Building a Hard-Disk Interface for an S-100 Bus System

Part 1: Introduction

How a Winchester disk drive and disk controller work, and what is needed to connect them to the S-100 bus and the CP/M operating system.

Andrew C. Cruce and Scott A. Alexander
ASC Associates Inc.
POB 615
Lexington Park, MD 20653

The development and availability of inexpensive, high-performance Winchester-technology disk drives offers us the opportunity to vastly expand the capability of microprocessor-based systems. The fact that these disk systems are both inexpensive and intrinsically highly reliable makes them extremely attractive as add-on devices for existing systems. Over the past several months we at ASC Associates have designed and constructed 5¼-inch Winchester disk subsystems for several microprocessor systems. In this and two subsequent articles we will describe in detail all the hardware and software necessary to integrate a standard, commercially available Winchester disk with an existing S-100-bus, CP/M-based computer system.

In terms of speed increase, a hard disk is to a floppy disk roughly what a floppy disk is to a cassette tape.

This month we'll review the general background information required to understand the following articles. Next month we'll explain the design steps required to interface the disk hardware with the system. In part 3 we will cover the software necessary to make CP/M aware that the disk is on the system, and we will describe the initial integration and debugging process. We intend that at the conclusion of this series you will have sufficient background information to be able to construct and integrate the disk system described in these articles with an S-100, CP/M-based computer system.

Why a Winchester?

The first question you might ask is why go to all the trouble of putting a Winchester disk on a microprocessor system in the first place. The answer is twofold: increased storage capacity and speed. Current state-of-the-art 5¼-inch floppy-disk-drive systems are limited to about 1 megabyte of storage per drive. The smallest Winchester systems, 5¼-inch drives, can today store over 10 megabytes per drive, and these storage capacities are only the beginning. The development of newer-technology thin-film read/write heads is expected to increase capacity by factors of four and more

About the Authors

Andrew Cruce has a Ph.D. in Aeronautical Engineering and has recently received an S.M. degree in management as a Sloan Fellow at MIT. Scott Alexander has an M.S. in Electrical Engineering. Both have extensive design and implementation experience with small computers and are full partners in the firm of ASC Associates, which markets the hardware described in this series of articles.

in the next several years.

The other advantage of a Winchester disk drive is its rapid operation. In terms of speed, a hard disk is to a floppy disk roughly what a floppy disk is to a cassette tape. For a Winchester disk system, maximum seek times (maximum time to find data on the disk) are on the order of 150 to 200 milliseconds (ms) rather than the several seconds associated with many floppy-disk systems. Also, once the data is located, it is transferred at 5 million bits per second, which is much faster than existing floppy-disk systems. At these rates a Winchester system can access data anywhere on the disk and load 64K bytes of information in under 1 second. The low access times, high data-transfer rates, and large storage capacities of Winchester drives allow us to realize the full processing power that is inherent in current microprocessor systems. Winchester drives open new vistas for such applications as large inventory systems, database management systems, and data analysis applications.

What Is a Winchester?

The term Winchester comes not from an inventor's name, but from the code name IBM assigned to the development of the Model 3340 disk memory, which was introduced in 1973. The industry as a whole has borrowed the Winchester name and now generally uses it to describe any disk drive using similar technology. The key element of the Winchester technology is that the head-to-disk assembly (HDA) is sealed from outside air and the disk is generally nonremovable.

In some ways, the Winchester technology is similar to conventional hard-disk drives. As with conventional hard disks, the read/write head floats over the recording medium on an air cushion that keeps the head from contacting the disk. In the case of the Winchester, however, the sealed and extremely clean environment of the HDA permits the disk designer to "fly" the read/write head closer to the disk surface. In typical removable-media hard-disk systems, the read/write head flies 60 to 70 microinches above the disk surface. The limitation on the distance the head flies above the disk is based on the minimum distance the head can fly safely above the disk and not risk contact with dust or any other contaminant on the disk. Any contact of this type causes the head to stop flying and crash on the disk surface. Such a crash normally ruins the read/write head and the surface of the disk medium, results in a complete loss of data, and necessitates an expensive repair job. Sealing the HDA in a Winchester drive provides a substantially cleaner environment than that of removable-media disks and allows the designer to fly the head about 20 microinches over the disk surface. This lower head altitude provides higher magnetic flux densities at the recording surface and thus higher recording densities on the disk.

During read/write/seek operations, the Winchester head flies above the surface of the disk on an air bearing, supported by carefully balanced aerodynamic forces. As the disk starts or stops, the head takes off or lands in a silicone-lubricated landing area. When the disk is not spinning, the head rests on and actually contacts the landing zone on the disk.

Winchester drives have a number of advantages over conventional hard-disk drives. First, they are very low cost both in absolute terms and in terms of cost per bit of storage capacity. In addition, the sealed environment of the HDA produces extremely high reliability with MBTF (mean time between failure) figures quoted in excess of 8000 hours. Winchester disk drives also require no preventive maintenance such as changing air filters or cleaning and aligning heads. This is of particular importance to owners of small, inexpensive computer systems who wish to have the capability associated with removable-media hard disks without the attendant maintenance hassles and expense. The primary disadvantage comes from the fact that the storage medium (the actual disk platter) is not removable. This prevents us from backing up data files in the conven-

INVESTMENT ANALYSIS FROM CENTENNIAL

CENTENNIAL SOFTWARE / 410 17TH ST. SUITE 1375 / DENVER, CO 80202 / (303) 595-9193

STOCK-FOCUS
Find out how low is low and how high is high. Using capital structure and performance data, Stock-focus objectively calculates the underlying value of a stock. The system was first developed by the management science department of a major money center bank, and is now in use by investment advisors, trust companies and brokerage houses. On your screen, Stock-focus will plot an estimate of lowest value, highest value and the current price. You then decide what to buy, sell or hold.

REAL-FOCUS
Exhaustively analyze potential real estate investments using the Wharton School's approach to real estate analysis. In minutes you can project profit, costs, and IRR for any project over a 10 year period. Real-focus accounts for amortization, debt, income, operating expenses, taxes, depreciation, and cash flows for both after-tax holding and the results of sale. With Real-focus you can analyze any potential investment from a single building project to a complex time-phased planned unit development.

THE FOCUS TECHNIQUE
FOCUS is CENTENNIAL SOFTWARE's new approach to Program Architecture, providing a natural interaction between microcomputers and users. It provides worksheet style input screens, free access to all program segments, and the ability to combine individual results files for portfolio analysis. You also receive a usable reference manual, menu helps, a program glossary, multiple report formats and spooling. With our FOCUS technique even the most complex programs are versatile and easy to use.

COMPUTER	MEMORY-K	REAL-FOCUS	STOCK-FOCUS
IBM PC	64	$179.00	$189.00
Apple (plus) II	48	149.00	159.00
TRS-80 I	48	149.00	159.00
TRS-80 II	64	179.00	189.00
TRS-80 III	48	149.00	159.00

ALL PROGRAMS IN DISK BASIC
IBM is a trademark of IBM, Apple is a trademark of Apple Computer Corp., TRS-80 is a trademark of Tandy Corporation.

ORDER NOW! FILL IN OR CALL 800-525-2003 (Toll Free)

PROGRAM NAME_____ COMPUTER_____ MEMORY_____
NAME _____ PHONE # _____
ADDRESS _____ ☐ VISA ☐ master charge
_____
ACCOUNT # _____ EXP. DATE _____

Circle 75 on inquiry card.

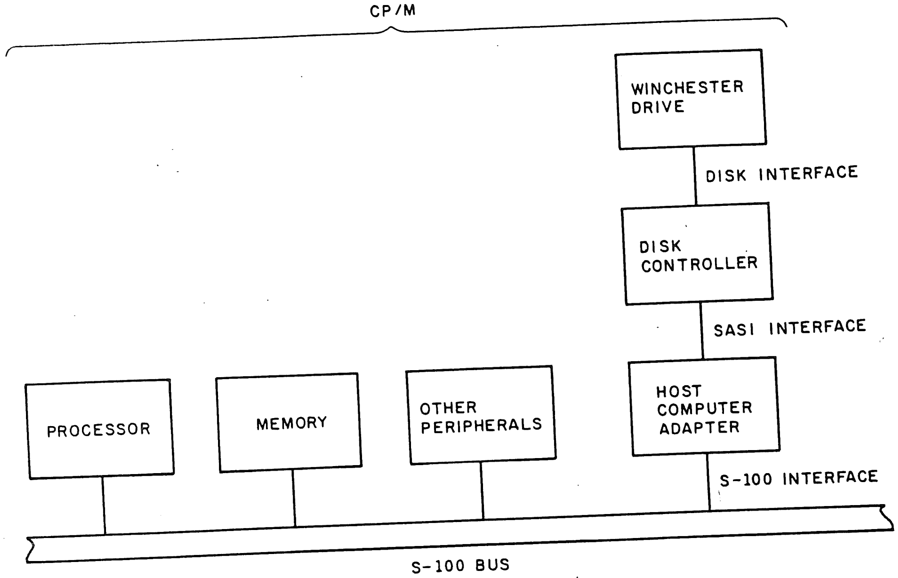


Figure 1: *A block diagram showing how a Winchester disk drive can be interfaced with an S-100-bus computer system.*

tional way (that is, by making and storing an exact copy of the disk to be backed up). However, this problem can be overcome in systems that have a floppy disk in addition to the Winchester drive. If you are willing to take the trouble, important files can be periodically backed up on floppy disks and saved in the event that a Winchester disk malfunctions. This may not be as convenient as standard backup procedures, but it can provide a large measure of data security.

Which Winchester?

During the design process of our system we first had to decide which of the available Winchester disk systems we should use. Currently, Winchester disks are available from a variety of manufacturers with disk platters in different sizes, the most common being 14-, 8-, and 5¼-inch diameters. We evaluated these three options by examining the requirements of a typical microcomputer user. As storage densities have gone up, the 14-inch systems have grown to the point where they can store a staggering amount of data at a relatively low cost. Currently, 14-inch systems have storage capacities in the multiple hundreds of megabytes. Although this leads to a very attractive cost per bit of storage capacity, it also leads to a relatively high absolute cost for microprocessor applications. In our opinion this level of capacity far exceeds the requirements of the typical microcomputer user. To a certain extent, the same logic also applies to the 8-inch drive systems. They are too big and too expensive for the highly price-sensitive microcomputer market. As a result, we homed in on the more recently available 5¼-inch drives as the best alternative. They are relatively inexpensive and are currently available in models that can store over 10 megabytes of data. Additionally, expected technology improvements in the near future will increase this storage capacity to over 40 megabytes. Thus the 5¼-inch format will not only satisfy most of today's requirements but also will provide a large potential for growth.

In addition to price and storage capacity there are a number of other features of the 5¼-inch drives that make them particularly attractive. One asset is a standardized drive interface that allows complete flexibility in switching from one manufacturer's drive to another in a completed system. This also allows companies to build standardized controller boards, which greatly ease the system integration problem. The major advantages of the 5¼-inch Winchester drive for microprocessor system applications are:

1. low cost
2. large storage capacity
3. rapid access time
4. high reliability
5. no need for preventive maintenance
6. common interfaces
7. small and compact size
8. low power requirements and low heat generation
9. availability from multiple vendors with standard interfaces

The Interface Problem

The block diagram in figure 1 presents a common approach to interfacing a Winchester disk with an existing computer system. The existing system contains a microprocessor, memory, and one or more peripherals that are all running under control of the CP/M operating system. All this hardware is plugged into and communicates via the S-100 bus. To add the Winchester system, the designer must provide an HCA (host computer adapter) that allows communication between the existing system bus and the disk controller. In addition, there must be a disk controller that accepts commands from the system via the HCA and in turn commands the Winchester disk to perform the desired functions. Finally, the designer must add software to the CP/M system to receive disk I/O (input/output) requests from application programs, such as "read a file" or "write a file," and translate these requests into commands for the HCA.

Now we'll discuss each of the elements in the Winchester system in more detail, concentrating on the operation of each element as well as the interfaces between the various elements.

The Disk and Disk Interface

A Winchester disk is similar to any other disk system in terms of operation and organization. The disk can be considered to be composed of concentric tracks of recorded information. Each track is further subdivided into sectors. A typical 5¼-inch Winchester drive system may contain upwards of 40,000 individual sectors,

 © BYTE Publications Inc

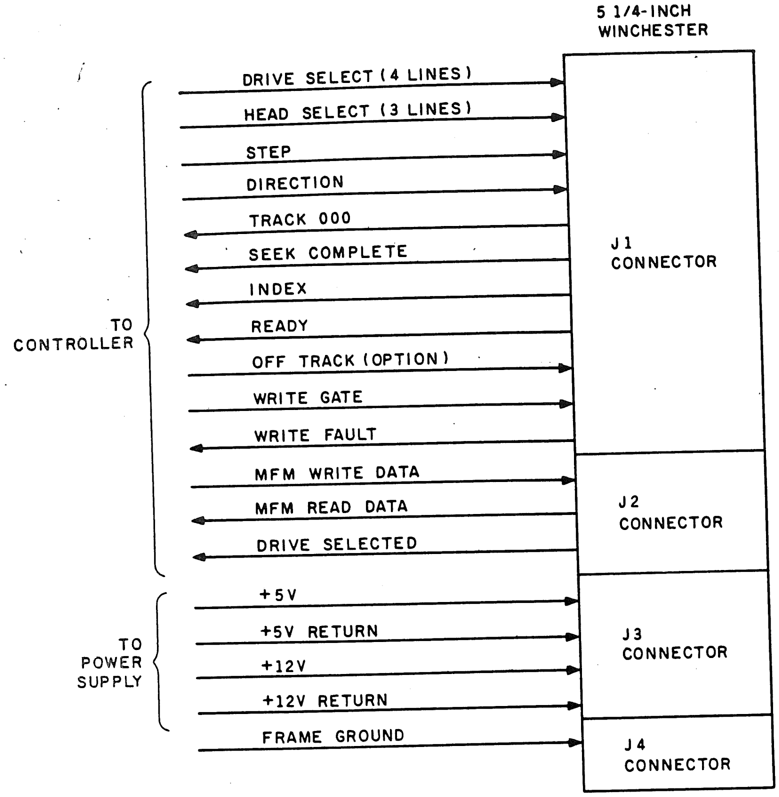


Figure 2: *The standard 5¼-inch Winchester disk-drive interface.*

each containing its own sector address information and data-storage space. As the following discussion will show, the operation of a Winchester disk is very similar to that of a standard floppy disk. The major difference is the speed of operation and the amount of data that a Winchester can hold. The speed of operation also requires that we use a dedicated hardware disk controller rather than have the controller functions performed by software as in a floppy-disk system.

Figure 2 illustrates the standard 5¼-inch Winchester disk drive interface, which connects the disk drive to the disk controller. Signals in this interface are of three basic types. The first type provides power required for disk operation, in this case +12 and +5 volts DC. Signals of the second type are data signals that transfer data between the disk and the controller. The data is transmitted serially at a 5-megabit-per-second rate in MFM (modified frequency modulation) format. The last type of signals are signals for control purposes that

Now, *here's* a printer for you.

The Silver-Reed EXP550 Electronic Bi-directional, Daisy Wheel Printer.

The new Silver-Reed EXP550 is one of the finest machines for the money on the market today. For starters, the EXP550 offers carrier feed in units of 1/120 inch and forward/reverse paper feed in units of 1/48 inch. Other features include: 16 CPS Shannon text · Subscript · Superscript · Bold · 17 inch paper capacity · 10, 12, 15, PS pitch · and much more. The EXP550 will provide you with letter quality printing at a cost that will amaze you. For more information, call (800) 421-4191 and ask for printer sales division.

SILVER-REED
SILVER-REED AMERICA, INC.
8665 Hayden Place, Culver City, CA 90230
(213) 837-6104, Outside of California (800) 421-4191

OEM & DISTRIBUTOR INQUIRIES WELCOME.

Circle 393 on inquiry card.

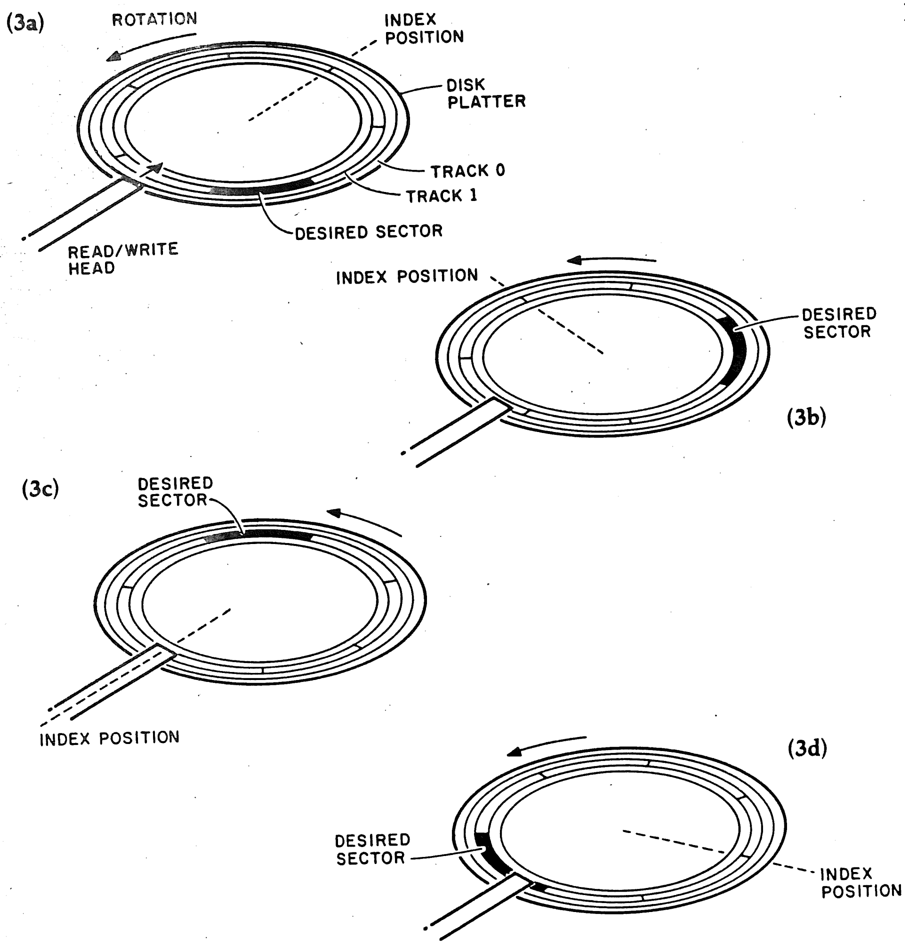


Figure 3: *Reading a sector on a hard disk. In figure 3a the read/write head moves to the proper track. In 3b the read/write head is positioned and waiting for the index pulse. When the index position passes under the read/write head (3c), the disk controller starts reading the first sector on the selected track and continues to read until the desired sector is reached. In figure 3d the desired sector is under the read/write head and the controller begins transferring data.*

allow selection of a particular drive, stepping of the read/write head in the selected drive, and control of other primitive disk functions.

Probably the easiest way to understand disk operation is to go through the steps involved in seeking and reading data on a particular sector of the disk. In our case, these are the functions performed by the controller. As the first step in the process, the controller moves the read/write head to the track containing the desired segment by sending control signals to the disk drive. When the read/write head is on the proper track, the controller then waits for a specific portion of the disk called the index position to pass under the head. This index position provides orientation information which identifies the start of a track. The controller then begins reading the serial data coming from the disk, looking at the sector-address information for each sector until it locates the address indicating the desired sector. The data immediately following this address is then captured and the read is completed. This sequence of events is shown diagrammatically in figure 3.

A disk-write operation is performed similarly. The same sequence of events occurs until the controller locates the proper sector. At this point, instead of reading data from the disk, the controller sends new data to the disk for recording.

The final point to be covered is how the sector-address information is put on the disk in the first place. This process is called formatting. When a disk is formatted, the controller starts on track 0 and, following the index position, writes the sector-address information for the first sector on the disk. It then fills the data area following the first address with nulls or other characters to reserve the data space for future use. As soon as it has filled the area, the controller begins the process over again for the next sector, writing the sector-address information and then reserving the data area. This process continues until all the sectors on the first track of the disk are formatted. The controller then steps the read/write head to the next track and repeats the process until it has formatted all the sectors on all the tracks.

Formatting is typically performed only once because creating the sector addresses and reserving the data areas would destroy any previously stored information on the disk. When formatting, we generally have to define the size of the data area associated with each sector. The size of this area affects the total number of sectors on the disk and thus the fraction of the available disk space that the sector-address information occupies. Typically, these data areas are set up to hold either 256 or 512 bytes of information, although special applications could require different allocations for optimum storage efficiency. For our case we will restrict consideration to the 256- or 512-byte cases.

Because of the need for formatting (i.e., placing sector-address information on the disk) manufacturers quote two storage-capacity measures for disk systems. The unformatted number refers to the total amount of data that can be stored on the disk. The formatted number refers to the total amount of data space that is available on the disk after it has been formatted. In general, the latter measure is of more importance to disk users.

The Controller and Controller Interfaces

Working backward from the disk drive toward the S-100 bus, the next device in the disk-drive subsystem is the disk controller. We just discussed

text continued on page 14

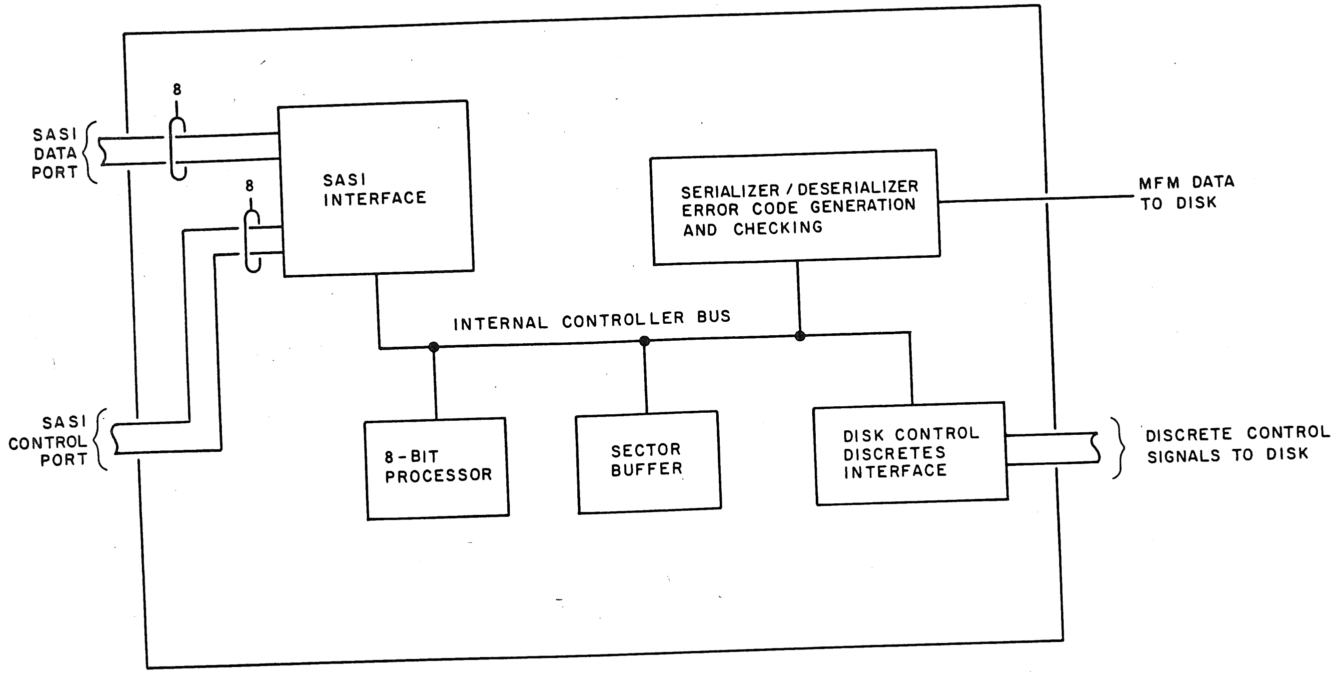


Figure 4: *A block diagram of the disk controller.*

the signals that the controller uses to access particular sectors on the disk. Now we'll discuss how these signals are generated and, in general, how a controller operates.

The controllers we will address are characteristically known as "smart" controllers. This means that they have some internal processing capability and use this capability to perform many of the interfacing chores with the disk without intervention from the host computer. The speed of the Winchester disk drive necessitates a dedicated controller to effectively handle all disk control and timing. Figure 4 presents a conceptual block diagram of this type of controller. The disk-drive interface, which we have already discussed, is on the right, and the interface to the HCA is on the left. A common interface between the controller and the HCA is based on that developed by Shugart Associates, known as the Shugart Associates System Interface (SASI). As shown, the SASI consists of two 8-bit connections. One set of 8 bits is for data and the other is for control signals. The control signals are split, with 5 bits used for controller-to-HCA signals and 3 bits for HCA-to-controller signals.

Internally, the controller is a bus-structured device with an 8-bit processor, a sector buffer, a serializer/deserializer, the disk interface, and the SASI interface connected to the internal bus. Again, the easiest way to understand the operation of the controller is to go through a typical sequence of operations. In this case, the controller will perform a read operation from a particular sector of the disk. The process starts when the host computer, using the HCA, generates a Select signal on the SASI interface. This alerts the controller that a command sequence will be coming in over the 8-bit data port. Through a series of handshakes, a command sequence consisting of 6 bytes of data is passed through the data port of the SASI. These 6 bytes contain the command to be executed by the controller—in this case, read data—and the sector address of the data to be read.

With this information, the controller begins to execute the requested command using its internal processor. It sends commands to the disk to move the read/write head to the track that contains the desired sector. Once the head arrives at the right track, it waits for the index pulse and then starts reading the data coming from the disk to find the appropriate sector. The 8-bit processor reads the data from the disk after it has gone through the serializer/deserializer. The deserializer portion of this device receives the MFM data directly from the disk, performs error checking and error correction on the data, and then passes the data to the 8-bit processor (via the internal controller bus) in parallel byte format. Once the controller locates the desired sector, it transfers the data from the disk into the sector buffer. This buffer is essentially a RAM (random-access read/write memory) chip that is used to store the information retrieved from the disk until it is requested by the host processor. The controller informs the host system, through the SASI port, when it has completed the data transfer. At this point the host can read the retrieved data out of the controller and take any appropriate action with it.

A write operation is performed in a similar manner. In this case, the host sends the Select command and the 6-byte command sequence to the controller that tells it to write data to a particular sector. The host then sends the controller the data to be written into the particular sector. The controller accepts this data and places it in the sector buffer. It then initiates the series of actions to find the sector to which the data is to be written. When the controller locates this sector, it passes the data from the sector buffer through the serializer, which adds error detection and correction bits to the data, and then sends the result to the disk in serial MFM form.

In addition to the read and write functions, a smart controller can perform a number of other functions, including formatting the disk, reformatting a particular track on a disk, and a variety of built-in test and loopback test functions. These functions are initiated exactly like the read and write functions but with a different set of commands passed to the controller.

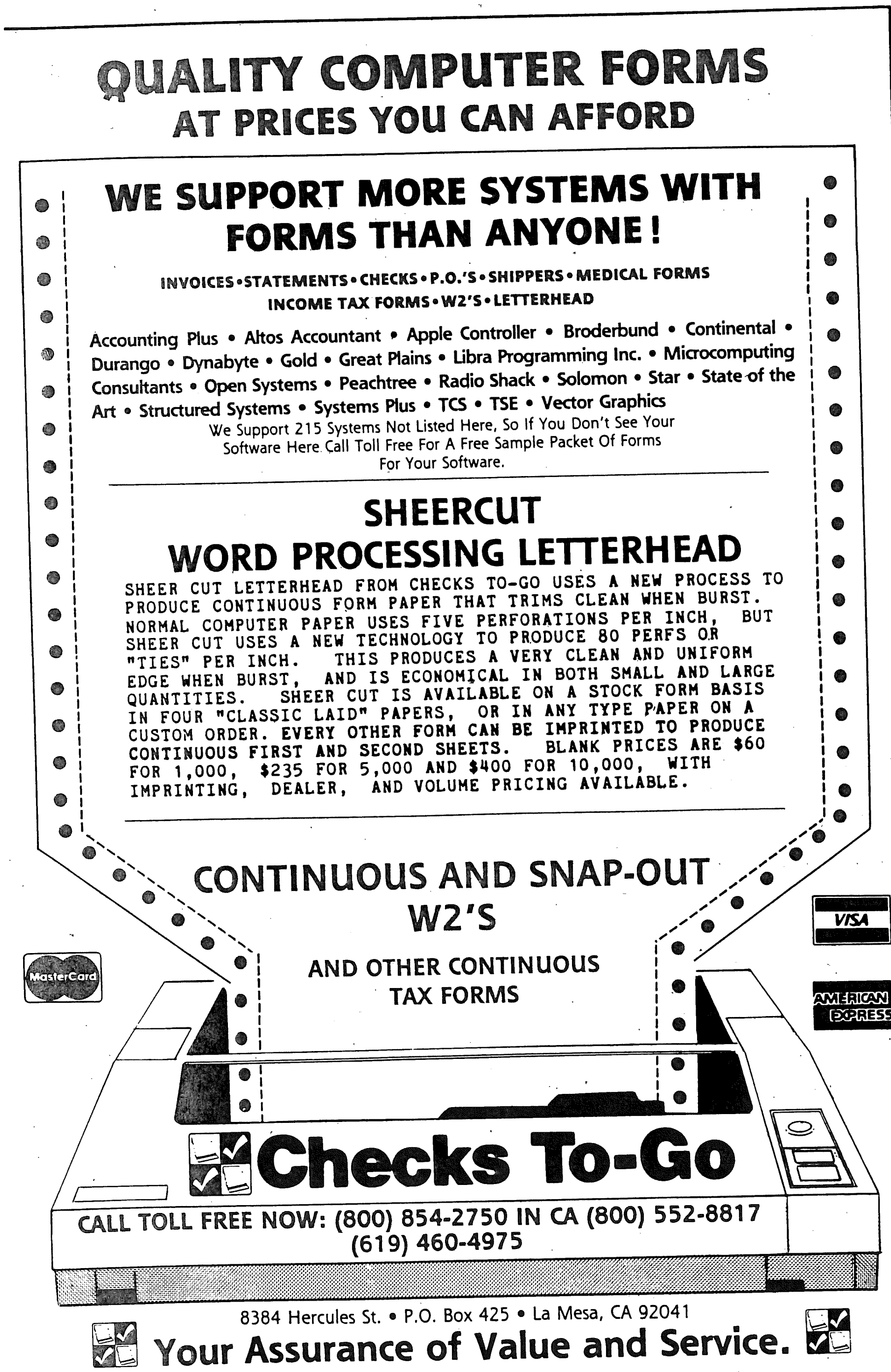
The Host Computer Adapter

The last piece of hardware required to complete the Winchester system interface is the host computer adapter (HCA). As figure 1 indicates, this adapter allows communication between the host computer S-100 bus and the SASI on the controller. A number of options are available in designing an HCA, but basically they boil down to the degree of intelligence that is to be incorporated into the HCA. In more simple designs, the HCA consists of only a couple of output ports on the S-100 bus with the proper address-decode logic. In this case, the two output ports on the S-100 bus correspond to the two 8-bit ports of the SASI interface, and the HCA is essentially a buffer device. The disk-driver software then manipulates these two ports to perform any required function exactly as if the controller were part of the system.

More complex designs would allow the HCA to perform some of the functions that would be performed by the host computer in the simpler design. Again, an example will best illustrate the process. Assume that a host system wishes to transfer a sector of 256 bytes from the host system to the disk. In the case of the simple HCA design, the driver software would be informed by the operating system of this required transfer and then would send the proper commands to the controller to initiate the transfer process. In addition, the driver software would sequentially fetch each of the 256 bytes of data to be transferred from the host memory and pass it through the SASI data port to the controller.

An alternate, more complex design of the HCA would eliminate much of this processing burden from the host system's processor. If the HCA were given DMA (direct memory access) capability, all the host processor would have to do would be to tell the HCA what sector to read or write to, where in host memory the data transfer was to begin, and how many bytes of data to transfer. The HCA would then take over the entire process of fetching the data from host memory and passing it to the controller and would simply inform the host processor when the process was complete.

As the description implies, providing the HCA with DMA capability increases the total system performance by reducing the load on the host processor. This increased performance carries with it a penalty in terms of increased cost and complexity of the HCA. In the design of our



Circle 77 on inquiry card.

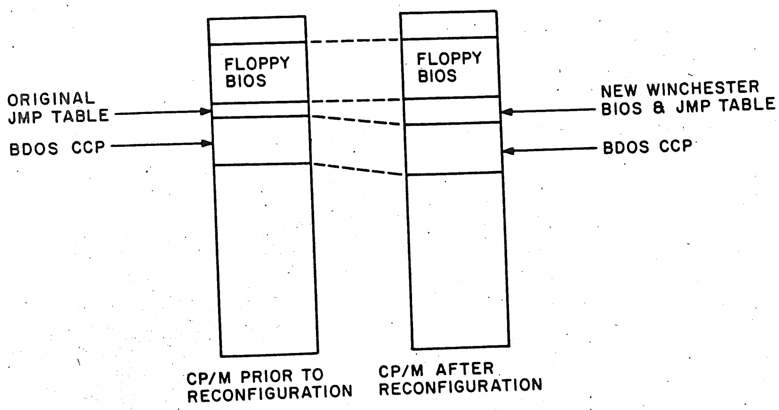


Figure 5: *A block diagram showing how the BIOS for the Winchester disk drive is inserted into the CP/M operating system.*

system, we considered this trade-off carefully. In next month's article on the hardware design, we will go through these trade-offs in detail and describe what system we chose and the reason for that choice.

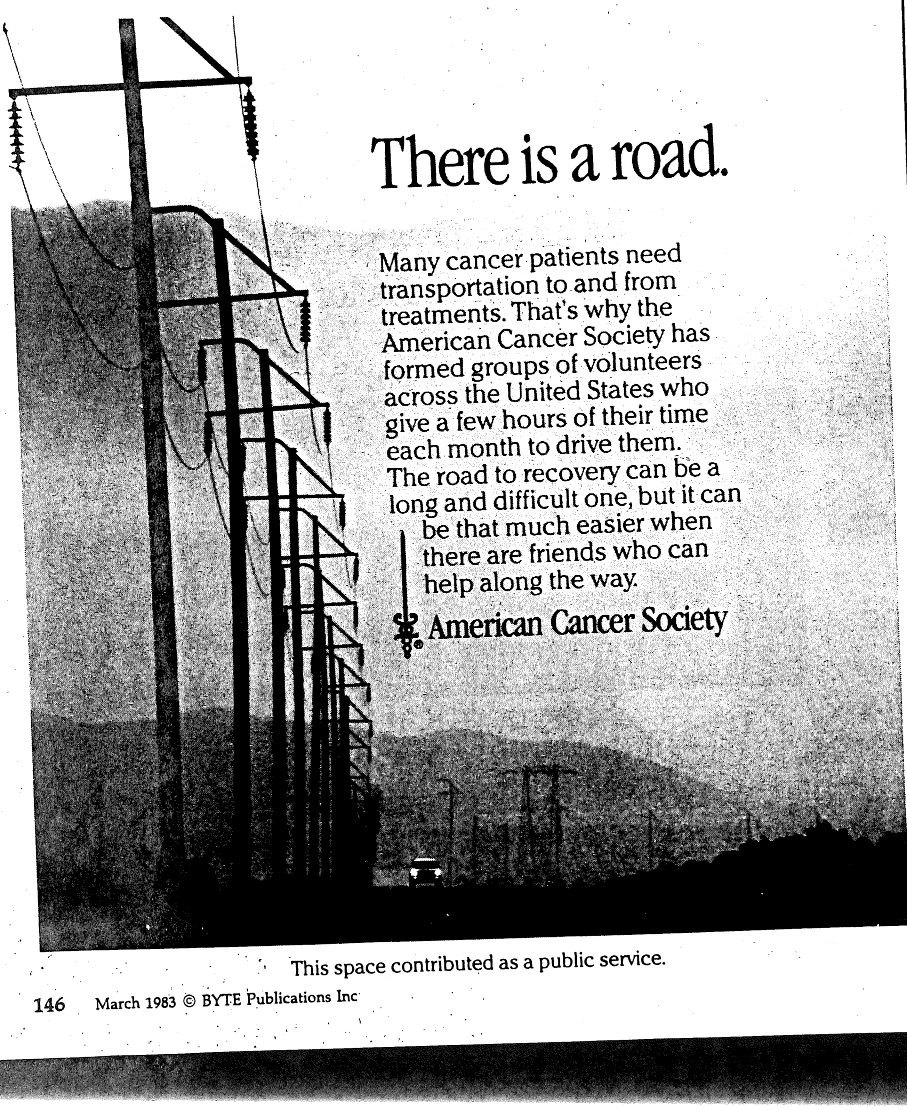
Variations

Up to now, we have described a general Winchester interface system that consists of a drive, a controller, and an HCA. Any given system must contain all these components. However, there is considerable latitude in how these components are packaged. One common packaging strategy is to put the controller and HCA functions on the same board. In this configuration, a single board plugs into the S-100 bus and a ribbon cable connects this board to the disk. In another strategy, the HCA is plugged into the S-100 bus and a ribbon cable connects the HCA to the controller and another ribbon cable connects the controller to the disk. This second configuration is likely to be more common because it allows builders of controllers to build one controller card that is applicable to many systems. In fact, as you will see next month, this is the configuration we chose.

In the previous discussions, we have not mentioned the possibility of adding multiple Winchester drives to a system. This is certainly possible and can be done with very little design effort. In most cases, the incremental cost of the second drive amounts to only the cost of the drive itself and the interconnection hardware. We will cover this option in detail next month when we discuss the specifics of the hardware implementation we chose and the particular controller hardware.

Operating System Considerations

The final step in integrating a Winchester disk into an existing S-100 CP/M-based system is to somehow make the CP/M operating system aware that the disk is part of the system. This is done by expanding the existing CP/M BIOS (basic input/output system) to include the new disk. The existing BIOS contains all the software necessary to run the current peripherals on the system. The modification we need would keep these existing routines and add the necessary routines to communicate with the new Winchester disk drive. The simplified memory map of CP/M both before and after the required modification, presented in figure 5, shows how this can be done. At the top of the existing BIOS is a jump table that points to the various primitive disk functions for an existing system. These functions include set track, set sector, select disk, read sector, write sector, etc. In order to add these functions for the new disk, the CP/M system is moved using the MOVECPM utility, and a new jump table is installed that points to the new disk routines. This new code, in addition to performing the required



This space contributed as a public service.

disk functions, keeps track of which disk is selected. If the Winchester is the selected disk, then these new routines perform any requested functions. On the other hand, if another disk or peripheral is selected, say the existing floppy disk, then the commands are passed directly to the old BIOS routines for that system. In this way, with a minimum of difficulty, the disk primitive routines for the new disk can be included in the CP/M system. We will cover the details of the BIOS routines for the Winchester system as well as the procedures for reconfiguring the existing system in part 3.

Summary

So far we have covered, in a general way, all the components required to interface a Winchester disk with an existing S-100, CP/M-based system. You should now have a fairly complete understanding of what a Winchester disk is, how it operates, and what some of the differences are between Winchester disks. In addition, you should now have a general grasp of the $5\frac{1}{4}$-inch drive interface, the Shugart Associates Standard Interface, the functions of a smart controller, and the host computer adapter. In parts 2 and 3 we will cover a specific example of the interfacing process in detail, using commercially available equipment: next month we will describe the hardware including the HCA, the controller, and a disk power supply; and in the final article we will describe the software aspects of writing new BIOS routines for CP/M and reconfiguring the system to include the new Winchester disk drive.

These articles will cover only the details of interfacing with S-100 CP/M-based systems. For interfacing with other computers and operating systems, however, the procedure is much the same. First, an HCA must be designed to allow communication between the host computer and the disk controller. Then the equivalent of the CP/M BIOS must be found in the operating system used, and new code must be generated to include the Winchester disk system. Depending on the availability of documentation on the hardware and operating system, this may or may not be an easy task. Hopefully, this series will provide a reference point from which to proceed. ■

The Winchester disk drive subsystem described in this series of articles is available as a completely assembled unit from ASC Associates of Lexington Park, Maryland. In addition to the S-100 version discussed, versions are also available for TRS-80 and Apple computers. The disk-drive systems for these computers use the same drive and controller hardware as the S-100 version but use a different host computer adapter and interface software. Until a nationwide dealer distribution network is established, these systems will be available by mail order for $1995. To order or obtain further information, write to ASC Associates Inc., POB 615, Lexington Park, MD 20653, or phone (301) 863-6784.

Serial ⟺ Parallel

only $89.95

Connector Option $10.00
CA Residents 6% tax
UPS Shipping $3.00

Engineering Specialties™
1501-B Pine Street
Post Office Box 2233
Oxnard, California 93030

Printers! Plotters! Punches! Robots!

Convert What You Have To What You Want!

★ RS232 Serial
★ 8 Baud Rates
★ Latched Outputs
★ Centronics Parallel
★ Handshake Signals
★ Compact $3\frac{1}{4} \times 4\frac{3}{8} \times 1\frac{1}{2}$

No longer will your peripheral choices be limited by the type of port you have available! Our new High Performance 700 Series Converters provide the missing link. Based on the latest in CMOS technology, these units feature full baud rate selection to 19.2K, with handshake signals to maximize transfer efficiency. Detailed documentation allows simplified installation. Order the Model 770 (Ser/Par) or Model 775 (Par/Ser) Today!

Buffer Products Coming Soon!

master charge — VISA

CALL (805) 487-1665 or 487-1666 For *FAST* Delivery

Circle 176 on inquiry card.

Building a Hard-Disk Interface for an S-100 Bus System

Part 2: The Hardware

Andrew C. Cruce and Scott A. Alexander
ASC Associates Inc.
209 Spring Valley Dr.
Lexington Park, MD 20653

In last month's article we described Winchester disk-drive technology and generally how drives of this type are integrated into microcomputer systems. This month we will be more specific and describe how a particular set of hardware can be assembled to provide Winchester disk capability for an S-100-bus microcomputer. After reading this article, you should understand the concepts involved in the design, fabrication, and integration of the hardware needed for a 10-megabyte S-100-based Winchester disk system.

The block diagram presented in figure 1 shows the various components of the Winchester disk subsystem. The functions of the host computer adapter (HCA), the disk controller, and the disk drive were described in last month's article. The only remaining component in the diagram is the power supply, which provides the disk drive and controller with power.

About the Authors

Andrew Cruce has a Ph.D. in Aeronautical Engineering and has recently received an S.M. degree in management as a Sloan Fellow at MIT. Scott Alexander has an M.S. in Electrical Engineering. Both have extensive design and implementation experience with small computers and are full partners in the firm of ASC Associates, which markets the hardware described in this series of articles.

This diagram suggests an approach to designing the disk system. The first step is to establish the data path; that is, you must determine how the HCA, the disk controller, and the disk drive are connected. Before you can do this you need to settle on the particular disk and controller to be used in the system as well as the functions that are to be included in the HCA. Once you complete this process, you'll have enough information available to specify power requirements for the system and to identify power supplies that could be used in the system. Finally, when the data interface is complete and you've chosen a power supply, you can consider how to package the system.

We'll now proceed through this design sequence, first discussing the availability of disk drives and controller cards and which ones we ultimately adopted. Next we'll cover the requirements for the HCA and go into the detailed logic design of this component. Finally, we will examine some of the options in choosing a power supply for the system and show the steps required to construct an attractive and reliable package for the system.

Disk Selection

The first decision we made in selecting a disk was on its size and storage capacity. As we stated last month, we chose the 5¼-inch disks because they offered what we consider to be more than sufficient storage capacity for most applications at a reasonable price. When we started this project, a number of manufacturers, including Seagate Technology, Shugart, Memorex, and Miniscribe, offered 5¼-inch Winchester drives. We needed a drive with a disk-to-controller interface equivalent to that of the ST-506 disk drive manufactured by Seagate Technology. This interface has become the de facto standard in the industry and provides the freedom to select from a number of potential controllers that are compatible with this interface.

In considering the storage capacities available on the 5¼-inch-drive systems, we found that 10-megabyte systems were becoming available at prices only marginally higher than the smaller 5-megabyte systems. Based on the attractive cost per bit of storage capacity of the 10-megabyte drives, we opted for the higher-capacity disk systems.

With these criteria in mind, and taking into account the price/performance ratios of the available units, we chose the 10-megabyte disk drive manufactured by the Miniscribe Corporation. Our initial choice could change because we made it at a time

Circle 299 on inquiry card. →

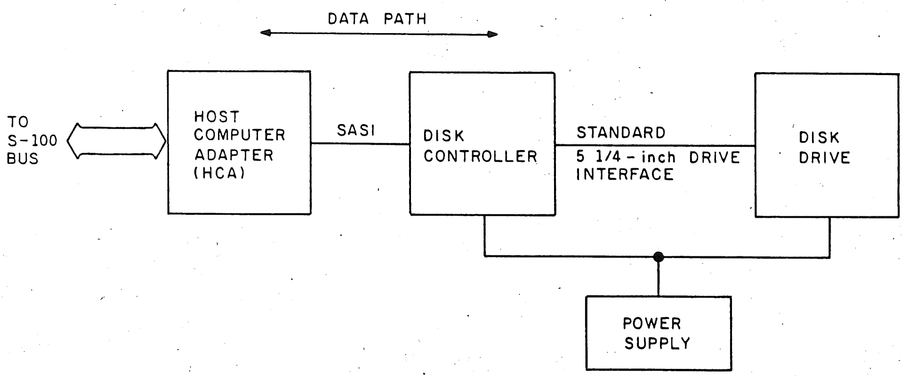


Figure 1: *Winchester disk-drive subsystem block diagram.*

when many disk-drive manufacturers had not yet commenced delivery of 10-megabyte hardware. However, now that the Seagate Technology interface has become the industry standard, virtually any new 5¼-inch drive currently being manufactured could be immediately plugged into whatever controller we selected. This also means that, for the purposes of this article, you could substitute any other 5¼-inch Winchester disk drive as long as it uses the Seagate Technology standard interface.

Controller Selection

Our decision on which controller to use in the system was somewhat more complicated. The first and most important requirement was that the controller be compatible with the Seagate Technology ST-506 interface (so as to be compatible also with any of the disk drives we were considering).

The second requirement was that the controller implement the Shugart Associates Systems Interface (SASI) to the HCA so that we could interface not only to S-100 systems, but also to a variety of other microcomputers such as the Apple, the Osborne, and the TRS-80. This criterion eliminated some of the available controllers, including those made by Morrow, which interfaced directly with the S-100 bus. A few of these devices would also have been eliminated because any system they are used in must conform to the IEEE standard for the S-100 bus. The problem here is that many of the older S-100 systems do not conform completely to this standard (for example, a number of systems do not implement the extended 24-bit addressing of the IEEE S-100-bus specification) and thus would not operate properly with the "S-100-compatible" controllers.

Working within these constraints, we considered the cost, performance, size, and power consumption characteristics of a number of commercially available Winchester disk-drive controller boards. At the time we made the selection, the Xebec S1410 controller appeared to offer the best performance per dollar expended. Again, for purposes of this article, the choice of the particular controller board does not largely influence the rest of the design. If you have a controller board that uses the SASI to connect to the HCA and a Seagate-Technology-compatible interface to connect to the Winchester drive, then the hardware and software designs covered in this series should work with little or no modification.

After we verified that the controller and disk were indeed compatible, all we needed to complete this interface was to fabricate a 34-conductor and a 20-conductor ribbon cable to connect the disk and the controller. The next step was to decide on the requirements for the HCA and to design the hardware accordingly.

HCA Interface Requirements

Several preliminary choices had to be made before we designed the host computer adapter for the Winchester disk system. The first was whether the disk subsystem should have DMA (direct memory access) capability. The alternative was to design the HCA to allow the host computer to perform data transfers between the disk and computer memory. Based on our assessment of most microprocessor system requirements, we considered that the additional speed gained by providing DMA capability in the HCA was not worth the additional complexity and expense that this capability would require. Thus we initially decided that the HCA would not perform DMA data transfers from the disk to memory.

The second decision concerning the design of the HCA was whether or not this device should contain a PROM (programmable read-only memory) chip and a new system-bootstrap program. Assuming that the system to which the Winchester disk is to be connected is operable, it should already have a bootstrap program. However, this program is designed to work with the original disk peripherals. It is possible to interface a Winchester disk with such a system and then use a separate "Winchester Boot" program to bring the Winchester disk BIOS (basic input/output system) routines into the system. However, this means that a hardware bootstrap will always go to the original disk system and that the Winchester Boot program must always be run to bring the Winchester drive into the system. We considered this to be a fairly cumbersome procedure and decided that we should provide the capability for a new bootstrap for the system that would bring up the Winchester disk and all the previously existing peripherals in the system.

Host Computer Adapter Design

Based on these requirements, the design of the HCA can be divided into three essential elements: a connection to the SASI on the disk controller board, a PROM subsystem on the HCA board to store a new bootstrap program, and an interface to the S-100 bus that provides all the signals required to support the first two elements. The combination of these elements results in the initial top-level HCA block diagram presented in figure 2.

The SASI interface on the disk controller is a 2-byte parallel interface.

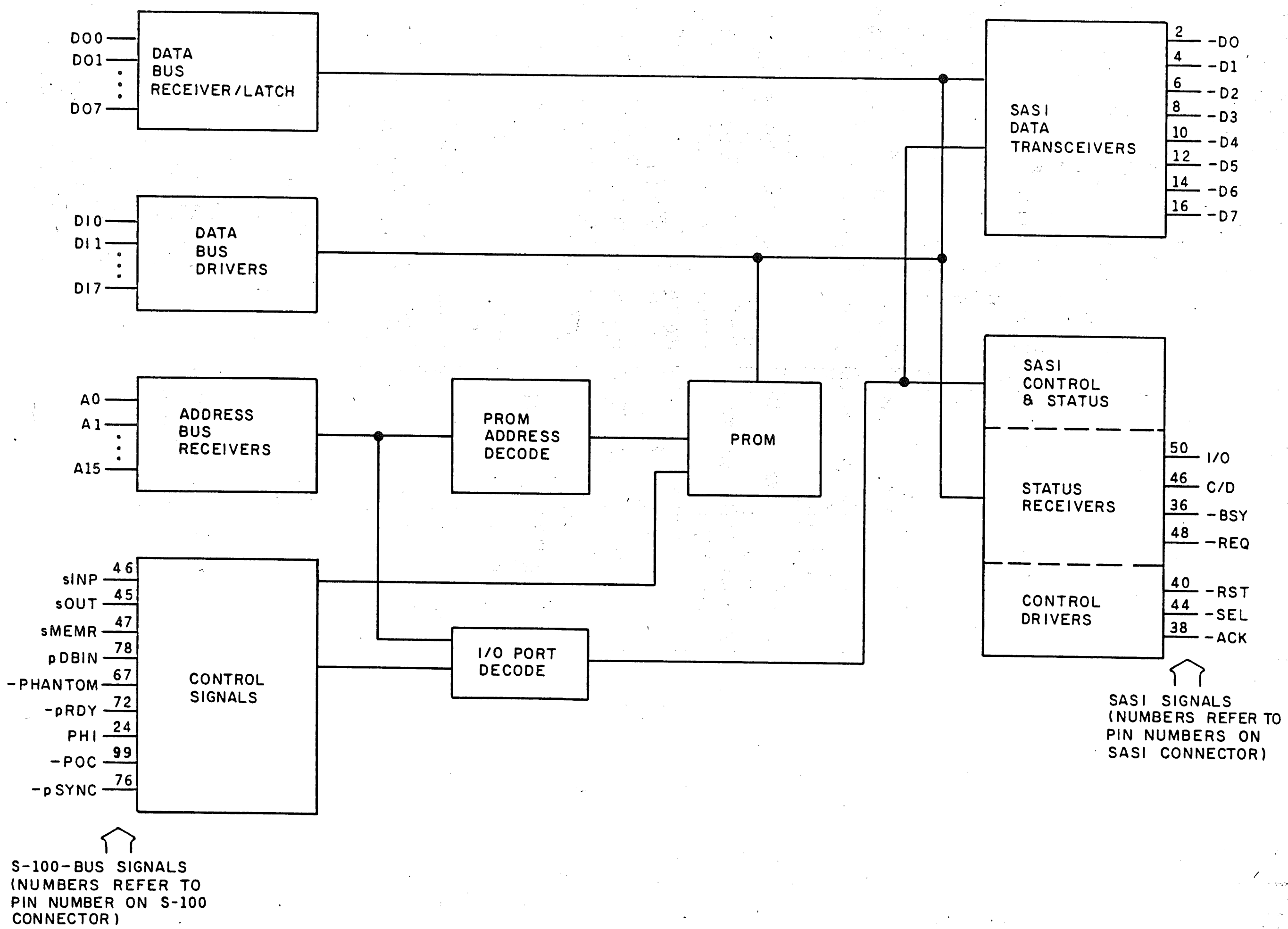


Figure 2: *Block diagram of the host computer adapter (HCA) for the Winchester disk drive.*

The first byte contains 8 bits of data being sent to or received by the controller. The second byte consists of control and status signals that are used by the host processor to determine the status of and provide control to the disk controller board. The controller board receives these signals on a standard 50-conductor ribbon cable. The signal names, descriptions, and pin numbers are presented in table 1. The portion of the HCA that communicates with the disk controller must send these signals to the controller, preferably via a 50-conductor cable.

The PROM portion of the HCA board is relatively simple in concept. Basically, it provides 2K bytes of nonvolatile memory to store a bootstrap loader. However, in considering the actual design, a number of questions arise. The first is whether to provide a wait state for access to the PROM. Depending on the speed of the processor being used, it may be necessary to make the processor wait during memory-access operations to allow the PROM time to access the requested data. Another question is whether to provide compatibility with memory systems that support phantom memory. In these systems the bootstrap PROM can be located in memory locations that overlay normal RAM (random-access read/write memory). During the bootstrap operation, the PROM system must assert a PHANTOM signal, which causes the RAM in the computer not to respond as long as the signal is asserted. Thus the PROM is accessed during the bootstrap, and then after the bootstrap is finished the PROM is disabled and conventional RAM memory replaces the PROM in memory. As we will show, neither of these two options is particularly difficult to implement, so we decided to design both a wait-state enable and phantom-memory support into the HCA. Again, figure 2 shows basically how these two capabilities fit into the system.

With the SASI interface and the PROM memory established, it is now possible to determine what signals from the S-100 bus must be used to communicate with these two elements. Accessing the PROM is like accessing normal memory. All that is required is some address-decode logic and some combinational logic using the S-100 bus signals sMEMR and pDBIN to gate the PROM data onto the S-100 bus. Several other signals are needed to support the memory wait state and the phantom-memory capability. These are the -pRDY signal going back to the processor (which keeps the processor in a wait state), the -PHANTOM signal back

Signal Name	Pin Number	Description
I/O	50	Input/Output: open collector output from controller to HCA. Low level indicates a controller-to-HCA data transfer. This signal is qualified by -REQ.
C/D	46	Command/Data: open collector output from controller to HCA indicating command or data on the data bus. Low level means command bytes. This signal is qualified by -REQ.
-BSY	36	Busy: open collector output from controller to HCA indicating a controller is ready to receive data or command information from the HCA. High level means controller is ready for data.
-MSG	42	Message: open collector output from controller to HCA indicating that the current command is complete. (This signal is not used in our design.)
-REQ	48	Request: open collector output from controller to HCA to initiate controller-host handshaking sequence.
-ACK	38	Acknowledge: host-generated signal that is asserted active low in response to controller -REQ when the host is ready to receive or transmit data. In order to complete the handshake, the host adapter must send -ACK in response to each -REQ from the controller.
-RES	40	Reset: host-generated signal, active low, that resets the controllers. This signal must remain low for at least 100 nanoseconds.
-SEL	44	Select: host-generated signal, active low to initiate a command transaction to a controller.
-DB0	2	Data Bus 0 through 7: tristate input/output bus to send data/command information from the host to the controller or for the host to receive data from the controller.
-DB1	4	
-DB2	6	
-DB3	8	
-DB4	10	
-DB5	12	
-DB6	14	
-DB7	16	

Table 1: *Signals and pin locations for the Shugart Associates Systems Interface (SASI) between the disk controller and the host computer adapter (HCA).*

to RAM (causing it not to respond), one phase of the system clock (PHI), and the Power On Clear signal (-POC or -RESET).

In addition to supporting the PROM, it is also necessary to provide for data communication between the S-100 bus and the SASI ports on the HCA. There are two ways of doing this. One is to make the SASI ports appear as memory to the processor and use memory-mapped I/O to communicate with these ports. However, this uses up memory space, and a better approach is to use the I/O ports that are accessed by the 8080 IN and OUT instructions. This again calls for some address-decode logic to determine which port is selected and some combinational logic on the S-100-bus signals sINP and sOUT. These signals indicate whether data is to be written to or read from the particular port selected; figure 2 shows generally how they are used in the HCA design. They are standard S-100 signals, and a detailed description of their use and required characteristics can be found in any complete description of the S-100 bus.

In the logic design process, the block diagram shown in figure 2 is expanded to include the digital logic necessary to perform the functions in-

PerSci

Peripherals a Generation Ahead

Voice Coil Actuated Dual 8" Floppy Disk Drive

- **FAST**
- **Apparent performance of a Hard Disc**
- **Equivalent of two standard drives**

Model 299B (DS DD/SD)
Model 270/277 (SS DD/SD)

PerSci, Inc.

12210 Nebraska Ave.
Los Angeles, CA 90025
(213) 820-7613
TELEX 687444

Circle 480 on inquiry card.

Full Size Keyboard Conversion

For Your ZX81/ TS1000

SUN KD-81

$69.95

Easy To Install
No Soldering — No Modifications

41 Keys with Full Size Space Bar • Allows Touch Typing
Same Key Layouts as Sinclair • Two Color Keytops
Keyboard Case Holds Both Keyboard and Computer
Commands and Graphics Spelled Out on Keytops

16K Piggyback RAM Module ... P/N MX-16 $49.95

SUNTRONICS CO., INC.

In Calif Call 213-644-1149
Outside Calif Call 1-800-421-5775
Or send Check or M.O. to:
P.O. Box 1957 Hawthorne CA 90250
California residents add 6½% sales tax.
Add $4.00 ship/handling. VISA/Mastercard include expiration date.

Circle 478 on inquiry card.

Exclusively UNIX* and C Information

That's UNIQUE, the oldest and largest independent journal covering the UNIX* and C marketplace.

- product reviews/listings/prices
- applications/tutorials/indexes
- industry inside information

Full year, $54 ($66 Foreign) - write, call or circle reader service number for full brochure and FREE sample copy.

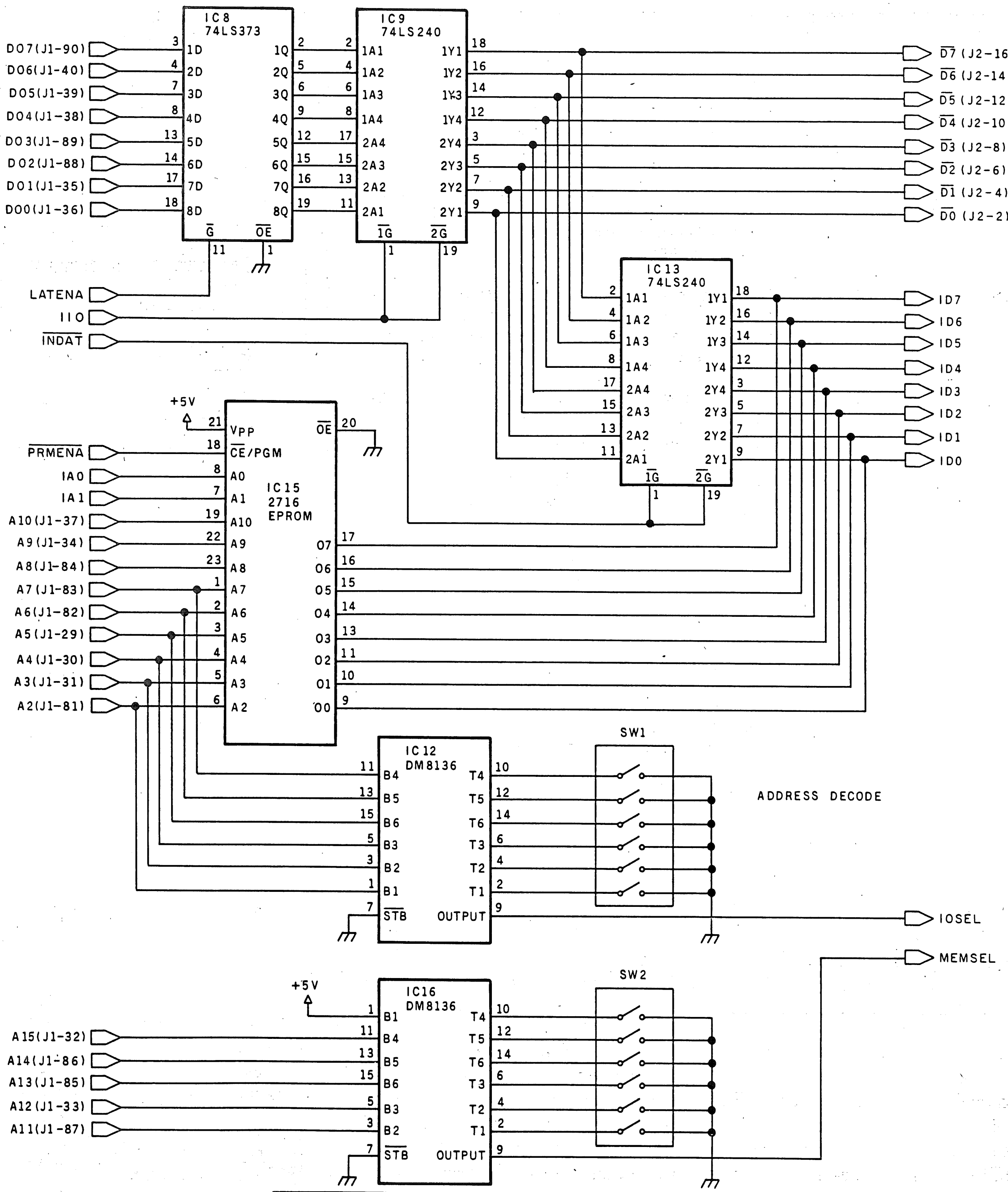
InfoPro Systems

Dept. B • Box 33 • E. Hanover NJ 07936
(201) 625-2925

*UNIX is a trademark of Bell Laboratories

Formerly The UNIX Software List

Circle 479 on inquiry card.



Number	Type	+5 V	GND	Number	Type	+5 V	GND
IC1	74LS10	14	7	IC10	74LS74	14	7
IC2	74LS139	16	8	IC11	74LS74	14	7
IC3	74LS244	20	10	IC12	DM8136	16	8
IC4	74LS00	14	7	IC13	74LS240	20	10
IC5	74LS04	14	7	IC14	74LS04	14	7
IC6	7438	14	7	IC15	2716	24	12
IC7	74LS240	20	10	IC16	DM8136	16	8
IC8	74LS373	20	10	IC17	74LS244	20	10
IC9	74LS240	20	10	IC18	74LS00	14	7

Power table applies to figures 4 and 5, as well.

Figure 3: *Diagram of the address-decode logic of the HCA.*

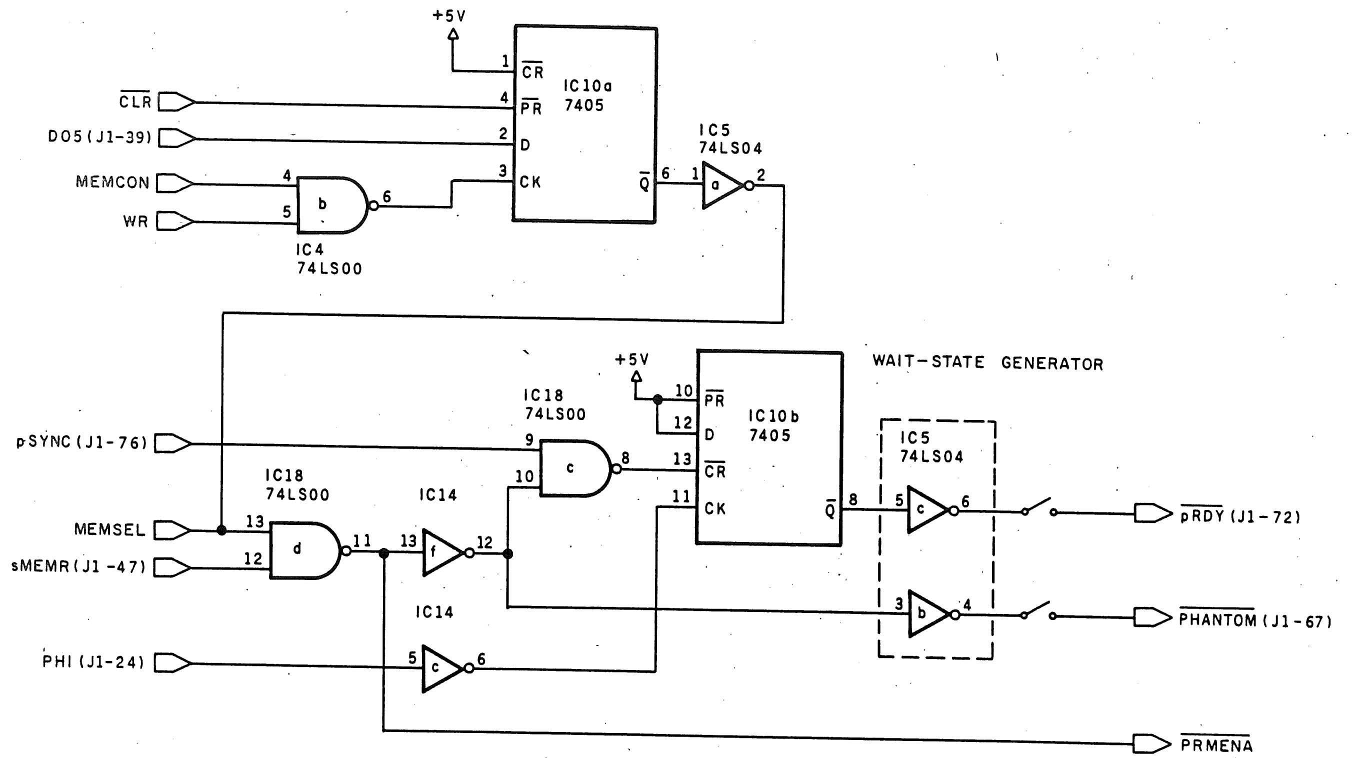


Figure 4: *Diagram showing how the PHANTOM signal and the memory wait state are generated in the HCA.*

Port Number	Action	Function
0	READ	reads SASI data onto S-100 data lines
	WRITE	writes S-100 data to SASI data port
1	READ	returns four status bits on S-100 data lines
	WRITE	generates SASI SELECT signal
2	READ	no action
	WRITE	with data bit 5 set turns PROM on
3	READ	no action
	WRITE	generates SASI RESET signal

Table 2: *I/O port read/write functions of the host computer adapter (HCA), relaying information between the S-100 bus and the disk controller.*

dicated in the diagram. Figures 3 through 5 show the result of this expansion. Figure 3 shows the address-decode logic and the S-100/SASI data bus buffering. Figure 4 shows how the PHANTOM signal and the memory wait state are generated, and figure 5 shows the port-decode logic and the remainder of the system.

The memory and port-address logic are set up so that the address of the PROM and the I/O ports are selectable using a set of switches on the HCA board. For the PROM, the upper five address bits are compared with a switch setting on the board and used to generate the MEMSEL signal, which is used by the logic in figure 4 to generate -PRMENA to enable the PROM. Similarly, bits 2 through 7 of the address are compared with switch settings on the HCA to determine if an I/O port on the HCA has been selected. If a match is found, IOSEL is asserted and bits 0 and 1 are used to determine which of four possible I/O ports has actually been selected. At this point there are eight possible operations because each possible I/O port may be either read or written depending on the status of the sINP and sOUT signals on the S-100 bus. Table 2 shows the results of each of these eight possibilities. If ports 0 or 1 (of the block of four selected I/O ports) are read, signals are generated that cause either SASI status or SASI data to be placed on the S-100 data bus. Reading ports 2 or 3 does not cause anything to happen. A write to port 1 generates a select signal at the SASI, and a write to port 0 writes 8 bits of data from the S-100 bus to the SASI data port. A write to port 3 generates a -RST signal at the SASI. Finally, a write to port 2 with data bit 5 set or reset enables or disables the PROM on the HCA.

Figure 4 shows how the wait-state and the phantom-memory options are handled. When the memory is selected, the output of the flip-flop U10 is set low and remains low until the next transition of the PHI clock signal raises it. This sequence generates the -pRDY signal, which is qualified by a switch setting. If the switch setting allows this signal to get to the S-100 bus, the result is a single wait state each time the PROM is selected. This allows use of PROM memory that would be too slow to respond to the high-speed Z80 processor boards on some S-100 systems. The upper portion of the figure also shows the combinational logic used to generate the PHANTOM signal.

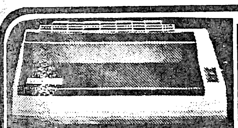
The PHANTOM signal is generated whenever the PROM is selected, depending on the state of flip-flop (U10). This flip-flop enables generation of the PHANTOM signal on receipt of the Power On Clear (-POC) signal and disables generation of PHANTOM when the host writes to port 2 of the four selected output ports with data bit 5 set. This whole operation is qualified by the switch on the board that either allows or disables PROM operation. For systems that support phantom memory, the intended mode of operation is to have the Power On Clear signal (which is the hardware boot signal) enable the PROM and the generation of the PHANTOM signal. The bootstrap in the PROM would then run and load a loader program into RAM. The loader program would then load the CP/M operating system, and (as a final instruction before transferring control to the operating system) the loader program would write data to port 2 with data bit 5 reset to disable the PROM from memory.

Figure 5 shows how this all comes together to perform the required HCA functions. For a PROM access, the upper five bits of the address, qualified as shown in figure 4, form the -PRMENA signal that enables the PROM. The lower eleven bits of the address are sent directly to the PROM to select the byte of data from the PROM that is to be read. This data is then placed on the S-100 data bus using the tristate bus driver (U17).

The way data and control information are transferred to and from the SASI depends on the particular portion of the SASI being addressed. The -RST and -SEL signals are generated by writing to ports 3 and 0, respectively. The -ACK signal is automatically generated if -REQ is asserted any time data is read from or written to the SASI. The control information from the disk controller to the host computer is read by sending an IN instruction to port 1. This places four bits of control information on the S-100 data bus. The data portion of the SASI is addressed by reads and writes to port 0. A data write to port 0 latches the data off the S-100 bus (U8) and passes this data through the data bus driver (U9) to the SASI if it is enabled by the SASI I/O control line. Data is received from the SASI data bus by U13. When the HCA is driving the data bus, the latched data from the S-100 bus will be turned around through U9 and U13 and will be available if a read is directed to port 0. This state exists whenever a controller is not active on the SASI bus and is useful for verifying the HCA data paths.

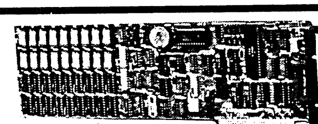
The next step in the design process is to take the logic diagrams shown in figures 3 through 5 and reduce them to a printed-circuit card that can be inserted in a standard S-100 bus 100-pin socket. Because the S-100 bus is fixed, the only choice in this operation is to decide how the HCA card will be physically connected to the disk controller card. For our controller card, this is done using a

Text continued on page 319

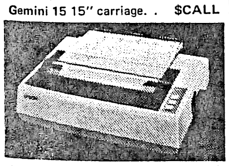


Star Gemini-10 100 CPS, 9x9 hi res. graphic, friction/tractor feed, 80 column, Parallel interface.. $CALL
Gemini 15 15" carriage. . $CALL

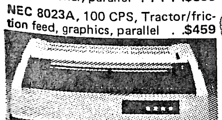
NEC SPINWRITER 3550
NEC's new letter-quality printer especially designed for the IBM PC. 203 columns, auto proportional space, justification, bi-directional, 350 words per min. Serial . $1,995



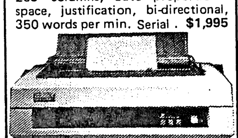
AST Multi-Function Enhancement Cards for IBM PC



The new Epson FX-80 in stock now! Has all the features of the famous MX-80 plus the speed of 160 CPS.
Epson MX100 III - *New lo price.* Call
C. Itoh ProWriter 8510AP, 120 CPS, 80 col., Graphics, Parallel . . $459
C. Itoh ProWrtier-II 1550, 120 CPS, 136 col., 2K buffer, parallel . . $699
Okidata ML82A, 120 CPS, 80 col., graphics, parallel $399
Okidata ML83A, 120 CPS, 136 col., bi-directional, friction/tractor feed, dual parallel/serial interface . $699
Okidata ML84, 200 CPS, 136 col., bi-directional, friction/tractor feed, parallel interface $999
Okidata ML84S, same as ML84 but serial interface. $1,099
Okidata ML92, The newest! 80 col. 160 CPS, friction/tractor feed, graphics, parallel interface. . . . $559
Okidata ML93, *The newest!* 160 CPS, 136 col, tractor/friction feed, bi-directional, parallel $899
NEC 8023A, 100 CPS, Tractor/friction feed, graphics, parallel . $459

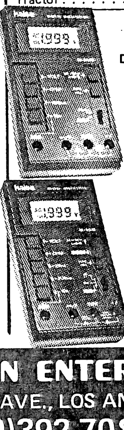


C. Itoh F-10 StarWriter
40 CPS daisywheel (parallel) $CALL
55 CPS daisywheel (parallel) $1,675



BROTHER HR-1
Letter quality daisywheel printer 15 CPS, bi-directional, friction feed, parallel interface. Super value $759
Serial interface $829
Tractor $119

COMREX CR-1-C
17 CPS daisywheel printer, bi-directional, Parallel $769
CR-1-S Above but serial . . . $799
Tractor $119



Kaise
Autoranging
Digital Multimeter
3.5-digit LCD w/ auto unit & sign indication. Full auto ranging on Volt & Ohm w/ 10Amp AC/DC range.
Model SK-6440
Special $89.50
Above features plus beeper for continuity tests, overload warning. Zero adj. cancels resistance value of test leads for more accurate measurements.
Model SK-6330
Special $129.50

ComboPlus Cards
ComboPlus cards offer a maximum of 6 features. User expandable parity memory from 64K to 256K and optional features can be added at a later date. Disk emulator (SuperDrive) and print spooler (SuperPrint) software programs are included with each card.

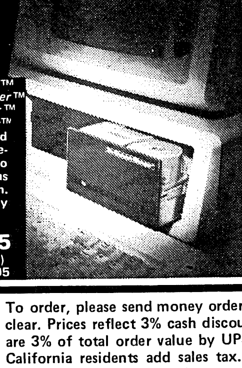
Amount of memory installed on card	64K	128K	192K	256K
with one option (serial or parallel port or clock)	$335	$447	$558	$669
with two options (SP or SC, or PC)	$375	$487	$598	$709
with all options (serial, parallel ports & clock)	$405	$517	$628	$739

MegaPlus Cards
MegaPlus cards offer a maximum of 8 features. User expandable parity memory from 64K to 512K. Serial port No. 1 and clock/calendar included with each card. Disk emulator & Print spooler programs are also included.

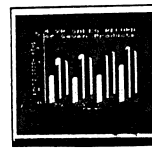
Amount of memory installed on card	64K	128K	192K	256K
w/ standard features indicated above	$395	$506	$617	$728
w/ either parallel or additional serial port No. 2	$435	$546	$657	$868
w/ both parallel and additional serial port No. 2	$465	$576	$787	$898
MegaPak option for 512K expansion				$399

I/O Plus Cards
I/O cards comes w/serial port No. 1 and clock/calendar as standard features
standard card with one serial port No. 1 & clock/calendar. $145
with either parallel port or additional serial port No. 2. . . . . $185
with both parallel port and additional serial port No. 2 . . . . $225

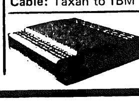
THE BUSINESS MANAGER™
The most complete and affordable business package on the market ... with all the software and storage you ever wanted ... just plug into your IBM PC and run your business!
★ Fully Integrated Accounting System
★ *GENERAL LEDGER*
★ *ACCOUNTS RECEIVABLE*
★ *BILLING*
★ *INVENTORY CONTROL*
★ *ACCOUNTS PAYABLE*
★ *PAYROLL*
★ *Word Processor - Perfect Writer™*
★ *Spelling Checker - Perfect Speller™*
★ *File Management - Perfect Filer™*
★ *Elec. Spreadsheet - Perfect Calc™*
All programs self-prompting and menu-driven. All software pre-installed on hard disk and ready to use. No more diskettes and tedious paper work. High speed operation. All software and hardware fully guaranteed and supported.



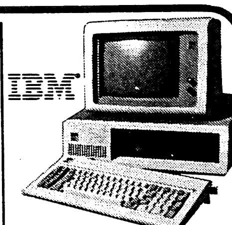
5MB HARD DISK plus above listed software---- $2,995
(Requires 128K and power supply)
Power Supply (required) . . . $195



MONITOR SALE
★★★ GREEN SCREEN ★★★
Dynax GM-120, 12" green, 600 line. 20 MHz. Very sharp looking. List $200.00. . . . . . $129.00
Taxan KG-12N, 12" green 800-dot, List $199.00. . . . . . . $149.99
Zenith ZVM121, 12" green, 15MHz List $139.00. . . . . . . $119.00
★★★ AMBER SCREEN ★★★
Dynax AM121, 12" umber, 600-dot 20 MHz. Excellent! List $250.00. . . . . . $149.00
Taxan KA12N, 12" amber, 800-dot, List $205.00. . . . . . . $150.00
★★★★ RGB COLOR ★★★★
Electrohome 13", Hi res., 580 dot, 16 colors, IBM PC direct drive. List $975.00. . . . . . . $795.00
NEC JC1203 RGB Color. Ideal for IBM PC. 690 dots hi res. 2,000 char. (8x8 dots). List $895.00 . . . . $749.00
Cable: NEC to IBM PC . . . $19.00
Princeton HX-12, 12" RGB color, designed with an NEC .31mm dot pitch color tube. Hi. res. 690 dots. 80 char. x 25 lines. Identical IBM cabinet. Most pleasing & best price-performance monitor for your IBM PC. List 795 . . . Best buy $CALL
Taxan RGBvision I, 12", 380-dot, 18 MHz. 2,000 char. (5x7 dots). List $399.00. . . . . . . $365.00
Taxan RGBvison II, 12", 510-dot, 18 MHz. 2,000 char (5x7 dots). List $599.00. . . . . . . $549.00
Taxan RGBvision III, 12", 630-dot, 18 MHz. 16 colors. List $699.00. . . . . $640.00
Cable: Taxan to IBM or Apple III $19



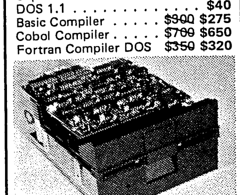
Hayes Smart Modem
300 ------ $225
1200------ $539



IBM Personal Computers
A. 64K system unit & keyboard, two 320K drives & controller, color/graphic card. $2,850
B. 64K system unit & keyboard, two 320K drives & controller, IBM monochrome display & display/printer adapter. $3,225
Other configurations available

SOFTWARE FOR IBM PC
NEW! TK!Solver — by the inventor of VisiCalc. A fantastic software that is so easy to use that requires no programming knowledge. It can also produce table & graphs.. . $249

VisiCalc	~~$250~~	$179
Desktop Plan	~~$300~~	$229
VisiDex	~~$250~~	$189
VisiFile	~~$300~~	$235
VisiSchedule	~~$300~~	$229
VisiTrend/Plot	~~$300~~	$229
SuperCalc		$199
DOS 1.1		$40
Basic Compiler	~~$300~~	$275
Cobol Compiler	~~$700~~	$650
Fortran Compiler DOS	~~$350~~	$320



5¼ TANDON DRIVES
The newest. IBM PC compatible. Installation instruction included.
TM100-1A, SSDD 160K . . . $175
TM100-2A, DSDD 320K. . . $259

MasterCard
EASTERN ENTERPRISES, INC.
2937 S. VAIL AVE., LOS ANGELES, CALIF. 90040
TOLL-FREE Order Desk (800)392-7081
Calif., Alaska, Hawaii (213)725-3080
VISA
TELEX 664477

To order, please send money order or cashier's check. Personal or company checks require 2–3 weeks to clear. Prices reflect 3% cash discount. Visa/Mastercard accepted. Shipping, insurance & handling charges are 3% of total order value by UPS Surface, 5% by UPS Air or Parcel Post. COD's requires 20% deposit. California residents add sales tax. No sales tax on out-of-state-orders. Prices & availability subject to change without notice. We are no responsible for typesetting errors. Send for catalog.

Circle 230 on inquiry card.

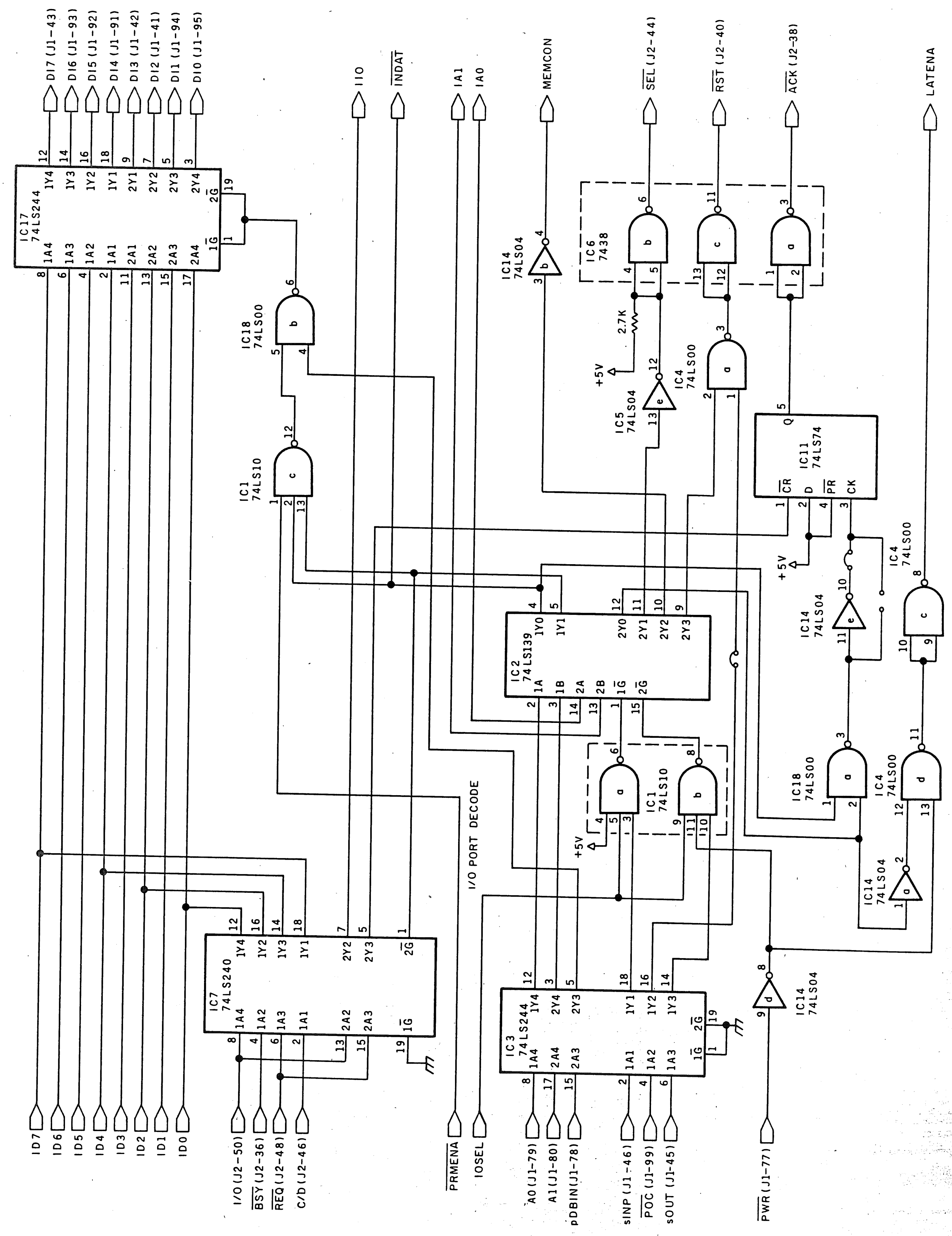


Figure 5: *Diagram of the port-decode and HCA-interface logic.*

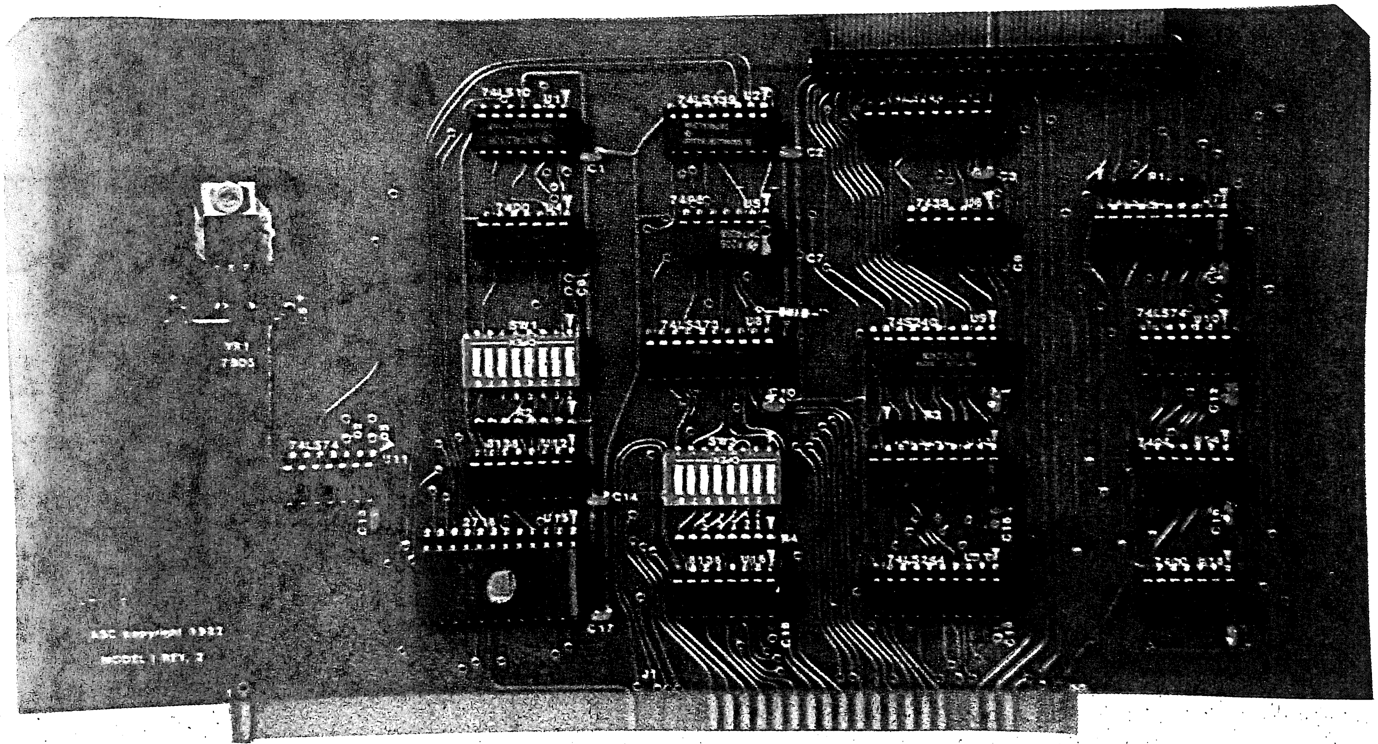


Photo 1: *The host computer adapter (HCA) printed-circuit card.*

50-conductor ribbon cable, which is the SASI standard. Photo 1 shows the printed-circuit card that results. For anyone interested in building only one of these devices, the HCA could also be implemented on a wire-wrap card instead of a printed-circuit card.

Power-Supply Characteristics

The HCA board obtains its power from the S-100 bus so the only elements of the Winchester disk subsystem that require external power are the disk drive and the controller card. Both these devices require +5 and +12 volts (V) DC. Adding up the power requirements for both devices at both voltages results in total power requirements of about 1.5 amps at +12 V and about 3.0 amps at +5 V.

Several options were available for providing this power. These included designing and constructing a power supply, buying a commercially available linear supply, or finding a commercially available switching supply. Fortunately we were able to find a commercially available switching supply that provided 2.0 amps at +12 V and 3.0 amps at +5 V. This exactly fit our power requirements. Because it was a switching supply it was small enough physically that we could include it in a small enclosure with the disk drive and controller card, and the supply was relatively efficient so that power dissipation and cooling problems in the disk enclosure would be minimized.

System Packaging

Choosing the power supply completed the overall design of the Winchester disk subsystem. The only remaining step was to devise a way to package the system. This was particularly important due to the characteristics of the Winchester disk drive. Photo 2 shows the Miniscribe disk as it is delivered from the manufacturer. From this view it is apparent that the disk drive, as is true of all Winchester disk drives, is intended to be mounted

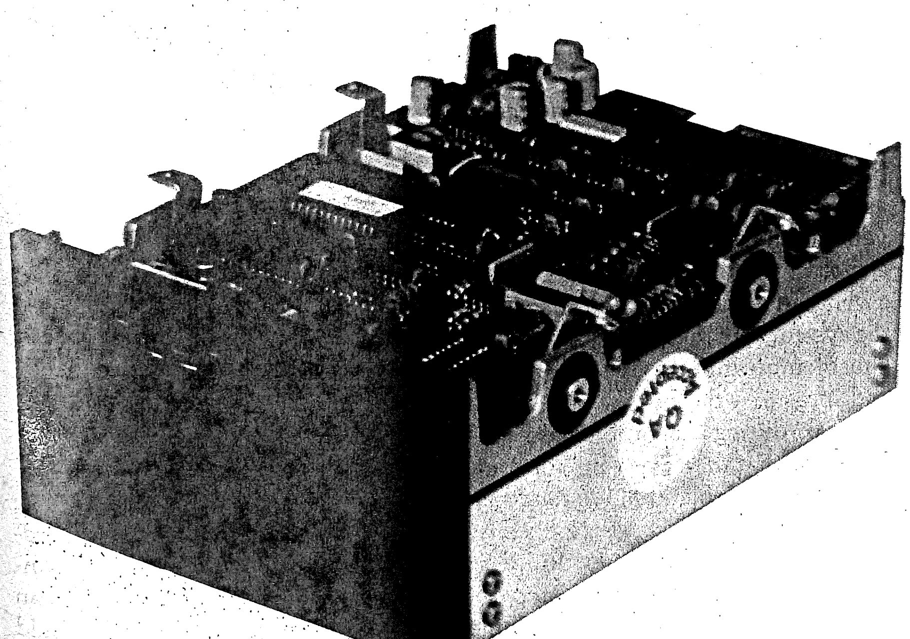


Photo 2: *The Miniscribe 5¼-inch Winchester disk assembly.*

in some type of enclosure. In fact it is even more important to install this disk in an enclosure than the photo would indicate. The disk is meant to be installed using the isolation points shown in the photo to isolate the disk mechanically from the rest of the system. These isolation points absorb high-frequency shocks occurring during handling that could result in damage to the very sensitive read/write head of the drive. Thus it is imperative to install the disk in an enclosure as soon as possible and to exercise extreme care in handling the disk prior to its installation.

The controller card and the power supply we chose had dimensions that made it feasible to mount both these elements in the same enclosure as the disk drive. Photo 3 shows all these elements before assembly. We chose to mount this equipment in a box 7 inches wide, 12 inches deep, and 7¼ inches tall. In our opinion, a box this size approached the median between having a small, compact system and still providing enough free space in the enclosure for adequate ventilation.

The enclosure selected had a removable top, and photo 4 shows how these various components were mounted in the box. The disk drive was bolted to channels attached to the box. The controller card was attached to the bottom of the box using angle brackets, and the power supply was bolted directly to the bottom of the box. We constructed a cable harness to route electrical power from the power switch to the power supply and on to the disk drive and controller. Also we fabricated three ribbon cables. One of these was used to connect the HCA to the controller board and the other two were used to connect the controller board to the disk drive. Photo 4 shows the installation with all the cables in place. As the photo shows, the installation is still rather compact but has sufficient room for ventilation.

One of the last but, from a reliability standpoint, one of the most important design considerations was to allow sufficient ventilation of the assembled product to prevent excessive heat buildup. Photo 5 shows

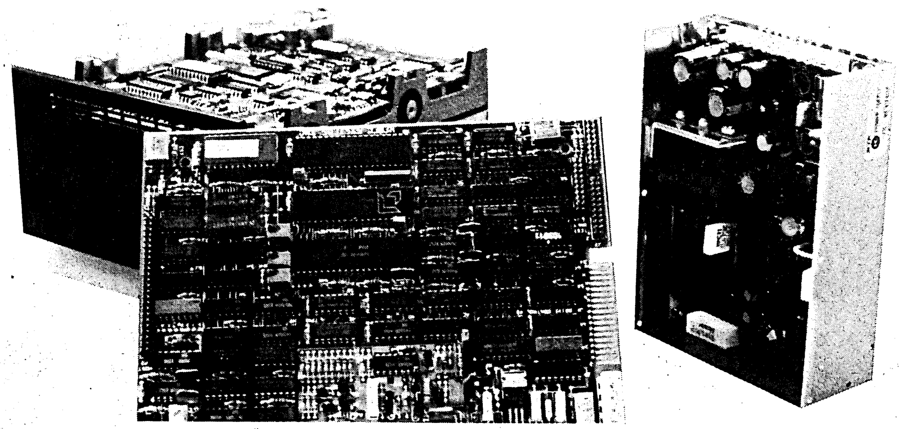


Photo 3: *The Winchester disk-drive subsystem components before assembly.*

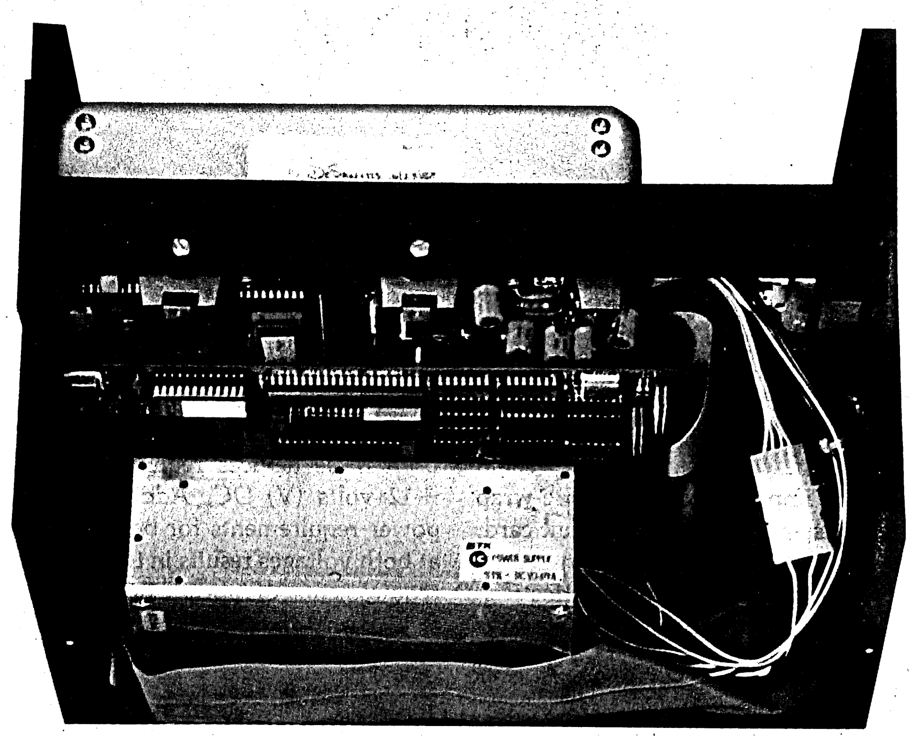


Photo 4: *The completed disk-drive assembly.*

both the front and back of the assembled enclosure. We punched ventilation holes on the front of the enclosure and on the rear we included a ventilation fan to force air through the system. We then measured the exhaust temperature of the air at the fan and measured the temperature inside the box during operation to assure ourselves that there were no hot spots in the box. The fact that we found no hot spots was not surprising because all the equipment in the enclosure is designed to be convectively cooled, and the fan we chose had a volume flow rate that replaced the air in the enclosure about 40 times every minute. The end result of the entire design operation was the set of hardware shown in photo 6, a forced-air enclosure containing the disk drive, controller, and power supply; an HCA card that plugs into an S-100 backplane; and a 50-conductor ribbon cable to connect these two elements.

Summary

In this article we have described the design and construction of the hardware necessary to interface a Winchester technology disk drive with an S-100 computer system. You should now have a detailed understanding of what is required to perform this type of integration, and with sufficient

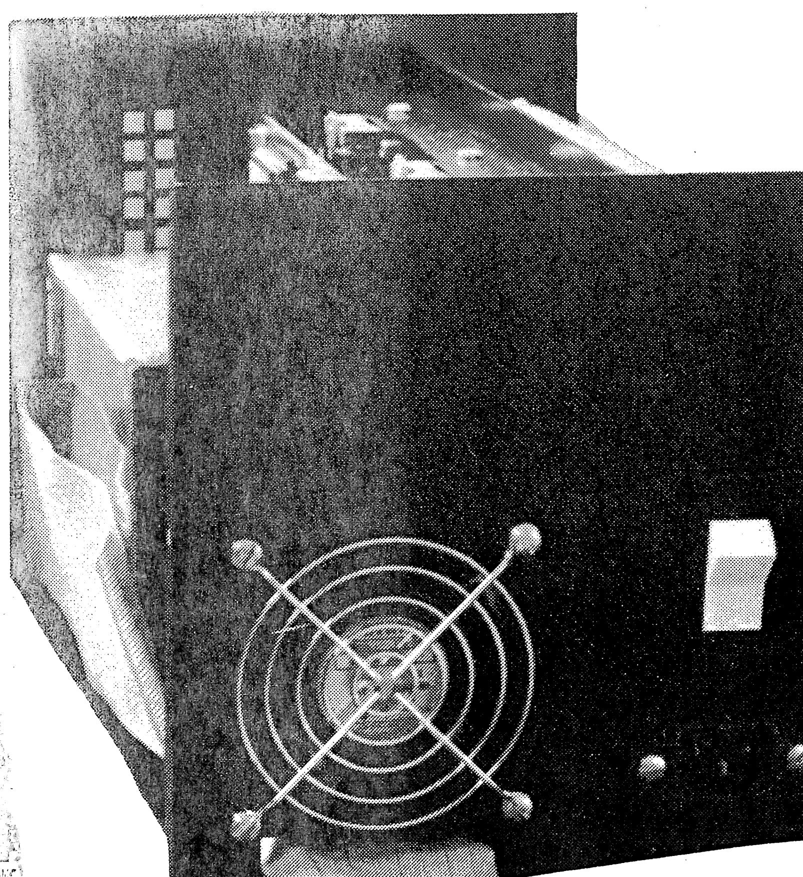


Photo 5: *Ventilation details of the disk-drive assembly.*

perseverance you should be able to design and construct one of these systems. Next month we will finish this series by describing the various software components that must be developed to interface the hardware with the system. Specifically, we will cover how to write a BIOS that combines the BIOS for the existing peripherals on the system with a set of BIOS routines to handle the hard disk. We will also discuss how to write a relocatable bootstrap loader for the HCA PROM and discuss some of the methodology involved in testing and debugging the various hardware interfaces in the system. ■

The Winchester disk-drive subsystem described in this series of articles is available as a completely assembled unit from ASC Associates of Lexington Park, Maryland. In addition to the S-100 version discussed, versions are also available for TRS-80 and Apple computers. The disk-drive systems for these computers use the same drive and controller hardware as the S-100 version but use a different host computer adapter and interface software. Until a nationwide dealer distribution network is established, these systems will be available by mail order for $1995. To order or obtain further information, write to ASC Associates Inc., POB 615, Lexington Park, MD 20653, or phone (301) 863-6784.

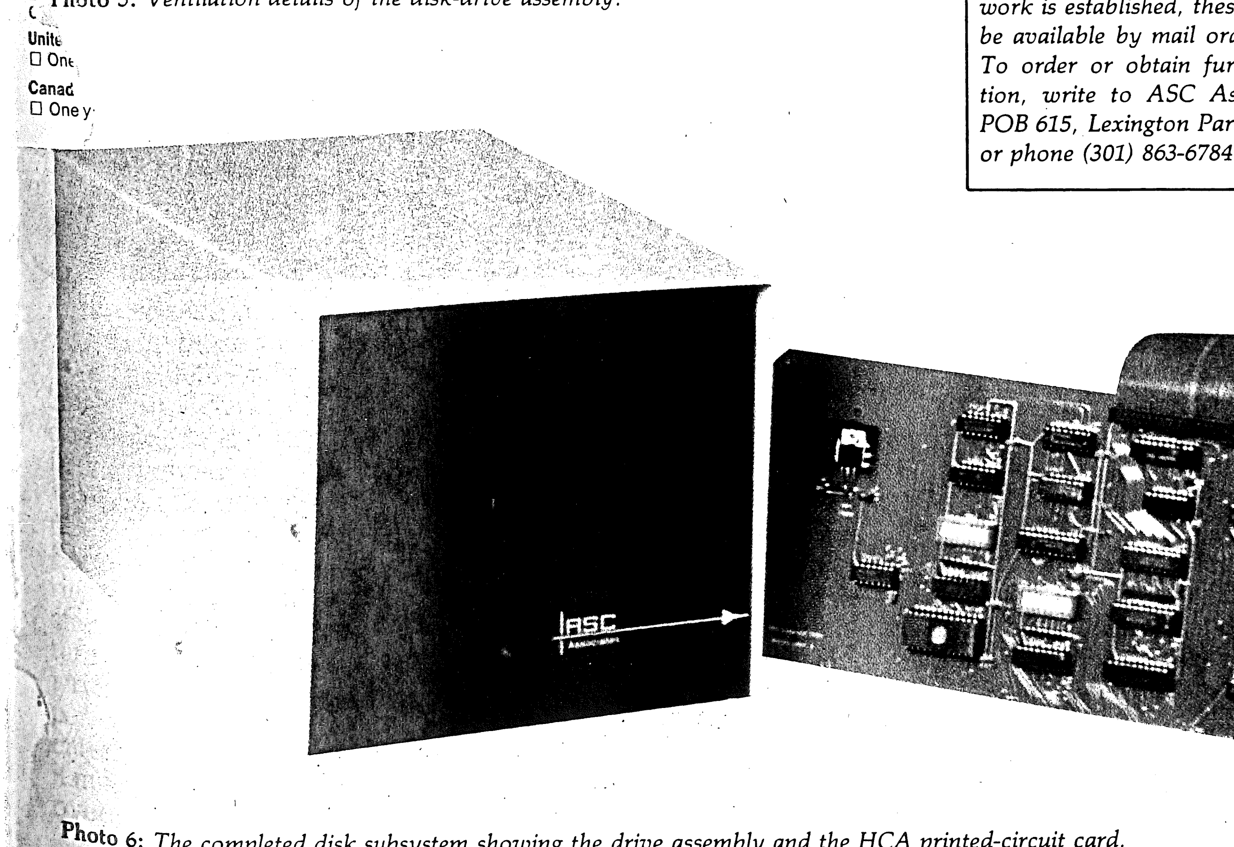


Photo 6: *The completed disk subsystem showing the drive assembly and the HCA printed-circuit card.*

Building a Hard-Disk Interface for an S-100 System

Part 3: Software

How to alter the CP/M operating system so that it will accommodate a Winchester disk drive and controller.

Andrew C. Cruce and Scott A. Alexander
ASC Associates Inc.
POB 615
Lexington Park, MD 20653

In part 1 of this series we described Winchester disk technology in general and the benefits it would provide for microcomputer systems. We then gave an overview of the work that would be required to add such a disk to an existing computer system. In part 2 we discussed the rationale you should use in choosing a particular Winchester disk drive and disk controller (we decided on a Miniscribe disk drive and a Xebec controller). We then described in detail the construction of the hardware interface—the host computer adapter—required to integrate the disk drive into the S-100 system. In this last article of the series we will describe the final ingredient needed for our Winchester disk-drive subsystem: the software.

First we'll review the operation of the CP/M basic disk operating system (BDOS) and basic input/output system (BIOS) software. Specifically we will describe the process through which application programs use the BDOS to access disk file information and how the BDOS uses the BIOS to obtain specific information from a particular peripheral. This will highlight what has to be done to write a BIOS to handle the Winchester disk system.

Next we will describe the operation of the disk drive and controller as accessed through the host computer adapter (HCA). We'll show how disk commands are initiated from software and what commands are available from the disk controller. With this background out of the way, we will identify requirements for a Winchester disk-subsystem BIOS and how these requirements can be satisfied. We will then describe how a BIOS is structured and how the BIOS routines for the Winchester disk system can be included with the original system BIOS routines to support the other peripherals. Finally, we will briefly review the CP/M procedures required to include the combined BIOS in a new CP/M system.

This will essentially complete the integration process. However, we can improve system performance even further by putting the system's bootstrap routine on the Winchester disk. We'll describe how a bootstrap program operates and how you can develop a new bootstrap and install it on the Winchester disk drive. We'll also spend some time discussing various debugging techniques that will probably be very helpful when you integrate the hardware and software.

BDOS and BIOS

The CP/M operating system basically consists of two separate elements called the BDOS and the BIOS. The BDOS is supplied by Digital Research Inc. and is the essence of the CP/M operating system. In addition to the BDOS, however, the system requires a set of routines known as the BIOS to handle the hardware-peculiar functions of each peripheral in the system. These routines are usually supplied by the disk-drive manufacturer and must be modified by the user to include other system peripherals.

Accesses to disks and other peripherals by application programs are usually handled by calls to the BDOS. These calls are made by loading specified registers with information required by the BDOS and the performing a call to location 05. In the case of disk accesses you must load a function code into the C register and a pointer to the file control

About the Authors

Andrew Cruce has a Ph.D. in Aeronautical Engineering and has recently received an S.M. degree in management as a Sloan Fellow at MIT. Scott Alexander has an M.S. in Electrical Engineering. Both have extensive design and implementation experience with small computers and are full partners in the firm of ASC Associates, which markets the hardware described in this series of articles.

Circle 112 on inquiry card.

BYTE	DESCRIPTION
00	POINTER TO SELECTED DISK
01 : 08	FILE NAME
09 : 11	FILE TYPE
12	CURRENT EXTENT NUMBER OF FILE
13 14	RESERVED FOR SYSTEM USE
15	READ COUNT FOR EXTENT
16 : 32	DISK ALLOCATION INFORMATION
33 : 35	OPTIONAL RANDOM RECORD NUMBER FOR DIRECT-ACCESS INPUT/OUTPUT

Figure 1: *A diagram of the file control block (FCB) format used by CP/M.*

block (FCB) of the desired file into the DE register pair. With this information, the BDOS determines what functions are to be performed and calls appropriate entry points in the BIOS with the information required to execute requested functions.

The BDOS disk functions are described in the standard CP/M documentation and include such functions as Open a File, Close a File, Read Next Record, and Write Next Record. The data structure that drives all these operations is the FCB, which is initially created by the application program and is updated by various BDOS functions. Figure 1 shows the structure of the FCB, which includes, among other things, the file name and file type along with 16 bytes of data that are used by CP/M in the calculation of a physical device address for access to the requested data. Additional data in the FCB is used to keep track of the drive the FCB is currently active on, a pointer to the current record, and a pointer to the current extent. If you are not familiar with the FCB construct, you can find additional information on it and the normal BDOS disk I/O functions in Digital Research's *CP/M Interface Guide*.

In order to understand the requirements for the Winchester disk BIOS, it is necessary to understand how this BIOS is used during the normal access of data from files contained on the Winchester disk. Figure 2 illustrates the steps that are performed in opening an existing disk file and reading the first 128 bytes of data from this file. This process is representative of the majority of the communications that occur between the BDOS and the BIOS during normal disk operations.

The process starts with the application program establishing which disk is to be active in the subsequent operations by loading the appropriate information in the C and E registers, as shown in the figure, and then calling the BDOS. The BDOS takes this information and passes it on to the BIOS, which then returns to the BDOS the address of a table that defines the physical characteristics of the disk that was selected. At this point, control is returned to the application program.

Next, the application program defines a DMA (direct memory access) buffer for subsequent disk operations by loading the DE register pair with the DMA buffer address, loading register C with 1A hexadecimal, and calling the BDOS. The BDOS in turn passes the DMA address to the BIOS for use in subsequent disk read/write operations.

After setting up the DMA buffer, the application program next opens the file that is to be read. First, the application program constructs an FCB for that file by reserving the required amount of space for the FCB, filling in the file-name and file-type portions of the FCB, and setting the Current Extent and Next Record fields to zero. The application program then calls the BDOS with the DE register pair pointing to the FCB and register C containing 0F hexadecimal. The BDOS must now search the file directory on the selected disk to determine if the file mentioned in the FCB is actually contained on the disk. In doing this the BDOS uses the BIOS to read each sector of the selected disk directory into memory and then searches for a match with the requested file name. When a match is found, the BDOS uses the information contained

COMPUTER SCIENCE PRESS

ESSENTIALS OF COBOL PROGRAMMING: A STRUCTURED APPROACH
Gerald N. Pitts and Barry L. Bateman

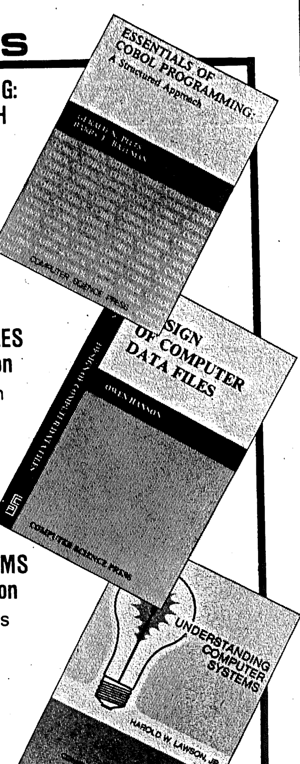
The COBOL language is presented in an easy-to-understand progression in this well-organized new book, providing a knowledge of the language and an understanding of how the various aspects of COBOL are interrelated. Questions at the end of each chapter make this an excellent choice for independent study. *1982. $14.95. 0-914894-34-X.*

DESIGN OF COMPUTER DATA FILES
Owen Hanson

This excellent new book explains how to design files for optimum performance and minimum cost. Intended for students of programming, data processing, and systems analysis, or as a reference work for practicing designers, this book brings together all the latest information necessary for the design of computer data files. *1983. $24.95. 0-914894-17-X.*

UNDERSTANDING COMPUTER SYSTEMS
Harold Lawson

Learning about computers can be enjoyable, as proved by this interesting new book by Harold Lawson. The author draws parallels between computing procedures and familiar everyday events to help the reader learn how computers work. A must for those who want to learn more about computer systems. *1982. $11.95. 0-914894-31-5.*



WRITE FOR A FREE CATALOG

All prices are subject to change without notice. Postage and handling is $1.50 for the first book, 50¢ for each additional book. Residents of Maryland add 5% sales tax.

COMPUTER SCIENCE PRESS, INC.
11 Taft Court, Dept. BY583 Rockville, Maryland 20850 (301) 251-9050

Circle 128 on inquiry card.

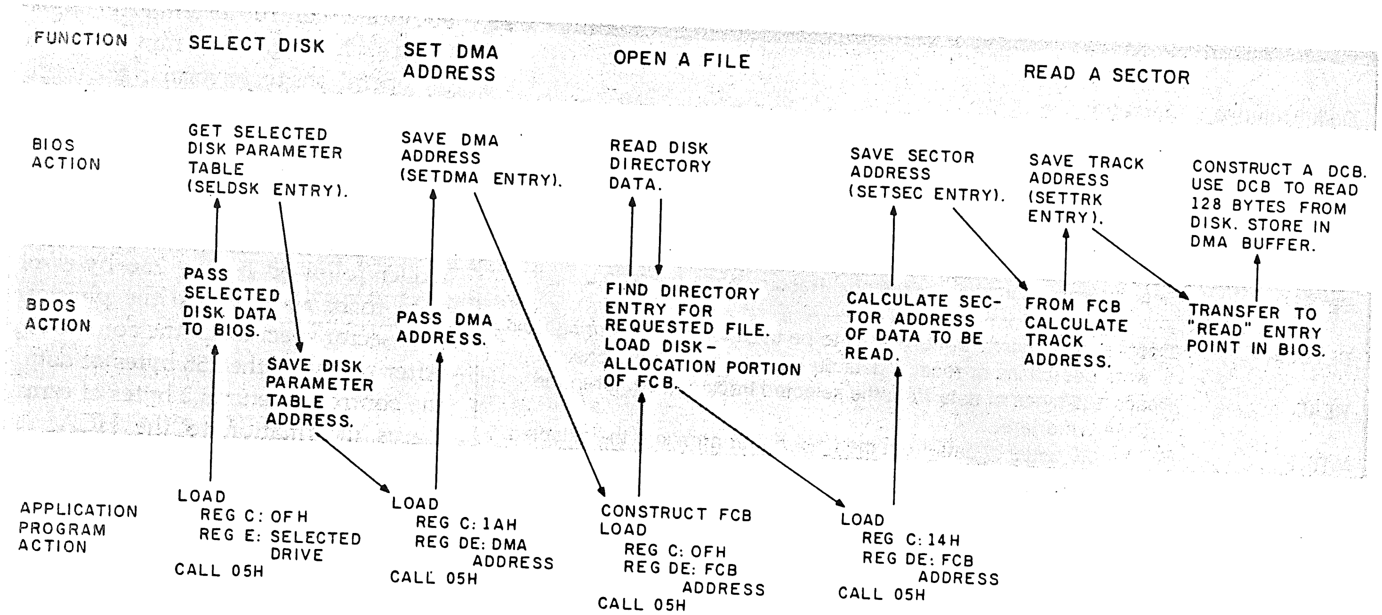


Figure 2: *A diagram of the input/output process for the Winchester disk subsystem.*

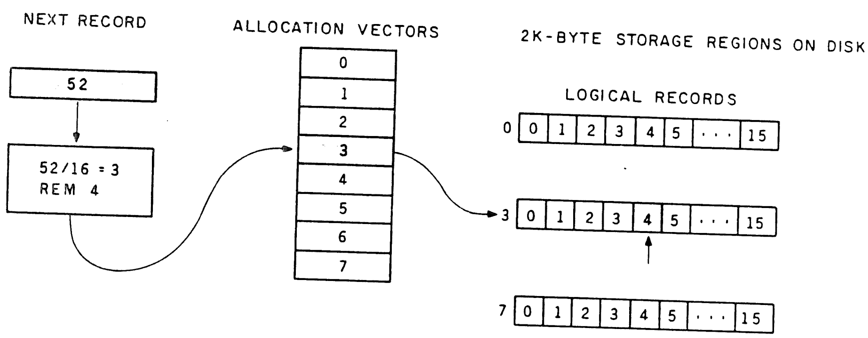


Figure 3: *A diagram showing how the Winchester disk address is calculated. First, the value of the Next Record field is divided by 16. The resulting quotient points to one of eight allocation vectors, which in turn points to one of several 2K-byte regions on the disk. The remainder of the division points to one of the 16 logical records in each 2K-byte region.*

in the directory entry to load the disk-allocation portion of the FCB. After filling in the required information in the FCB, the BDOS returns control to the application program.

The function of the disk-allocation portion of the FCB is to determine the physical disk address of particular records within a disk file. In the case of our Winchester disk system, the 16 bytes of disk-allocation information correspond to eight pointers, each containing 16 bits. These pointers refer to 2K-byte storage areas on the disk. Each of these 2K-byte storage areas is in turn made up of sixteen 128-byte logical sectors of disk information as required by CP/M. When a particular record is accessed, the value in the Next Record field of the FCB is divided by 16 to determine which disk-allocation vector to use in the disk-address calculation, and the remainder from this division is used to determine which logical sector in the storage region is required. This process is shown in more detail in Figure 3. It should be noted that a single file entry only provides access to 16K bytes on the disk. To access a larger file you must use extents, which are duplicate file entries containing unique pointers for different portions of the file.

The information derived from these calculations is used by the BDOS when the application program next issues the Read command to the BDOS. In this case the DE register pair is loaded with the address of the FCB, and register C is loaded with 14 hexadecimal when the BDOS is called. The BDOS then uses the data in the FCB to calculate a sector and track address for the requested data. First it passes the sector-address information to the BIOS (through the SETSEC entry point) followed by the track address information (through the SETTRK entry point). We'll discuss the design of the BIOS and the required entry points later. After the sector and track information have been passed, the BDOS then asks the BIOS to perform a Read operation of the identified sector.

At this point the BIOS takes the sector- and track-address information and constructs a device control block (DCB) that commands the Xebec controller to read the data from the requested sector of the disk. As we explained earlier in this series, the commands go from the computer, through the host computer adapter (HCA), to the disk controller. The disk controller then performs the requested Read operation from the disk, placing the data in a local memory area on the controller card. The BIOS can then begin to access the data being read from the disk and move it to the DMA memory buffer specified by the original SETDMA command. Once

Disk Primitive	Operation
SELDSK	Selects a particular disk in the system as the "active" disk. The routine must keep track of which disk is selected and pass the address of a disk-characteristics table describing the selected disk to the BDOS.
SETTRK	Sets the track number for the next Read or Write operation.
SETSEC	Sets the sector number for the next Read or Write operation.
SETDMA	Defines the 128-byte buffer that is to be used to get data during disk-write operations or receive data during disk-read operations.
READ	Reads 128 bytes of data from the selected track and sector in the DMA buffer area.
WRITE	Writes 128 bytes of data from the DMA buffer area into the selected track and sector.
SECTRAN	Performs logical-to-physical sector translation to improve overall CP/M disk response.
HOME	Moves the head on the selected disk to sector 0, track 0.

Table 1: *A list of disk-related primitive functions that have to be performed by the Winchester disk BIOS in order to be compatible with CP/M.*

all 128 bytes of the requested logical sector are moved into the DMA area, the Read operation is complete and control is returned to the application program.

In this overview of the combined operation of an application program, the BDOS, and the BIOS in accessing and reading information contained on the disk, some of the intricacies of the process, such as extents of disk files and detailed operation of the BDOS, have been glossed over. However, the illustration should provide enough information for you to understand the design and construction of a BIOS for a Winchester disk subsystem, and we refer you to CP/M documentation to gain a more in-depth understanding of the CP/M file control services (FCS) and the operation of the BDOS.

As you can see from the above example, only a few primitive functions have to be performed by a Winchester disk BIOS for it to be compatible with a CP/M system. Table 1 presents a complete list of these primitive functions along with a brief description of each function. In a complete BIOS, each of these functions is a separate entry point into the BIOS, and all you have to do to establish a BIOS is write code for each entry point to perform the necessary function. But first you must understand the hardware operation and software interface to the Winchester disk, which is our next topic.

Hardware Operation and Software Interface

As we discussed last month, the communications interface between the BIOS routines for the Winchester disk drive and the drive itself consists of four I/O ports on the S-100 bus. The addresses of these ports are selected in a contiguous block of four ports by switch settings on the HCA card. We will refer to ports 0 through 3 to indicate particular ports in this block.

Commands and data are sent to the controller by writing to these ports. The HCA retrieves data from the controller by reading from these ports. Each command to the controller consists of a block of 6 bytes of command information. This 6-byte block is the device control block mentioned earlier. Figure 4 shows the general format of the DCB and how this format is used for three different disk commands. The DCB always contains the op code and command types as shown in the general description of the data structure. In addition, the DCB may contain up to 21 bits of physical address information when an actual Read or Write operation is being performed. It also may contain additional control information specifying such things as the number of retry operations to perform in the event that an error condition is detected. For more information on the details of the DCB, see the Xebec controller manual. For a Write command, the 6 bytes of the DCB are sent to the controller followed by the 256 bytes of data to be written into the physical disk sector specified in the command. After receiving the 256 bytes of data, the controller returns 2 bytes of error status information to the HCA. A Read operation works similarly in that the 6 bytes of the DCB are sent to the disk, which then returns 256 bytes of data from the sector specified in the command block. The controller also tags on an extra 2 bytes of error status information at the end of the transfer. In addition, the Xebec controller offers a Request Sense Status function that returns 4 bytes of more detailed error status information at the end of the command.

In typical operation, the disk and controller would be reset on a cold boot by performing an output to port 3. Once this is completed, the disk controller would be put in a command mode by outputting a 1 to the data port (port 0) followed by a Write to the status port (port 1), which activates the SEL signal. (This and many of the signals referred to in this section were described last month.) The disk-access routine then examines the REQ line by performing an input from port 1 and testing the proper status line. When the REQ line becomes active the routine outputs the first of the 6 command bytes to port 0. This output automatically generates the ACK signal to complete the handshake between the HCA and the controller. The software then monitors the REQ line until it again becomes active and then sends the second command byte. This process is repeated until all 6 command bytes have been sent to the disk controller. The routine then uses the same process to read or write the appropriate number of data bytes at port 0 depending on the particular command that was sent. In the case of a BIOS routine, the disk address that is to be accessed is derived by the BDOS from

(4a) GENERAL DCB FORMAT

BYTE	Bits 7 6 5	Bits 4 3 2 1 0
0	COMMAND CLASS	OP CODE
1	LUN	HIGH ADDRESS
2	MIDDLE ADDRESS	
3	LOW ADDRESS	
4	SECTOR COUNT	
5	CONTROL FIELD	

(4b) DCB FOR A READ OPERATION

BYTE	7	6	5	4	3	2	1	0	
0	0	0	0	0	1	0	0	0	- CLASS = 0, OP CODE = 8
1	0	0	0	HIGH ADDRESS					-D = 0 OR 1 FOR SELECTED DRIVE
2	MIDDLE ADDRESS								
3	LOW ADDRESS								
4	0	0	0	0	0	0	0	1	- READ A SINGLE SECTOR
5	0	0	0	0	0	0	0	0	

(4c) DCB FOR A WRITE OPERATION

BYTE	7	6	5	4	3	2	1	0	
0	0	0	0	0	1	0	1	0	- CLASS = 0, OP CODE = AH
1	0	0	D	HIGH ADDRESS					
2	MIDDLE ADDRESS								
3	LOW ADDRESS								
4	0	0	0	0	0	0	0	1	- WRITE A SINGLE SECTOR
5	0	0	0	0	0	0	0	0	

(4d) DCB FOR A REQUEST SENSE STATUS COMMAND

BYTE	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	1	- CLASS = 0, OP CODE = 3
1	0	0	0						NOT USED
2									NOT USED
3									NOT USED
4									NOT USED
5									NOT USED

Figure 4: *The device control block (DCB) used by the Xebec hard-disk controller. Figure 4a shows the general format for the DCB. Figures 4b through 4d show how the DCB is used for a Read operation, a Write operation, and a Request Sense Status command, respectively. Byte 0 contains information on the Command Class and the Op Code. LUN (logical unit number) indicates which drive you have selected of the two possible drives that can be connected to the controller. The high, middle, and low addresses form a 21-bit physical sector address. Byte 4 indicates the number of sectors to be read or written (for CP/M this is always 1). Byte 5 contains the following control fields: bit 7, when set, disables the four re-tries by the controller on all disk-access commands (during normal use, this is always 0); bits 3 through 6 are not yet used and are set to 0; and bits 0 through 2 select half-step options for other disk drives (for the Miniscribe drive, this is set to 0).*

the information contained in the FCB. This information is then passed to the BIOS routines at the SETTRK and SETSEC entry points and the BIOS uses this information to construct the DCB for the actual disk access.

Generating a Combined BIOS

The BIOS structure is very simple. A table of jump vectors is placed at the beginning of the BIOS code. This includes jumps to each of the primitive functions that the BIOS performs. Table 2 shows that these functions include the disk-oriented primitives presented in more detail in table 3 as well as the necessary functions to handle the other peripherals in the system, such as terminals and printers. The BIOS routines for peripherals other than disks are used without modification in the construction of a new combined BIOS. The existing disk and boot functions are implemented as shown in figure 5. The boot functions for both the existing disk system and the new Winchester disk are combined to ensure that all peripherals on the system are properly initialized when the system is booted. A new SELDSK function is written that keeps track of which disk system is selected and passes the proper disk-definition-table address back to the BDOS. Finally, a series of disk-handling routines are written for the Winchester disk. These routines are used as the initial vector addresses for the initial jump table. Depending on the particular entry point, these routines either perform the indicated function for the Winchester drive and then transfer to the appropriate function for the existing floppy drive or test to see which drive is selected and simply transfer control to the appropriate routine for the selected drive. An example of the first case is the SETDMA function, which causes the DMA buffer address to be set for both the Winchester and any other disk drives in the system. In the case of an actual READ function, however, a test is made for the selected disk and then control is passed to the proper disk-read routine. Listing 1 on page 386 presents a skeleton of a combined BIOS that includes all the BIOS routines for the Winchester disk drive and comments indicating where routines from an existing BIOS should be placed.

The code presented in listing 1 has been tested and will operate with a Xebec controller connected to a Miniscribe disk using the HCA described in last month's article. The SELDSK entry point keeps track of which disk is selected and uses register pair HL to pass the address pointer of the selected disk characteristics table to the BDOS. This table is located at DPBASE in the BIOS code and has been created using two macros, DISKS and DISKDEF, supplied by Digital Research. These macros and the structure of this table are described in Digital Research's *CP/M System Alteration Guide*. The code for the cold boot entry point, BOOT, assumes that a new copy of the sys-

Entry Point	Function
BOOT	System initialization on hardware boot.
WBOOT	Reads a new copy of BDOS into memory and initializes the system.
CONST	Samples and returns the status of the current console device.
CONIN	Reads the next character from the current console device.
CONOUT	Writes a character to the current console device.
LIST	Sends a character to the list device.
PUNCH	Sends a character to the punch device.
READER	Reads next character from the read device.
HOME	Returns the read/write head of the currently selected disk to track 0.
SELDSK	Selects the current disk.
SETTRK	Sets the track for the next Read/Write operation.
SETSEC	Sets the sector for the next Read/Write operation.
SETDMA	Sets the DMA buffer area for the next READ/WRITE operation.
READ	Reads data from the selected disk.
WRITE	Writes data to the selected disk.
LISTST	Gets the status of the list device.
SECTRAN	Translates logical sector to physical sector.

Table 2: *A complete list of primitive functions for the BIOS in CP/M. Note that this includes the disk-related primitives in table 1.*

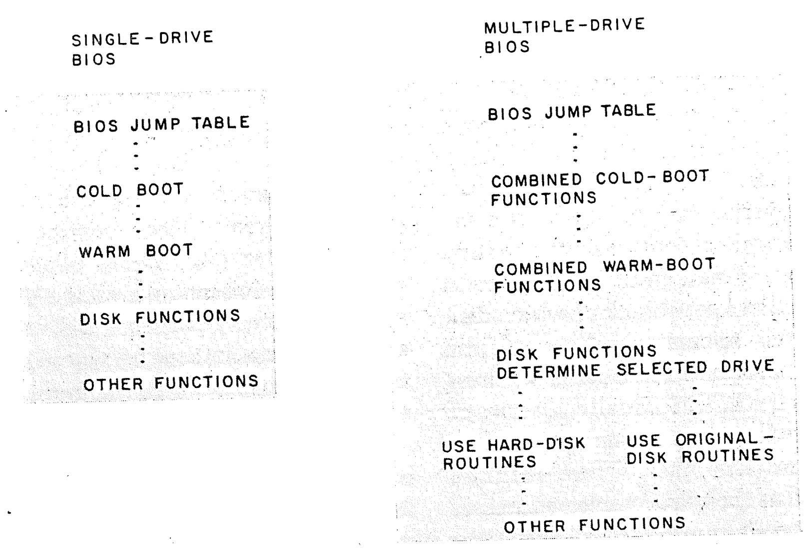


Figure 5: *The combined BIOS structure.*

tem has already been booted into memory and simply performs any required initialization before transferring control to CP/M. The warm boot, WBOOT, reloads a new copy of the BDOS and the console control program (CCP) before transferring control back to CP/M. The HOME, SETTRK, SETSEC, and SETDMA entry points perform the commanded function for both the Winchester disk system and for the existing disk system. Finally, the READ and WRITE entry points determine which disk is currently selected and transfer control to the appropriate routines to read or write data from or to the selected disk.

Another complication concerning Read and Write operations on the Winchester disk drive arises from the fact that the Winchester disk is set up to read and write 256-byte physical sectors, whereas CP/M operates in logical sectors of 128 bytes. This means that each disk-read operation places two logical CP/M sectors into the local memory on the controller card and that each disk-write operation must write two logical CP/M sectors to the disk. The READ and WRITE routines keep track of which of these two logical sectors is currently being pointed to in the controller memory and determine what action to take to properly read or write the data on the disk. For example, in the case of a Write, because CP/M requests transfer of only 128 bytes of data to the disk, the BIOS must read the appropriate 256-byte record from the disk, place the 128-byte CP/M buffer over the proper half of this longer record, and then write the combined 256-byte record back out to the disk.

Additional Utilities

Two other software utilities must also be written prior to completing the installation of the hard-disk system. The first of these is a formatting utility that will format the disk and check for any bad tracks on the disk. The second is a system-generation utility that will write the new operating system beginning at track 0, sector 0 on the Winchester disk after the combined BIOS has been integrated into a CP/M system. This last process places a hardware-bootable system on the hard disk.

A new Winchester disk drive is delivered in an unformatted condition. This means that the disk has no information identifying the beginning and end of each sector. Once the HCA and other disk hardware is integrated into the system, you must create a routine to write this formatting information on the disk. A formatting routine is then used to issue a DCB to the controller, commanding it to format the disk. The controller takes care of the rest by writing the required formatting information onto the disk.

When the disk formatting is complete, the formatting routine should then read each sector on the disk to determine if the drive hardware was delivered with any bad sectors on the disk. For Winchester disk drives, it is not uncommon for a disk to have several bad sectors. The test program reads all sectors on the disk and saves the location of any bad sectors identified during the Read process. The routine then identifies these bad sectors to the controller, which constructs an alternate track assignment for each of the bad sectors on the disk. After this alternate assignment is complete, the existence of bad sectors on the disk is transparent to the system. The controller keeps the alternate-track data on the disk and, when a disk access is made to a bad sector, the controller automatically switches to the alternate track to read or write the data.

The second utility, the system-generation program, is used to write the new operating system to the Winchester drive or to read an existing operating system from the Winchester drive into memory. Installation procedures for a combined Winchester drive BIOS are the same as for any typical CP/M BIOS and will be covered in the next section. At the end of the configuration process, a system image will reside in memory starting at location 900 hexadecimal. The locations 900 to 980 hexadecimal contain the system loader, and the locations from 980 hexadecimal on contain the newly configured operating system. A WRITE routine must be designed to take the data from these

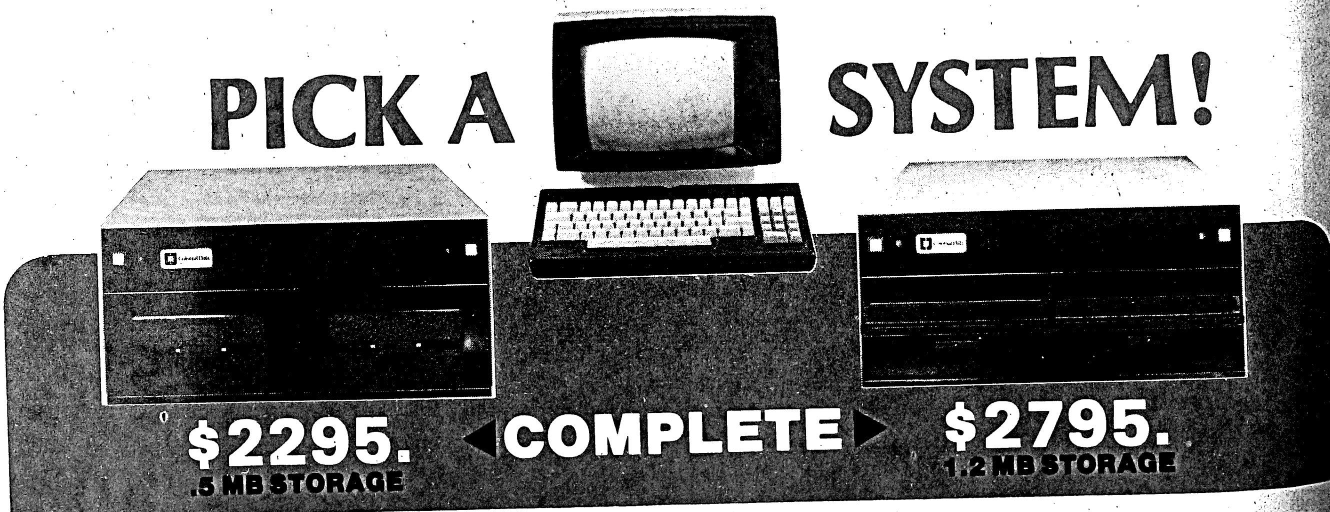
No matter how careful you are in building the system, something is bound to be wrong.

locations in memory and write them to the Winchester disk drive starting at track 0, sector 0. Similarly, a READ routine must be designed to take information from track 0 of the disk and place it into memory starting at location 900 hexadecimal.

Building a New BIOS

One other program must be written before a new BIOS can be built and installed in a CP/M system. This is the system loader that is used to initially read in the system from the disk. This program is written to run at location 80 hexadecimal and is restricted to 128 bytes in length. It is used to load the system during the bootstrap process that we will describe later. The program is designed to read the system starting at track 0, sector 1 of the Winchester disk and, when the load is complete, to transfer control to the proper entry point of CP/M to start the operating system running.

Once the system loader and the BIOS code are complete, the next step is to assemble both of these programs and remove any assembly or syntax errors. In the case of the BIOS, the symbolic variable MSIZE must also be defined before assembly to correspond to the size of the system being generated. Assuming that you have achieved an error-free assembly, the next step is to build a new CP/M system that contains the new system loader and BIOS and to write this



We're offering you our SB-80 system in either 5 1/4" or 8" disk drives, your choice. Either way your system comes with a full size (12" diagonal) non-glare tiltable green screen with 24 lines by 80 character format. Its multi-character set offers blinking cursor, underlining, reverse video, and half and zero intensity. The movable, detachable keyboard has a numeric pad with cursor control and function keys.

■ Single Board Technology ■ CP/M® Operating System
■ 4 MHz Z80A CPU ■ 64K 200ns Main Memory
■ 8-Inch Dual Density Floppy Drives
■ 5 1/4-Inch Dual Density Floppy Drives
■ 2-Serial Ports ■ 2-Parallel Ports
■ 4-Counter/Timers ■ Expandable

For further information about this limited offer call or write:

Colonial Data

Nationwide on-site and depot repair service through the professionals at INDESERV.

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

Colonial Data Services Corp., 105 Sanford Street, Hamden, Conn. 06514 • (203) 288-2524 • Telex: 95601

Circle 98 on inquiry card.

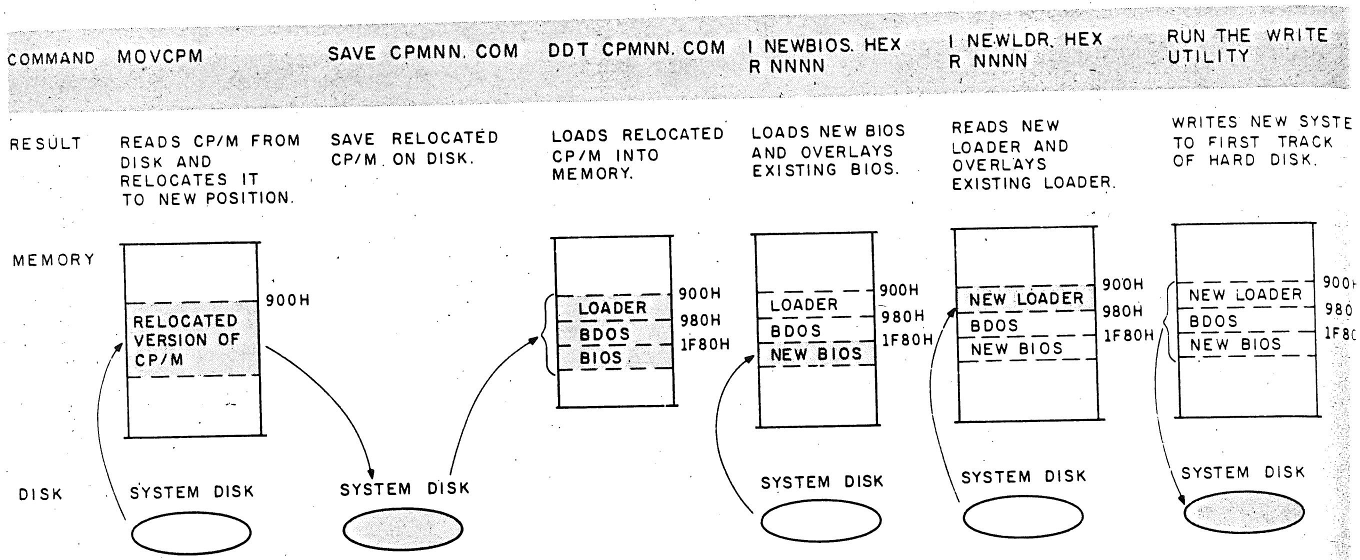
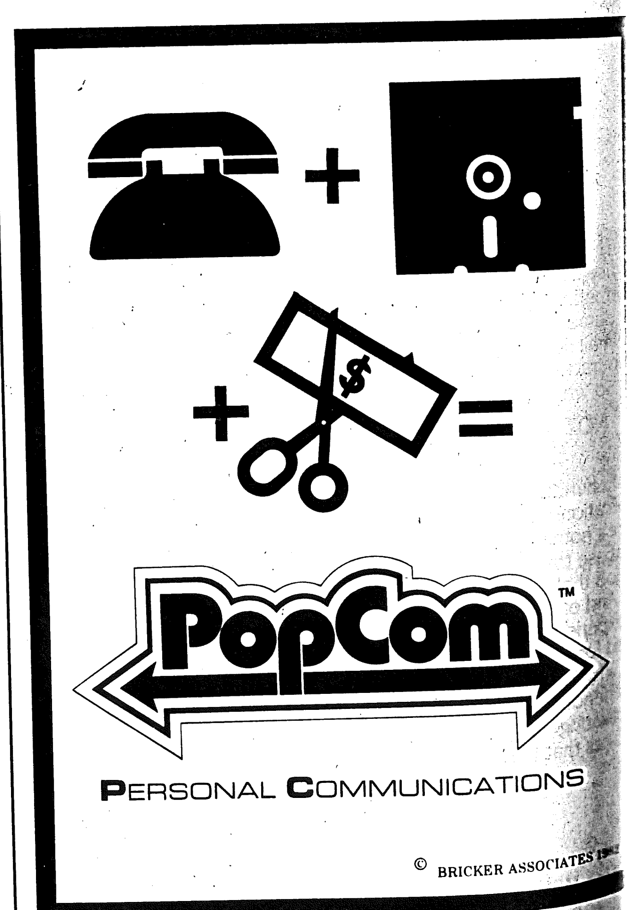


Figure 6: *Installing the new operating system. First, the MOVCPM instruction reads the current version of CP/M from the disk int memory at location 900 hexadecimal and relocates it to run at a location specified in the command. The SAVE instruction then place the relocated version of CP/M onto the disk under the filename CPMNN.COM. The DDT command reloads the relocated version c CP/M into memory at location 900 hexadecimal and transfers control to the DDT program. Next, the command I NEWBIOS.HE reads the new BIOS from the disk and overlays the existing BIOS. The command I NEWLDR.HEX then reads the new system load from the disk and overlays the existing loader. Finally, the WRITE utility program writes the properly configured operating system t the first track of the Winchester disk.*

new system to the Winchester disk drive starting at track 0, sector 0. In this process you should first use the MOVCPM utility (provided by CP/M) to create a new copy of CP/M that is properly sized for the system. Then, as instructed in the MOVCPM utility, this new copy of the operating system must be saved on the disk fc later retrieval by using CP/M's star dard SAVE utility. Once the ne copy of the system is saved, the ne

Circle 127 on inquiry card.



Circle 363 on inquiry c

step is to read the system back into memory using the DDT utility. If the system was saved under the name CPMNN.COM you can use the following CP/M command to initiate DDT and read the new copy of the system into memory:

DDT CPMNN.COM

After this command is executed, the new system will be located in memory starting at location 900 hexadecimal, and DDT will be running. The next step is to overlay the new system loader and new BIOS on top of the copy of CP/M in memory by using two DDT commands: I (insert) and R (read). These commands place the new system loader into memory at location 900 hexadecimal and place the BIOS at location 1F80 hexadecimal. Once these overlays are complete, the system is properly configured for the Winchester disk drive. Next we must get out of DDT and return to CP/M while leaving the newly created copy of the system in memory. First, issue a G0 command to DDT that transfers control to CP/M. Once you are back in CP/M, the final remaining task is to run the system-generation utility to load the newly constructed system, which now resides in memory starting at location 900 hexadecimal, onto the Winchester disk starting at track 0, sector 0. Figure 6 graphically summarizes these steps.

Building a New Bootstrap

At the completion of the process we just described, a copy of a properly configured CP/M system and system loader is on the first track of the Winchester disk. The one remaining task is to develop the software to bootstrap this system into memory and to configure the hardware so that this bootstrap is executed when a hardware boot command occurs. The actual bootstrap code is very simple. All it has to do is read the first 128 bytes from track 0, sector 0 on the Winchester disk drive into memory starting at location 80 hexadecimal and, when the read is complete, transfer control to the beginning of this code. This bootstrap code should be written to run at whatever location is going to be assigned to the PROM (programmable read-only memory) chip on the HCA, which we mentioned last month. When properly assembled and linked, the bootstrap code is burned into the HCA PROM and the software installation is essentially complete. The processor board is then "restrapped" so that the bootstrap address corresponds to the beginning of the PROM on the HCA.

Once this modification has been completed, a hardware boot command (i.e., pressing the RESET button) results in the sequence of events presented in Figure 7. At the boot command, the processor begins executing code at the beginning of the HCA PROM. This code reads the first 128 bytes of data from the first physical sector of the first track on the disk into memory starting at location 80 hexadecimal. When the read is complete, the code in the PROM

THE BENCHMARK®

A Word Processor Worthy of Comparison . . .

FEATURE	BENCHMARK 2.0	Wordstar 3.21	BENCHMARK 3.0
Ease of Operation	YES	NO	YES
Plain English Commands	YES	NO	YES
Use Cursor Keys	YES	NO	YES
Control Characters	NO	YES	NO
Function Keys	YES	NO	YES
Computer Aided Tutorial	YES	YES	YES
Descriptive Directory	YES	NO	YES
29 Character Title	YES	NO	YES
Author & Operator ID	YES	NO	YES
Document Size	YES	NO	YES
Creation Date	YES	NO	YES
Revision Time & Date	YES	NO	YES
Standard Editing Features	YES	YES	YES
Go To Any Page	YES	NO	YES
Interactive Printing	YES	NO	YES
Automatic Reformatting	YES	NO	YES
Automatic Repagination	YES	NO	YES
Headings and Footings	YES	YES	YES
Multi-line	YES	NO	YES
Keyboard Phrases	YES	NO	YES
Proportional Printing	NO	NO	YES
Business Graphics	NO	NO	YES
Paragraph Assembly	NO	NO	YES
Edit Marking	NO	NO	YES
Auto Widow/Orphan Protect	NO	NO	YES
Auto Footnoting	NO	NO	YES
Built In Calculator	NO	NO	YES
Price	$249	$495	$499

DEMAND A COMPARISON!

Call or Write:
Metasoft Corporation, 711 E. Cottonwood, Suite E, Casa Grande, AZ 85222 • (602) 961-0003

Circle 521 on inquiry card.

Circle 162 on inquiry card.

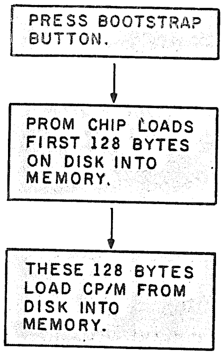


Figure 7: *The Bootstrap process. Pressing the hardware bootstrap (Reset) button causes the microprocessor to transfer control to the PROM chip on the host computer adapter. The code in the PROM reads the first 128 bytes of data from track 0 of the hard disk into memory starting at location 80 hexadecimal and then transfers control to that location. This code in turn loads the CP/M system off the first track of the Winchester disk. Then, when the load is complete, the code transfers control to location 0 hexadecimal to start CP/M running.*

transfers control to location 80 hexadecimal, which now contains the system loader. This code now reads the remainder of the CP/M system from the first track on the disk into memory and then transfers control to the CP/M system. The result is that a properly configured system, which includes the Winchester disk drive, is left running in memory, waiting to respond to any user commands.

System Debugging

No matter how careful you are in building the hardware and software we have just finished describing, it is a fact of life that when plugged into the system there will be something wrong. This is when the really interesting portion of the system integration process begins, namely, finding and correcting the inevitable bugs in the hardware and software. This debugging process can be broken into three separate areas. The first is debugging the HCA. The second is debugging and examining the HCA, disk controller, and disk system. The last is debugging the CP/M interface software. We will now describe some of the techniques that can be used in each of these areas.

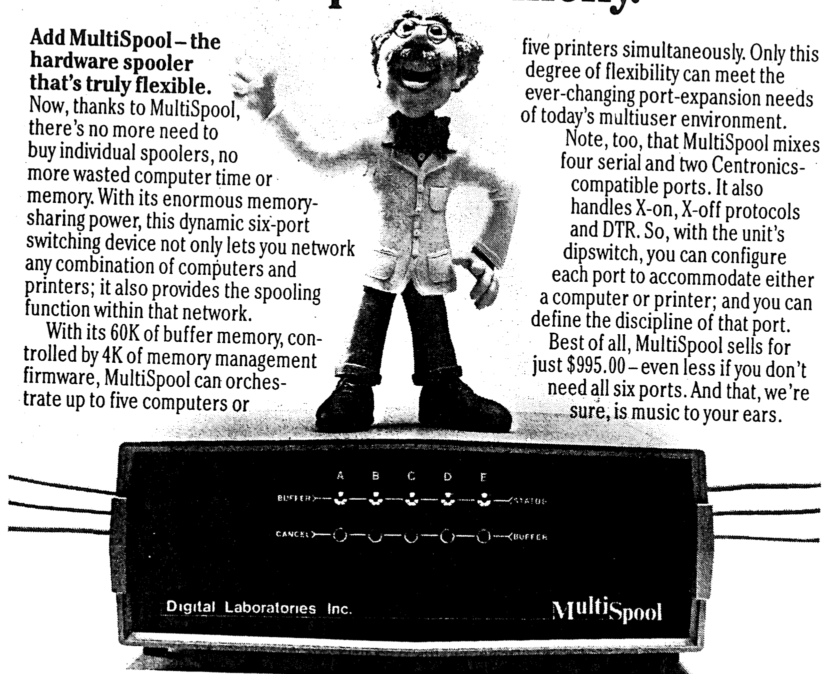
Make your spooling network sing in six-port harmony.

Add MultiSpool – the hardware spooler that's truly flexible. Now, thanks to MultiSpool, there's no more need to buy individual spoolers, no more wasted computer time or memory. With its enormous memory-sharing power, this dynamic six-port switching device not only lets you network any combination of computers and printers; it also provides the spooling function within that network.

With its 60K of buffer memory, controlled by 4K of memory management firmware, MultiSpool can orchestrate up to five computers or five printers simultaneously. Only this degree of flexibility can meet the ever-changing port-expansion needs of today's multiuser environment.

Note, too, that MultiSpool mixes four serial and two Centronics-compatible ports. It also handles X-on, X-off protocols and DTR. So, with the unit's dipswitch, you can configure each port to accommodate either a computer or printer; and you can define the discipline of that port.

Best of all, MultiSpool sells for just $995.00 – even less if you don't need all six ports. And that, we're sure, is music to your ears.



Digital Laboratories Inc.
600 Pleasant Street, Watertown, MA 02172. (617) 924-1680

BUY MONITORS DIRECT!!

$109.95 plus $4.75 shipping
Model TM-12PC-GX
12" P31 GRAPHICS
* 1200 line resolution
* 18.9 MHz band width
® Apple III is a registered trademark of Apple Computer.

$299.95 plus $9.50 shipping
Model TN-1440
13" RGB COLOR
* 360/230 lines resolution
* Up to 80 characters
* Apple III and IBM customers please add $29.50 for factory installed mod board and cable for 16 colors.

Monitors by: TECO
dataled
VISA, MasterCard & American Express accepted. No C.O.D.
Call Toll Free 1-800-258-6370
18 Bridge Street, Salem, NH 03079

The first thing to debug is the HCA card. Debugging this device involves making sure that the various combinations of possible input conditions result in the expected outputs to the disk controller. You can do this most efficiently by writing a set of driver routines in assembly language to write messages to the various output ports on the HCA. If you write these routines to loop continuously, you can use an oscilloscope to check the output registers on the HCA to ensure that the card is acting properly.

When you've verified proper operation of the HCA to the maximum extent possible, next connect the disk controller and the Winchester disk drive to the HCA. This simple connection begins the debugging process for the Miniscribe disk drive, which has extensive self-test features and continuously monitors its own operation to check for faults. If a fault is detected, the drive communicates with the user by flashing a Morse code letter (using the drive select light) to identify the particular fault that was detected. Once the disk is operating properly, the way to debug and test the rest of this total system is to pass various DCBs to the disk controller and see if the disk responds to these DCB inputs as expected. For example, you can test the disk by performing a controller self-test, formatting the disk, writing information to a particular sector, reading the information back, and testing for error conditions. This type of testing not only verifies proper disk operation but also provides you with valuable insight into the disk operation, which you can use during the remainder of the integration process.

Once the disk subsystem hardware is operating properly, the next step is to integrate the software. We described this integration process earlier. However, during the initial testing it is convenient to modify this procedure to allow use of some of the CP/M debugging tools to debug the interface software. To do this, you should create a "false" CP/M system that runs inside (i.e., at a lower memory address than) the current CP/M system running on the computer. This keeps the initial user system intact and allows you to run the new CP/M system under control of an existing debugged CP/M system, so you have access to the standard debugging tools from CP/M, such as DDT, to aid in debugging the new version of the BIOS.

Conclusion

We have now described the entire process of adding a Winchester disk subsystem to an S-100 computer system running CP/M. As we have tried to show, this is a rather substantial undertaking and should not be started lightly. The main advantage in performing such an integration project is the learning that takes place during the project. If your main goal is to obtain additional storage capability for a microcomputer, you could obtain this storage less expensively and certainly with less effort by buying a commercially available unit such as the one we manufacture. Because of quantity pricing and volume discounts, complete Winchester disk-drive subsystems are commercially available at a cost comparable to what a hobbyist would spend for the hardware portion of the system alone.

However, if you are interested in doing the work yourself and you have the necessary time and expertise, we encourage you to attempt this project and use this series of articles as a guide.■

The Winchester disk drive subsystem described in this series of articles is available as a completely assembled unit from ASC Associates of Lexington Park, Maryland. In addition to the S-100 version discussed, versions are also available for TRS-80 and Apple computers. The disk-drive systems for these computers use the same drive and controller hardware as the S-100 version but use a different host computer adapter and interface software. Until a nationwide dealer distribution network is established, these systems will be available by mail order for $1995. To order or obtain further information, write to ASC Associates Inc., POB 615, Lexington Park, MD 20653, or phone (301) 863-6784.

DOCUMATE/PLUS T.M.

Indexes *and* Sorts WordStar T.M.

DocuMate/Plus™ Features . . .
- 16 Level Indexing
- Table of Contents Generation
- Self-Contained Sort Feature
- "See" and "See Also" References
- Output Format Control
- Cross-Document Index Generation

DocuMate/Plus generates completely collated tables of contents and alphabetical multi-level indexes from WordStar™ text files of any length . . . as detailed as you desire.

DocuMate/Plus is refreshingly easy to use, and its internal sort feature can save you the cost of an expensive stand-alone sort program. Just type simple DocuMate commands right into your text file. DocuMate/Plus automatically finds and places all index references in a separate work file for sorting and collating.

DocuMate/Plus builds master indexes for many documents, such as research notes, books and articles, speeches, hardware and software documentation, and business correspondence.

DocuMate/Plus is Today's Most Flexible Text-Based Indexing System for Professional and Business Information Processing.

Try Documate/Plus at your local Osborne Computer Dealer, or call us for the name of your local authorized Orthocode textware dealer.

Documate/Plus is available on standard 8" CP/M and 5¼" Apple, Northstar, Osborne and Otrona diskettes.

- DocuMate/Plus Program with Manual $165.00
- Manual Only .... $ 25.00

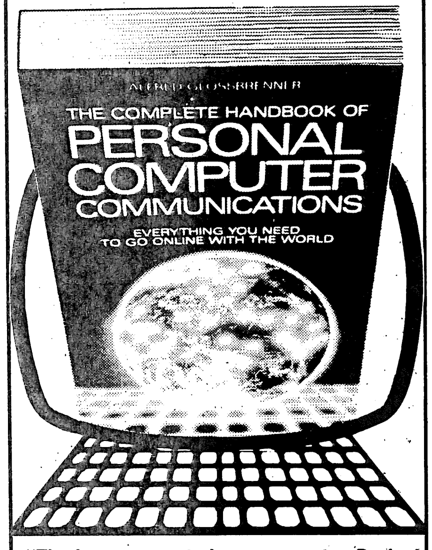
ORTHOCODE
The Textware Company

THE ORTHOCODE CORPORATION
P.O. Box 6191 • Albany, CA 94706
(415) 753-3222

• WordStar is a trademark of MicroPro International • CP/M is a trademark of Digital Research

Circle 343 on inquiry card.

Bring thousands of dollars worth of information home for only $14.95!



"The best reason to buy a computer. Packed with useful information—it's an absolute must for computer owners everywhere."
—Gary G. Reibsamen, Vice-President of NewsNet Inc.

Your passport to the electronic universe—how to turn any personal computer or word processor into a communication/information machine of incredible versatility and power, even without a knowledge of programming. Here's just a sampling of what you get—✱ **How to select a modem and communications software** ✱ How to send electronic letters, telexes and MailGrams™ ✱ **Money-saving techniques for using The Source, Dow Jones News/Retrieval, DIALOG, BRS and more** ✱ Get *free* software! Download thousands of programs ✱ **Get privileged information from $30,000 worth of newsletters** ✱ Get discounts of 10% to 40% by shopping on-line ✱ **Trouble-shooter's guide to problem-free communications** ✱ Hundreds of *free* computer bulletin boards (CBBS) ✱ **Access vast libraries (Library of Congress, *N.Y. Times*, magazines, and whole encyclopedias)** ✱ Get real-time computer games and tournaments ✱ **Electronic banking, investment management, and then barter** ✱ Get toll-free numbers, valuable short-cut commands ✱ **Get on-line advice from experienced users of your equipment.**

Plus much more! In just two days you'll see what this unique handbook can do for you—but we'll give you two weeks to decide. Send for your copy *now*; if you're not satisfied return it in two weeks for a full refund.

Main Selection of The Small Computer Book Club
Alternate Book-of-the-Month Club/Science Selection

Please return this coupon to your bookseller or direct to:
St. Martin's Press
175 Fifth Avenue, New York, N.Y. 10010 Attn: ML BY

Please send me______copy(ies) of *The Complete Handbook of Personal Computer Communications* @ $14.95 paperback each. Please add $1.50 per book for postage and handling. My check or money order is enclosed in the amount of $______. I may examine the book for two weeks and, if not 100% satisfied, I can return it for a complete refund.

Name______________________
Address______________________
City______________________
State______________Zip______________

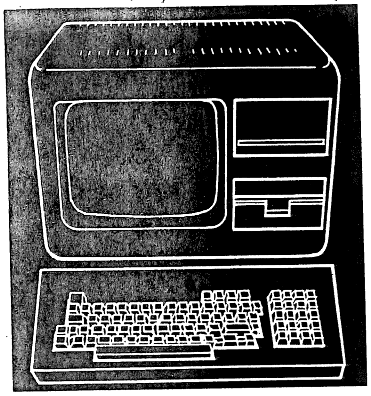
Listing 1: *A complete listing of the BIOS needed to integrate a Winchester disk drive into a CP/M-based microcomputer.*

```
CP/M MACRO ASSEM 2.0     #001     BIOS FOR A.S.C. ASSOC. HARD DISK SUBSYSTEM. 7/18/82

                     TITLE 'BIOS FOR A.S.C. ASSOC. HARD DISK SUBSYSTEM. 7/18/82'
                     ;
                     ;        COMBINED BIOS INDICATES WHERE THE USER IS TO UPDATE
                     ;        SPECIFIC SUBROUTINES IN ORDER TO MERGE HIS EXISTING
                     ;        CP/M BIOS WITH THE A.S.C. ASSOCIATES CUSTOM BIOS.
                     ;
                     ;               COPYRIGHT (C) A.S.C. ASSOCIATES 1982
                     ;
                     ;        THIS BIOS USES THE DIGITAL RESEARCH CBIOS, DISKDEF LIB, AND
                     ;        SECTOR BLOCKING AND DEBLOCKING ROUTINES SUPPLIED WITH THE
                     ;        STANDARD CP/M 2.2 SOFTWARE PACKAGE AND IS THEN COMBINED WITH
                     ;        THE A.S.C. ASSOCIATES HARD DISK SOFTWARE DRIVERS.
                     ;
                     ;        THROUGHOUT THIS CODE COMMENTS DELIMITED USING THE FOLLOWING
                     ;        FORMAT ARE TO BE CHANGED WHEN ADDING USER DISK ROUTINES.
                     ;
                     ;                        >>>>> COMMENT <<<<<
                     ;
                     ;
003A =               MSIZE    EQU     58       ;CP/M VERSION MEMORY SIZE IN KILOBYTES
                     ;
                     ;
9800 =               BIAS     EQU     (MSIZE-20)*1024
CC00 =               CCP      EQU     3400H+BIAS          ;BASE OC CCP
D406 =               BDOS     EQU     CCP+806H        ;BASE OF BDOS
E200 =               BIOS     EQU     CCP+1600H       ;BASE OF BIOS
0004 =               CDISK    EQU     0004H           ;CURRENT DISK NUMBER 0=A,...,15=P
0003 =               IOBYTE   EQU     0003H           ;INTEL I/O BYTE
0001 =               NUDISKS  EQU     1               ;NUMBER OF DISKS IN SYSTEM
                     ;
                     ;
E200                          ORG     BIOS            ;ORGIN OF THIS PROGRAM
                     ;
002C =               NSECTS   EQU     (BIOS-CCP)/128  ;WARM START SECTOR COUNT
                     ;
                     ;
                     ;        JUMP VECTOR FOR INDIVIDUAL SUBROUTINES
E200 C385E2                   JMP     BOOT            ;COLD START
E203 C394E2          WBOOTE:  JMP     WBOOT           ;WARM START
E206 C3F3E2                   JMP     CONST           ;CONSOLE STATUS
E209 C3F6E2                   JMP     CONIN           ;CONSOLE CHARACTER IN
E20C C3F9E2                   JMP     CONOUT          ;CONSOLE CHARACTER OUT
E20F C3FCE2                   JMP     LIST            ;LIST CHARACTER OUT
E212 C302E3                   JMP     PUNCH           ;PUNCH CHARACTER OUT
E215 C305E3                   JMP     READER          ;READER CHARACTER OUT
E218 C308E3                   JMP     HOME            ;MOVE HEAD TO HOME POSITION
E21B C30EE3                   JMP     SELDSK          ;SELECT DISK
E21E C32AE3                   JMP     SETTRK          ;SET TRACK NUMBER
E221 C332E3                   JMP     SETSEC          ;SET SECTOR NUMBER
E224 C339E3                   JMP     SETDMA          ;SET DMA ADDRESS
E227 C34BE3                   JMP     READ            ;READ DISK
E22A C366E3                   JMP     WRITE           ;WRITE DISK
E22D C3FFE2                   JMP     LISTST          ;RETURN LIST STATUS
E230 C341E3                   JMP     SECTRAN         ;SECTOR TRANSLATE
                     ;
                     ;DUMMY LIST OF JUMP VECTORS, USED TO TRANSFER CALLS TO USERS FLOPPY
                     ;DISK SYSTEM.  THIS TABLE IS OVERLAYED WITH THE USERS JUMP TABLE
                     ;
                     ;
                     ;       >>>>> REMOVE THIS SECTION OF DUMMY JUMP VECTORS. <<<<<
                     ;
                     ;
E233 0000C9          XBOOT:           NOP!NOP!RET     ;COLD START
E236 0000C9          XWBOOT:          NOP!NOP!RET     ;WARM START
E239 0000C9          XCONST:          NOP!NOP!RET     ;CONSOLE STATUS
E23C 0000C9          XCONIN:          NOP!NOP!RET     ;CONSOLE CHARACTER IN
E23F 0000C9          XCNOUT:          NOP!NOP!RET     ;CONSOLE CHARACTER OUT
E242 0000C9          XLIST:           NOP!NOP!RET     ;LIST CHARACTER OUT
E245 0000C9          XPUNCH:          NOP!NOP!RET     ;PUNCH CHARACTER OUT
E248 0000C9          XREADR:          NOP!NOP!RET     ;READER CHARACTER OUT
E24B 0000C9          XHOME:           NOP!NOP!RET     ;MOVE HEAD TO HOME POSITION
E24E 0000C9          XSLDSK:          NOP!NOP!RET     ;SELECT DISK
E251 0000C9          XSETRK:          NOP!NOP!RET     ;SET TRACK NUMBER
E254 0000C9          XSTSEC:          NOP!NOP!RET     ;SET SECTOR NUMBER
E257 0000C9          XSTDMA:          NOP!NOP!RET     ;SET DMA ADDRESS
E25A 0000C9          XREAD:           NOP!NOP!RET     ;READ DISK
E25D 0000C9          XWRITE:          NOP!NOP!RET     ;WRITE DISK
E260 0000C9          XLSTST:          NOP!NOP!RET     ;RETURN LIST STATUS
E263 0000C9          XSCTRN:          NOP!NOP!RET     ;SECTOR TRANSLATE
                     ;
                     ;
                              MACLIB  DISKDEF
                              DISKS   1
E266+=               DPBASE   EQU     $          ;BASE OF DISK PARAMETER BLOCKS
E266+00000000        DPE0:    DW      XLT0,0000H      ;TRANSLATE TABLE
E26A+00000000                 DW      0000H,0000H     ;SCRATCH AREA
E26E+B5E676E2                 DW      DIRBUF,DPB0     ;DIR BUFF,PARM BLOCK
E272+99E935E7                 DW      CSV0,ALV0       ;CHECK, ALLOC VECTORS
                              DISKDEF 0,0,63,,2048,4896,612,0,1
E276+=               DPB0     EQU     $               ;DISK PARM BLOCK
E276+4000                     DW      64              ;SEC PER TRACK
E278+04                       DB      4               ;BLOCK SHIFT
E279+0F                       DB      15              ;BLOCK MASK
E27A+00                       DB      0               ;EXTNT MASK
E27B+1F13                     DW      4895            ;DISK SIZE-1
E27D+6302                     DW      611             ;DIRECTORY MAX
E27F+FF                       DB      255             ;ALLOC0
E280+C0                       DB      192             ;ALLOC1
```

Listing 1 continued on page 388

Circle 136 on inquiry card.



LOW-COST 68000 MULTI-USER

You can have a powerful six-user WICAT computer with the acclaimed Motorola 68000 processor, hard disk and plenty of memory for only **$12,850.**

BUSINESS SOFTWARE AVAILABLE NOW

Integrated Accounting System
- General Ledger
- Accounts Receivable
- Accounts Payable
- Order Entry with Inventory
- Payroll

ULTRACALC Spread Sheet
Integral Database
MenuSystem

SERVICE

You can count on Concurrent *after* the sale as well as before. There's a telephone hotline for your convenience. On-site service is also available. And we are always ready to discuss your needs and particular applications.

Concurrent™ Corporation
1870 Madison Road Cincinnati, Ohio 45206
(513) 281-1270

HIGH PERFORMANCE MICRO COMPUTERS FOR SCIENCE AND INDUSTRY

Listing 1 continued:

```
E281+0000               DW      0                ;CHECK SIZE
E283+0100               DW      1                ;OFFSET
0000+=          XLT0    EQU     0                ;NO XLATE TABLE
                ;
                ;       >>>>>  ADD YOUR HAND CODED OR MACRO GENERATED  <<<<<
                ;       >>>>>  DISK PARAMETER TABLES HERE.             <<<<<
                ;
                ;
                ;
                ;       UTILITY MACRO TO COMPUTE SECTOR MASK
                        SMASK   MACRO   HBLK
                ;;      COMPUTE LOG2(HBLK), RETURN @X AS RESULT
                ;;      (2 ** @X = HBLK ON RETURN)
                @Y      SET     HBLK
                @X      SET     0
                ;;      COUNT RIGHT SHIFTS OF @Y UNTIL = 1
                        REPT    8
                        IF      @Y = 1
                        EXITM
                        ENDIF
                ;;      @Y IS NOT 1, SHIFT RIGHT ONE POSITION
                @Y      SET     @Y SHR 1
                @X      SET     @X + 1
                        ENDM
                        ENDM
                ;
                ;*****************************************************
                ;*                                                   *
                ;*         CP/M TO HOST DISK CONSTANTS               *
                ;*                                                   *
                ;*****************************************************
                ;
0800 =          BLKSIZ  EQU     2048             ;CP/M ALLOCATION SIZE
0100 =          HSTSIZ  EQU     256              ;HOST DISK SECTOR SIZE
0020 =          HSTSPT  EQU     32               ;HOST DISK SECTORS/TRK
0002 =          HSTBLK  EQU     HSTSIZ/128       ;CP/M SECTS/HOST BUFF
0040 =          CPMSPT  EQU     HSTBLK * HSTSPT  ;CP/M SECTORS/TRACK
0001 =          SECMSK  EQU     HSTBLK-1         ;SECTOR MASK
                        SMASK   HSTBLK           ;COMPUTE SECTOR MASK
0001 =          SECSHF  EQU     @X               ;LOG2(HSTBLK)
                ;
                ;*****************************************************
                ;*                                                   *
                ;*         BDOS CONSTANTS ON ENTRY TO WRITE          *
                ;*                                                   *
                ;*****************************************************
0000 =          WRALL   EQU     0                ;WRITE TO ALLOCATED
0001 =          WRDIR   EQU     1                ;WRITE TO DIRECTORY
0002 =          WRUAL   EQU     2                ;WRITE TO UNALLOCATED
                ;
                ;
                ;*****************************************************
                ;*                                                   *
                ;*      A.S.C. HOST INTERFACE TO XEBEC HARD          *
                ;*      DISK CONTROLLER CONSTANTS.                   *
                ;*                                                   *
                ;*****************************************************
                ;
00A0 =          CNTRL   EQU     0A0H             ;COMMAND CONTROL PORT
00A1 =          DATA    EQU     0A1H             ;DATA I/O PORT.
00A2 =          ASCPROM EQU     0A2H             ;BIT 5=1 PROM ON, BIT 5=0 PROM OFF
00A3 =          RSTPT   EQU     0A3H             ;CONTROLLER RESET PORT
000A =          WRTCMD  EQU     0AH              ;WRITE COMMAND
0008 =          READCMD EQU     08H              ;READ COMMAND
0003 =          ERRCMD  EQU     03H              ;RETREIVE ERROR COMMAND
0001 =          SELECT  EQU     01H              ;SELECT DISK COMMAND
0002 =          CMDAT   EQU     02H              ;COMMAND/DATA BIT
0004 =          BUSY    EQU     04H              ;CONTROLLER BUSY BIT
0010 =          REQ     EQU     010H             ;REQUEST COMMAND BIT
                ;
                ;
                ;       INDIVIDUAL SUBROUTINES TO PERFORM EACH FUNCTION
                BOOT:   ;SIMPLEST CASE IS TO JUST PERFORM PARAMETER INITIALIZATION
E285 AF                 XRA     A                ;ZERO IN THE ACCUM
E286 320400             STA     CDISK            ;SELECT DISK ZERO
E289 32A4E5             STA     HSTACT           ;HOST BUFFER INACTIVE.
E28C 32A6E5             STA     UNACNT           ;CLEAR UNALLOC COUNT.
E28F D3A2               OUT     ASCPROM          ;TURN OFF BOOT PROM
                ;
                ;       >>>>>  ADD ANY SPECIFIC SYSTEM HARDWARE INITALIZATION  <<<<<
                ;       >>>>>  CODE HERE (EG. USART INITIALIZATION CODE)       <<<<<
                ;
E291 C3D2E2             JMP     GOCPM            ;INITIALIZE AND GO TO CP/M
                ;
                ;
                ;
                WBOOT:  ;SIMPLEST CASE IS TO READ THE DISK UNTIL ALL SECTORS LOADED
E294 AF                 XRA     A                ;
E295 32A4E5             STA     HSTACT           ;HOST BUFFER INACTIVE.
E298 32A6E5             STA     UNACNT           ;CLEAR UNALLOC COUNT.
E29B 318000             LXI     SP,80H           ;USE SPACE BELOW BUFFER FOR STACK
E29E 0E00               MVI     C,0              ;SELECT DISK 0
E2A0 CD0EE3             CALL    SELDSK
E2A3 CD08E3             CALL    HOME             ;GO TO TRACK 00
                ;
E2A6 062C               MVI     B,NSECTS         ;B COUNTS # OF SECTORS TO LOAD
E2A8 0E00               MVI     C,0              ;C HAS THE CURRENT TRACK NUMBER
E2AA 1601               MVI     D,1              ;D HAS THE NEXT SECTOR TO READ
                ; NOTE THAT WE BEGIN BY READING TRACK 0, SECTOR 1 SINCE SECTOR 0
                ; CONTAINS THE COLD START LOADER, WHICH IS SKIPPED IN A WARM START
E2AC 2100CC             LXI     H,CCP            ;BASE OF CP/M (INITIAL LOAD POINT)
                LOAD1:
                        ;LOAD ONE MORE SECTOR
E2AF C5                 PUSH    B        ;SAVE SECTOR COUNT, CURRENT TRACK
E2B0 D5                 PUSH    D        ;SAVE NEXT SECTOR TO READ
```

Listing 1 continued on page 39

Circle 161 on inquiry card.

!!!!FANTASTIC PRICES!!!! FROM DIGITAL DIMENSIONS

OKIDATA
ML-80..80 COL., 80 CPS $317
ML-82A..80 COL., 120 CPS $395
*ML-83A..136 COL., 120 CPS $639
*ML-84..PARALLEL, 136 COL., 200 CPS $969
*ML-84..SERIAL, 136 COL., 200 CPS $1,083
**ML-92..80 COL., 160 CPS $509
**ML-92..SERIAL, 80 COL., 160 CPS .... CALL
*ML-93..136 COL., 160 CPS $832
*ML-93..SERIAL, 136 COL., 160 CPS .... CALL
PACEMARK 2410..350 CPS .... CALL
PACEMARK 2350..SERIAL, 350 CPS .... CALL
*Includes TRACTOR FEED
**Inc., OKIGRAPH™ DOT ADDRESSABLE GRAPHICS

IDS
PRISM 80..3.4K & 200 Sprint $1,036
Inc. sheet feed, color & graphics $1,429
PRISM 132..3.4K & 200 Sprint $1,195
Inc. sheet feed, color & graphics $1,591
MICROPRISM 480 .... $569

STAR MICRONICS
GEMINI-10..100 CPS, 2.3K BUFFER $409
GEMINI-15 .... $519

C. ITOH
Prowriter..80 Col., 120 CPS, Parallel $395
Prowriter 2..80 Col., 120 CPS, Parallel $570
Starwriter F-10..40 CPS $1,425
Printmaster F-10..55 CPS $1,595

SMITH CORONA
TP-I..12 CPS $649

MODEMS
Hayes 300 Baud Smart Modem $219
Hayes 1200 Baud Smart Modem $515

BUFFERS
Quadram Microfazer
16K Parallel to Parallel $169
32K Parallel to Parallel $199

MONITORS
AMDEK Color I $319
AMDEK Color II $719
NEC JB1201 Green $169

INTERFACES
Practical Peripherals
Grappler Plus $149
Buffer Board 64K $135

FOR THE IBM P.C.
NEC Spinwriter 3550..33CPS $2,045
Quadram Quadboard w/64k $469
Quadram Quadboard w/128K $519
Quadram Quadboard w/192k $609
Quadram Quadboard w/256k $699
Monte Carlo Card 64K $429

E-Z COLOR BOARD
For the Apple II or Apple II Plus.
Includes demo software and
E-Z COLOR Editor $199
S-100 Systems $279
TRS-80 .... $239

INTEX TALKER text-to-speech synthesizer.
Serial and parallel interface included $280

DIGITAL DIMENSIONS
190 Chapel Rd., Manchester, CT 06040
Info & Orders Call 203-649-3611
Orders Only 1-800-243-5222

C.O.D. Welcome. Allow 2-3 Weeks For Checks. MC/VISA OK.
All Prices Include UPS Ground Freight In U.S.
CT Residents Add 7½% Sales Tax.
Prices Subject To Change Without Notice.

Listing 1 continued:

```
E2C0 C294E2             JNZ     WBOOT    ;RETRY THE ENTIRE BOOT IF AN ERROR OCCURS
                ;
                ;       NO ERROR, MOVE TO NEXT SECTOR
E2C3 E1                 POP     H        ;RECALL DMA ADDRESS
E2C4 118000             LXI     D,128    ;DMA=DMA+128
E2C7 19                 DAD     D        ;NEW DMA ADDRESS IS IN H,L
E2C8 D1                 POP     D        ;RECALL SECTOR ADDRESS
E2C9 C1                 POP     B        ;NUMBER OF SECTORS REMAINING, CURRENT TRK
E2CA 05                 DCR     B        ;SECTORS=SECTORS-1
E2CB CAD2E2             JZ      GOCPM    ;TRANSFER TO CP/M IF ALL HAVE BEEN LOADED
                ;
E2CE 14                 INR     D
E2CF C3AFE2             JMP     LOAD1    ;LOAD MORE BDOS CODE
                ;
                GOCPM:
                ;       END OF LOAD OPERATION, SET PARAMETERS AND GO TO CP/M
E2D2 3EC3               MVI     A,0C3H   ;C3 IS A JMP INSTRUCTION
E2D4 320000             STA     0        ;FOR JMP TO WBOOT
E2D7 2103E2             LXI     H,WBOOTE         ;WBOOT ENTRY POINT
E2DA 220100             SHLD    1        ;SET ADDRESS FIELD FOR JMP AT 0
                ;
E2DD 320500             STA     5        ;FOR JMP TO BDOS
E2E0 2106D4             LXI     H,BDOS   ;BDOS ENTRY POINT
E2E3 220600             SHLD    6        ;ADDRESS FIELD OF JUMP AT 5 TO BDOS
                ;
E2E6 018000             LXI     B,80H    ;DEFAULT DMA ADDRESS IS 80H
E2E9 CD39E3             CALL    SETDMA
                ;
E2EC 3A0400             LDA     CDISK    ;GET CURRENT DISK NUMBER
E2EF 4F                 MOV     C,A      ;SEND TO THE CCP
E2F0 C300CC             JMP     CCP      ;GO TO CP/M FOR FURTHER PROCESSING
                ;
                ;
                ;       >>>>>  THE FOLLOWING SIMPLE I/O ROUTINES MUST BE  <<<<<
                ;       >>>>>  FILLED WITH THE USERS OWN SYSTEM ROUTINES  <<<<<
                ;
                CONST:  ;CONSOLE STATUS, RETURN 0FFH IF CHARACTER READY, 00H IF NOT
E2F3 C339E2             JMP     XCONST   ;SENT TO USER JMP VECTOR
                ;
                ;
                CONIN:  ;CONSOLE CHARACTER INTO REGISTER A
E2F6 C33CE2             JMP     XCONIN   ;SEND TO USER JMP VECTOR
                ;
                ;
                CONOUT: ;CONSOLE CHARACTER OUTPUT FROM REGISTER C
E2F9 C33FE2             JMP     XCNOUT   ;SEND TO USER JMP VECTOR
                ;
                ;
                LIST:   ;LIST CHARACTER FROM REGISTER C
E2FC C342E2             JMP     XLIST    ;SEND TO USER JMP VECTOR
                ;
                LISTST: ;RETURN LIST STATUS (0 IF NOT READY, 1 IF READY)
E2FF C360E2             JMP     XLSTST   ;SEND TO USER JMP VECTOR
                ;
                PUNCH:  ;PUNCH CHARACTER FROM REGISTER C
E302 C345E2             JMP     XPUNCH   ;SEND TO USER JMP VECTOR
                ;
                ;
                READER: ;READ CHARACTER INTO REGISTER A FROM READER DEVICE
E305 C348E2             JMP     XREADR   ;SEND TO USER JMP VECTOR
                ;
                ;
                ;       I/O DRIVERS FOR THE DISK FOLLOW
                ;       FOR NOW, WE WILL SIMPLY STORE THE PARAMETERS AWAY FOR USE
                ;       IN THE READ AND WRITE SUBROUTINES
                ;
                HOME:   ;MOVE TO THE TRACK 00 POSITION OF CURRENT DRIVE
                ;       TRANSLATE THIS CALL INTO A SETTRK CALL WITH PARAMETER 00
E308 010000             LXI     B,0      ;SELECT TRACK 0
E30B C32AE3             JMP     SETTRK   ;WE WILL MOVE TO 00 ON FIRST READ/WRITE
                ;
                ;
                SELDSK:
                        ;SELECT DISK
E30E 79                 MOV     A,C                     ;SELECTED DISK NUMBER
E30F 329BE5             STA     SEKDSK                  ;SEEK DISK NUMBER
E312 0D                 DCR     C                       ;LOAD ALTERNATE DRIVE
E313 B7                 ORA     A                       ;CHECK FOR HARD DISK
E314 C24EE2             JNZ     XSLDSK                  ;GO SELECT ALTERNATE DRIVE
E317 0C                 INR     C                       ;RESTORE HARD DISK NUMBER
E318 210000             LXI     H,0000                  ;LOAD ERROR CODE
E31B FE01               CPI     NUDISKS                 ;CHECK AGAINST MAX # DISKS
E31D D0                 RNC                             ;NO CARRY IF GREATER
E31E 6F                 MOV     L,A                     ;DISK NUMBER TO HL
E31F 2600               MVI     H,0
                        REPT    4                       ;MULTIPLY BY 16
                        DAD     H
                        ENDM
E321+29                 DAD     H
E322+29                 DAD     H
E323+29                 DAD     H
E324+29                 DAD     H
E325 1166E2             LXI     D,DPBASE                ;BASE OF PARM BLOCK
E328 19                 DAD     D                       ;HL=.DPB(CURDSK)
E329 C9                 RET
                ;
                SETTRK:
                        ;SET TRACK GIVEN BY REGISTERS BC
E32A 60                 MOV     H,B
E32B 69                 MOV     L,C
E32C 229CE5             SHLD    SEKTRK                  ;TRACK TO SEEK
                ;
                ;       >>>>>  ENTER YOUR SYSTEM SET TRACK ROUTINE HERE  <<<<<
E2B1 E5                 PUSH    H        ;SAVE DMA ADDRESS
E2B2 4A                 MOV     C,D      ;GET SECTOR ADDRESS TO REGISTER C
E2B3 CD32E3             CALL    SETSEC   ;SET SECTOR ADDRESS FROM REGISTER C
```

Listing 1 continued on page 392

Circle 346 on inquiry card.

HOT PRICES!

COMPUTERS

K-Pro w/software $1649
Superbrain Jr, S/S, D/D $1875
Superbrain QD, D/S, D/D $2250
Franklin Ace 1000 .... $1049
NEC A.P.C. .... Call

DISK DRIVES

Tandon 100-2 (IBM) $269
Apple Compatable w/cabinet ... $239
Corona 10MB/IBM HD $1799
Corona 5MB HD $1719

MONITORS

NEC JB1201 Hi Res Green $169
BMC Green 12" $85
Princeton Graphics R6B $629
Taxan I R6B $315
Amdek Amber $159

PRINTERS

Gemini 10 100cps $359
Gemini 15 100cps $519
Okidata 92, 160cps $529
NEC 7710, 7730 .... $2190
IDS Prism 132 .... $1179
Daisywriter $1099
Brother HR-1 .... $829
Other Brands .... Call

EXPANSION PRODUCTS

AST Boards 64k, Ser, Par, clock $389
Mountain Computer .... Call
Practical Peripherals .... Call
Micro Soft .... Call

MODEMS

Hayes Smartmodem $219
Hayes 1200 baud modem $529

ORDERS ONLY 1-800-531-3133
INFORMATION: (206) 641-7233
Mail & Phone Orders Only.
Prices reflect 3% cash discount.
Sorry, no COD's.

PACIFIC COMPUTERS
13256 Northup Way #7
Bellevue, WA 98005

Listing 1 continued:

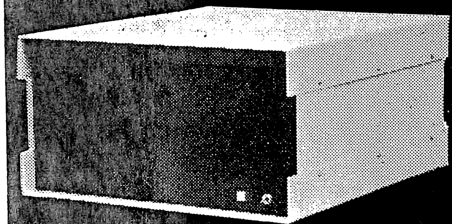
```
E2B6 C1                 POP     B          ;RECALL DMA ADDRESS TO B,C
E2B7 C5                 PUSH    B          ;REPLACE ON STACK FOR LATER RECALL
E2B8 CD39E3             CALL    SETDMA     ;SET DMA ADDRESS FROM B,C
                ;
                ;       DRIVE SET TO 0, TRACK SET, SECTOR SET, DMA ADDRESS SET
E2BB CD4BE3             CALL    READ
E2BE FE00               CPI     00H        ;ANY ERRORS?
                ;
                ;MAYBE THIS CALL IS FOR ANOTHER DRIVE
                ;BETTER GO SET THE TRACK NUMBER
E32F C351E2             JMP     XSETRK
                ;
                SETSEC:
                        ;SET SECTOR GIVEN BY REGISTER C
E332 79                 MOV     A,C
E333 329EE5             STA     SEKSEC              ;SECTOR TO SEEK
                ;
                ;       >>>>>  PLACE YOUR SYSTEM SET SECTOR ROUTINE HERE.  <<<<<
                ;
                ;MAYBE THIS CALL IS FOR ANOTHER DRIVE
                ;BETTER GO SET THE SECTOR NUMBER
E336 C354E2             JMP     XSTSEC
                ;
                SETDMA:
                        ;SET DMA ADDRESS GIVEN BY BC
E339 60                 MOV     H,B
E33A 69                 MOV     L,C
E33B 22B3E5             SHLD    DMAADR              ;LOAD DMA ADDRESS
                ;
                ;       >>>>>  PLACE YOUR SYSTEM SET DMA ROUTINE HERE.  <<<<<
                ;
                ;MAYBE THIS CALL IS FOR ANOTHER DRIVE
                ;BETTER GO SET THE DMA ADDRESS
E33E C357E2             JMP     XSTDMA
                ;
                SECTRAN:
                        ;TRANSLATE SECTOR NUMBER BC
E341 3A9BE5             LDA     SEKDSK              ;GET DISK NUMBER
E344 B7                 ORA     A                   ;CHECK FOR HARD DISK
                ;
                ;       >>>>> CHANGE THE FOLLOWING JNZ XSCTRN CODE TO  JUMP TO  <<<<<
                ;       >>>>> YOUR SYSTEM SECTOR TRANSLATION ROUTINE.           <<<<<
                ;
E345 C263E2             JNZ     XSCTRN              ;TRANSLATE OTHER SECTOR NUMBER
E348 60                 MOV     H,B
E349 69                 MOV     L,C
E34A C9                 RET
                ;
                ;*****************************************************
                ;*                                                   *
                ;*      THE READ ENTRY POINT TAKES THE PLACE OF      *
                ;*      THE PREVIOUS BIOS DEFINTION FOR READ.        *
                ;*                                                   *
                ;*****************************************************
                READ:
                        ;READ THE SELECTED CP/M SECTOR
E34B 3A9BE5             LDA     SEKDSK              ;GET DISK NUMBER
E34E B7                 ORA     A                   ;CHECK FOR HARD DISK A:
                ;
                ;       >>>>>  CHANGE THE FOLLOWING JNZ XREAD TO JUMP TO YOUR  <<<<<
                ;       >>>>>  SYSTEM READ ROUTINE.                            <<<<<
                ;
E34F C25AE2             JNZ     XREAD               ;READ ALTERNATE DISK
E352 AF                 XRA     A
E353 32A6E5             STA     UNACNT
E356 3E01               MVI     A,1
E358 32B1E5             STA     READOP              ;READ OPERATION
E35B 32B0E5             STA     RSFLAG              ;MUST READ DATA
E35E 3E02               MVI     A,WRUAL
E360 32B2E5             STA     WRTYPE              ;TREAT AS UNALLOC
E363 C3D8E3             JMP     RWOPER              ;TO PERFORM THE READ
                ;
                ;*****************************************************
                ;*                                                   *
                ;*      THE WRITE ENTRY POINT TAKES THE PLACE OF     *
                ;*      THE PREVIOUS BIOS DEFINTION FOR WRITE.       *
                ;*                                                   *
                ;*****************************************************
                WRITE:
                        ;WRITE THE SELECTED CP/M SECTOR
E366 3A9BE5             LDA     SEKDSK              ;GET DISK NUMBER
E369 B7                 ORA     A                   ;CHECK FOR HARD DISK A:
                ;
                ;       >>>>>  CHANGE THE FOLLOWING JNZ XWRITE CODE TO JUMP TO  <<<
                ;       >>>>>  YOUR SYSTEM WRITE ROUTINE.                       <<<
                ;
E36A C25DE2             JNZ     XWRITE              ;WRITE ALTERNATE DISK
E36D AF                 XRA     A                   ;0 TO ACCUMULATOR
E36E 32B1E5             STA     READOP              ;NOT A READ OPERATION
E371 79                 MOV     A,C                 ;WRITE TYPE IN C
E372 32B2E5             STA     WRTYPE
E375 FE02               CPI     WRUAL               ;WRITE UNALLOCATED?
E377 C291E3             JNZ     CHKUNA              ;CHECK FOR UNALLOC
                ;
                ;       WRITE TO UNALLOCATED, SET PARAMETERS
E37A 3E10               MVI     A,BLKSIZ/128        ;NEXT UNALLOC RECS
E37C 32A6E5             STA     UNACNT
E37F 3A9BE5             LDA     SEKDSK              ;DISK TO SEEK
E382 32A7E5             STA     UNADSK              ;UNADSK = SEKDSK
E385 2A9CE5             LHLD    SEKTRK
E388 22A8E5             SHLD    UNATRK              ;UNATRK = SECTRK
E38B 3A9EE5             LDA     SEKSEC
E38E 32AAE5             STA     UNASEC              ;UNASEC = SEKSEC
                ;
                CHKUNA:
```

Listing 1 continued on page 39

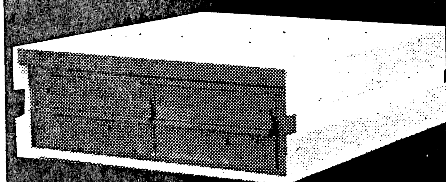
Circle 231 on inquiry card.

Main/Frames
Main/Frames
from
$200

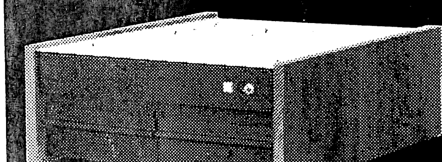
- 30 Models of Enclosures
- Assembled and tested
- Quasi-Coax Motherboards
- Power Supply
- Card cage and guides
- Fan, line, card, fuse, power & reset switches



8" Floppy Main/Frame
$482



8" Disc Enclosure
$250



$525
Phase/80 8" Floppy Mainframe

$900
Phase/80 Desk + Mainframe

Write or call for our brochure which includes our application note: "Building Computers — A Recipe"

INTEGRAND
8620 Roosevelt Ave.•Visalia,CA 93291
209/651-1203
We accept BankAmericard/Visa and MasterCharge

Listing 1 continued:

```
                        ;CHECK FOR WRITE TO UNALLOCATED SECTOR
E391 3AA6E5             LDA     UNACNT          ;ANY UNALLOC REMAIN?
E394 B7                 ORA     A
E395 CAD0E3             JZ      ALLOC           ;SKIP IF NOT
                ;
                ;       MORE UNALLOCATED RECORDS REMAIN
E398 3D                 DCR     A               ;UNACNT = UNACNT-1
E399 32A6E5             STA     UNACNT
E39C 3A9BE5             LDA     SEKDSK          ;SAME DISK?
E39F 21A7E5             LXI     H,UNADSK
E3A2 BE                 CMP     M               ;SEKDSK = UNADSK?
E3A3 C2D0E3             JNZ     ALLOC           ;SKIP IF NOT
                ;
                ;       DISKS ARE THE SAME
E3A6 21A8E5             LXI     H,UNATRK
E3A9 CD73E4             CALL    SEKTRKCMP       ;SEKTRK = UNATRK?
E3AC C2D0E3             JNZ     ALLOC           ;SKIP IF NOT
                ;
                ;       TRACKS ARE THE SAME
E3AF 3A9EE5             LDA     SEKSEC          ;SAME SECTOR?
E3B2 21AAE5             LXI     H,UNASEC
E3B5 BE                 CMP     M               ;SEKSEC = UNASEC?
E3B6 C2D0E3             JNZ     ALLOC           ;SKIP IF NOT
                ;
                ;       MATCH, MOVE TO NEXT SECTOR FOR FUTURE REF
E3B9 34                 INR     M               ;UNASEC = UNASEC+1
E3BA 7E                 MOV     A,M             ;END OF TRACK?
E3BB FE40               CPI     CPMSPT          ;COUNT CP/M SECTORS
E3BD DAC9E3             JC      NOOVF           ;SKIP IF NO OVERFLOW
                ;
                ;       OVERFLOW TO NEXT TRACK
E3C0 3600               MVI     M,0             ;UNASEC = 0
E3C2 2AA8E5             LHLD    UNATRK
E3C5 23                 INX     H
E3C6 22A8E5             SHLD    UNATRK          ;UNATRK = UNATRK+1
                ;
                NOOVF:
                        ;MATCH FOUND, MARK AS UNNECESSARY READ
E3C9 AF                 XRA     A               ;0 TO ACCUMULATOR
E3CA 32B0E5             STA     RSFLAG          ;RSFLAG = 0
E3CD C3D8E3             JMP     RWOPER          ;TO PERFORM THE WRITE
                ;
                ALLOC:
                        ;NOT AN UNALLOCATED RECORD, REQUIRES PRE-READ
E3D0 AF                 XRA     A               ;0 TO ACCUM
E3D1 32A6E5             STA     UNACNT          ;UNACNT = 0
E3D4 3C                 INR     A               ;1 TO ACCUM
E3D5 32B0E5             STA     RSFLAG          ;RSFLAG = 1
                ;
                ;*****************************************************
                ;*                                                   *
                ;*      COMMON CODE FOR READ AND WRITE FOLLOWS       *
                ;*                                                   *
                ;*****************************************************
                RWOPER:
                        ;ENTER HERE TO PERFORM THE READ/WRITE
E3D8 AF                 XRA     A               ;ZERO TO ACCUM
E3D9 32AFE5             STA     ERFLAG          ;NO ERRORS (YET)
E3DC 3A9EE5             LDA     SEKSEC          ;COMPUTE HOST SECTOR
                        REPT    SECSHF
                        ORA     A               ;CARRY = 0
                        RAR                     ;SHIFT RIGHT
                        ENDM
E3DF+B7                 ORA     A               ;CARRY = 0
E3E0+1F                 RAR                     ;SHIFT RIGHT
E3E1 32A3E5             STA     SEKHST          ;HOST SECTOR TO SEEK
                ;
                ;       ACTIVE HOST SECTOR?
E3E4 21A4E5             LXI     H,HSTACT        ;HOST ACTIVE FLAG
E3E7 7E                 MOV     A,M
E3E8 3601               MVI     M,1             ;ALWAYS BECOMES 1
E3EA B7                 ORA     A               ;WAS IT ALREADY?
E3EB CA12E4             JZ      FILHST          ;FILL HOST IF NOT
                ;
                ;       HOST BUFFER ACTIVE, SAME AS SEEK BUFFER?
E3EE 3A9BE5             LDA     SEKDSK
E3F1 219FE5             LXI     H,HSTDSK        ;SAME DISK?
E3F4 BE                 CMP     M               ;SEKDSK = HSTDSK?
E3F5 C20BE4             JNZ     NOMATCH
                ;
                ;       SAME DISK, SAME TRACK?
E3F8 21A0E5             LXI     H,HSTTRK
E3FB CD73E4             CALL    SEKTRKCMP       ;SEKTRK = HSTTRK?
E3FE C20BE4             JNZ     NOMATCH
                ;
                ;       SAME DISK, SAME TRACK, SAME BUFFER?
E401 3AA3E5             LDA     SEKHST
E404 21A2E5             LXI     H,HSTSEC        ;SEKHST = HSTSEC?
E407 BE                 CMP     M
E408 CA2FE4             JZ      MATCH           ;SKIP IF MATCH
                ;
                NOMATCH:
                        ;PROPER DISK, BUT NOT CORRECT SECTOR
E40B 3AA5E5             LDA     HSTWRT          ;HOST WRITTEN?
E40E B7                 ORA     A
E40F C47FE4             CNZ     WRITEHST        ;CLEAR HOST BUFF
                ;
                FILHST:
                        ;MAY HAVE TO FILL THE HOST BUFFER
E412 3A9BE5             LDA     SEKDSK
E415 329FE5             STA     HSTDSK
E418 2A9CE5             LHLD    SEKTRK
E41B 22A0E5             SHLD    HSTTRK
E41E 3AA3E5             LDA     SEKHST
E421 32A2E5             STA     HSTSEC
```

Listing 1 continued on page 39

NO LIMIT **Epson Giveaway** NO LIMIT

Epson MX70/80 Cartridges

$5.00 EACH

Min. 3 of same color

Reloads $2.50 each Min. 12

$30.00 a Doz. of Same Color

Cartridges and Reloads Available In Black, Red, Green, Blue, Brown

B.T. Enterprises
Dept. 15C
10B Carlough Rd.
Bohemia, N.Y. 11716

Order Toll Free 800-645-1165 NYS Orders Call 516-567-8155

Prices Subject to Change

N.Y.S. Residents Add Tax. Add $2.00 Shipping & Handling

B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

Circle 74 on inquiry card.

ELIZA IS HERE!

AT LAST! A FULL IMPLEMENTATION of the original ELIZA program is now available to run on your microcomputer!

Created at MIT in 1966, ELIZA has become the world's most celebrated artificial intelligence demonstration program. ELIZA is a non-directive psychotherapist who analyzes each statement as you type it in and then responds with her own comment or question — and her remarks are often startlingly appropriate!

Designed to run on a large mainframe, ELIZA has hitherto been unavailable to personal computer users except in greatly stripped down versions lacking the sophistication which made the original program so fascinating.

Now, our new microcomputer version possessing the **FULL** power and range of expression of the original is being offered at the introductory price of only $25. And if you want to find out how she does it (or teach her to do more) we will include the complete **Source Program** for only $20 additional.

Order your copy of ELIZA today and you'll never again wonder how to respond when you hear someone say "Okay, let's see what this computer of yours can actually do!"

ELIZA IS AVAILABLE IN THE FOLLOWING DISK FORMATS:

1. Standard 8 inch single density for all CP/M based computers $25 for ELIZA.COM - add $20 for Microsoft BASIC-80 Source
2. 5¼ inch CP/M for Apple II equipped with Z-80 SoftCard $25 for ELIZA.COM - add $20 for Microsoft BASIC-80 Source
3. 5¼ inch for 48K Apple II with Applesoft ROM and DOS 3.3 $25 for Protected File - add $20 for Applesoft Source
4. 5¼ inch for 64K IBM Personal Computer $25 for Protected File - add $20 for IBM Disk BASIC Source
5. 5¼ inch for Osborne I Microcomputer $25 for Protected File - add $20 for Microsoft BASIC-80 Source

ARTIFICIAL INTELLIGENCE RESEARCH GROUP
921 NORTH LA JOLLA AVENUE
LOS ANGELES, CALIFORNIA 90046
(213) 656-7368 (213) 654-2214
MC, VISA and CHECKS ACCEPTED

Circle 46 on inquiry card.

5¼" Floppy Disk Drives

(Direct IBM® Plug-in)

TANDON Model TM 100-1 $189* ea.
TANDON Model TM 100-2 $249* ea.
12" Green Phospher Zenith Monitor $109* ea.

IBM EXPANSION BOARDS

64K Memory .... $275* ea.
128K Memory .... $375* ea.
192K Memory .... $450* ea.
256K Memory .... $510* ea.
Floppy Controller........ $180* ea.
16K System BD Memory . $ 25* ea.

MC / VISA or C.O.D. with certified check or money order. Arkansas residents add sales tax.

For information or orders, call
501-777-9859

G-H Computer Systems, Inc.
P.O. Box 871 • Hope, AR 71801

* Plus Shipping.

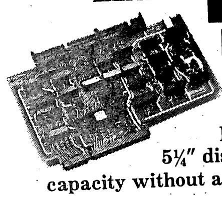
Circle 203 on inquiry card.

Listing 1 continued:

```
E424 3AB0E5           LDA     RSFLAG          ;NEED TO READ?
E427 B7               ORA     A
E428 C487E4           CNZ     READHST         ;YES, IF 1
E42B AF               XRA     A               ;0 TO ACCUM
E42C 32A5E5           STA     HSTWRT          ;NO PENDING WRITE
                ;
                MATCH:
                      ;COPY DATA TO OR FROM BUFFER
E42F 3A9EE5           LDA     SEKSEC          ;MASK BUFFER NUMBER
E432 E601             ANI     SECMSK          ;LEAST SIGNIF BITS
E434 6F               MOV     L,A             ;READY TO SHIFT
E435 2600             MVI     H,0             ;DOUBLE COUNT
                      REPT    7               ;SHIFT LEFT 7
                      DAD     H
                      ENDM
E437+29               DAD     H
E438+29               DAD     H
E439+29               DAD     H
E43A+29               DAD     H
E43B+29               DAD     H
E43C+29               DAD     H
E43D+29               DAD     H
                ;     HL HAS RELATIVE HOST BUFFER ADDRESS
E43E 11B5E5           LXI     D,HSTBUF
E441 19               DAD     D               ;HL = HOST ADDRESS
E442 EB               XCHG                    ;NOW IN DE
E443 2AB3E5           LHLD    DMAADR          ;GET/PUT CP/M DATA
E446 0E80             MVI     C,128           ;LENGTH OF MOVE
E448 3AB1E5           LDA     READOP          ;WHICH WAY?
E44B B7               ORA     A
E44C C255E4           JNZ     RWMOVE          ;SKIP IF READ
                ;
                ;     WRITE OPERATION, MARK AND SWITCH DIRECTION
E44F 3E01             MVI     A,1
E451 32A5E5           STA     HSTWRT          ;HSTWRT = 1
E454 EB               XCHG                    ;SOURCE/DEST SWAP
                ;
                RWMOVE:
                      ;C INITIALLY 128, DE IS SOURCE, HL IS DEST
E455 1A               LDAX    D               ;SOURCE CHARACTER
E456 13               INX     D
E457 77               MOV     M,A             ;TO DEST
E458 23               INX     H
E459 0D               DCR     C               ;LOOP 128 TIMES
E45A C255E4           JNZ     RWMOVE
                ;
                ;     DATA HAS BEEN MOVED TO/FROM HOST BUFFER
E45D 3AB2E5           LDA     WRTYPE          ;WRITE TYPE
E460 FE01             CPI     WRDIR           ;TO DIRECTORY?
E462 3AAFE5           LDA     ERFLAG          ;IN CASE OF ERRORS
E465 C0               RNZ                     ;NO FURTHER PROCESSING
                ;
                ;     CLEAR HOST BUFFER FOR DIRECTORY WRITE
E466 B7               ORA     A               ;ERRORS?
E467 C0               RNZ                     ;SKIP IF SO
E468 AF               XRA     A               ;0 TO ACCUM
E469 32A5E5           STA     HSTWRT          ;BUFFER WRITTEN
E46C CD7FE4           CALL    WRITEHST
E46F 3AAFE5           LDA     ERFLAG
E472 C9               RET
                ;
                ;*****************************************************
                ;*                                                   *
                ;*      UTILITY SUBROUTINE FOR 16-BIT COMPARE        *
                ;*                                                   *
                ;*****************************************************
                SEKTRKCMP:
                      ;HL = .UNATRK OR .HSTTRK, COMPARE WITH SEKTRK
E473 EB               XCHG
E474 219CE5           LXI     H,SEKTRK
E477 1A               LDAX    D               ;LOW BYTE COMPARE
E478 BE               CMP     M               ;SAME?
E479 C0               RNZ                     ;RETURN IF NOT
                ;     LOW BYTES EQUAL, TEST HIGH 1S
E47A 13               INX     D
E47B 23               INX     H
E47C 1A               LDAX    D
E47D BE               CMP     M       ;SETS FLAGS
E47E C9               RET
                ;
                ;*****************************************************
                ;*                                                   *
                ;*      WRITEHST PERFORMS THE PHYSICAL WRITE TO      *
                ;*      THE HOST DISK, READHST READS THE PHYSICAL    *
                ;*      DISK.                                        *
                ;*                                                   *
                ;*****************************************************
                WRITEHST:
                      ;HSTDSK = HOST DISK #, HSTTRK = HOST TRACK #,
                      ;HSTSEC = HOST SECT #. WRITE "HSTSIZ" BYTES
                      ;FROM HSTBUF AND RETURN ERROR FLAG IN ERFLAG.
                      ;RETURN ERFLAG NON-ZERO IF ERROR
E47F 3E0A             MVI     A,WRTCMD        ;LOAD WRITE COMMAND
E481 CD8FE4           CALL    COMCODE         ;SEND COMMAND TO CONTROLLER
                WRTBUFF:
                      ;THIS LOOP SENDS BUFFER TO CONTROLLER
E484 C3ECE4           JMP     OUTLOOP         ;SEND OUT THE BUFFER
                ;
                READHST:
                      ;HSTDSK = HOST DISK #, HSTTRK = HOST TRACK #,
                      ;HSTSEC = HOST SECT #. READ "HSTSIZ" BYTES
                      ;INTO HSTBUF AND RETURN ERROR FLAG IN ERFLAG.
```

Listing 1 continued on page 5

ZENITH/ Heath Users



Double Your 5¼″ disk storage capacity without adding a drive.

Get twice as much from your H88 or H89 microcomputer. Our FDC-880H floppy disk controller, in conjunction with your 5¼″ drives, for example, expands memory capacity from 256 bytes to 512 bytes per sector.

And it handles single and double-sided, single and double-density, 8″ and 5¼″ drives — simultaneously.

C.D.R. Systems Inc.

Controlled Data Recording Systems Inc.
7210 Clairmont Mesa Blvd., San Diego, CA 92111
(714) 560-1272

Circle 89 on inquiry card.

MicroFrame
has new life

New Sources of S-100 Bus
Mainframes & Disk Enclosures

Power Regulation meeting previous standards

Ann Arbor Terminals
60 Lines Display
48 Programmable Keys
1,499 prepaid

CALL FOR PRICES

MICRO MIDWEST
10205 W. 69 TERR.
MERRIAM, KS. 66203
913-362-3462

Circle 299 on inquiry card.

TeleVideo™ USERS!

COGITATE Fast Dump/Restore System for TS 802H, 806.
- Back-up files twice as fast as PIP.
- Double the storage, up to 700K/diskette, with multiple diskette capability **$90.00**

COGITATE Type Ahead With Print Screen
- Print key prints screen.
- 64 character type ahead buffer **$90.00**

TurboDOS
- MULTI-USER **$750.00**
- SEMINARS **$250.00**

CP/M® is registered trademark of Digital Research, Inc.
TurboDOS is a registered trademark of Software 2000, Inc.

PLUS OTHER GOOD TELEVIDEO STUFF!

COGITATE, Inc.
SPECIALISTS IN UNIQUE TELEVIDEO SOFTWARE
24000 Telegraph Road, Southfield, MI 48034
(313) 352-2345
VISA/MASTER CARD Accepted

Circle 97 on inquiry card.

Listing 1 continued:

```
E487 3E08              MVI   A,READCMD        ;GET READ COMMAND
E489 CD8FE4            CALL  COMCODE          ;SEND COMMAND TO CONTROLLER
                READBUF:
                       ;ROUTINE READS DATA INTO BUFFER
E48C C3FDE4            JMP   INLOOP           ;READ THE DATA
                ;
                COMCODE:
                       ;COMMON CODE FOR READ AND WRITE TO CONTROLLER
E48F 3295E5            STA   CMDBYT0          ;STORE COMMAND BYTE
E492 3A9FE5            LDA   HSTDSK           ;GET DISK NUMBER
                       REPT  5                ;MOVE TO BIT POSITION
                       RAL                    ;
                       ENDM
E495+17                RAL                    ;
E496+17                RAL                    ;
E497+17                RAL                    ;
E498+17                RAL                    ;
E499+17                RAL                    ;
E49A E6E0              ANI   0E0H             ;MASK OFF HI ADDRESS DATA
E49C 3296E5            STA   CMDBYT1          ;SAVE IN COMMAND BLOCK
E49F 2AA0E5            LHLD  HSTTRK           ;GET TRACK NUMBER
                       REPT  5                ; * 32 SECTOR/TRACK =# SECTORS
                       DAD   H                ;
                       ENDM
E4A2+29                DAD   H                ;
E4A3+29                DAD   H                ;
E4A4+29                DAD   H                ;
E4A5+29                DAD   H                ;
E4A6+29                DAD   H                ;
E4A7 3AA2E5            LDA   HSTSEC           ;GET SECTOR #
E4AA 5F                MOV   E,A              ;SET UP FOR ADD
E4AB AF                XRA   A                ;
E4AC 57                MOV   D,A              ;
E4AD 19                DAD   D                ;ADD IN SECTOR NUMBER
E4AE 7D                MOV   A,L              ;
E4AF 3298E5            STA   CMDBYT3          ;SAVE IN COMMAND BLOCK
E4B2 7C                MOV   A,H              ;MSB OF ADDRESS
E4B3 3297E5            STA   CMDBYT2          ;SAVE IN COMMAND BLOCK
E4B6 3E01              MVI   A,01             ;
E4B8 3299E5            STA   CMDBYT4          ;LOAD SECTOR COUNT
E4BB AF                XRA   A                ;DESELECT HALF STEP OPTION
E4BC 329AE5            STA   CMDBYT5          ;SELECT HALF STEP OPTION
E4BF 2195E5            LXI   H,CMDBYT0        ;START OF COMMAND BLOCK
E4C2 CDCCE4            CALL  SENDCMD          ;SEND COMMAND BLOCK TO CONTROLLER
E4C5 CDDFE4            CALL  CLRCMD           ;CLEAR THE COMMAND BUFFER
E4C8 21B5E5            LXI   H,HSTBUF         ;POINT TO TRANSFER BUFFER
E4CB C9                RET
                ;
                SENDCMD:
                       ;SEND THE COMMAND BLOCK POINTED TO BY HL
E4CC CD11E5            CALL  SELCT            ;SELECT THE CONTROLLER
E4CF CD26E5            CALL  WAIT             ;WAIT FOR REQUEST
E4D2 0606              MVI   B,6              ;COMMAND BYTE COUNT
                SENDCD:
E4D4 7E                MOV   A,M              ;GET COMMAND BYTE
E4D5 D3A1              OUT   DATA             ;SEND IT TO CONTROLLER
E4D7 23                INX   H                ;NEXT BYTE
E4D8 05                DCR   B                ;KEEP TRACK
E4D9 C2D4E4            JNZ   SENDCD           ;
E4DC C326E5            JMP   WAIT             ;WAIT FOR REQUEST
                ;
                CLRCMD:
                       ;CLEAR COMMAND BLOCK
E4DF AF                XRA   A                ;LOAD ZERO
E4E0 0606              MVI   B,6              ;BYTE COUNT
E4E2 2195E5            LXI   H,CMDBYT0        ;POINT TO FIRST BYTE
                CLRCMD1:
E4E5 77                MOV   M,A              ;CLEAR BYTE
E4E6 23                INX   H                ;
E4E7 05                DCR   B                ;SUB COUNT
E4E8 C2E5E4            JNZ   CLRCMD1          ;
E4EB C9                RET                    ;
                ;
                OUTLOOP:
                       ;SEND OUT DATA UNTIL C/D BIT SIGNALS FINISH
E4EC DBA0              IN    CNTRL            ;READ STATUS
E4EE E602              ANI   CMDAT            ;CHECK C/D BIT
E4F0 CA2EE5            JZ    CKERR            ;DONE, CHECK FOR ERRORS
E4F3 7E                MOV   A,M              ;GET OUTPUT BYTE
E4F4 D3A1              OUT   DATA             ;TO CONTROLLER
E4F6 CD26E5            CALL  WAIT             ;REQUEST
E4F9 23                INX   H                ;NEXT BYTE
E4FA C3ECE4            JMP   OUTLOOP          ;KEEP LOOPING
                ;
                INLOOP:
                       ;READ DATA UNTIL C/D BIT SIGNALS FINISH
E4FD DBA0              IN    CNTRL            ;READ STATUS
E4FF E602              ANI   CMDAT            ;CHECK C/D BIT
E501 CA2EE5            JZ    CKERR            ;CHECK FOR ERRORS
E504 DBA1              IN    DATA             ;READ BYTE
E506 77                MOV   M,A              ;TO MEMORY
E507 CD26E5            CALL  WAIT             ;REQUEST
E50A 23                INX   H                ;NEXT LOCATION
E50B C3FDE4            JMP   INLOOP           ;KEEP LOOPING
                ;
                RESET:
                       ;RESET THE XEBEC DISK CONTROLLER
E50E D3A3              OUT   RSTPT            ;SEND ANYTHING TO THE RESET PORT
E510 C9                RET
                ;
                SELCT:
                       ;SELECT DISK 0 OF THE XEBEC CONTROLLER
```