPE AT LOAD POINT	0
7	TAPE UNIT READY	0

Table 4-35: I/O PORT 0DB HEX - TAPE DRIVE STATUS

BIT	BIT NAME	TRUE
0	MOVE TAPE REVERSE	1
1	MOVE TAPE FORWARD	1
2	MOVE TAPE AT HIGH SPEED	1
3	REWIND TAPE	1
4	COMMAND VALID STROBE	0
5	not used	
6	not used	
7	not used	

Table 4-36: I/O PORT 0DD HEX - TAPE MOTION CONTROL

4.9. Interrupt Controller

All maskable interrupts in the ACS 8600 are fully vectored. The vectors are generated by the Intel 8274 Serial Interface device and the Intel 8259 Programmable Interrupt Controller. Three 8259s are configured in a master slave fashion to resolve the priority of inputs processed by the 8259. The priority of the interrupts is listed below.

0. System Call
1. Time Slice Interrupts
2. Rigid Disk Interrupts
3. Flexible Diskette Interrupts
4. Interrupts from the 8087
5. 8089 Channel One Interrupts
6. 8089 Channel Two Interrupts
7. Parallel Interface Port A Interrupts
8. Parallel Interface Port B Interrupts
9. Multibus Interface Interrupt Level Zero
10. Multibus Interface Interrupt Level One
11. Multibus Interface Interrupt Level Two
12. Multibus Interface Interrupt Level Three
13. Multibus Interface Interrupt Level Four
14. Multibus Interface Interrupt Level Five
15. Multibus Interface Interrupt Level Six
16. Multibus Interface Interrupt Level Seven
17. Interrupt Request from Tape Controller
18. Interrupt Request from Intelligent I/O Controller #1
19. Interrupt Request from Intelligent I/O Controller #2
20. Interrupt Request from Serial Channels Zero and One

The 8259 may be programmed to disable specific interrupts or to acknowledge the end of a specific interrupt service routine. The following tables give the specific details of interrupt source and interrupt controller levels. Note that some interrupt request are assigned to a slave 8259 and therefore require two end of interrupt commands -- one for the master and one for the slave. As the 8259s are used in the ACS 8600 they are intended to be operated in edge triggered mode and there is no required sequence for issuing the two commands. For more specific detailes on the programming of the 8259 consult Intel literature on the 8259 or Application Note AP-59.

BIT	BIT NAME	ENA	DIS
0	SYSTEM CALL	0	1
1	SYSTEM TIMER	0	1
2	SLAVE INTERRUPT CONTROLLER # 1	0	1
3	SLAVE INTERRUPT CONTROLLER # 2	0	1
4	TAPE CONTROLLER INTERRUPT	0	1
5	I/O CHANNEL # 1 INTERRUPT	0	1
6	I/O CHANNEL # 2 INTERRUPT	0	1
7	CPU SERIAL CHANNELS (PORTS 048 - 04E)	0	1

Table 4-37: MASTER INTERRUPT CONTROLLER MASK REG. (PORT 05AH)
Control Register 1

BIT	BIT NAME	TRUE
0	INTERRUPT REQ. LEVEL BIT 0 (4-46)	1
1	INTERRUPT REQ. LEVEL BIT 1 (4-46)	1
2	INTERRUPT REQ. LEVEL BIT 2 (4-46)	1
3	MUST BE ZERO	0
4	MUST BE ZERO	0
5	COMMAND BIT 0 -- SEE TABLE 4-47	1
6	COMMAND BIT 1 -- SEE TABLE 4-47	1
7	COMMAND BIT 2 -- SEE TABLE 4-47	1

Table 4-38: MASTER INTERRUPT CNTLR. - COMMAND REG. 2 (PORT 058H)

BIT	BIT NAME	TRUE
0	READ REGISTER CMD 0 SEE TABLE 4-48	1
1	READ REGISTER CMD 1 SEE TABLE 4-48	1
2	POLL COMMAND	1
3	MUST BE ONE	1
4	MUST BE ZERO	0
5	SPECIAL MASK MODE 0 SEE TABLE 4-49	1
6	SPECIAL MASK MODE 1 SEE TABLE 4-49	1
7	NOT USED DON'T CARE	

Table 4-39: MASTER INTERRUPT CNTLR.- COMMAND REG. 3 (PORT 058H)

BIT	BIT NAME	ENA	DIS
0	HARD DISK INTERRUPT	0	1
1	FLEXIBLE DISKETTE INTERRUPT	0	1
2	8087 INTERRUPT	0	1
3	CHANNEL 1 OF 8089 INTERRUPT	0	1
4	CHANNEL 2 OF 8089 INTERRUPT	0	1
5	CH A PARALLEL PORT INTERRUPT (040H)	0	1
6	CH B PARALLEL PORT INTERRUPT (042H)	0	1
7	NONE	0	1

Table 4-40: SLAVE INTERRUPT CNTLR. 1 - MASK REG. (PORT 062H)
Control Register 1

BIT	BIT NAME	TRUE
0	INTERRUPT REQ. LEVEL BIT 0 (4-46)	1
1	INTERRUPT REQ. LEVEL BIT 1 (4-46)	1
2	INTERRUPT REQ. LEVEL BIT 2 (4-46)	1
3	MUST BE ZERO	0
4	MUST BE ZERO	0
5	COMMAND BIT 0 -- SEE TABLE 4-47	1
6	COMMAND BIT 1 -- SEE TABLE 4-47	1
7	COMMAND BIT 2 -- SEE TABLE 4-47	1

Table 4-41: SLAVE INTERRUPT CNTLR. 1 - COMMAND REG. 2 (PORT 060H)

BIT	BIT NAME	TRUE
0	READ REGISTER CMD 0 SEE TABLE 4-48	1
1	READ REGISTER CMD 1 SEE TABLE 4-48	1
2	POLL COMMAND	1
3	MUST BE ONE	1
4	MUST BE ZERO	0
5	SPECIAL MASK MODE 0 SEE TABLE 4-49	1
6	SPECIAL MASK MODE 1 SEE TABLE 4-49	1
7	NOT USED DON'T CARE	

Table 4-42: SLAVE INTERRUPT CNTLR. 1 - COMMAND REG. 3 (PORT 060H)

BIT	BIT NAME	ENA	DIS
0	MULTIBUS INTERRUPT REQUEST 0	0	1
1	MULTIBUS INTERRUPT REQUEST 1	0	1
2	MULTIBUS INTERRUPT REQUEST 2	0	1
3	MULTIBUS INTERRUPT REQUEST 3	0	1
4	MULTIBUS INTERRUPT REQUEST 4	0	1
5	MULTIBUS INTERRUPT REQUEST 5	0	1
6	MULTIBUS INTERRUPT REQUEST 6	0	1
7	MULTIBUS INTERRUPT REQUEST 7	0	1

Table 4-43: SLAVE INTERRUPT CNTLR 2 - MASK REG. (PORT 06AH)
Control Register 1

BIT	BIT NAME	TRUE
0	INTERRUPT REQ. LEVEL BIT 0 (4-46)	1
1	INTERRUPT REQ. LEVEL BIT 1 (4-46)	1
2	INTERRUPT REQ. LEVEL BIT 2 (4-46)	1
3	MUST BE ZERO	0
4	MUST BE ZERO	0
5	COMMAND BIT 0 -- SEE TABLE 4-47	1
6	COMMAND BIT 1 -- SEE TABLE 4-47	1
7	COMMAND BIT 2 -- SEE TABLE 4-47	1

Table 4-44: SLAVE INTERRUPT CNTLR. 2 - COMMAND REG. 2 (PORT 068H)

BIT	BIT NAME	TRUE
0	READ REGISTER CMD 0 SEE TABLE 4-48	1
1	READ REGISTER CMD 1 SEE TABLE 4-48	1
2	POLL COMMAND	1
3	MUST BE ONE	1
4	MUST BE ZERO	0
5	SPECIAL MASK MODE 0 SEE TABLE 4-49	1
6	SPECIAL MASK MODE 1 SEE TABLE 4-49	1
7	NOT USED DON'T CARE	

Table 4-45: SLAVE INTERRUPT CNTLR. 2 - COMMAND REG. 3 (PORT 068H)

BIT 2	BIT 1	BIT 0	INTERRUPT REQUEST LEVEL ACTED UPON
0	0	0	INTERRUPT REQUEST LEVEL ZERO
0	0	1	INTERRUPT REQUEST LEVEL ONE
0	1	0	INTERRUPT REQUEST LEVEL TWO
0	1	1	INTERRUPT REQUEST LEVEL THREE
1	0	0	INTERRUPT REQUEST LEVEL FOUR
1	0	1	INTERRUPT REQUEST LEVEL FIVE
1	1	0	INTERRUPT REQUEST LEVEL SIX
1	1	1	INTERRUPT REQUEST LEVEL SEVEN

Table 4-46: INTERRUPT REQUEST LEVEL BIT DEFINITION

BIT 2	BIT 1	BIT 0	COMMAND
0	0	1	NON SPECIFIC END OF INTERRUPT
0	1	1	SPECIFIC END OF INTERRUPT **
1	0	1	ROTATE ON NON SPECIFIC EOI
1	0	0	ROTATE IN AUTOMATIC EOI MODE (SET)
0	0	0	ROTATE IN AUTO EOI MODE (CLEAR)
1	1	1	ROTATE ON SPECIFIC EOI **
1	1	0	SET PRIORITY COMMAND **
0	1	0	NO OPERATION

Table 4-47: INTERRUPT CONTROLLER COMMANDS
** Uses Interrupt Request Level Bits (Table 46)

BIT 1	BIT 0	READ REGISTER COMMAND
0	1 or 0	NO OPERATION
1	0	READ INT REQUEST REG ON NEXT READ
1	1	READ INT SERVICE REG ON NEXT READ

Table 4-48: READ INTERRUPT CONTROLLER REGISTER COMMANDS

BIT 1	BIT 0	SPECIAL MASK MODE
0	1 or 0	NO OPERATION
1	0	RESET SPECIAL MASK MODE
1	1	SET SPECIAL MASK MODE

Table 4-49: INTERRUPT CONTROLLER SPECIAL MASK MODE CONTROL

4.10. Multibus Compatible Expansion Interface

The ACS 8600 system provides a pair of rear panel connectors that permit connection to an additional enclosure containing a standard Multibus card cage. This expansion capability allowes system users to add special or custom application dependent interfaces to the system. The interface supports both eight bit and sixteen bit bus masters that use the daisy chain technique for resolving the priority of bus access. The interface supports the addressing of up to one megabyte of memory. Since the ACS 8600 has provisions for a full megabyte of memory, the Multibus interface does not permit the main CPU to access any memory outside of the ACS 8600. To fully protect the integrity of the ACS 8600, the operating system and the other users that may be using the system, the Multibus interface does not permit bus masters external to the ACS 8600 to access any I/O ports internal to the ACS 8600. The design of the Multibus interface includes socketed termination resistors for all Multibus signals that must be terminated on one of the bus masters. All signals in the interface meet the electrical and driver-receiver requirements of the Multibus Specification published by Intel with two exceptions. The open collector signals Bus Busy (BUSY/) and Common Bus Request (CBRQ) have only twenty milliamperes of drive instead of the thirty-two milliamperes of drive specified in the specification. The two clocks supplied to the Multibus Interface (BCLK/ and CCLK/) are 9.8304 megahertz. One further limitation that must be noted is that the Multibus specification limits total bus length to twelve inches. As the ACS 8600 is currently packaged approximately eight inches of cable are required to run from the CPU Board to the rear panel. The Multibus signals listed on the next page are supported by the interface.

Address Lines - ADR0/ through AD13/

Data Lines - DAT0/ through DATF/

Control Lines - IORC/, IOWC/, MRDC/, MWTC/, INTA/ and XACK/

Interrupt Lines - INT0/ through INT7/

Clock Lines - BCLK/ and CCLK/

System Initialization Line - INIT/

5. PHYSICAL, ELECTRICAL AND ENVIRONMENTAL SPECIFICATIONS

The lists on this page and the next page give the electrical and environmental requirements of the ACS 8600 system. These figures are preliminary estimates and should be taken as estimated worst case maximum values.

Main System Enclosure

Size
19" Wide, 23" Deep, 7" High

Weight
60 Pounds - Free standing weight
80 Pounds - Shipping weight

Front Panel Controls
AC ON/OFF Switch
RESET Switch

Rear Panel Connections
AC Power
Fuse
Console RS-232C Connectors (DB 25)
Parallel Interface Connector (DB 37)
Multibus Interface Connector (72 Pins)
Magnetic Tape Expansion (50 Pin)
Rigid Disk Expansion (50 Pin & 20 Pin)

Maximum AC Inrush Current
8.5 A at 115 V, 60 Hz (3 sec)

Maximum Input Power
4 A at 115 V, 60 Hz

Power Dissipation
300 Watts
1024 BTU per hour

Ambient Temperature Range
59 to 90 Degrees Farenheit
15 to 32 Degrees Celsius

Relative Humidity
20% to 80% Non-condensing

Peripheral Enclosure

Size
19" Wide, 23" Deep, 7" High

Weight
55 Pounds - Free standing weight
75 Pounds - Shipping weight

Front Panel Controls
AC ON/OFF Switch

Rear Panel Connections
AC Power
Fuse
Expansion Connectors (Depends on peripherals used -- refer to Main System Connectors)

Maximum AC Inrush Current
6 A at 115 V, 60 Hz (3 sec)

Maximum Input Power
3 A at 115 V, 60 Hz

Power Dissipation
275 Watts
940 BTU per hour

Ambient Temperature Range
59 to 90 Degrees Farenheit
15 to 32 Degrees Celsius

Relative Humidity
20% to 80% Non-condensing

6. SOFTWARE

Altos Computer Systems offers the following operating systems for use on the ACS 8600 system:

Single User
CP/M-86

Multiple User
XENIX
OASIS-86
MP/M-86

Altos Computer Systems offers the following languages for support of end user applications on the ACS 8600 system:

CIS-COBOL
RM-COBOL
PASCAL/M-86
C-BASIC-86
MICROSOFT COBOL
MICROSOFT PASCAL
MICROSOFT BASIC
MICROSOFT FORTRAN

A. RIGID DISK CONTROLLER DOCUMENTATION

WARNING: THE BOARD DESCRIBED IN THIS SPECIFICATION IS USED ON SEVERAL SYSTEMS.

WARNING: THIS SPECIFICATION MAY DESCRIBE FEATURES THAT ARE NOT IMPLEMENTED OR NOT AVAILABLE ON ACS 8600 SYSTEM.

WARNING: I/O PORTS REFERENCED IN THIS SPECIFICATION MAY NOT AGREE WITH THE I/O PORTS DEFINED IN THE SECTION 4.2 OF THE ACS 8600 SYSTEM SPECIFICATION. THE I/O PORTS DEFINED IN SECTION 4.2 ARE CORRECT WHILE I/O PORTS IN THIS SPECIFICATION REFER TO ANOTHER SYSTEM THAT UTILIZES THIS BOARD.

USER INFORMATION: EIGHT INCH SOFT SECTORED DRIVES AND THE ALTOS CONTROLLER

Specifications for these drives are as follows:

Model Number	Disks	Heads	Cylinders	Total Capacity
Shugart				
SA1002	1	2	256	5 megabytes
SA1004	2	4	256	10
Quantum				
Q2010	1	2	512	10
Q2020	2	4	512	20
Q2030	3	6	512	30
Q2040	4	8	512	40

Aside from capacity, there are two significant differences between these drives: the Quantum drives have twice the number of cylinders that the Shugart drives have and the Quantum drives have a REZERO capability that is neither provided for nor needed on the Shugart drives.

On the controller there is a jumper at S1 that needs to be set to coincide with the type of hard disk in the system. This must be set to S for the Shugart drives and to Q for the Quantum drives. This jumper sets the cylinder number at which write precompensation takes place (at cylinder 128 for the Shugart drives and 256 for the Quantum).

Data is stored on the disks in blocks or sectors of either 256 or 512 bytes (not programmable) identified by cylinder, head and sector numbers. The address for each sector is written previously (by a FORMAT command) ahead of the data field. When reading from or writing to the disk, this address is read and verified before the data field is read or written. A correct match must be found for the address and its check bytes before the sector can be written or read.

The general procedure to read or write data is to first position the heads over the desired cylinder via a SEEK command, load the head and sector numbers into the appropriate ports and finally issue a READ or WRITE command. When the correct sector is found, the data is transferred between disk and system memory under control of the DMA by way of a FIFO buffer.

The controller is addressed by way of four I/O ports (020H-023H).

Addresses:

Port	Read	Write
020H	X	Drive and head numbers
021H	Data	Old cylinder number (in two bytes), sector number and data
022H	X	New cylinder number (in two bytes)
023H	Status	Commands

COMMANDS

Port 023H is the command port when writing and the status port when reading. There are six commands that may properly be issued:

RESET initializes the controller circuit. A reset is done automatically at power on and is not normally required thereafter.

RECAL commands the heads to be positioned over track zero. This must be done for each drive following power on before any disk access is attempted and thereafter as part of the error recovery routine. A recal requires about 15ms (Quantum) to 18ms (Shugart) per cylinder for a maximum time of about 8 seconds for the Quantum drives and about 4 seconds for the Shugart drives.

SEEK causes the heads to move from the programmed "old cylinder" to the programmed "new cylinder".

READ is the command to read the addressed sector off of the disk.

WRITE is the command to write to the addressed sector on the disk.

FORMAT is the command to format an entire track. This consists of writing out the address or header for each sector on the specified (by cylinder and head) track.

For the Quantum drives there is an additional command, REZERO. This causes the Quantum drives to do an internal recalibration of the head positioning circuitry. This takes about 2 seconds to complete.

Commands (Port 023H):

Bit number	7	6	5	4	3	2	1	0	Hex
NULL	0	0	0	0	0	0	0	0	00
READ	0	0	0	0	X	0	0	1	01
WRITE	0	0	0	0	X	0	1	0	02
FORMAT	0	0	0	0	X	1	0	0	04
SEEK	0	0	0	1	X	0	0	0	10
RECAL	0	0	1	0	X	0	0	0	20
REZERO*	0	1	0	0	X	0	0	0	40
RESET	1	0	0	0	X	0	0	0	80

No more than one command may be issued at one time. For all the commands except NULL and RESET, when the command has been completed, the controller will issue an interrupt (via the PIO at port 008H bit 7). Following this interrupt, the status port must be read. This clears the status register and is needed to permit subsequent interrupts. If more than one command is outstanding (a second issued before the first interrupts back), both commands will be aborted.

STATUS

The status byte (which should be read at the completion of every operation as noted above) reports on the status of the drive or controller.

READY indicates that the selected drive is powered and up to speed. No operation will take place until a drive is selected and ready is true. Normally being READY indicates that the drive is ready to be read from or written to but a special case exists with the Shugart drives: upon being powered on these drives require a two minute stabilization time before they should be read from or written to. Unfortunately, READY will go true before these two minutes are up (in about five seconds) and does not allow for this warm-up time.

WRITE FAULT indicates a disk error during writing. To clear a write fault, the selected drive must be de-selected then re-selected.

CRC ERROR indicates that either the header or data field read off the disk was in error. If the operation was a write, the error was incurred in reading the header and the sector was not written to. If the operation was a read, then the error could have occurred in either the header or data field. If it occurred in the header, the operation was aborted and the data field was not read. If it occurred in the data field, the data was already transferred before the error was detected.

RECORD NOT FOUND indicates that during a read or write operation the specified header was not found in four revolutions of the disk. This may be caused by a seek error or incorrectly

specified head or sector. RNF during a read may also indicate that the data sync field (zeros, "A1" or "F8") was not found even after the header was located. This is most likely caused by trying to read a sector after it had been formatted, but before it was written to.

BAD SECTOR indicates that the sector to be read or written was previously specified as a bad sector. A sector is flagged as bad during formatting with a one at bit 3 of the head byte. Good sectors have a 0 there.

TASK COMPLETE indicates that the requested command was carried out or aborted. Analysis of the status byte will determine whether the operation was aborted.

BUSY indicates that the controller is in the process of carrying out an operation. This signal will be a 0 at the completion of an operation.

Status byte (Port 023H):

Bit	Mnemonic	Meaning
7	RDY	Selected drive is ready
6	WR FLT	Write fault (from drive)
5	(none)	
4	CRC ERR	CRC error encountered
3	RNF	Record not found
2	BD SECT	Bad sector encountered
1	TC	Task complete
0	BUSY	Busy

OTHER PORTS

Port 022H is for writing the new cylinder number. This is entered in two consecutive bytes, the low order eight bits first then the high order three bits.

Cylinder bytes:	7	6	5	4	3	2	1	0
Least significant	X	X	X	X	X	A	9	8
Most significant								

Port 021H has multiple functions. Before doing a SEEK the old cylinder is entered (in two bytes as for the new cylinder). Before doing a READ or WRITE the sector number is entered into this port, and when data is being transferred between disk and memory it is done through this port.

Port 020H is for selecting the drive and head to be used. The heads are selectable in a binary sequence (limited by the number of heads on the drive). Either one or the other but not both drives should be selected.

Drive/head byte (Port 020H):

Bit	Name	Function
7	HS3	Head select bit 3
6	HS2	Head select bit 2
5	HS1	Head select bit 1
4	HS0	Head select bit 0
3		(none)
2		(none)
1	DS2	Drive select 2
0	DS1	Drive select 1

FORMATTING

As this drive is soft-sectored, an entire track must be formatted at one time to keep the proper timing relationship among the sectors. Formatting of a track destroys any data on the track, so if the header for a sector ever needs to be changed (i.e. to mark a sector as a BAD SECTOR) all the data on that track will need to be backed off of the disk before formatting and then restored afterward.

The formatting procedure is as follows: first the heads must be positioned to the desired cylinder using the SEEK command. Then a header image consisting of four bytes: an "0FEH", cylinder number, head number, and sector number must be placed in memory and written out to the disk by the DMA as each sector arrives under its head during one rotation of the disk. The cylinder number in the header must agree with the cylinder that the heads are on and the head number must agree with the selected head. However, the sector number can be any unique number between 0 and 255. This allows for staggering the sectors if required. The DMA must be programmed for a four byte continuous transfer.

Header Image:

Byte name	Description
Sync byte	"0FEH"
Cylinder byte	Low order bits (7-0) of the cylinder number
Head byte	Head bits (3-0) in the top four bits, a bad sector flag at bit 3, and high order cylinder bits (A-8) at the bottom three bits
Sector byte	Sector number in hex

To write out the header, the FORMAT command must be issued. When that header has been written out, an interrupt is generated. Then the header image must be updated to the next sector number (only the sector numbers change down the track) and a new FORMAT command issued before the next sector arrives at the head (464 microseconds for a 256 byte sector). The program must also keep count of the number of sectors that have been formatted and stop issuing the FORMAT command when the last one for that track has been written. When the interrupt occurs following the writing of the last sector's header, the status must be read and when the next interrupt occurs, formatting of that track is complete and another command may be issued. (Following the next to last interrupt, a dummy or null command may be issued if necessary).

To format the entire disk, the above procedure must be repeated for each head of each cylinder.

Two sector sizes are provided for: 256 bytes and 512 bytes. One track will accommodate 31 sectors at 256 bytes and 17 at512. The sector size is determined by jumpers on the controller at S2. Two jumpers are used. Both must be in either the 256 or the 512 position.

At the time of formatting, the controller fills in every byte of the data field with "Ø4EH". Following the formatting of a track, any attempt to read a sector on that track that has not been specifically written to will result in a record not found (RNF) as the sync bytes will be missing for these sectors. Each sector must subsequently be written to with some other data (i.e. "ØE5H" to initialize the sector for CP/M, MP/M or AMEX). Any data stored on a track gets destroyed when that track is formatted.

The sequence of sectors on a track is under control of the formatting program. Physically adjacent sectors should not be read or written sequentially. This is because the data is transferred between disk and main memory under DMA control. System memory (composed of dynamic RAMs) requires periodic clocking to maintain its data. This refreshing is normally done by the CPU, but when the DMA is active, the CPU's refresh cycles are held off. To allow the memory to get properly refreshed, then, the CPU must regain control and generate the refresh cycles to keep the memory alive.

The soft sectoring of the drive precludes the ability to perform a specific READ ID command. This is because the location of the sector on a track is unknown. When READING or WRITING, a sector is located by reading all the headers encountered along the track until the correct one is found. In performing a READ, when the correct header is found, the header bytes are read off the disk and transferred into memory. As a diagnostic, this can be used to verify that the data read was from the specified sector.

READING OR WRITING A SECTOR

WARNING: upon being powered on, the Shugart drive requires a two minute stabilization time before reading or writing. The READY signal from the drive will go true about five seconds following power-on, so it is up to the user to wait these two minutes. The Quantum drive will go ready in about ten seconds. It is not yet known whether it will require a similar stabilization time.

Assuming the drive has been properly initialized following power on by selecting and RECALING the drive, seek to the desired cylinder by loading the old cylinder number at port 021H (following a RECAL this would be 000H), loading the new cylinder number at port 022H, then issuing a SEEK command. An interrupt will occur when the head has arrived at the new cylinder and settled down (i.e. no additional head settle time is required).

To write or read a sector, the sector must first be found. The controller does this by comparing the header images (which are written on the disk during formatting) as they pass under the head to the bytes programmed into ports 020H-022H (020H has the head number, 021H has the sector number and 022H has the new cylinder number). As the sector register gets overwritten by data during a write, before the next read or write command the sector number must be reissued to port 021H. Also, in switching between drives, even if a seek is not necessary port 022H must be reloaded with the new cylinder number.

To read a sector, prepare a buffer area in main memory to which the data from disk can be placed. Next, program the DMA to transfer the desired number of bytes (either 256 or 512 bytes for WRITE or 259 or 515 for READ) from the controller's I/O port (021H) to the buffer in memory using the CONTINUOUS mode and standard memory and I/O timing (to properly pace the FIFO). In doing a READ there are additional bytes that are read off the disk: the three bytes of header information that preceded the data (cylinder, head/drive and sector, just as they are written out during formatting). Along with the 256 or 512 bytes of data, these three bytes must be transferred by the DMA. There are also two bytes of CRC that follow the data and these may but do not need to be transferred. Also note that the DMA transfers one more byte than programmed, so that the number of bytes the DMA is programmed to transfer will end up being two or four bytes longer than the sector length.

Next, load the address (cylinder, head/drive and sector) of the sector to be read. The cylinder number should be already correct from having done a SEEK. Load the head number to port 020H without changing the drive number (they share the same byte). Then load the sector number at port 021H. (A 20microsecond delay is required between selecting a drive or a head and issuing a read or write command, but the inherent programming delays normally take care of this). Finally, issue the READ command.

When the drive/controller locates the desired sector, the controller signals the DMA and the data transfer is carried out by the DMA directly into memory. When the transfer is complete, the controller generates another interrupt and clears out the READ command. The status register must then be read. If a RNF is indicated, the data was not transferred to memory. If a CRC ERR or BAD SECTOR is indicated, the data was transferred but its accuracy is questionable.

To WRITE a sector, a similar procedure is used with the buffer area first being filled with the data to be written to the disk. The header and CRC bytes are provided by the hardware, so only the data bytes need be transferred by the DMA. Again, the status register must be read. If a RNF, CRC ERR or BAD SECTOR is indicated the data was not written out to disk. When writing, all the sync bytes are provided by the controller (in contrast to the technique used for the SA4000 which required a sync byte ahead of the data). However, when FORMATTING the drive, a sync byte of "0FEH" must preceded the header bytes.

CONTROLLER RESPONSES

When a command is issued without a drive being ready or selected, the controller will wait until the selected drive becomes ready before the operation is carried out. Then at the completion of the operation an interrupt is generated.

RESET does not generate an interrupt neither does selecting or deselecting a drive.

Reading and writing from/to the disk cannot be done by the CPU because of the high data rate. Except for some diagnostic tests, the data must be transferred by the DMA.

To prevent extraneous writes to the disk when the power goes off, the controller has a sensing circuit that deselects the drives when the +5 volt supply falls below +4.5 volts. This circuit does not guarantee, however, that data being written to a sector will not be lost if power goes off. To insure against such a loss, power down the system only when no writing is being done to the disk.

BAD SECTORS

As noted before, this controller has the capability of flagging bad sectors. A sector may be flagged as being bad by entering the appropriate bit (bit 3 in the head byte - see the header image info on page 3) in the header during the format operation. Thereafter, when such a sector is accessed, writes and reads will be aborted and a BAD SECTOR status reported. Note that a BAD SECTOR status is not reported just because the accessed sector is bad, but because it has been flagged as being bad. The usual status returned when reading a sector that has a defect but

is not flagged is a CRC error.

The reason for flagging a sector as being bad is that the recording media occasionally contains flaws that do not permit reliable writing and reading in certain areas. By marking those sectors containing known flaws as BAD SECTORS the diagnostic that tests the disk drive can distinguish between defective media and a failure of the drive or controller to read properly.

Each drive manufacturer does an analog (and therefore very sensitive) test of each drive that produces a "media scan" record indicating where the defective areas are. This map can be used to identify the sectors that should be marked as BAD SECTORS and that the operating system should not attempt to use.

Shugart provides an SA1004 MEDIA ERROR MAP with each drive which gives the track number (TRK), head number (HD), beginning location with respect to the index pulse (BYTE COUNT) and length of the defect in bits (LENGTH [BITS]). To convert these numbers to nomenclature that the controller recognizes, only the sector number needs to be deciphered. The cylinder number is the same as Shugart's track number and the head number needs no translation.

The number of the bad sector(s) is determined from the table below. This gives the range in bytes for each sector relative to the index pulse. The defective area given by the media scan spans from the BYTE COUNT to the BYTE COUNT plus the result of LENGTH divided by 8. Any sector whose range falls within the defective area is bad. Defective areas may extend across more than one sector. In addition, the possible speed variation of the disk during formatting (+/-3%) may result in an ambiguity as to which sector a defect will fall into. This is reflected in the table where the sector ranges overlap. In such a case, all potentially defective sectors need to be identified. This table assumes no skewing and 512 byte sectors.

CONVERSION TABLE - BYTE COUNT TO SECTOR NUMBER

BYTE COUNT	PHYSICAL SECT NO	LOGICAL SECT NO
15- 596	0	0
588- 1204	1	6
1162- 1813	2	12
1735- 2422	3	1
2308- 3031	4	7
2881- 3639	5	13
3455- 4248	6	2
4028- 4857	7	8
4601- 5466	8	14
5175- 6074	9	3
5748- 6683	10	9
6321- 7292	11	15
6894- 7901	12	4
7468- 8509	13	10
8041- 9118	14	16
8614- 9727	15	5

9187-10335 16 11

To avoid the delay in seeking to a sector only to find that it is bad, and to get around the problem that may occur if a defect occurs in the header of a sector (in which case the BAD SECTOR flag may not be recoverable but rather a RNF status would be returned), it has been proposed that cylinder 0 (guaranteed to be defect-free) be used to store the bad sector table which the operating system would check before writing or reading a sector. (See ALTOS Format for Hard Disk Systems 11/19/80). Another approach is to assign dummy files to the sectors that are bad so that the operating system would not try to access them.

DISK DRIVE SPECIFICATIONS

Nominal track capacity:		10416 bytes
Min track capacity (-3% speed):		10102 bytes
Formatted capacity (at 512 bytes):		
	SA1002	4,456,448 bytes
	SA1004	8,912,896 bytes
	Q2010	8,912,896 bytes
	Q2020	17,825,792 bytes
	Q2030	26,738,688 bytes
	Q2040	35,651,584 bytes
Transfer rate:		230 nanosec per bit 1.84 microsecper byte
Access time:		
Track to track:	(Shugart)	19ms
	(Quantum)	15ms
Average:	(Shugart)	70ms
	(Quantum)	50ms
Maximum:	(Shugart)	150ms
	(Quantum)	100ms
Average latency	(Shugart)	9.6ms
	(Quantum)	10ms

Disk Data Format:

No of bytes	Character	
(Start at Index Pulse)		
16	"4E"	
14	"00"	Written out 31 times for 256 byte sectors or 17 times for 512 byte sectors during format
1	"A1*"	
1	"FE"	
1	cylinder byte	
1	head byte	
1	sector byte	
2	CRC bytes	
10	"00"	
12	"00"	Written out during a write operation
1	"A1*"	
1	"F8"	
256 or 512	(Data)	
2	CRC bytes	
3	"00"	
29	"4E"	
(as reqd)	"4E"	
(Stop at Index Pulse)		

6/05/81 WS

USER INFORMATION: EIGHT INCH HARD DISK CONTROLLER FOR THE ACS8600 (ASSEMBLY NUMBER 11123)

To permit this controller to operate with both the Z80 and the 8086/8089 systems, the original circuit (assembly number 10635) was modified to delay the DMA ready signal during data transfers. So that the controller may be used with both of these systems, a jumper selection has been added to adjust the DMA ready signal. For a Z80 system, the jumper at S3 should be set to 2, and for an 8086/8089 system it should be set to 1. This is the only change that is necessary to allow the controller to move from a Z80 to an 8086/8089 system.

In addition to the above, the following functional changes were made to the operation of the controller:

1) A new command has been defined, read without header bits. This command allows a sector to be read off of the disk without the data being preceded by the cylinder, head and sector numbers. This new command is READWOHD, 00001001 or 09H. The original READ command, 01H, works as it did before in transferring three bytes of sector ID before the data. These ID bytes are invaluable in running diagnostics on the drive/controller.

2) Status bits (TASK COMPLETE, BAD SECTOR, RNF and CRC ERR) do not get cleared by reading status unless TASK COMPLETE is true at the beginning of the read cycle. Thus, polling status no longer runs the risk of hanging up the controller.

3) BUSY no longer depends upon drive READY (which in turn depends upon a drive being selected). BUSY is now only a controller status bit. It is true if the controller is in the process of doing a READ, WRITE, FORMAT, SEEK or a RECAL.

4) The write precompensation circuit has been changed to allow for individual compensation of each hard disk drive in a two drive system. Thus the external drive does not have to be the same make as the internal drive as is the present requirement. This is handled by a second jumper at S1. S1a sets the compensation for drive 1, and S1b sets it for drive 2.

A couple of miscellaneous items not covered in the previous write up on the controller: 1) initiating a SEEK or RECAL without the drive being READY can hang up the controller in that no interrupt will be returned, and 2) because port 021H is multiplexed (used for Old Cylinder Number, Sector number and data), it is necessary to insure that the appropriate byte is in this register for the operation in process. Specifically, old

cylinder should be there prior to SEEKING and sector number prior to a READ or WRITE. This needs to be adhered to even during error recovery.

11/05/81 WS

B. INTELLIGENT I/O CHANNEL DOCUMENTATION

WARNING: THE BOARD DESCRIBED IN THIS SPECIFICATION IS USED ON SEVERAL SYSTEMS.

WARNING: THIS SPECIFICATION MAY DESCRIBE FEATURES THAT ARE NOT IMPLEMENTED OR NOT AVAILABLE ON ACS 8600 SYSTEM.

WARNING: I/O PORTS REFERENCED IN THIS SPECIFICATION MAY NOT AGREE WITH THE I/O PORTS DEFINED IN THE SECTION 4.2 OF THE ACS 8600 SYSTEM SPECIFICATION. THE I/O PORTS DEFINED IN SECTION 4.2 ARE CORRECT WHILE I/O PORTS IN THIS SPECIFICATION REFER TO ANOTHER SYSTEM THAT UTILIZES THIS BOARD.

Intelligent I/O Controller Specification

Date: 1/28/82

By: Norm Kelly

INTRODUCTION

The Intelligent I/O board consists basically of a Z80 processor with eight serial channels under its control. The processor communicates with the host via main memory and a pair of interrupt lines. The host may interrupt the Z80 and the Z80 may interrupt the host. This board is intended to offload the host processor such that the majority of character interrupt processing may be handled by the Intelligent I/O. This should allow the host processor to devote more of its power to higher level tasks, and possibly even handle more users in a more timely fashion.

FUNCTIONAL DESCRIPTION

The Intelligent I/O controller consists of a Z80A microprocessor, four Z80A SIO dual serial controllers, 4KBytes of EPROM, 2 Kbytes of static ram, 2 AM9513 timer ICs, a Multibus interface, and associated control logic. The hardware supports eight serial channels capable of operating in asynchronous mode at all standard baud rates between 110 baud and 38.4 Kbaud. Each serial channel has a corresponding programmable timer for baud rate generation to allow for independent baud rates. The EPROM is intended to hold the control program for the Z80A, since execution from main memory is not possible. Extra control logic is required for the Multibus interface to allow the full 16 Megabyte addressing provided by the bus specification.

Interrupts on Z80A

The Z80A processor may be interrupted by a total of 10 sources. The eight serial channels are all connected to the INTR line, and each can be considered a source of interrupts. The four SIO devices are daisy-chained together such that the Z80A vectored interrupt mechanism may be used to differentiate these interrupts. This assumes that the SIOs are programmed to generate vectors, and that the Z80A is programmed for interrupt MODE 2.

Two other interrupt sources are tied to the NMI pin. These two sources consist of a decoded address from the host processor expected to be used as a channel attention, and a 'memory error' signal generated by hardware in the main memory controller when an uncorrectable memory error is detected while accessing main memory. The memory error signal is also connected to the SOURCE 4 input of the second AM9513 timer IC. This allows differentiation between the two NMI sources by polling the timer to see if a memory error has occurred.

BAUD RATE TIMERS

Each serial channel has a dedicated timer for generating its transmit/receive baud rate clock. The ICs used for these timers are AM9513 system timing controllers. The crystal inputs of both timer chips are driven at 1.8432 MHz. This input frequency allows operation of the serial channels at all standard baud rates between 50 baud and 38.4 Kbaud. The divisors to be used for these rates are listed in the table below.

BAUD RATE	DIVISOR
38.4 K	3
19.2 K	6
9600	12
7200	16
4800	24
3600	32
2400	48
2000	58
1800	64
1200	96
600	192
300	384
150	768
135	853
110	1047
75	1536
50	2304

NOTE: These values assume that the SIO channels have been programmed for clock rates 16 times the baud rate.

In addition to the standard baud rates available as listed above, serial channel #6 has provisions for external clocking on both transmit and receive to support Bisync.

MULTIBUS INTERFACE

A Multibus interface on the Intelligent I/O board allows communication with the host processor via main memory. All memory accesses above the address of 32K (8000H), are mapped into main memory. Address bit 15 from the Z80A is used for decoding this mapping, while all of the lower address bits (A0 - A14) are passed to the bus. All upper address bits on the bus (A15 - A23) must be generated by external hardware. A write-only port (HIGH BYTE) is used to generate the upper 8 address lines (A16 - A23), while a bit out of the control port is used for address bit 15. This allows the Intelligent I/O to access all of the memory in the Multibus memory address space of 16 megabytes. In addition, a bit out of the control port is used to lock the system bus when a test-and-set function is required. This bit causes the Multibus Arbiter to lock the bus whenever possession is gained (i.e., either immediately or on the next Multibus access). This means that no other bus master will be allowed on the bus until this bit is reset. Caution is advised when using this function as system throughput may be affected.

I/O PORT ADDRESSES

Serial Channel 0:	
Data Register	0000H
Command Register	0001H
Serial Channel 1:	
Data Register	0002H
Command Register	0003H
Serial Channel 2:	
Data Register	0004H
Command Register	0005H
Serial Channel 3:	
Data Register	0006H
Command Register	0007H
Serial Channel 4:	
Data Register	0008H
Command Register	0009H
Serial Channel 5:	
Data Register	000AH
Command register	000BH
Serial Channel 6:	
Data Register	000CH
Command Register	000DH
Serial Channel 7:	
Data Register	000EH
Command Register	000FH
Timer (9513) 0:	
Data Register	0010H
Command Register	0011H
Timer (9513) 1:	
Data Register	0014H
Command Register	0015H
Control Port (write only):	0018H
High Address Port (write only):	001CH

Note: No other I/O addresses may be used, as these addresses are not uniquely decoded and other addresses may cause conflicts.

SERIAL CHANNEL DESCRIPTIONS

Serial Channel	SIO	Port	AM9513	Timer	Priority
0	0	A	0	1	Highest
1	0	B	0	2	
2	1	A	0	3	
3	1	B	0	4	
4	2	A	0	5	
5	2	B	1	1	
6	3	A	1	2	
7	3	B	1	3	Lowest

CONTROL PORT BIT ASSIGNMENTS

BIT	NAME	FUNCTION
Ø	ADR15	This bit is used to replace A15 from the Z8Ø when accessing the Multibus.
1	INTOUT	This bit is used to generate an interrupt to the host processor. Since the interrupts are edge triggered, this bit should be toggled to guarantee that a new interrupt is seen by the host.
2	LOCK	This bit is used to lock the Multibus so that a test-and-set function may be implemented. It must be reset to allow other masters to acquire the bus.

APPENDIX A.

ALTOS WARRANTY, SHIPPING DAMAGE INSPECTION, AND REPAIR PROCEDURES

ALTOS COMPUTER SYSTEMS 90-DAY, LIMITED WARRANTY

Altos Computer Systems warrants each of its products to be free from defects in materials and workmanship for a period of 90 days from date of purchase by the end-user. During the warranty period, Altos, at its option, will repair or replace components in the products that prove to be defective at no charge other than shipping and handling, provided the product is returned prepaid to an authorized Altos Service Center or to Altos Computer Systems, 2360 Bering Drive, San Jose, California 95131.

This warranty will not be effective if, in the opinion of Altos Computer Systems, the Altos product has been damaged by accident, misuse, misapplication, or as a result of service or modification by other than an authorized Altos service center.

This warranty is expressly exclusive and in lieu of all other warranties or guarantees either express or implied. including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. In no event shall Altos be liable for lost profits, loss of good will, or any other special or consequential damages.

Return of the End-User Registration Card is required for this warranty to be valid. Warranty claims are void unless the End-User Registration Card and Dealer Registration Card are on file at Altos Computer Systems.

SHIPPING DAMAGE INSPECTION

If an Altos computer is received which has been damaged in shipment, or the possibility that it could have been damaged judging by the appearance or condition of the shipping container, note it on the waybill and require the delivery agent to sign the waybill. Notify the transfer company immediately and submit a damage report to the carrier, your dealer, and to Altos Computer Systems., Customer Service Dapartment, Attention: CLAIMS. If no exterior damage exists, unpack the microcomputer and inspect for hidden damage. All insurance claims for damage must be filed by the shipper.

REPAIR AND SERVICE PROCEDURES

If you experience technical problems with your Altos computer, perform the steps in the following procedure:

1. Contact your Authorized Reseller for assistance. They will be able to fill most of your repair or service needs.

   If your Authorized Reseller is unable to assist you and you are located in the United States, perform either step 2 or 3.

2. Contact Moore Business Systems for nationwide service.

   Moore Business Systems
   13615 Welch Road
   Suite 100
   Dallas, TX 75234

   Phone: (800) 527-1058 (National Watts Line)
   (800) 442-1160 (Texas Watts Line)

   They will be able to offer you on-site service contracts, on-site time and material services, and depot service for your computer both under warranty and out of warranty. Call the toll-free number for price information and procedures.

3. Contact Altos Computer Systems Customer Service Department for factory depot service.

   Altos Computer Systems
   Customer Service Department
   2360 Bering Drive
   San Jose, CA 95131
   (408) 946-6700
   Telex: 171562 ALTOS SNJ or 470642 ALTO UI

ALTOS FACTORY REPAIR PROCEDURES

Follow the steps in the following procedure to obtain Altos factory repair services:

1. Contact Customer Service for assistance in ensuring that your system needs factory service, and to minimize the occurance of "no problem found" when your system is returned for repair.

2. Obtain Warranty or standard Return Authorization Number. At the time the Return Authorization Number is assigned you will need to provide the following information:

   a. Model Number of your system.
   b. Serial Number
   c. Date Purchased
   d. Specific Problem
   e. Purchase Order Number or Authorization for any repair charges
   f. Name, address and telephone/telex number of your company and name of a responsible technical person whom Altos Customer Service may contact in the event of any question or problems

3. Ship defective Altos computer in original shipping container, with the Return Authorization Number clearly labeled on the outside of the container. Include a detailed description of the symptoms of the malfunction, and any special instruction. Follow all shipping and packing instructions contained in your user manual.

   a. Warranty freight:

      The End-User pays freight to Service Center.Altos Service Center pays return freight via like carrier

   b. Non-Warranty Freight:

      The End-User pays freight both ways

CAUTION

Alto Customer Service is not responsible for the loss of data on any system returned for repair. A system returned for repair will be fully tested, and this testing will destroy all data on any hard disk product.

APPENDIX B.

TROUBLE SHOOTING PROCEDURES

(THESE PROCEDURES WERE IN DEVELOPMENT AT THE TIME THIS REVISON WAS PRINTED.)

APPENDIX C.

COMMON CRT TERMINAL AND PRINTER INTERFACE CONFIGURATIONS

(THESE PROCEDURES WERE IN DEVELOPMENT AT THE TIME THIS REVISON WAS PRINTED.)

APPENDIX D.

INSTRUCTIONS FOR PINNING CIRCUIT BOARDS

(THESE PROCEDURES WERE IN DEVELOPMENT AT THE TIME THIS REVISON WAS PRINTED.)

APPENDIX E.

MATRIX MAPS

(THESE MATRIX MAPS WERE IN DEVELOPMENT AT THE TIME THIS REVISON WAS PRINTED.)

APPENDIX F.

SCHEMATIC DIAGRAMS

(THESE SCHEMATIC DIAGRAMS WERE IN DEVELOPMENT AT THE TIME THIS REVISON WAS PRINTED.)

APPENDIX G.

PREVENTIVE MAINTENANCE PROCEDURES

(THESE PROCEDURES WERE IN DEVELOPMENT AT THE TIME THIS REVISON WAS PRINTED.)

Altos Computer Systems
World Headquarters
2360 Bering Drive
San Jose, CA 95131
U.S.A.
(408) 946-6700
Telex: 171562 ALTOS SJN