FEATURES Z/H-100 SYSTEM

HOW TO USE: ZENITH/HEATH COMPUTERS

S-A DESIGN

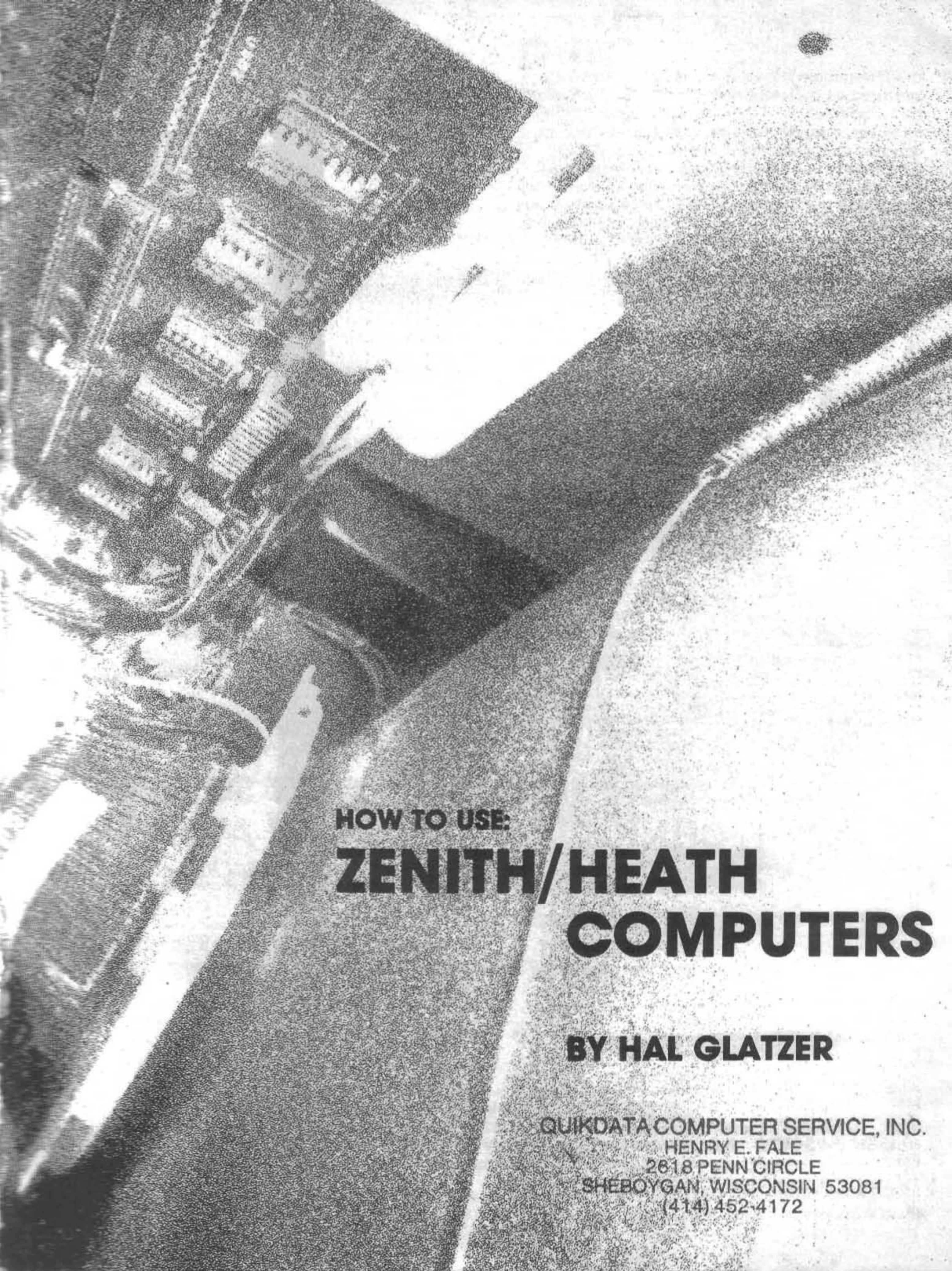
YOUR GUIDE TO:

- GETTING STARTED
- KITS & ACCESSORIES
- AVOIDING MISTAKES
- PERSONAL COMPUTING
- WORD PROCESSING
- BUSINESS ACCOUNTING
- DATA MANAGEMENT
- TELEPHONE NETWORKS
- GAMES & EDUCATION

BY HAL GLATZER

INCLUDES 16-BIT, CP/M ZENITH Z-100 SYSTEMS

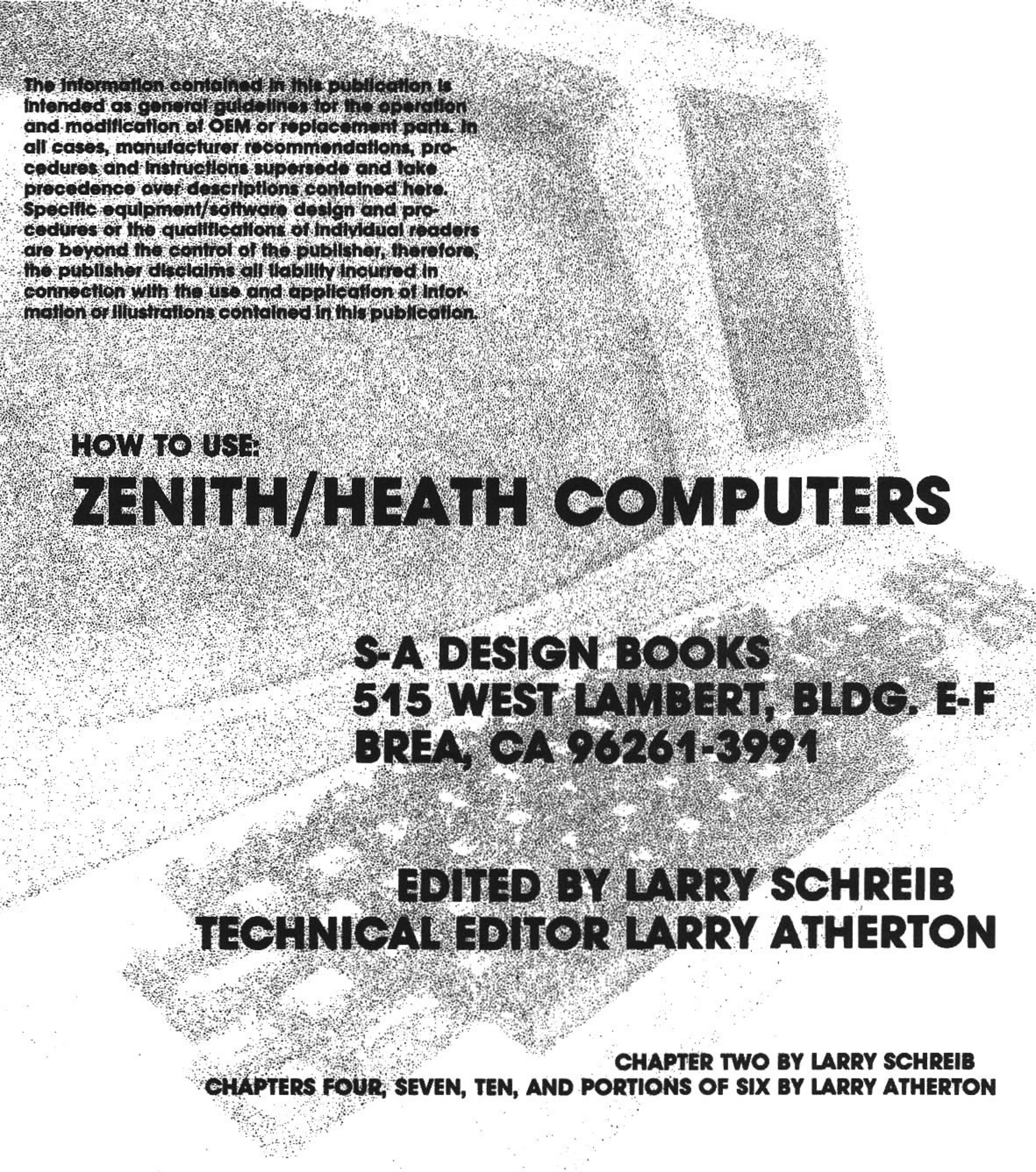
INCLUDES ALL-IN-ONE HEATH H-89 SYSTEMS AND ACCESSORIES



HOW TO USE:

ZENITH/HEATH COMPUTERS

BY HAL GLATZER



The information contained in this publication is intended as general guidelines for the operation and modification of OEM or replacement parts. In all cases, manufacturer recommendations, procedures and instructions supersede and take precedence over descriptions contained here. Specific equipment/software design and procedures or the qualifications of individual readers are beyond the control of the publisher, therefore, the publisher disclaims all liability incurred in connection with the use and application of information or illustrations contained in this publication.

HOW TO USE: ZENITH/HEATH COMPUTERS

S-A DESIGN BOOKS
515 WEST LAMBERT, BLDG. E-F
BREA, CA 96261-3991

EDITED BY LARRY SCHREIB
TECHNICAL EDITOR LARRY ATHERTON

CHAPTER TWO BY LARRY SCHREIB
CHAPTERS FOUR, SEVEN, TEN, AND PORTIONS OF SIX BY LARRY ATHERTON

ISBN 0-931472-18-0

TABLE OF CONTENTS

THIS IS AN INDEPENDENT PUBLICATION PRODUCED BY S-A DESIGN BOOKS AND IS NOT IN ANY WAY ASSOCIATED WITH THE HEATH OR ZENITH DATA SYSTEMS CORPORATIONS.

INTRODUCTION

Someday, we may come to think of computers as we now think of automobiles: so much a part of our lives that owning one, or knowing how to operate one, is taken for granted. It's practically impossible to earn a living in many parts of North America without access to a car, and it's often hard to cash a check without showing a driver's license.

If that happens to computers—if someday a person can't really flourish without owing a computer, or at least being able to work with one—then it may be because a handful of companies did for computers between 1975 and 2000 what the auto manufacturers did in the first half of the 20th century.

But this book isn't about Apple, or Radio Shack, or IBM or, for that matter, about Heath or Zenith per se. It is about small computers, and it focuses on one group of computers that has a loyal following: those made originally by the Heath Company, of Benton Harbor, Michigan, which was acquired in 1979 by Zenith Data Systems, a division of the Zenith Radio Corporation, of Chicago, Illinois.

Many studies have been made showing why people prefer one brand of automobile to another, and often the reason involves advertising. Yet neither Heath nor Zenith has engaged in massive advertising campaigns. Nothing resembling Heath's quarterly kitbuilders' catalog, nor the magazine spreads that promote Zenith's color television sets has—until recently—been used to sell computer products. Beyond advertising, though, many car buyers determine early in their lives that a particular brand *suits* them, somehow. Styling and accessories may play a part, but when the going gets tough, most car owners want reliability, sturdiness, and the security of knowing that there are a lot of dealers and service people around.

Chrysler loyalists have their Plymouths, General Motors' fans have their Chevrolets, Ford people swear by their Fords, and Volkswagens seem to run forever. These have been many people's "first" cars, and they continue to be taken care of, and kept running as solid *transportation* machines, even after many years. Why? Because all of them give a sense of reliability, ease of operation, and a feeling of their "permanence" to a driver.

Like the owners of those cars, people who use the Zenith/Heath computers have a strong loyalty to the brand. It is my belief that there are some good reasons why: the computers are reliable, easy to operate, and are built so well that they are likely to endure as functioning machines long after many of their competitors have been traded in on newer models.

To imagine that a computer is just a fancy adding machine with a TV screen and a typewriter attached is to imagine that a car is just an internal-combustion engine with four wheels on the ground and one to steer it with. A manufacturer decides how to put a car together, and those decisions affect its customers' lives in profound ways. In my opinion, Heath and Zenith made a lot of the "right" choices:

1) The terminal—especially the keyboard which the user handles—is fundamentally well-designed, and has offered more user-friendly features than most of its competitors;

2) The microprocessor chip—the gizmo inside that actually does the work—has proven to be so popular, among other manufacturers as well, that thousands of people have learned how to write programs for it;

3) The "plain vanilla" option—a fully-functioning computer with no frills, at a low price—has enabled small business owners, professional and technical people, as well as hobbyists, to start small and build their systems gradually, expanding them to meet their needs.

4) The co-processor—which, in the new Z-100s, runs the existing 8-bit software until new, 16-bit programs come along—is in step with the rest of the computer industry, neither leading nor lagging in the stage-of-the-art.

If a few words of jargon send you running for the glossary, don't be alarmed. This book will explain everything *in context* as it goes along. But I still want to share some feelings I've had about "my" Zenith computer before I introduce the actual text.

As a professional writer, I was not looking for a computer so that I could learn about computers, or so that I could write programs, or even to do any of the mathematical or computational tricks for which computers have become popular. Frankly, I wanted a better typewriter. I had used office word processors, the kind that don't do anything else, and I longed for that kind of power over print: manipulating, editing and remaking words, phrases, pages and even books. I needed a word processor, but the office-size, stand-alone variety was far too expensive for a freelance journalist.

The best alternative was to buy a microcomputer and some software that would make it process words instead of data. Before I went shopping, though, I did quite a bit of homework, and I recommend that other people do likewise. I went to a few computer shows, I bought practically every magazine on the subject, read a few general books from the library, and I talked to a dozen people who had computers of their own. I set my price range early on—I knew how much money I could spend—and I stayed within my limits. When the time came to call on vendors, to see what they could offer me, I had drawn up a list of requirements—the absolute minimum that I would expect from a computer—and a wish-list of desirable features—those that would enhance my work.

The winnowing process depended heavily on my needs *as a writer*, and people in other businesses should draw on their *own* needs. But three things impressed me about the Zenith that I eventually bought:

It had *function keys* that inserted text, or moved the cursor around the screen, etc., the same way that the expensive stand-alone word processors did.

It was sold and serviced by a reputable dealer who understood its hardware and its software thoroughly. The dealer, in this case, had on-staff engineers who had actually enhanced the basic computer's capabilities, and I knew that I would be able to expand my system, with their help and "handholding," as far and as long as I wished.

Today, after two years of working with my Zenith, I can say that it has served me well. If I could change anything about it it would be to have a detachable keyboard. But the fundamental design, construction, and features have never given me trouble, and over 24 months the system has spent less than two days in the shop being serviced.

Therefore, I want to thank Brad Gjerding, Kay Gjerding, Doug Sauby and John Michael, of Magnolia Microsystems, Inc., for selling me the tools of my trade, and for being willing to share their knowledge and experience with me and with my readers.

I want to thank Andy Czernek, of Zenith Data Systems, who arranged for me to meet and interview the people at Heath and Zenith who have contributed so much to this book: Mike Brenner, Thomas Dornback, Tom Glover, Dave Green, Mark Groeger, Carl Goy, Skip Gwyer, Gerry Kabelman, Richard Kellerman, Myron Kukla, Larry Liddle, Jim Blake, Donald Moffet, Robert Reid, and Barry Watzman. Thanks, too, to Gordon Letwin, of Microsoft.

Special thanks go to Kay Green, Terry Murphy and Margaret Elwood, for their help.

Finally, the idea for writing this book was suggested by two dedicated Zenith/Heath users who found in their computer not only a great word processor and publishing aid, but a source of inspiration. They actually hooked up most of the hardware, and ran most of the programs, that are described in these pages. The nitty-gritty reviews are theirs, and I concur with their findings. I am very grateful to them for giving me the chance to write this book: Larry Atherton and Larry Schreib, of S-A Design Books.

ABOUT THIS BOOK

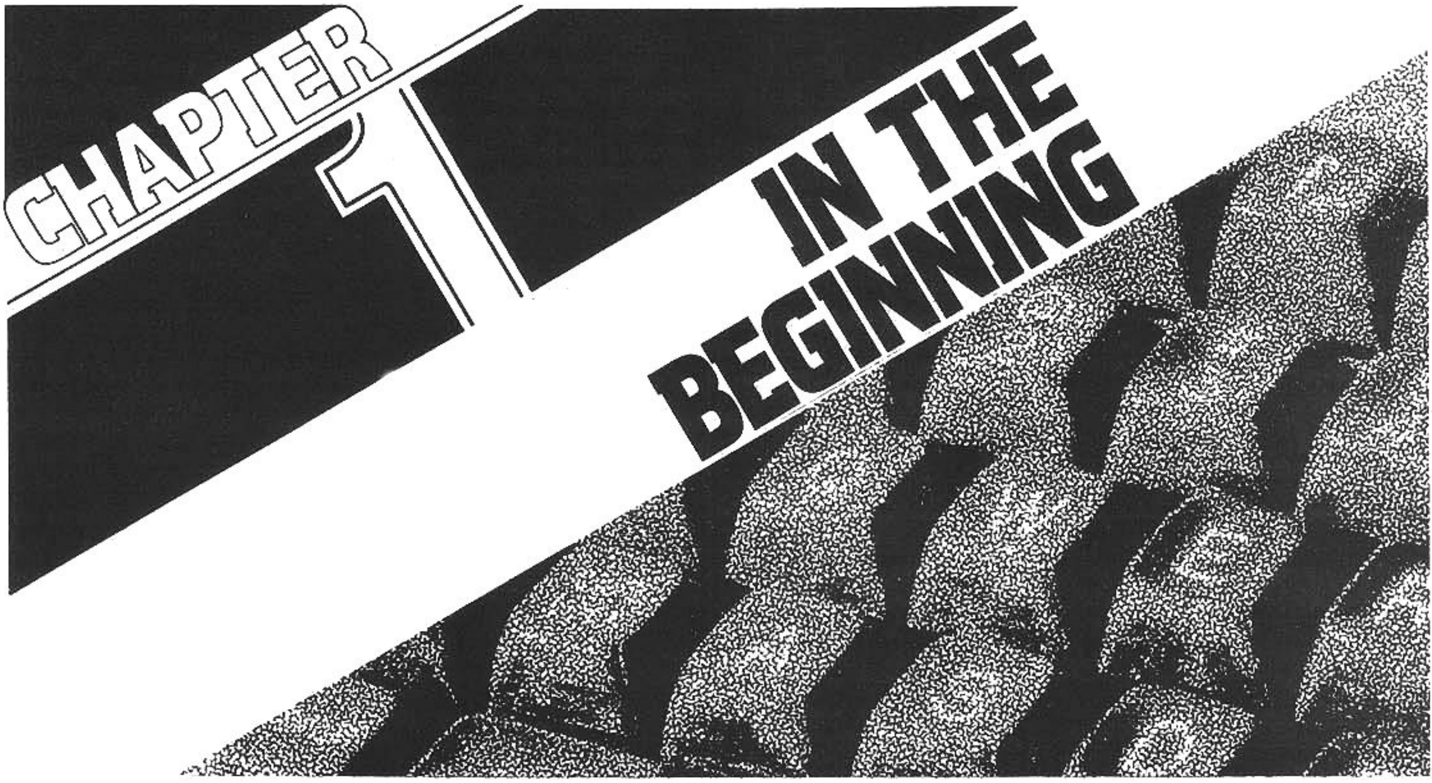
This book presents a wide-ranging overview of the Zenith/Heath family of microcomputers. However, we feel it is not a typical computer book. Most computer texts are either blandly generalized or crippled with cryptic details. This book strives to find a useful middle ground. It is filled with specific information about how to use one kind of computer (Zenith/Heath), but it also contains a generous volume of helpful background material.

The book is organized in a progressive manner. The beginning chapters assume the reader has no (or very little) knowledge of computers, and additional details are presented in later discussions, but the reader does not have to be an electronic engineer to understand what is being said. Though many interesting technical topics are presented, the writer aproaches them in a non-technical manner. In this way the beginner should be able to develop a useful understanding of general microcomputer functioning, while becoming very familiar with the powerful capabilities of Zenith/Heath computers.

Chapters one and two examine the relatively short history of modern microcomputers (as typified by the development of the Heath all-in-one H-89 computer), and introduce a simplified view of how a microcomputer works. Chapters three and four look inside the basic "hardware" of Zenith/Heath computers, and further explore elementary computer operating principles. Chapters five and six present a broad overview of computer "programming" (software), and highlight this all-important subject with detailed reviews of many popular and practical programs. Chapter seven returns to the subject of hardware, but with a view toward the wide range of upgrades and auxiliary hardware that expand the practical value of a microcomputer. Chapter eight is a potpourri of interesting, though sometimes challenging, minutiae. Chapter nine illustrates some of the diverse special applications for Zenith/Heath computers. And, chapter ten is a pictorial chronology of the interesting and unique Heath "build-it-yourself" microcomputer and printer kits.

The publishers express special gratitude to Andy Czernek of Zenith Data Systems for his generous assistance during the final production of this book.

CHAPTER 1 IN THE BEGINNING



THE FIRST HEATH H-8 COMPUTER

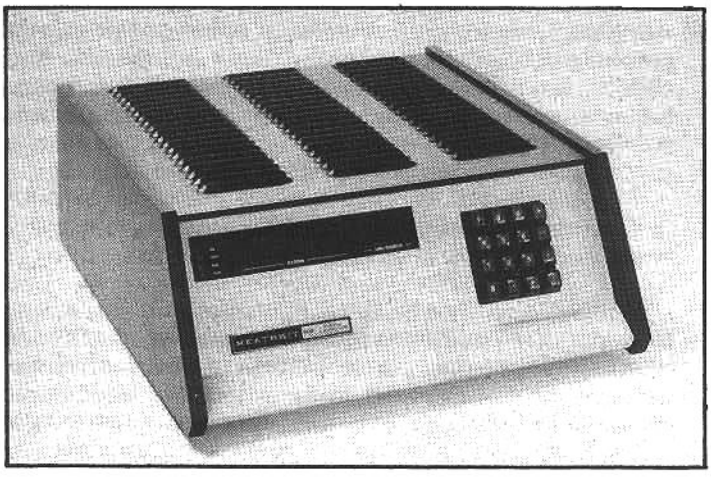
The first Heath computer, designated the H-8, was a typical Heath do-it-yourself kit. When introduced in their catalog in the fall of 1977, it appeared to be a plain box with an array of toggle switches and red lights across the front panel. The kit contained an imposing collection of circuit boards and paper bags laden with electronic components; and though it was undoubtedly intimidating for a complete novice, it also included the extensively detailed, step-by-step assembly instructions that have become the hallmark of Heath kits.

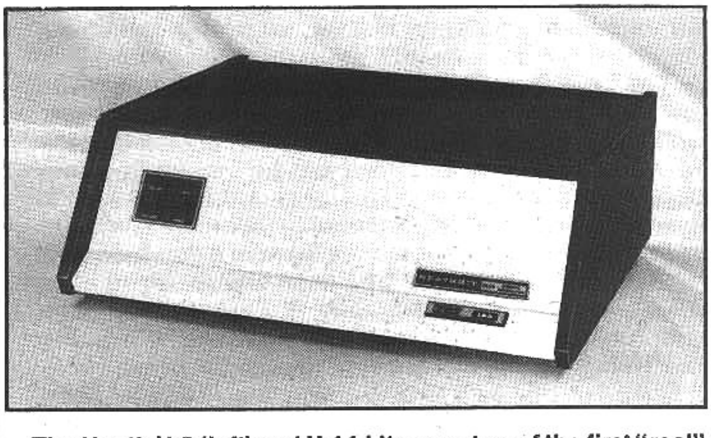
Once assembled, the H-8 could be laboriously programmed to play tic-tac-toe or perform some elementary math, but because there were virtually no major prewritten programs available, it had very little, if any, immediate practical value. Nonetheless, it was a "real computer." And though it had limited appeal, early computer buffs developed an affection for it. Today the H-8 remains largely a hobbyist-oriented system, but through the years it has introduced many enthusiasts to the world of personal computing.

In the same 1977 catalog, Heath introduced a second computer kit, an unusual 16-bit system called the H-11. (Most microcomputers currently use 8-bit technology, though the more powerful 16-bit data-handling architecture is becoming increasingly available on reasonably inexpensive small machines like the Zenith/Heath Z/H-100 series: see chapter 4.) It has the same microprocessor chip—a super-miniaturized computer processing unit constructed on a thin wafer of silicon—that Digital Equipment Corporation (DEC) uses in some of its highly regarded commercial computer systems.

Both of these early Heath computers were aimed primarily at what was then a relatively small number of dedicated enthusiasts. They were rarely purchased and assembled by people interested only in the practical services of a computer. Instead, they appealed to computer hobbyists who enjoyed probing the inner workings of the machine.

Currently, computer users who want a "turn-key"





The Heath H-8 (left) and H-11 kits were two of the first "real" computers available to early hobbyists. Throughout the 1970s, thousands of these microcomputers were built by clever hobbyists who developed and traded software, accessories and enhancements for them.

computer—something they can buy (or assemble), plug in and use immediately—tend to purchase the Zenith or Heath Z/H-89, Z/H-90 or Z/H-100 systems. These computers were designed with later microprocessor technology, and they are more versatile and somewhat easier to use—especially for first-time owners—than the early hobbyist systems.

The history of Heath computers is not long, since microprocessor chips were only developed in the mid-1970s, but the pivotal events in the evolution of the original H-8 into the Heath H-89 are typical of the incredibly rapid development that is common in microcomputer design.

FURTHER DEVELOPMENTS

Today many computer users take it for granted that computers have always had a keyboard and a video display screen. They are accustomed to hitting a few keystrokes to manipulate lines on and off the screen, or pressing a few keys to move elements on the display up, down or sideways. It seems so simple, so natural, that a novice may conclude that this was the idea from the start. It was not.

The H-8 was a functional computer, however it was simply a box containing the primary computer circuitry—the central processing unit or *CPU*, volatile random-access memory or *RAM*, and additional supporting elements. It did not have any of the sophisticated "peripherals" that are generally associated with current microcomputers.

The hobbyist who assembled one was faced with the somewhat laborious process of programming it (i.e., loading into the random-access memory a series of instructions designed to perform a sequence of data manipulations or computations). At first the only viable way to load a program was to log it in step-by-step, using various configurations of the switches or push-buttons across the front panel. This, in effect, wrote a series of program instructions into the computer memory. However, when the machine was switched off the program would disappear from the volatile computer memory, and each time the user wanted to run a different program, he had to go through the tedious process of hand-loading and "debugging" the program, which might consist of several hundred instructions.

THE H-9 TERMINAL

Though the H-8 performed exactly as advertised, it was basically just a computer in a box. For it to be a practical tool, the user needed some way to communicate with the "box" more readily. Specifically, the H-8 lacked a typewriter-like keyboard to allow the user to easily enter data and program instructions, and it lacked a display to permit the user to view data input and output from the computer. In short, the H-8 needed to be connected to a suitable interactive communications device—like a *video terminal*.

Heath developed such a terminal, called the H-9, to complement the H-8, and made it available in kit form. Heath designer Carl Goy recalls the elementary nature of the H-9. "Our products had always been designed for tinkerers. A Heath customer wouldn't mind spending eons to build a keyboard, so it was a pretty basic kit," says Goy, who is now Zenith's manager of computer products engineering. "Our first terminal was like a glass teletype. That is, it showed on a CRT video screen what would have appeared on paper if the terminal had been a typewriter—then the most readily available keyboard-type electronic communication device."

Though the H-9 had a video display, it had many of the same limitations as a conventional teletype. The user could only enter characters (data) in sequence—one right after the other—and the line of data was automatically scrolled up as soon as it was filled. As a result, the user had only limited control over the display, and changing or correcting text was not easy.

One young engineer, Gordon Letwin, brought a background in writing computer software (programs) to the project. At the time, Letwin was a recent college graduate who had developed a talent for writing software, and found an opportunity to do it for Heath. "The prototype H-9 was a rat's nest of wires wrapped around white plastic perf-boards," recalls Letwin. "It had so many connections that dirt was a factor in the failure rate; if you dropped it a few inches, it worked better! But over and above this, it had serious problems. There were internal problems that kept it from realizing its potential. You couldn't go from the bottom of the screen to the top, and you could only display capital letters.

"And it was too slow. The people who developed the H-9 were limited by the technology of those days. Unfortunately, if you made some of the electronics work too fast, the letters would scroll right off the screen! Most people didn't realize that the screen display was controlled by the software, not the hardware," Letwin explains.

Heath's engineers designed the H-9 terminal to complement the H-8 computer. Though it incorporated a television-like CRT video screen, it had a limited display. By comparison, a modern terminal can display up to 2000 characters on the screen.

Heath was now considering a redesign of the H-8 computer, but it became obvious that a more sophisticated terminal would greatly enhance the capabilities of the new computer. So Goy, Letwin and the other Heath engineers set about to design a new video terminal that would eliminate the shortcomings of the H-9.

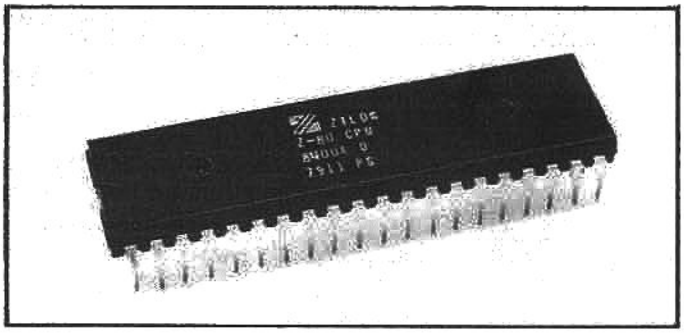
They proposed to include many advanced features—like direct cursor addressing (the *cursor* is a "pointer" that indicates on the screen where data is being entered or deleted). This would permit the user to instantly move the cursor to any spot on the video screen. They also wanted to add a separate numeric keypad and "programmable special-function keys." This would greatly increase the ability of the keyboard to communicate with the computer, and simplify text editing and data entry. Under program control these additional keys could delete a character from the text, insert a character between two others, add or delete entire lines of copy, split open the displayed text so the user could easily insert and remove entire blocks of copy, or perform specified complex functions with a single keystroke. But these features would complicate the design.

Letwin explains, "At the time I was working with another engineer on a master program to direct the main microprocessor to load and unload data to and from a floppy disk—what's called a *disk operating system*, or DOS, or simply an operating system. When I considered the design of the new terminal, I felt that what it really needed was a separate processor to control the terminal functions, and we knew where we could find one; a microprocessor like the one in the main computer would be perfect."

The Heath engineers were quick to grasp the extensive power of the miniature processor: the then-new "computer-on-a-chip." It answered the needs of the new terminal perfectly. In fact, they would eventually put so much "intelligence" into the terminal—by incorporating in the new design a powerful microprocessor dedicated entirely to terminal functioning—that it could do many advanced tasks without the aid of the microprocessor in the main computer CPU!

THE ALL-IN-ONE H-89 COMPUTER

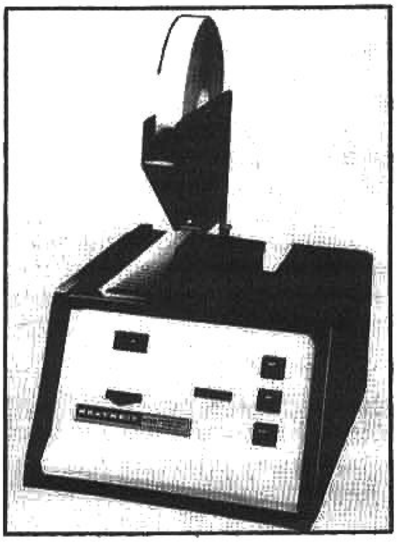
During the redesign of the H-8 computer a new chip became available that could execute instructions more efficiently than earlier designs. This new microprocessor was the now-famous Zilog Z80. It was part of the same processor "family" as the Intel 8080 being used in the existing H-8, and had many of the same operational characteristics. It could store in RAM and retrieve all or any portion of 65,535 pieces of data (numbers, letters, or characters) at a time. Switching from the 8080 to the Z80 seemed like a natural upgrade to increase the capabilties of the new computer.



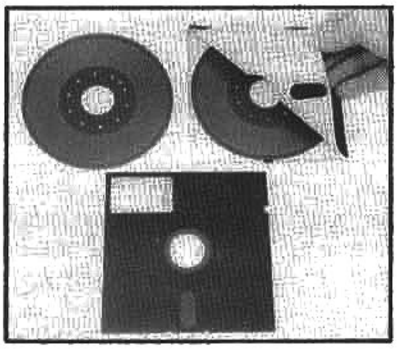
This is a microprocessor, a "computer-on-a-chip." The Zilog Z80 processor, used in Z/H-19 terminals and the Z/H-89 and Z/H-90 computers, is one of the most popular processors in the microcomputer industry, and *contains nearly 100,000 electronic components—on a wafer of silicon.*

Meanwhile, another problem with early microcomputers was being solved by a technological evolution that would dramatically change computer history: the floppy disk. Even with the aid of a terminal, loading programs was still awkwardly slow and tedious, and finding a program bug—created by hitting a wrong key while entering the many hundreds of data elements in a typical computer program—could be an excruciating and time-consuming task. This was severely limiting the overall capability of the small computer. (High-speed magnetic tape drives could readily load huge amounts of data into large mainframe computers, but these drives were far too expensive for microcomputers.)

But things were improving. A technique had been developed to permanently record program instruction sequences with an inexpensive device that punched holes in a narrow paper tape. When the tape was fed back through the device, it would rewrite the program into the computer memory. This was faster than hand-loading programs and it was somewhat more reliable; however, switching between programs and entering large amounts of data was still relatively cumbersome.



Paper-tape punches were among the first low-cost data-storage devices, and were used by early computer hobbyists to record and "playback" large programs. Data from the computer memory was transmitted to the punch and "coded" in the hole patterns punched in the tape. Later this paper tape was fed through the reader channel and the patterns were decoded and reentered in the computer memory.



A *floppy disk* is simply a thin platter of mylar plastic, coated with a magnetic-sensitive oxide. One disk can store millions of bits of magnetic data. The disk is permanently enclosed in a flexible envelope that protects the oxide surface and wipes it clean by trapping dust particles on a porous interior surface.

Another derivation of this recording technique allowed programs and data to be stored on ordinary audio tape cassettes. Original programs still had to be keyed into the computer memory and debugged by hand, but once the program was perfected, it could be dumped—through a hardware interface and a connecting cable—to an ordinary monaural cassette recorder, where it was stored as magnetic impulses on an audio tape cassette (just as music is recorded on a cassette). This provided for low-cost, reasonably reliable storage, and the programs could be easily, albeit slowly, reloaded from the tape cassette through the recorder to the computer memory. In addition, cassette recorders were inexpensive and the tapes could be duplicated to allow even very complicated programs to be readily swapped and sold.

Paper and cassette tape storage were steps in the right direction (and they are still used in special situations or as a low-cost storage alternative for budget-minded users), but for most practical and business applications tape storage was virtually swept aside by the emergence of the flexible or *floppy disk*, which has far greater potential for semi-permanent mass storage of programs and data at comparatively low cost.

Floppy disks are made of thin mylar plastic, coated with a magnetic-sensitive oxide (similar to audio recording tape), and permanently enclosed in a protective jacket. These disks are extremely portable, relatively inexpensive, and allow a great deal of information to be recorded and retrieved very rapidly.

The user can store many (even hundreds) of floppy disks in a relatively small space. He selects a disk from this "library" and inserts it into the *disk drive*, where a motor-driven spindle engages the central hole in the mylar disk through an access hole in the jacket. The spindle turns the disk at a fixed and carefully-regulated speed, much like a phonograph record. Magnetically encoded data is recorded or retrieved from the disk surface by a tiny recording head that can move laterally across the rotating disk surface as directed by the disk operating system software and the disk-controller (a hardware interface between the CPU and the disk drive.) This head corresponds somewhat to the arm and needle of a phonograph player.

The beautiful simplicity of floppy-disk storage, perhaps as much as anything, has made modern microcomputers possible. By rotating the disk and scanning the surface with the head, the disk drive can access any portion of the recording surface in an extremely short time. It is possible, therefore, to quickly and accurately record or retrieve large amounts of data with a device that is reasonably simple and reliable.

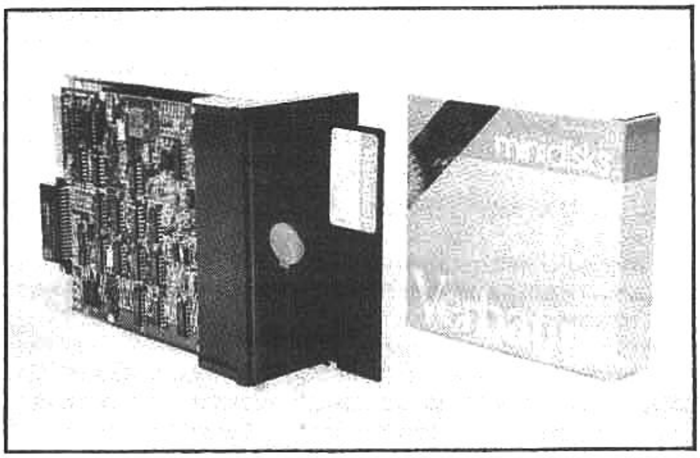
Most computer companies, Heath included, do not build their own disk drives; they buy them from manufacturers who specialize in disk drives. "At that time," Gordon Letwin recalls, "Heath was likely to be the biggest single purchaser of floppy disk drives ever, so manufacturers' representatives descended on us in droves."

After sorting through the available models, Heath engineers selected a compact drive that used a convenient 5.25-inch "minifloppy" disk, but since it was not supplied with a self-contained enclosure, it needed to be mounted in some sort of housing to protect it. Some of the Heath engineers had suggested mounting the disk drive in the same enclosure with the "computer" (the CPU, RAM storage, power supply and support elements). However, the new terminal design was nearing completion and the decision had already been made to redesign the terminal housing. The terminal, computer unit and disk drive would work together to make a complete, self-contained computer system, so by mid-1977 an idea began to form: why not put them all in one box!

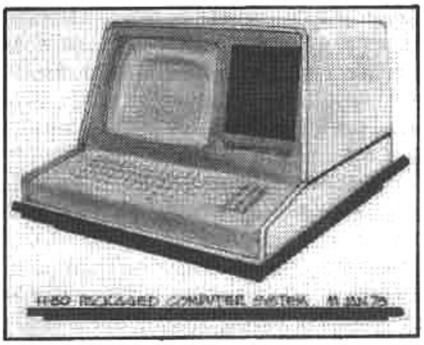
Says Goy, "It saved us quite a few dollars in packaging, and made the system work more efficiently. We had already decided to use a Z80 processor in the new computer, but we were worried that it would be overworked if it had to run the terminal too, so we added a second one to run the terminal—they aren't very expensive—and that removed a lot of the hardware constraints we had necessarily imposed on the machine. And the two chips could communicate directly with each other through a simple internal connection, eliminating a lot of external wiring."

By September, 1978, the Heath engineers had a prototype of the new terminal, called the H-19, working. It was to be offered as a separate component for use as a conventional communications device. The first full-functioning all-in-one computer, designated the H-89, with a single, onboard, 5.25-inch disk drive followed a few months later.

In June, 1979, at the National Computer Conference (a major trade convention) the H-19 terminal and H-89 computer were officially unveiled. At the time Heath was one of the first major manufacturers to offer an all-in-one computer with the major components integrated into a single cabinet. "We didn't know what anybody else was doing," Goy says, "but we knew that if a customer wanted a computer in those days, he had to have a lot of boxes and



This floppy-disk drive holds one 5.25-inch disk, and on each disk it can store the equivalent of approximately 70 pages of typewritten text. The drive shown is manufactured by Siemens, and fits into the cabinet of the all-in-one Z/H-89 or Z/H-90 computer.



Heath was a pioneer of the all-in-one microcomputer concept. With all of the major components in one enclosure, the system is compact and functional. This preliminary concept drawing depicts the computer with two disk drives, but Heath ultimately decided to use only one drive in the H-89 enclosure.

In 1977, the H-89 was offered as a do-it-yourself kit or as a "wired-and-tested" system, called the WH-89. When Zenith purchased Heath, the ready-to-run system became available through Zenith Data Systems as the Z-89, but the H-89 and WH-89 are still sold in Heath Electronics Centers and through the Heath catalog.

cables strung all over the place. The whole focus of our design, once we decided to do it, was to integrate the terminal, computer and disk drive into one enclosure. Maybe it wouldn't be something you could tuck under your arm, but at least it wouldn't be so big that it had to cover an entire desk."

THE HEATH-ZENITH CONNECTION

At this point it may be helpful to explain the relationship between Heath and Zenith. When Heath became involved in kit-based, hobby-oriented computers, the company was not directly associated with Zenith. This was before the incredible explosion of micro-technology made the inexpensive "personal computer" a practical reality. But as the market began to grow, Zenith, the largest domestic manufacturer of color televisions, was aware of the steadily increasing number of black-and-white CRT (cathode ray tube) displays being sold for use as video monitors for small computers.

(In computer jargon a CRT video display and the immediate hardware required for it to function is often called a *monitor.* A monitor is similar in appearance to a television set, but it is functionally quite different. In some systems the monitor is sold as a separate component, housed in an enclosure that is connected to the main computer processor unit by a wire. In such a design the keyboard may also be a separate component (also wired to the main computer) or it may be part of the enclosure for the main computer. A *terminal* combines the monitor and keyboard hardware in one enclosure. It is primarily a modular device used to communicate with the main computer. A single terminal may be a component part of a small system, or several terminals may be employed in a large multi-user system—where many users communicate with a single, very-powerful centralized computer.)

Eventually the cost of microprocessor chips—the centerpiece of all small computer design—plummeted as large-scale manufacturing techniques were developed. The concurrent introduction of reliable, low-cost floppy disk storage made it possible to record, erase, manipulate and permanently store large amounts of magnetically-encoded data on easily transported and relatively inexpensive floppy disks. These events, along with other contributing factors, made small computers—what are now generally called microcomputers—possible. In a relatively short time the sales of these small but awesomely powerful desktop computers took off like a rocket.

Zenith already had the capability to manufacture the CRT video displays. They also had considerable experience with electronic circuitry and the techniques required to build computers, and they were aware of the sales potential, so they moved to extend their long-established manufacturing and marketing capabilities into the computer market. To speed this entry, Zenith purchased Heath in 1979, and created a separate corporate entity, Zenith Data Systems, to market computers under the Zenith brandname.

Initially Zenith Data Systems sold Heath-designed products—though only in fully assembled, ready-to-run form. Later, as Zenith began to focus on the huge office-equipment market, some ZDS-labeled products—though maintaining an obvious link to their Heath origins—were configured for the specific requirements of business-type applications: the Z/H-90 is an example. Heath continued to sell their existing line of computers, terminals and accessories as kits, and they also offered some items—like the all-in-one H-89 computer—in fully assembled form.

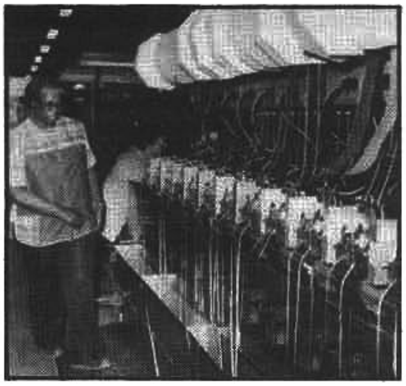
As the capabilities of microprocessor technology continued to expand, Zenith Data Systems established a set of design criteria and began work on an all-new computer—from the desktop up. This system, the Z-100 series, was introduced in 1982, and with these powerful and flexible machines it is obvious that ZDS has now completed a full-fledged expansion into the extremely competitive and lucrative office-equipment and scientific markets.

Hobbyists and avid kitbuilders will be happy to hear that the new Z-100 is available from Heath in kit form, continuing Heath's long-standing commitment to budget-minded computer enthusiasts. And as of this writing, Heath indicates they will sustain the H-8 and H-11, and both Heath and Zenith will continue sales and support of the Z/H-89 family of computers and accessories.

HOW THEY'RE BUILT

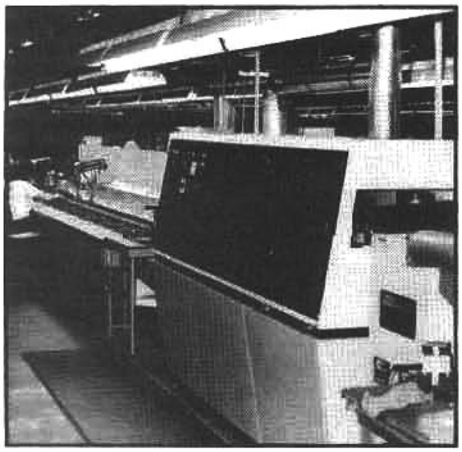
Today a kitbuilder can follow Heath's detailed instructions and build a computer on the kitchen table (see chapter 10), but on the assembly line things are done a little differently.

Kit versions of the H-89 begin on the Heath "bag line." Individual components are hand-selected from trays and placed in labeled bags. Each kit is complete with all components and detailed step-by-step instructions (see chapter 10).



For building "wired" computers, and for the preassembled circuit boards included in computer kits, most of the component selection is performed on custom-made machinery. Here resistors are being removed from "tape strips" and readied for installation, which will be done with automated equipment.

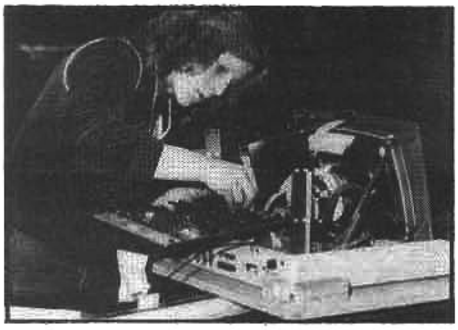
A printed circuit board is positioned on the work surface of a machine which will insert components in their proper positions on the board. Automation significantly reduces assembly errors.



At this station, all components are simultaneously soldered to the printed circuit boards. The cabinet in the foreground uses a wave-soldering technique to ensure high quality and reliability of board assemblies.

After the sub-assemblies are completed, the main assembly is begun. Here a worker guides the CRT section into an assembly support fixture. In an earlier step, the video boards and connecting cables were installed.

With the installation of a keyboard, the computer begins to take on an identifiable shape. At this point in the assembly process, temporary clamps (arrow) hold the CRT section to the support fixture.



Next, the "intelligence" is installed. The terminal logic board (shown here) goes in first, followed by the CPU board. The boards have already passed a 48-hour burn-in test.

With all of the internal components in place, the power is switched on for the first time. The completed computers are then moved to test and alignment areas where the video displays are adjusted.

Every completed computer receives a 48-hour burn-in test—a series of programs designed to rigorously exercise every aspect of performance and to identify any imperfect components.

After the 48-hour board test and the 48-hour assembly test, each computer is individually inspected and hand-tested for a final time. After it receives an "OK," the computer is packaged and readied for shipment.

A Heath manager recalls that when Zenith bought Heath, there was not enough room at the assembly plant to produce 100 units a day. Now there are new facilities and equipment—with substantial expansion still underway—that can manipulate and automatically test 500 circuit boards at a time. Boards that fail this extensive testing are repaired, then fed back into the test procedure. Boards that pass testing are used in kits or sent to the production assembly line.

Assembled Zenith/Heath computers are left running for 48 hours, continuously—a period called the "burn-in." During that time, they are subjected to internal tests of display and data handling. The test routines are stored on special floppy disks that Zenith/Heath service technicians also use to field-test customer machines.

After the burn-in and checkout is complete, the computers are packaged for shipment; however, some are pulled off the line for quality inspection. In addition, a few are later removed from their shipping cartons to check on packing procedures, materials, and handling. Some units are actually tested to destruction—that is, they are simply turned on and operated continuously. Most will run virtually indefinitely; they just do not fail.

The factory's commitment to quality in manufacturing has helped build a loyal base of customers. Those people, in turn, have helped Zenith and Heath develop a wide range of programs to run on their computers.

THE SOFTWARE DILEMMA

A group of instructions that a computer can understand and execute is called a *program*, or in computer jargon it is generally called *software*. It is software that makes a computer more than simply a bunch of electrical connections, just as recorded music is what makes a phonograph more than a piece of furniture.

In the beginning, Heath decided not to supply any software for its computers. This decision seemed to suit the market at the time; hobbyists and kitbuilders enjoyed tinkering and exploring the limits of their machines. Many of them delighted in learning to write in computer languages, playing games they had created themselves, or taking occasional excursions into more sophisticated or business-oriented programs. For all practical purposes this seemed like a reasonable arrangement; Heath could supply the hardware and the customer would have to develop the software.

"Heath vowed not to be in the software business, beyond our Basic-language programming course," says Jim Blake wistfully. He is now Zenith's software project manager, but for many years he was the company's link with the Heath Users' Group (HUG)—a loosely knit "club" of Heath computer users that had been established with minimal support from Heath to provide a forum for exchanging software and technical information. "We had always been a customer-oriented company, so in the early days we actively encouraged our customers to write software for their H-8s and swap it among themselves."

This gathering of a local Zenith/Heath users' group is typical of many throughout the U.S. and around the world. Typically, a guest speaker may be invited to give a technical presentation, or members may just trade shop talk.

However, the explosion in hardware technology made microcomputers readily available to users who did not want to painstakingly develop their own programs. This steadily increasing demand for "canned" software began to conflict with the Heath do-it-yourself approach, and eventually Heath began receiving substantial pressure from users to correct this incongruous shortfall.

HEATH USERS RESPOND

Blake admires the Heath customers' loyalty, but he laments what it produced. "Our customers were reading the computer magazines and knew that some other manufacturers were providing software support for their machines. So many of them just waited for us or HUG to come out with programs of our own. However we didn't have the resources to write our own software to run on HDOS—professional software jockeys are $30-to-40,000-a-year folks."

The pressure from users was unrelenting, and finally Heath directed Blake to set up a company project, called Softstuff, to pay a fee or royalty to anyone who would contribute original software. This was an incentive for independent software developers to produce high-quality programs that Heath could market to answer the demand from both beginners and advanced users for applications software.

Many of the initial programs came from HUG members. "Some of those HUG guys were pretty smart," Blake recalls, "and we offered their programs to more than just other HUG members; we sold them to anybody who wanted them. Within 18 months software became Heath's number three seller, after the hardware itself and the educational materials like the Basic-language course.

"This helped considerably, but being married to our own unique disk operating system was still a problem," continues Blake. "At first, our little HDOS-oriented marketplace was comfortable and we resisted change—because using a non-Heath operating system, like CP/M, required modifications to the hardware. But eventually we realized that we had to consider the possibility of using another operating system."

HDOS AND CP/M

An *applications program* is a program designed to do a specific user-defined task or set of tasks (e.g., a word processing program, an accounting program, etc.). However, an applications program is not universal—it must be designed to coordinate with a specific operating system (see chapter 5). And to a large extent the operating system is tied to the hardware, particularly the type of microprocessor and the disk-controller (the hardware interface between the CPU and the disk drives). Consequently, Heath's early commitment to HDOS and their subsequent incapacity to produce applications software was severely restricting the overall "usefulness" of Heath computers—especially to business and professional users.

HDOS is an excellent operating system, and at the user level it is more "friendly" and forgiving of mistakes than many operating systems. This can be a boon to novice users, but the hardware specificity of HDOS limits the appeal for professional programmers to produce new software that is compatible with it. HDOS is considered to be no more obstreperous than any other operating system, but producing any computer applications program requires a dedication of considerable time and effort to learn the peculiarities of the instruction or command structure—the specific internal programmable manipulations—that the computer processor can understand and the hardware can perform. As a result, after the programmer has made the considerable commitment to develop a program that is compatible with a specific DOS, it can only be used on a computer that uses this same operating system—a limitation that applies to nearly all operating systems. In other words, if a programmer writes an HDOS-compatible program, it can be used with only one brand of hardware: Zenith/Heath.

The development in 1975 of CP/M—Control Program for Microcomputers—by Gary Kildall (the software engineer who would subsequently become the founding force behind Digital Research) would change all of this. CP/M has a tremendous advantage over most other operating systems: to a high degree—though not totally—it is independent of hardware. (CP/M accomplishes this through a feature that allows it to be "configured" for different types of hardware: see chapter 5.)

CP/M's machine-independence allows it to work on virtually any make or brand of microcomputer that uses a

The Heath Users' Group (HUG) provides a wealth of information. Members may purchase software from a library of low-cost programs, and they receive an informative magazine called REMark.

Z80 or 8080 microprocessor, and programs written to coordinate with CP/M can likewise be used on these many different computers. This is important for a software designer because his programs can be sold to any computer owner who is able to use CP/M on his computer. It is important to the user because the attractive universal nature of CP/M has resulted in a substantial pool of software from which to choose: consequently, CP/M has rapidly become one of the most prevalent microcomputer operating systems.

At Heath there were some early advocates of CP/M, but the first commercial sales of CP/M for Zenith/Heath computers came from outside the company. A few independent dealers, who had the engineering know-how, modified the existing Z/H-89 hardware and began selling CP/M-compatible '89s with great success.

As CP/M continued to emerge as the prominent Z80-based operating system, Zenith and Heath made an important decision: they abandoned their strict dedication to HDOS and began selling Z/H-89s that could be used with either CP/M or HDOS. This greatly extended the practical applications of the Z/H-89, as it could now be used with the numerous CP/M programs that had been developed by software professionals who appreciated the broad appeal of CP/M.

Today, many users remain loyal to HDOS, and some dedicated programmers continue to write software for it. (There is currently a large pool of quality HDOS software: see chapter 6). HDOS is still available as an optional operating system for the Z/H-89 and very useful in many applications (particularly diagnostic work on the Z/H-89 hardware), but CP/M is now considered the primary operating system for the Z/H-89. And a version of CP/M is one of the primary operating systems provided for the Z/H-100.

Jim Blake is philosophical about the change, "Operating systems should be transparent to the user, anyway. They should help him solve his problem without calling attention to themselves. The user shouldn't have to be a software jock. And software isn't a static, unchanging thing. Software is like ice cream: if you don't do something with it, it melts away."

CURRENT SOFTWARE DEVELOPMENT

The Heath Users Group is now much more than just a software exchange. HUG publishes a user-oriented magazine called "REMark." HUG also sponsors an electronic bulletin board on a major consumer-oriented telephone network (see chapter 3), and helps people set up and support local HUG clubs around the world. And HUG provides various "user services." According to Gerry Kabelman, one of HUG's professional software developers, "We offer technical support and advice, and answer a lot of questions about hardware and software."

The tradition of involving users runs deep in the company, because it runs deep in the country. "Americans are, by their nature, tinkerers," says Tom Dornback, Zenith's vice president for software development. Under his direction, Zenith has extended the concept of a network of "HUGgies" to a network of outside testers. "We look for people who have expertise in one particular field: word processing, math, education, and so on," he says. "Then we send them new software and they give it a full-time test. This is not a strenuous test under actual working conditions, but these people have nothing but malicious intent—they get paid by how well they break the product! In this way we are able to work out the bugs and sell a reliable product."

BUSS #51

The Independent Newsletter of Heath Co. Computers

BUSS is an independent newsletter dedicated entirely to Zenith/Heath hardware and software. This feisty publication is considered "required reading" by most enthusiasts, and is always packed with helpful, timely, easy-to-understand information. This issue detailing the new Z/H-100 system was mailed just hours after the first public showing.

At Zenith, there are also staff members who initially review all new software written and submitted by outside suppliers to see if Zenith should buy the product and offer it for sale to their customers. Each product goes through an intensive evaluation before Zenith finally approves it for use on its computers. "The testers give us very critical reports," says Dornback, "but we have to make sure that the final version works the first time, and every time a user tries it. We don't want any recalls."

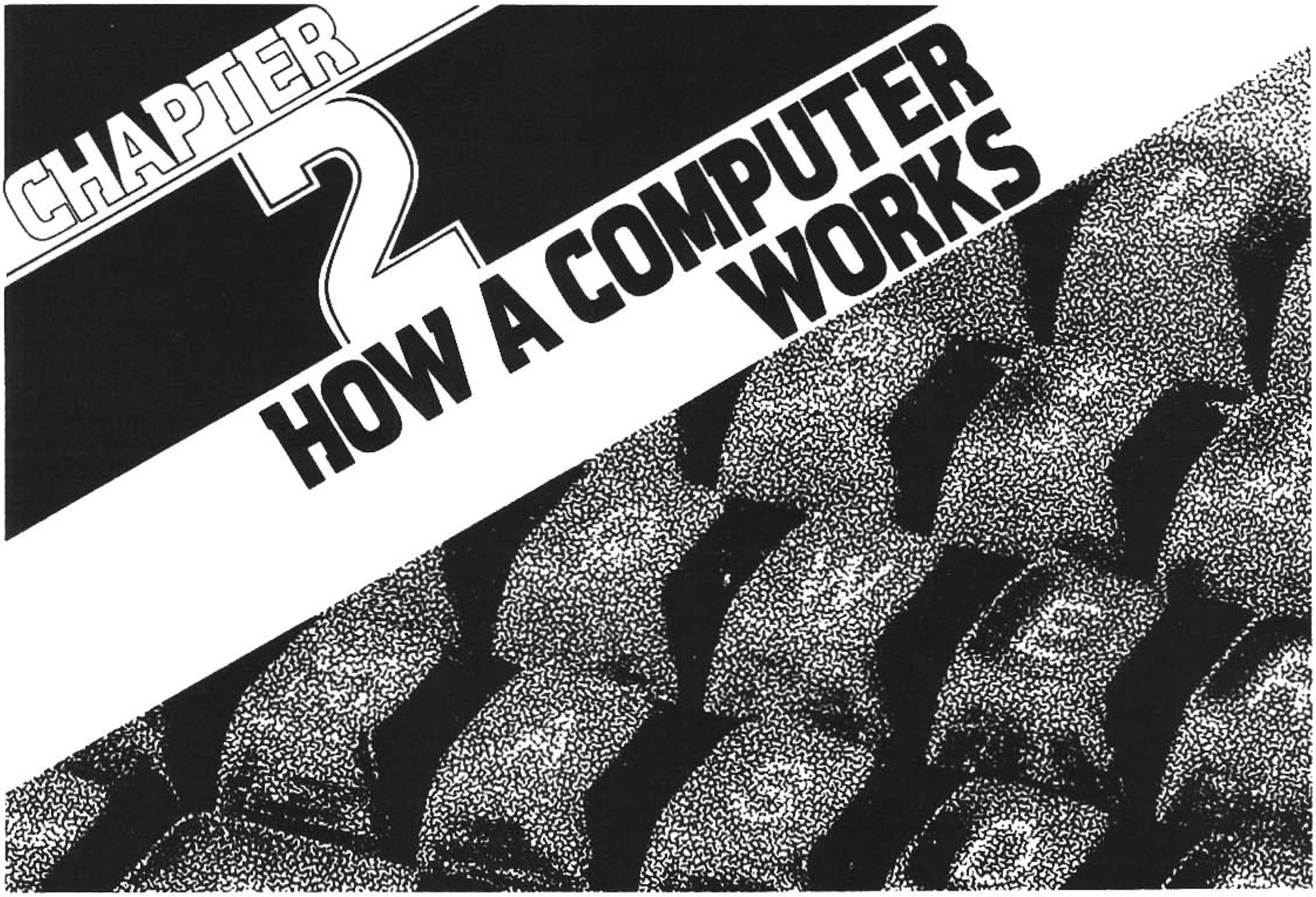
Besides the software, the Heath instruction manuals are subjected to the same process. "The best manual would be no manual at all," he says, philosophically. "Our long-term objective is to develop products that teach the users by themselves, perhaps with on-screen graphics and pictures. Eventually, voice synthesis may be part of the program—announcing to the user that the software has caught a user error. Right now, the best example of self-teaching software is SuperCalc (see chapter 6), where a simple keystroke gets you a help-menu (a listing on the video screen that prompts the user for correct usage of the program) any time you need it, without interrupting the rest of the job."

With this level of commitment it seems certain that Heath and Zenith are completely aware of the importance of software support for their machines. Selling software has become extremely profitable for the company; and throughout the computer industry it is predicted that software sales will soon rival the annual multi-billion-dollar hardware sales. The reason is simple: sooner or later hardware will saturate the marketplace, yet people will continue to need new and more efficient software. In one analogy, it is like a customer paying one time for a stereo system and hundreds of times for records; or buying a safety razor and having to fill it with a lifetime supply of new blades.

This also means there is sure to be continued demand in the forseeable future for software that makes use of existing hardware. The door will remain open for any bright, ambitious programmer and entrepreneur to design, test and market innovative software. (The more computers that this software will accommodate, the greater the potential for sales; and considering the current state of the market, this means that the software should run under the CP/M operating system.)

Without the active network of users groups and a worldwide base of customers, Zenith and Heath could not have learned nearly as much about what people want in a computer. The latest developments in hardware and software indicate that the company has given considerable thought to the feedback provided by these groups.

CHAPTER 2 HOW A COMPUTER WORKS



A computer is a complex machine, but it operates on simple principles. It isn't necessary for the user to understand every detail, but a knowledge of overall concepts is essential. To help the beginner, Zenith and Heath sell excellent self-education courses. One of these programs, "Computer Concepts," provides helpful instructions for the small-businessperson who faces the problems of implementing a new computer or computer service.

GENERAL STRUCTURE AND FUNCTIONING OF A COMPUTER

In the preceding chapter we touched on some of the general nomenclature of a microcomputer. In this chapter we will try to give the beginning user an elementary understanding of general microcomputer functioning and terminology. It is not our intent, nor is it possible in the space available, to provide the reader with more than a limited description of these matters.

To keep the discussions reasonably simple, we will in many cases be forced to draw wide-ranging generalizations and summarize complex technical issues with statements that may seem absurdly brief to the reader who already has some knowledge of these matters. We hope, however, that the average reader will gain a useful understanding of the important inner workings of a Zenith/Heath microcomputer.

To successfully use a computer it is not necessary to know every minute detail of how it works. Hardware and software designers spend considerable effort to make their products "user friendly;" as a result, a neophyte can learn to "push the right buttons" very quickly. Undoubtedly, some guidance from an experienced user will be helpful, but with well-documented software it only takes a few hours to learn the important basics. On the other hand, without some understanding of the internal structure and functioning, the user may never feel fully at ease with a computer.

A microcomputer is a complex machine with many interdependent elements, but it operates in a relatively simple manner. It is important for the beginner to form a clear and accurate picture of overall operating concepts and avoid technical details. This is not easy. The computer world is awash with jargon, but it is generally not necessary for the day-to-day user to know what all of this tech-speak means. Once the fundamental concepts are firmly understood, the user can wisely evaluate software and consider the real value of an upgrade or accessory, and in those rare instances when things don't work the way the book says, some elementary knowledge and an analytical approach may rescue the user from an otherwise hopeless situation.

For those who have an urge to know more computer science, Zenith and Heath market several excellent self-teaching courses of instruction. These range from studies in elementary computer functioning to specialized courses in computer programming. Many advanced reference books are also available at computer stores, book stores and local libraries. There is certainly no lack of information about computers; the dedicated user need only spend the time and effort necessary to learn as much as he or she wishes.

HOW IT WORKS

Among those who have absolutely no knowledge of computers, there is a popular belief that a computer "thinks." All the user has to do is push a few buttons and the computer can answer any question. Sadly, this is not true.

In more accurate terms, *a computer can store information and manipulate or retrieve this information.* The user must put information—be it numbers, words or some other type of symbols—into the computer, and later this information can be called back in its original form, or through a series of manipulations this information may be restructured and returned to the user in another useful form.

The information inside a computer is often referred to as data. The entry, manipulation and retrieval of data is controlled within the computer by the microprocessor. The processor is directed to perform these functions by a set of rigidly structured instructions. A group of instructions designed to accomplish a specific task or set of tasks is called a *program*; the more commonly used general term is *software*.

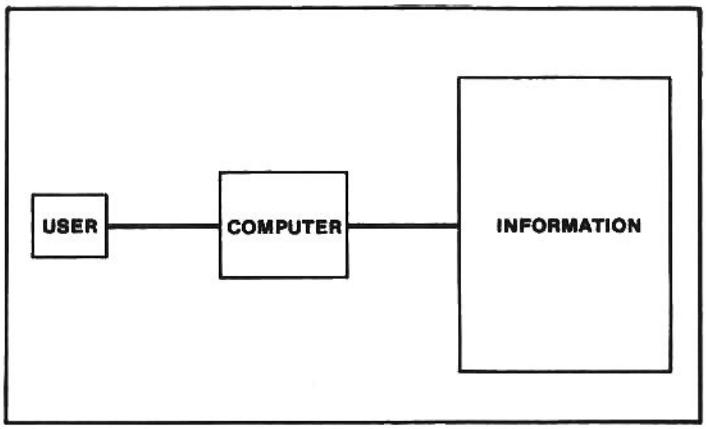
The instructions are written in what is called *machine code*. This code is usually expressed in writing as a series of 1s and 0s. Each 1 or 0 is referred to generally as a *bit*. A 1 may be called a *high bit* and a 0 may be called a *low bit*. (A high bit indicates an "on" or "positive" electrical state and a low bit indicates an "off" or "negative" electrical state.) Because there are only two numbers (electrical states) in this code, it is sometimes called a *binary code*.

The individual bits are generally formed into logical groupings of eight. Each of these groupings is called a *byte*. A byte is the most fundamental meaningful segment of computer code. Every letter, number or symbol of our written language is represented by a unique 8-bit byte. Each byte in a computer may represent a piece of data or it may be part of a program instruction.

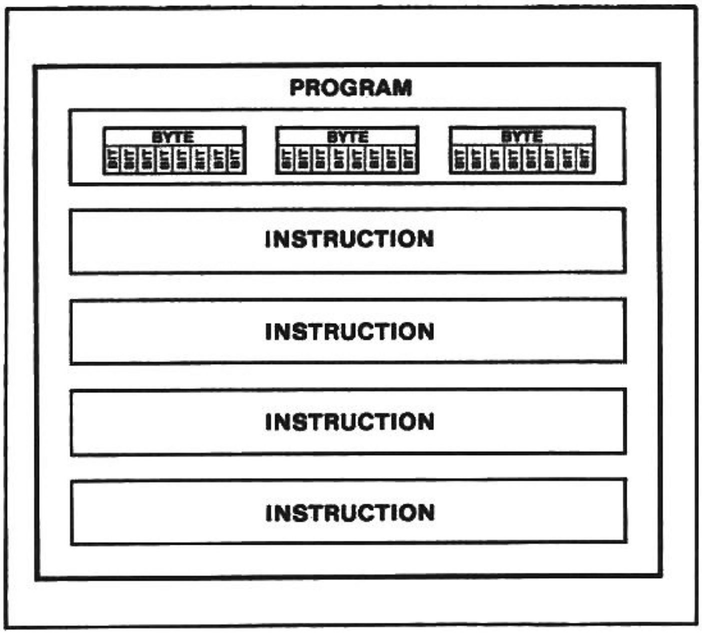
Virtually everything that happens within a computer is controlled by a program of one form or another. By "loading a program" into the computer and then issuing specific commands to that program, the user causes hundreds of instructions to be directed to the microprocessor, which can read and execute these machine-level tasks at the rate of hundreds of thousands of instructions per second.

The limit to what a computer can do is determined by the capabilities of the hardware and the design of the software in use at the time. If the user desires a specific result, but the program structure does not allow for the internal manipulations needed to gain that result, the user will not achieve his goal. The user can, however, achieve different purposes by choosing from many different programs. Generally only one program can be used at a time (though complex software can link program "modules" together), But the abilities of the computer are determined by the number and kind of programs the user can obtain or create.

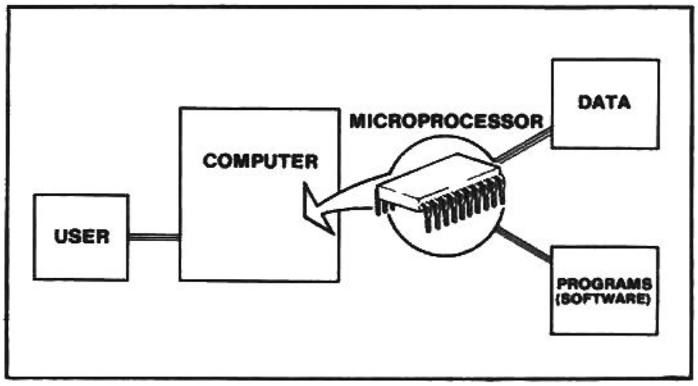
In a common analogy, a computer can be viewed like an office filing cabinet. The user puts data into the computer, like a secretary puts documents into a filing cabinet. This



Contrary to popular belief, a computer cannot "think." A microcomputer is capable of performing very complex tasks, but all of these tasks are based on information—of one form or another—that the user has previously recorded into the machine. The value of a computer depends on how effectively the user enters, manipulates and retrieves data.



All information—programs or data—inside a computer is made up of tiny electrical pulses, called bits. The elemental bits are combined in groups of 8 to form bytes. Each byte represents one of 128 possible characters: letters, numbers or symbols. Several bytes may be grouped together to form one instruction of a program, and several instructions combine to make a complete program.



The information in a computer can be divided into two general categories: data and programs. Data is simply words, numbers or symbols that a user wishes to store and manipulate within the computer. The manipulation of data inside the computer is controlled by extremely simple instructions. A group of these instructions, designed to perform a specific function or series of functions, is called a program or (in a general sense) software.

data can be withdrawn at a later date, like a secretary takes documents out of a filing cabinet. In a computer, the filing, retrieval and various data manipulations are performed by the programs—under the direction of the user. To a large extent, the effectiveness of a computer is determined by the manner in which the data is stored and retrieved by the program(s), just as the effectiveness of a filing cabinet is often determined by the ingenuity of the secretary.

In the final analysis, the value of a computer lies not in its complexity but in its speed. A computer can only execute simple program instructions, but the computer can execute these instructions so quickly and efficiently that the end results often appear complex. It can, therefore, accomplish in minutes what might otherwise take the user hours or even days to complete.

THE HARDWARE

There are thousands of individual pieces of hardware in a Zenith/Heath microcomputer. These pieces are put together to form the several major components of the machine. For the purposes of this discussion we are not going to study every component or subassembly in detail. Instead, we are going to limit our discussion to four major constituents: the *terminal assembly*, the central processing unit or *CPU*, the random-access memory or *RAM*, and the onboard *disk drive*. (These are not precise entities and they often operate with the aid of supporting elements, but for describing the general operation of a microcomputer, these groupings will suffice.)

THE TERMINAL

A *terminal* is a communications device. In the all-in-one Z/H-89 the terminal assembly is incorporated in the computer enclosure. It is dedicated to transmitting communications between the user and the computer central processing unit (see below, CPU).

(The same terminal hardware that is in the Z/H-89 is sold by Zenith and Heath as a separate stand-alone unit, called the Z/H-19 terminal. It is not a computer, it is only a device to communicate with a central processing unit and may be used as a remote terminal in certain large commercial or multi-user systems or as a terminal for a small system that has a separate CPU assembly.)

The two major components of the terminal assembly are the *keyboard* and the television-like cathode ray tube, or *CRT*, display. The user issues commands to the computer by pressing the keys on the keyboard. The terminal "encodes" these commands and transmits them to the CPU.

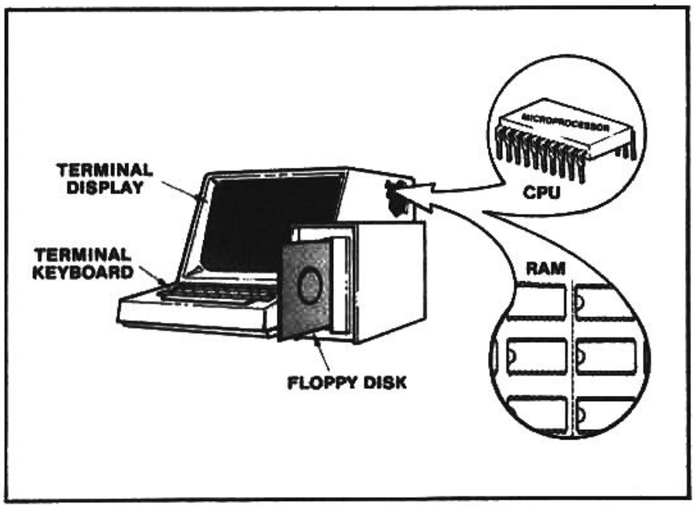
The video display gives visual prompting to the user. It displays the program commands or data as the user types on the keyboard, and it displays messages or data that is transmitted from the terminal to the CPU. It also returns and displays on the CRT messages or data that the CPU wishes to communicate to the user.

THE CPU

The main "computing" element of a computer is the *central processing unit*, often called the *CPU*. Several components make up the CPU but the main constituent is the *microprocessor*, sometimes just called the processor or the *MPU* (for microprocessor unit). It is, in effect, the central manager of all that happens inside the machine. For periods of time the microprocessor may turn control of various jobs over to other components but it is, nonetheless, the main operating element within the computer. It is the heart or, if you prefer, the brain of the machine.

In the all-in-one Z/H-89 the CPU is incorporated in the same enclosure as the terminal section and the rest of the main components. In systems that are not designed with the all-in-one principle, the main components may be in separate enclosures. In either case, the CPU element may be referred to loosely as "the computer" (as opposed to the instance when this same term is used to refer to the entire computer system, which would be composed of the CPU, terminal, disk drive, and support elements).

The microprocessor used in the CPU of the Z/H-89 is the Zilog Z80. There are other types of microprocessors.



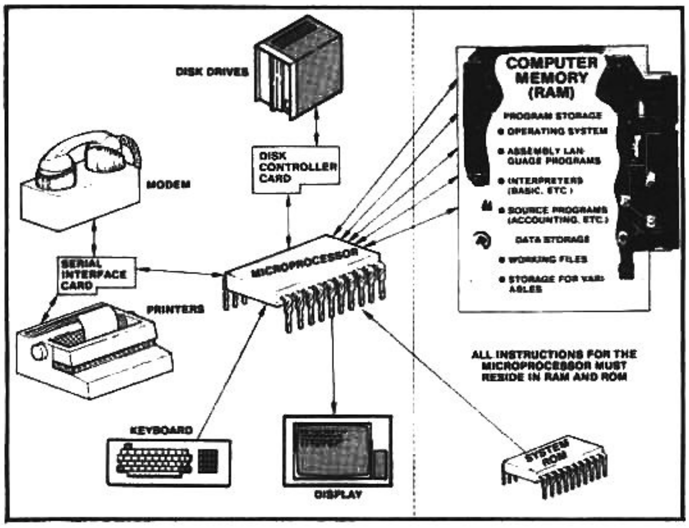
Though there are hundreds of components inside a microcomputer, the major groups (sub-assemblies) are: the terminal section, the central processing unit (CPU), the random-access memory (RAM) and the floppy disk. The user communicates with the computer through the terminal section; the user enters data and commands on the keyboard and the computer delivers information to the user via the CRT display. The CPU is the heart of the computer—all internal functions are initiated by the microprocessor, according to software (or firmware) instructions. The RAM is a dynamic work area where program instructons are stored prior to execution by the CPU and where all data usually resides while it is being manipulated. The floppy disk provides permanent storage for program and data files.

Each type of processor understands a specific set of instructions, and one type is generally not interchangeable with another. As we shall see later, only software designed to work with Zilog Z80 or the Intel 8080 (a predecessor of the Z80) microprocessors can be used in Zenith/Heath systems.

The microprocessor is also tied directly to the primary piece of computer software—the *operating system*. In a sense, the operating system is a master control program. It interacts with the microprocessor to perform the most elementary user-oriented machine functions, such as loading an applications program.

SYSTEM AND APPLICATIONS SOFTWARE

At this point, it may be helpful to interrupt our hardware discussion for a brief look at the important role of software in computer functioning. *An operating system is a program (or a collection of programs) that allows the user to control the elementary functioning of the computer.* As soon as a computer is switched on, the operating system must be loaded into the computer memory from some storage source (see: The Disk Drive). Instructions for this transfer are usually contained in the *System Monitor Program*, which is permanently recorded in a *ROM* chip (read-only memory—the processor can only read and execute the limited number of instructions contained in a ROM, but the processor cannot alter the contents of a ROM).



The microprocessor controls all of the primary computer functions. These functions are defined and executed by program instructions received from the random-access memory (RAM) or the read-only memory (ROM). Many types of hardware make the computer system useful (typical examples are represented to the left of the dotted line). However, all of the interaction (data transfer) between this "peripheral equipment" is controlled by the processor, acting on instructions from programs residing in RAM. Almost any type or brand of hardware can be used with the processor, but the software (in RAM) used to "drive" (control) the equipment must be designed to send correct instructions to the processor. That is, the processor must be able to communicate with the peripheral hardware. Note that communication with the disk drive is established through a special piece of hardware called the disk-controller card (interface) and that communication with modems and printers is established through the serial (or parallel) ports. Note: this drawing is not a totally accurate depiction of the complex communication channels inside a microcomputer system, but it should help the beginner visualize the importance of the microprocessor as the primary link between the user, the software, and the hardware.

When the operating system is in RAM, it allows the user to control overall machine-oriented functions. But to make the computer perform a specific task, such as writing a letter, the user instructs the operating system to load into main memory (from a storage source) a separate *applications program* designed to accomplish this particular type of job—in this case a word processing program.

The user can only "use" an applications program when it is in memory (it is loaded to a location separate from the operating system) and after an applications program is in memory, *control of the computer is generally turned over to the applications program*, and the operating system stands aside (though the applications program may call on the operating system, if necessary, to perform those tasks that the operating system normally handles). The user controls the computer by sending commands to the applications program. However, as soon as the specific task is completed, the user signals the applications program that he wants to return to the operating system. The link to the applications program is then broken and the user is once again directly in contact with the operating system (actually this is a simplification, but for our purposes it will suffice).

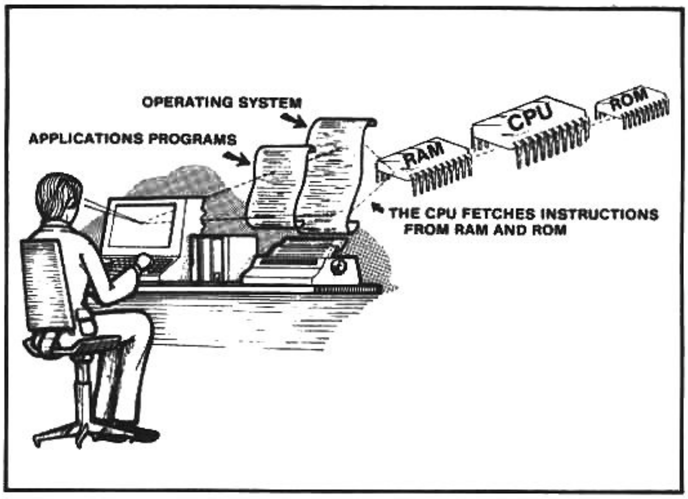
In effect, the user is in contact with the computer on one of two levels. Overall, the user manages machine-oriented tasks (e.g., transferring, duplicating, or erasing programs or data, etc.) with the operating system. On a specific level, the user accomplishes specialized tasks with an applications program. Generally, the user is active on one level or the other—*either he is communicating directly with the operating system or with an applications program.*

As a result of this interrelationship, the processor is tied to the operating system and the operating system is tied to the program. Consequently, when a programmer develops an applications program, he must design it to work with a specific microprocessor and a specific operating system. Thus, *the only applications software that can be used with Zenith/Heath computers is that which has been designed (or modified) specifically for use with the operating systems that can be implemented on Zenith/Heath hardware.*

THE RAM

Next to the microprocessor, the *random-access memory* or *RAM* is probably the most important hardware component of a microcomputer. The term "memory" is often used to refer to the RAM. A memory is a place where programs or data are stored. It is called random-access because, at the direction of the processor, *program instructions and data can be randomly stored and retrieved from any legitimate location within RAM.* All of this happens in a very structured manner. When a program is being executed by the processor, the instructions are sequentially retrieved by the processor and acted upon. Or, if the processor is accessing a data file, the processor (under software control) places the data into a specific location in the RAM; the processor keeps constant track of every byte that is stored there; and it can randomly (as directed by the software) retrieve, manipulate or erase any portion of the stored data.

All of this activity is directed toward fulfilling the user's needs. At any one time the RAM may be filled with thousands of program instructions and pieces of data, all working in unison with the CPU to accomplish a complex



The disk operating system (DOS) is the most important program in a floppy-disk-based computer system. The DOS may be viewed as the "master control program." It allows the user to manipulate applications software and data files between the disk and the RAM.

When the computer is first switched on, a copy of the DOS must be booted (loaded) from permanent storage on the disk into RAM. The user does this by issuing the "Boot" command, which causes a small program (the boot instructions) permanently embedded in the system ROM to be executed by the CPU. When the process is completed, the operating system is copied into RAM and computer control is turned over to the DOS.

The user can issue a variety of commands to the DOS, but one of the most important functions is to load an applications program into the RAM. After the user instructs the DOS to load and execute a program, the DOS is transparent (in effect) and the user communicates directly with the applications program to accomplish a specific task (e.g., word processing, accounting, programming, etc.). During this time, however, the operating system is still in RAM and the applications program may occasionally call on the DOS to perform disk-oriented tasks (i.e., loading data to and from the disk, viewing disk directories, etc.).

task. But in a few seconds the user can wipe the RAM clean and refill it with a new contingent of program instructions or data, designed to do something entirely different.

This concept is what makes modern computers so versatile. A microcomputer can be used to write letters, store recipes, perform mathematical calculations, balance a checkbook, play chess, inventory and evaluate a stamp or coin collection, simulate an airplane landing on an airfield, turn the lights and heating on or off throughout a house, and much more. But each of these jobs must be direct by an applications program.

If a computer had to permanently retain every program it was capable of executing, it would need a huge memory bank, but the concept of random-access memory sidesteps this limitation. Hundreds, or even thousands, of programs can be readily stored on lightweight floppy disks, and when the user wishes to perform a specific task with the computer, the applicable program is loaded from the disk into the RAM. The program is executed—with or without additional input from the user, as required by the design of the program—and when the job is done, the RAM is freed for additional use. Then, another job can be selected by the user, the appropriate program loaded from the floppy disk to the RAM, and so on. In this way the user can make his computer do virtually anything for which a program can be designed, yet the entire machine remains compact and reasonably inexpensive.

THE DISK DRIVE

The development of compact, low-cost floppy disk drives was one of the technological breakthroughs that made powerful microcomputers possible. Large amounts of data can be accurately and efficiently retrieved or stored upon these small, inexpensive oxide-coated disks. As an example, it would be possible to store the entire text of this book on one double-sided, high-density 5.25-inch floppy disk.

The floppy disk is the permanent repository for the programs and data used in a microcomputer. When a microcomputer is turned off, all but the most elementary operational instructions are cleared from the machine. When it is turned back on, it has enough permanent "intelligence" in the System Monitor ROM to load a copy of the operating system from the floppy disk to RAM. Note that we said "a copy," in fact, *a duplicate of the operating system is formed in main memory, while the original remains intact on the floppy disk* (this is true when any program is loaded from the disk).

THE RELATIONSHIP BETWEEN RAM AND THE FLOPPY DISK

The relationship between the floppy disk and the RAM is one of the most important microcomputer concepts. To understand this relationship, the user must realize that the computer accesses data in RAM in one manner, and when the same data is on the floppy disk, it can only be accessed in a more restricted way.

The microprocessor can *individually* access every byte that is in the user portion of the RAM. This access is controlled by the structure of the program that is sending instructions to the processor. For instance, if the user is writing a letter with a word processing program, it is easy to write sentences and paragraphs, erase words, split words apart, insert new words and perform complex text-handling tasks. This is possible because the user—through interaction with the applications program and the microprocessor—can instantly manipulate any portion of the data (in this case, the words of the text) stored in the RAM and displayed on the screen for the user to see.

This is not possible when data is stored on the disk. When data is recorded on the disk, it can only be accessed as a contiguate grouping. The user—through interaction with the processor and the operating system—can only store or retrieve a grouping of data (called a data file) in its entirety. It is not possible to go into a disk file and manipulate a few pieces of data.

Initially, it may be easier to view the RAM as a "working area" in the computer and the floppy disk as a "storage area." When the user wishes to work with a program or a data file, it must be in RAM (working area). The user must transfer a copy of the program or data file from the floppy disk (storage area) to the RAM before he can actively manipulate the contents. In the example of the word processing program, if the user wishes to create a new document (e.g., write a letter), he creates it in the RAM with the aid of the word processing program; to keep the document as a permanent record, he must transfer it to the floppy disk, where it is retained as a disk file with a specific name.

This sounds simple, but we must add an important clarification: the user's access to the RAM is rigidly restricted. Generally, the only way the user can gain access to the RAM space is through an applications program. At any one time, several different programs or files may be resident in the RAM. When the operating system is loaded, the RAM is "organized" and specific space is allocated for the disk operating system and other operational software. At the same time a "user space" is allocated to accept programs and files that the user transfers into the RAM. All programs must coordinate with the operating system so they are delivered to the proper RAM location, otherwise one program could easily "crash" into another and create total mayhem.

From the foregoing we can also see that the "size" of the RAM and floppy disk are important. When the RAM is larger, more space is allocated for the user, and bigger data files can be put into this working space. When the floppy disk size is larger, the user can put many more programs or files on each disk.

The largest amount of RAM space that can be used with microprocessors like the Z80 is "64K." This means that 65,535 bytes of data—characters, numbers, symbols, etc.—can be stored in the RAM (for ease of reference the number is rounded to 64K). It is the maximum possible RAM size because the Z80 microprocessor cannot address (keep track of) more data than this at one time. Of course, the computer can work with less RAM, but the user is then further limited in the amount of data that can be stored in the RAM.

The 5.25-inch floppy disk is, like the RAM, a highly organized and structured storage area. There are different *disk data formats*—the disk format is the specific way data is recorded and organized on the disk surface—and some formats will allow a standard 5.25-inch disk to hold more data. However, there are restrictions on the type of disk format that can be used, because the format is specified by the operating system, the disk drive and the disk-controller hardware. The user can upgrade the system to accept a format that allows more storage, but the hardware associcated with greater data density is almost always more expensive (see chapter 7 for disk upgrades and formats).

TYPES OF INFORMATION

Up to this point we have referred to computer information as "data," "programs" or "files." Before we continue, it will help if the reader understands the precise meaning of these terms.

In a general sense, *data* refers to all of the electrically-encoded information stored inside a computer. As human beings we communicate with words and numbers. Inside of a computer all letters, numbers and symbols (data) are represented by electrical pulses (bits) that are organized and transmitted in precise groups (bytes).

A collection of bytes that form instructions to make the computer perform specific functions is generally called a *program*. Programs may be further classified according to their purpose. A program that controls internal functioning of the computer hardware, such as a portion of the disk operating system, may be called a *system program*. If the program is designed to perform a user-designated task, it may be called an *applications program*.

If a collection of data does not perform any sort of computer function, it is often called a *data file*. For instance, The user may use a word processing software (an application program) to create a document (data file), such as a letter. A user must use an applications program or a system program to manipulate a data file or control the functioning of the computer. However, a data file is a collection of words or numbers that cannot perform any computer manipulations whatsoever: a data file is just a reservoir of information.

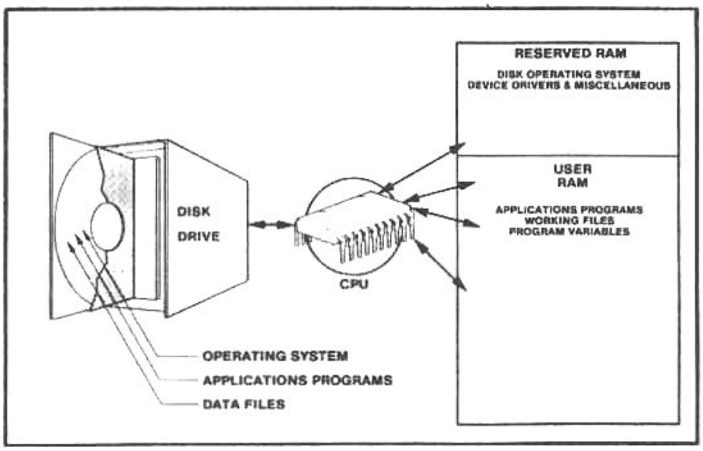
In another sense, a file—either a program or a data file—when it is stored on the floppy disk may be referred to as a *disk file*. In the same way, a file in RAM may be referred to as a *RAM file*.

PUTTING IT ALL TOGETHER

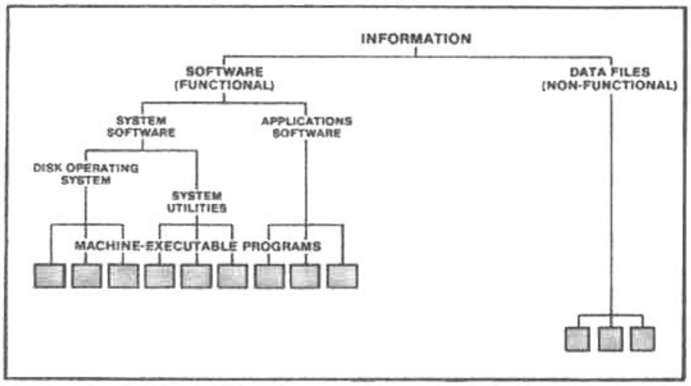
Earlier we likened a computer to a filing cabinet. This is a useful comparison but it may now be helpful to look at this analogy in greater detail. Let's say that the computer user is like the president of a big company. In many ways the user will control the complex activities of the computer just as the president directs the activity inside a big company.

When our company president arrives at the office in the morning, he or she calls the administrative manager over the office intercom, and when the manager responds, the president asks, "come into my office, we have work to do." In a similar way, the computer user turns on his computer, and when the computer responds with a prompt on the display, the user says, in effect, "load the operating system into RAM." (Note: during this discussion the interaction betwen the user and the computer is represented in a conversational way, in real life this discourse is an exchange of brief messages and commands; see chapter 5.)

In this analogy the administrative manager is similar to the operating system and the executive's office is similar to the RAM memory inside the computer. The executive will control the operation of his business through an intermediary, the administrative manager, and the user controls the overall operation of his computer through a



The transfer of information from the floppy disk to the system RAM is one of the most important computer functions. This is accomplished by the CPU, acting on instructions from the disk-operating system. After the user has booted the DOS into RAM (accomplished with the boot instruction in the system ROM) and is in contact with the DOS, the user issues a command to load a program from the disk to the RAM. The DOS sends the proper instructions to the CPU, which copies the program from the disk into the space allocated in RAM for user programs. Some space is always reserved in RAM for the DOS and "housekeeping" system software. The system RAM is called "dynamic" because the information stored there is not permanently recorded. The disk, however, is used for permanent storage. The CPU cannot access individual bytes on a disk, it can only cause a group of bytes (programs or data files) to be loaded to the RAM (from the disk) where it can then be accessed byte by byte. The information recorded on a disk will remain there when power is turned off, but all program/data bytes recorded in RAM will be lost.



It is easy to distinguish between functional programs (instructions that cause the computer to perform specific tasks) and non-functional data files (a grouping of words, number or symbols that is simply stored information). But sub-classes of software are less apparent. This chart assumes a functional hierarchy, with more complex and more important software placed higher on the chart and simple data files placed lower. (The relationship between machine-executable programs and high-level programming languages is defined in chapter 6.)

similar intermediary, the operating system.

In our office analogy the administrative manager arrives in the executive's office and says, "OK, I'm ready to work, what do you want me to do." The executive responds, "We've got many things to do. First, I want to write a letter, call a stenographer from the central office pool and wait here in the office until I need you again." In our computer scenario, the operating system is loaded into RAM and responds with a prompt that, in effect, says "OK, I'm ready to work, what next." At this time the user decides, like our company president, to write a letter, and responds by saying, "Go to the floppy disk and load a word processing program into RAM." Here, the floppy disk has been likened to a large, central office pool filled with many secretaries and job specialists.

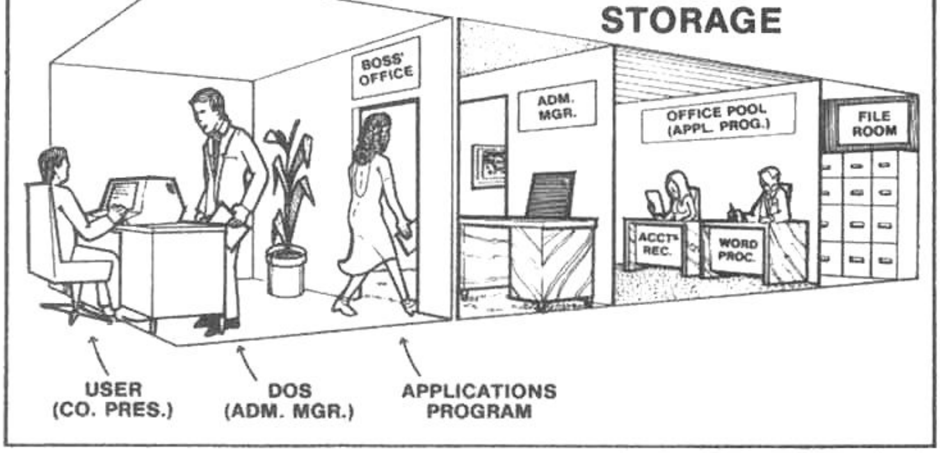
When the stenographer arrives, the administrative manager waits patiently, and the president says, "I want to write a letter to Bill Jones. I will probably have to review this document in a few days, so I want you to put a copy in the file cabinet under the index: Letter.BJ." Similarly, the computer signals the user when the word processing program is loaded, and the operating system steps aside while the user communicates directly with the program. In our example, the user wants to write a letter to Bill Jones, but he must give this document a unique name so that it does not become confused with any others that may be stored on the disk drive. He decides—according to the specific conventions governing the naming of computer files (the Zenith/Heath conventions are discussed in chapter 5)—to title the document "Letter.BJ." Here we have pointed out that anything stored on the disk drive must be uniquely identified. This naming system is similar, in our office analogy, to a structured index system for a filing cabinet.

(This brings up the question of file organization on the disk. For now we will say simply that a *directory* is created on the disk by the operating system. As files are written to the disk or removed, this directory is automatically updated. To find out what is on the disk, the user types a command to the operating system; the operating system reads the disk directory and displays a listing of the directory on the video screen.)

So, our executive dictates the letter and finishes up by saying to the stenographer, "The letter is done, put a copy in the main filing cabinet. You may now leave." He then turns to the administrative manager and says, "Next, I want to check the company financial status, send in the accountant." In the same way, the user produces a letter with the word processing program, and when he finishes, he commands, "Write the letter to the floppy disk, remove the word processing program from RAM and return control to the operating system." In effect, the word processor creates a unique file called "Letter.BJ," on the disk drive and transfers the document into this disk file. Then, the word processing program turns control of the computer back to the operating system. When the operating system signals that is is once again in control, the user commands, "I want to load an accounting program to RAM," and the system moves on to the next task.

Not every applications program will interact with the user in the exact manner described above, but in Zenith/Heath systems the relationship between the CPU, the RAM, the operating system, applications programs and the floppy disk will follow these general principles.

In chapter 5 we will look more closely at the operating system and file manipulation, but first we will examine the hardware inside the Z/H-89 and Z/H-100 computers



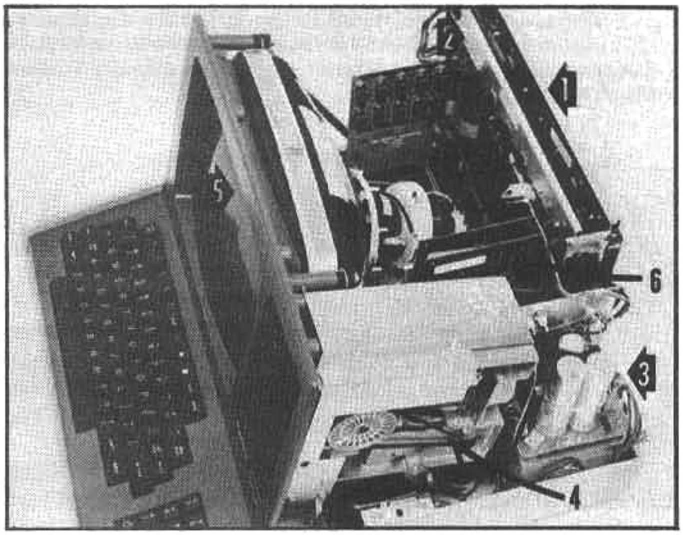
The important relationship between the disk-operating system, applications software and data files is demonstrated in this analogy. As shown here, the computer user is like the president of a large company. The user direct the activities within his computer through an intermediary, the disk operating system, just as the company president might use an administrative manager. Through the operating system, the user calls up applications programs to accomplish specific tasks (e.g., word processing, accounting, database management, etc.). Similarly, the president calls on stenographers, accountants, researchers, etc. When raw data must be stored in a computer, it is collected in a data file and permanently stored on a floppy disk. This is similar to storing letters and documents in carefully indexed filing cabinets.

CHAPTER 3 THE ALL-IN-ONE Z/H-89

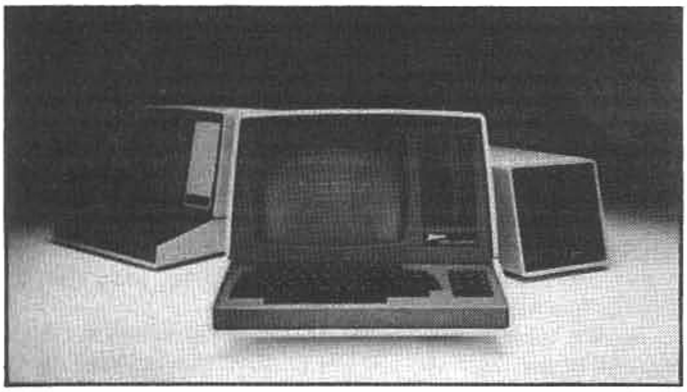
THE ALL-IN-ONE COMPUTER

Heath was among the first American microcomputer manufacturers to take an "all-in-one" approach. At the time, other makers favored enclosing the major individual components—the terminal, the CPU and the disk drive(s)—in separate cabinets. Currently the trend has shifted throughout the industry toward the all-in-one concept, and Heath continues to be a strong proponent of the compact, all-in-one microcomputer.

During the redesign of the H-9 into the H-19 terminal the designers were aware that the new design could be readily upgraded from a terminal, suitable only for communications, to a full-fledged microcomputer. By adding an input/output (I/O) device and central processing unit (CPU), the H-19 would become a complete, functional computer, and with this goal in mind, the basic enclosure was designed to accept these additional components. Space was allocated at the rear of the enclosure for the addition of a CPU circuit board, and room was provided to the right of the CRT display to provide internal mounting space and an access opening for a 5.25-inch floppy disk drive.



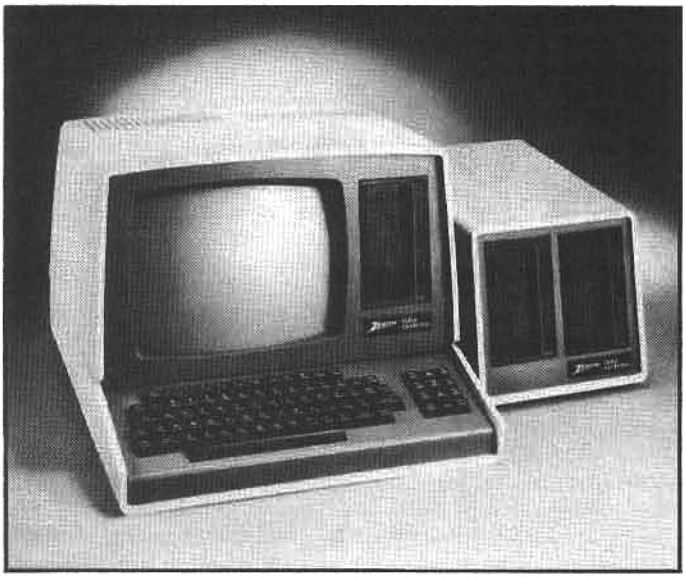
With the cover removed, the primary components of a Z/H-89 can be seen: (1) the terminal logic board (TLB), (2) the central processor unit board (CPU), (3) the power supply, (4) the disk drive (5) the CRT screen, and (6) the disk-controller card.



The Heath H-89 *all-in-one* microcomputer incorporates a fine keyboard design, a full size CRT display, a powerful Z80 microprocessor, 48 Kbytes of random-access memory and a built-in 5.25-inch floppy-disk drive—all wrapped in a compact, handsome cabinet. Both Heath and Zenith sell the basic Z/H-89, and Zenith offers an enhanced version (with a higher-capacity disk and more RAM) called the Z-90.

In 1977 the prototype all-in-one computer was completed and the first H-89 kits were available by 1978. In the ensuing years microcomputer design has developed at a whirlwind pace, yet the original Heath H-89 has consistently been upgraded to take advantage of new technology. As this is written the H-89—in its various forms and configurations—remains a powerful, versatile and extremely useful microcomputer.

In chapter 7 we will discuss many of the hardware upgrades available for the basic H-89. In chapter 5 we will discuss the evolution of CP/M as a primary operating system for the Zenith Z-90 and Z/H-89, displacing the Heath-developed HDOS operating system. (Though Heath's HDOS is still an option on all Zenith and Heath machines.) This CP/M software support is particularly significant because it gives the Z/H-89 capabilities far beyond the initial hobbyist-oriented expectations of the designers. In effect, this seemingly simple change makes the Z/H-89 microcomputer system an outstanding choice for small-business and commercial applications. This successful evolution is an unspoken tribute to the solid design of the H-89 and the ingenuity of the Heath and Zenith designers.



The Z/H-89 system readily supports multiple disk drives. Pictured here with the Z/H-89 is the Z/H-17-2 expansion, which includes two 5.25-inch drives. This three-drive system is a very useful business tool because it is more flexible than a system with just one drive.

THE Z/H-89 AND THE Z/H-90

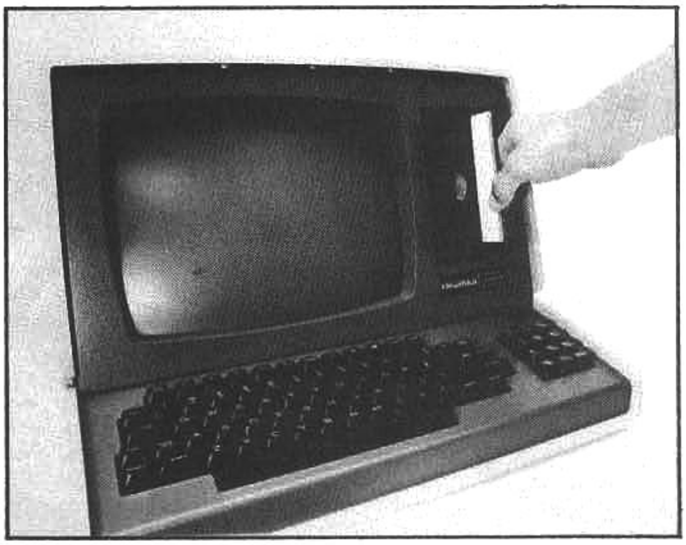
The H-89, WH-89 (assembled version of the H-89) and Z-89 are self-contained microcomputer systems. They have a central processing unit using a powerful Z80 microprocessor; a reliable 5.25-inch floppy-disk drive built by Siemens—with provisions to add two additional drives; and 48 Kbytes of random-access memory (RAM), enough to store 49,152 bytes of data. In the rear of the chassis, there are interface ports for attaching a printer and other devices.

For the first few years this "package" met most demands of the marketplace. Soon, however, customers needed more: additional disk drives, more internal RAM, and extra ports to accommodate more peripheral equipment. Zenith and Heath designers developed modifications to accommodate most of these features, and eventually, the most-requested upgrade features were added to the basic Z/H-89 and the upgraded computer was designated the Z/H-90.

The Z/H-89 has a standard 48 Kbytes of RAM and an option for 16 Kbytes of additional memory, but the Z/H-90 comes with the full 64K of RAM. Both the Z/H-89 and the Z/H-90 have three communications ports in the rear panel: two for printers and one for a modem (telephone connection) or a remote terminal. The Z/H-89 is designed to use 5.25-inch hard-sectored disk drives with 100K bytes of storage, the Z/H-90 uses soft-sectored 5.25-inch disk drives with 160 Kbytes of storage space. In addition, both the Z/H-89 and the Z/H-90 can be expanded to use even higher density 5.25-inch drives (the optional Z/H-37 drive package) with 640 Kbytes of disk space. Beyond this, either machine can also be expanded with 8-inch floppy- and hard-disk drives.

USING MORE THAN ONE DISK

Up to two additional disk drives can be added without modification to the Z/H-89. External disk drives are typically attached to the computer by a flat ribbon cable. One or two additional drives, enclosed in a separate cabinet, can be readily added to increase mass storage of programs and data. In addition, both the Z/H-89 and Z/H-90, with the appropriate modifications, will support two different types of drives at once—up to five drives total—with no more than three of one kind. For example, a hard-sector 5.25-inch drive can be used with a soft-sector 5.25-inch drive; a soft-sector 5.25-inch with an 8-inch; an 8-inch with a hard disk; or any other combination of two types of drives (see chapter 7).



The floppy disk is inserted into the drive with the indicator arrow on manufacturer's label pointing toward the drive and the label facing the left side of the drive. If there is any resistance, the disk may be misaligned; remove it and insert it again. After the disk is installed, close the drive door gently; if it will not close easily, check to ensure that the disk is completely inserted (forcing the drive door will damage the disk and can damage the drive assembly).

Systems which have more than one disk drive are considerably more versatile than those which have only one. There will usually be an improvement, albeit modest, in data-access speed when upgrading to a high-density 5.25- or 8-inch disk drive or a hard disk.

Having two or three disk drives has several advantages. Some of the most important are: (1) time can be saved when the user wants to run several different large programs, since time is not lost removing and inserting disks; (2) it is easier to copy programs or backup files from one disk to another, and (3) larger programs and/or data files may be stored.

USING A Z/H-89

For the purposes of this discussion, the Z/H-89 and Z/H-90 can be considered functionally identical. Though they may differ in some technical respects, they both utilize disk drives for mass storage, they both have relatively large RAM, and they both connect to similar peripheral devices. For all practical purposes they work in exactly the same manner.

As the user faces the machine, the power switch is located at the right-rear, in the back of the cabinet. After the power is applied, the electronics become functional almost immediately, and the fan begins to circulate air up through the vents on top of the chassis. The disk drive that is built into the front of the cabinet is now ready to accept a disk.

The user inserts a disk that contains an operating system—HDOS or CP/M—closes the disk drive door, presses the **B** key and touches **RETURN**. The computer then examines the disk to find the operating system program, loads the operating system program into the RAM, and displays a *prompt* showing that it is ready to begin work. The HDOS prompt looks like this: **>**. The CP/M prompt looks like this: **A>**.

If the disk contains a directly-executable applications program—for example: a word processor, or an assembly-language accounts receivable program—the user starts the program by typing the program name: for example, if LEDGER is the name of the program he wishes to run, he only types **LEDGER** and pushes the **RETURN** (signaling the end of the command).

To run a program written in the high-level computer language known as BASIC (assuming the appropriate programs reside on the disk) he types **BASIC** (or **MBASIC** if the MicroSoft derivation of BASIC is being used) followed by the name of the program. For example, **BASIC ACCOUNT.BAS** would be a valid command—where ACCOUNT.BAS is the filename of an accounting program, written in the BASIC computer language.

From this point on the user communicates directly with the applications program. He enter commands on the terminal keyboard according to the command structure found in the documentation manual provided with each program. (See chapters 5 and 6 for further details of software usage.)

THE TERMINAL SECTION

COMMUNICATING WITH THE TERMINAL

The user primarily communicates with the Z/H-89 all-in-one computer system through the *terminal*. Like a railroad terminal, a computer terminal is a starting- or ending-point for data. The terminal is the user's connection with the CPU and the rest of the system components. The all-in-one Z/H-89 has a built-in terminal assembly. This same assembly is also sold as a separate communications terminal (without the CPU and support components that makeup a complete computer), and is called the Z/H-19 terminal. *The Z/H-89 is a Z/H-19 terminal to which has been added hardware and software that make it into a complete, functional microcomputer.*

This chapter will describe some of the ways the Z/H-19 terminal is used to communicate with a computer, be it the internal processing unit in the Z/H-89 or a separate (remote) computer system. Initially we will discuss the structure and operation of Zenith/Heath terminals, and in the following sections we will examine communication between a terminal and a computer.

THE Z/H-19

In 1981 the University of Washington purchased more than 300 Z/H-19 terminals for the students and faculty to use. The terminals were recommended by UW's Academic Computing Center because they could be easily connected to the school's large Digital Equipment Corporation (DEC) computer.

This "DEC-compatibility" has made the Z/H-19 a popular choice in organizations where a DEC computer is installed. DEC's own terminal, called the VT-52, is widely used with large DEC PDP-8 and PDP-10 computers, but the Z/H-19 can be made to appear (to the computer) electrically identical with the VT-52, yet it costs less. (The Z/H-19 emulates the DEC VT-52 to a great extent because Heath designers used DEC computers and terminals while developing their microcomputer products.)

Though most readers will be interested in communicating with a Zenith/Heath computer, the internal functioning of the terminal section of the Z/H-89 is identical to that of the Z/H-19. In the following discussion we will often refer to the Z/H-19, but the descriptions will also apply (except where noted) to the Z/H-89.

MAKING A Z/H-89 OUT OF A Z/H-19

Although it is not a common conversion, it is possible to physically convert a Z/H-19 into a Z/H-89: Heath, in fact, sells an upgrade kit for this purpose. Though we don't believe that many readers will contemplate this modification, the differences between the two machines will become clearer if we look at the individual components required to make a Z/H-89 computer out of a Z/H-19 terminal. Here is what it takes to make the change:

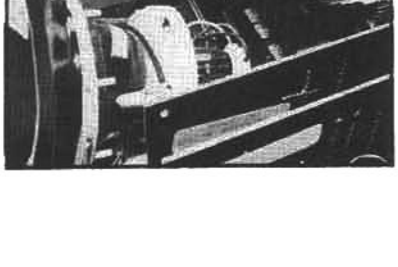
First, you need a larger power transformer, because the '89 needs more current. The power-supply printed circuit board has to be changed, and a fan has to be installed to carry away the heat.

Next, a CPU card must be plugged in and wired to communicate with the Z80 microprocessor on the terminal logic board.

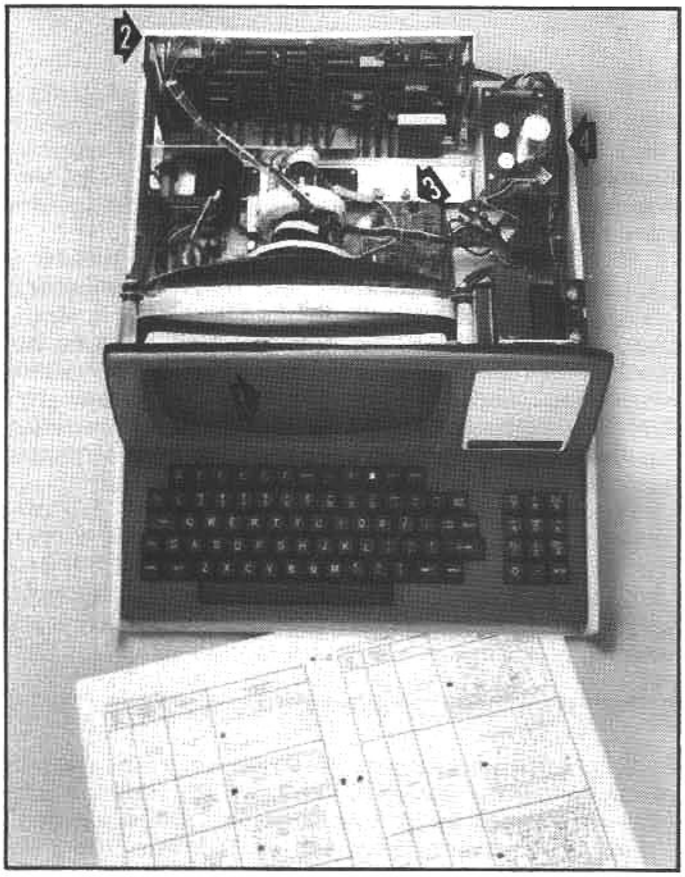
Finally, the lower back panel of the cabinet must be replaced. The new panel has two RS-232C communication ports and allows for future options as they become available. Inside, a circuit-card slot is provided to add the disk-controller card (required when a disk drive is added to the system), and additional slots and sockets are available for memory expansion and other upgrades.

Though this conversion is not extremely common, it is worth noting that if a used Z/H-19 terminal can be purchased for a bargain price, it can be upgraded to a Z/H-89, and the user can generally save a significant

The powerful Z-19 terminal is widely used for communication with remote computers and in multi-user environments. It has a Zilog Z80 microprocessor (to control internal functioning), an extremely readable display, fully-addressable cursor, a reverse-video mode, nine software-definable keys, 33 graphics characters, switch-selectable protocols and baud rates, and a reasonable price.



This printed-circuit board is the 16-Kbyte RAM upgrade, which expands the basic Z/H-89 RAM from 48-Kbytes to a full 64 Kbytes—the maximum memory space that the Z80 microprocessor can support (see chapter 9 for a complete discussion of memory expansion).



With the cabinet cover removed the primary components of the Z-19 can be seen: (1) the video display tube (CRT), (2) the terminal logic board (TLB), (3) video circuit board, and (4) the power supply. With the addition of a CPU board (mounted in front of the TLB) and an enhanced power supply, the H-19 can be upgraded to an all-in-one Z/H-89 computer.

amount of money.

INSIDE THE TERMINAL

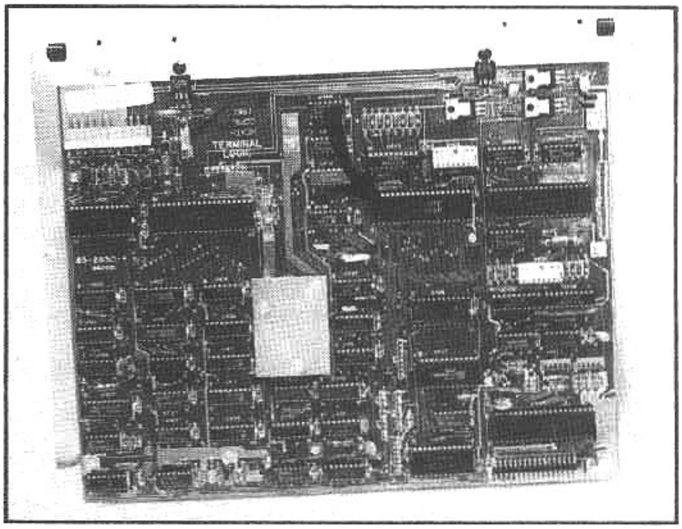
Although the activity inside a computer terminal is similar to a railroad terminal, another analogy to keep in mind is a telephone switchboard, because signals that are generated in one place have to be sent to their proper destination. Like a switchboard, the terminal is directly involved in generating and transmitting communications throughout the computer system.

THE TERMINAL LOGIC BOARD

Inside the terminal there is a circuit board called the *terminal logic board* (TLB). This board holds the main operating elements of the terminal. Two of the most important of these components are the microprocessor and the terminal ROM.

THE TERMINAL PROCESSOR

The *terminal microprocessor* is a Zilog Z80. It is identical to the processor in the main CPU of the Z/H-89; however, the terminal processor is dedicated to scheduling and dispatching data coming into or originating within the terminal section. Like all processors, it needs a program of instructions to perform these functions, and these instructions are contained in the terminal ROM.



Most of the electronics of the Z/H-19 terminal and the terminal section of the Z/H-89 are contained on the terminal logic board (TLB). This is a powerful design because of the dedicated Z80 microprocessor (arrow), which controls internal functioning and external communication to the computer CPU.

THE TERMINAL ROM

The *terminal ROM* (read-only memory) holds a series of permanent instructions that direct the terminal processor to produce specific responses to events within the terminal assembly. These instructions are permanently stored, in the sense that they do not vanish if the power is interrupted. They are not part of the wiring, and so can not be called hardware. They are not easily changed, as computer programming typically is, so they can not be called software, in the normal sense. They are firmly enough embedded in the chip, however, that the typical user can not change them (it takes special equipment), so they are called *firmware.*

There are 4,096 memory circuits in this ROM chip, so it is generally called a 4K ROM. The group of instructions (a complex program) inside this 4K ROM is known as the *terminal monitor,* and at times this ROM may also be called *the terminal monitor ROM.*

THE KEYBOARD ENCODER

Each key on the keyboard is a simple switch. When a key is pushed down, the switch is closed. This is detected by the *keyboard encoder*—an integrated circuit chip mounted on the TLB that constantly scans the keyboard to determine the status of all key switches. When the encoder determines which switch was activated, it sends a stream of data to the terminal processor, telling it what has happened. The Monitor program in the microprocessor analyzes this data, checks other contingent conditions inside the terminal and determines what to do with the data.

THREE MODES OF OPERATION

Before the terminal processor can act on the information from the keyboard, it checks the *terminal mode.* There are three modes: off-line, full-duplex, and half-duplex.

The terminal mode is controlled by the key in the upper left corner of the keyboard that is marked OFF-LINE. This is a *latching* key: when it is pressed and latched in the down position, it places the keyboard and CRT display in a closed system. That is, the keystroke data is not sent anywhere except to the CRT screen. The *off-line mode* is useful for changing screen display modes, such as going from normal white-on-black to reverse-video (black on white). To return the terminal to the *on-line mode*, the OFF LINE key is depressed one more time, and it returns to the unlatched (up) position.

When the terminal is on-line, it will be in either the full- or half-duplex mode. *Full-duplex* means that the terminal display is under the complete control of an external computer, called the *host*. The code from the keystroke is carried along a cable that connects the back of the Z/H-19 terminal to the host computer through an external connector (in the Z/H-89, the signals are passed internally to the main CPU board located next to the TLB). This external connector—a standard multi-pin *DB-25 plug*—is used in the microcomputer industry as a serial data connection, and is often called an *RS-232C interface* (see below, serial transmission).

When the host computer receives a keycode through this cable, and that code represents a "displayable" character, the computer sends a signal back along the cable that instructs the terminal to display the character on the CRT screen. That is, the TLB does not display the character that is typed until the host "echoes" that same character back to the terminal—this is the nature of full-duplex communication.

In *half-duplex* the keycode is sent in two directions: to the host computer and to the display. A displayable character is always reproduced on the screen, regardless of whether the host (remote) computer echoes the code or not.

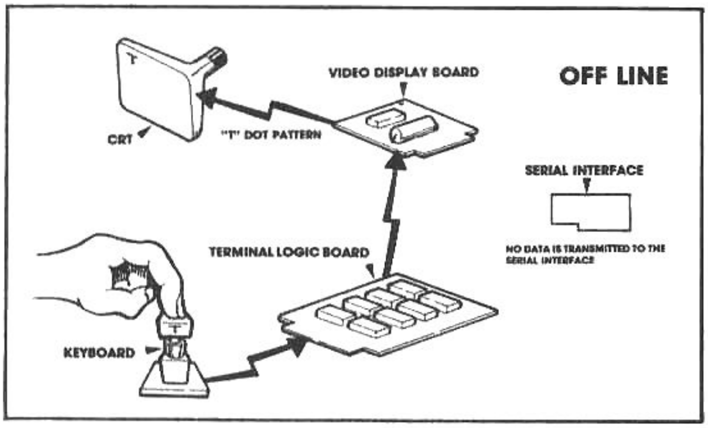
In all three modes, the character displayed on the screen corresponds to the key that was depressed. To the user, the path between the key and the screen is invisible, but very important. In half-duplex mode, for example, *the user can not assume that the keyed-in character actually reached the host computer,* the screen display will come from the terminal.

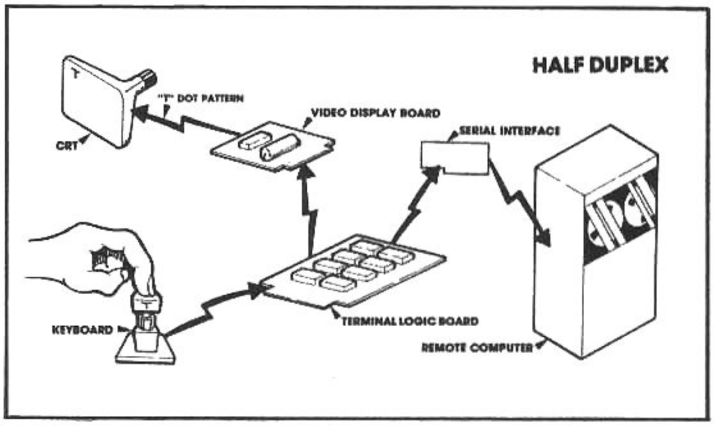
Once the terminal recognizes what was typed on the keyboard, it finds a place on the screen to put the data. A firmware chip is responsible for this.

On a typical full-size CRT screen, as used by Zenith and Heath, there are 80 possible character positions—usually called *columns*—on each line of the display, and there are 24 *lines* (there is a 25th line, but it can only be accessed by a special instruction). The terminal will only "print" a character on the screen at the cursor position—where the flashing box or underscore is located. When the terminal is switched on, or RESET, the first character will be displayed in the upper left corner of the screen. The chip examines this location (i.e., the top line, first column) to see if the cursor is located there. If so, this is where the character will be placed; if not, it will go on until it finds the cursor position. When the cursor is located, the character will be formed at that point on the display.

THE CHARACTER-GENERATOR ROM

Other firmware is responsible for generating the

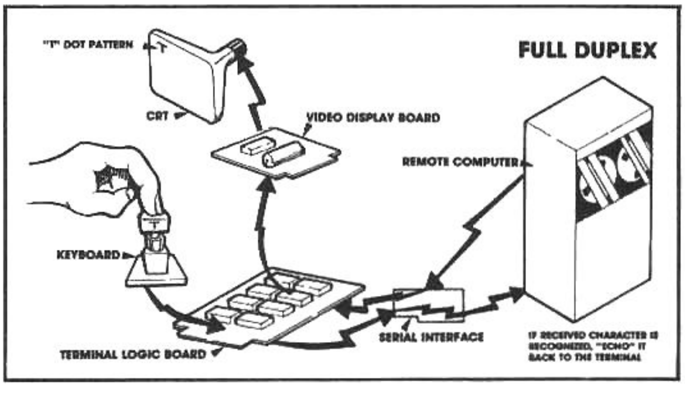


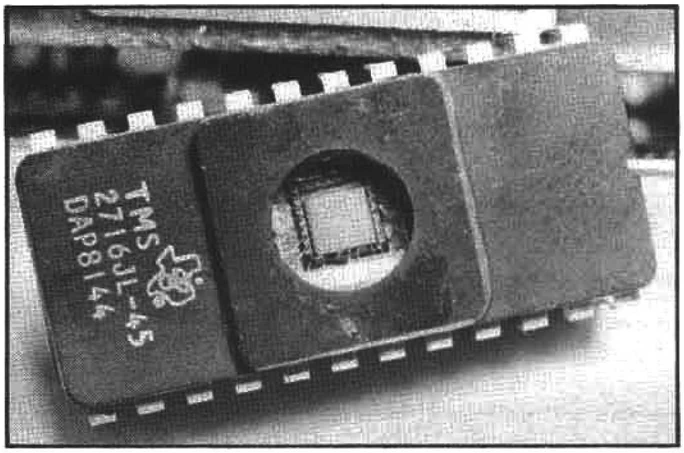


The terminal section of a Z/H-89 and the Z/H-19 can be configured in three ways: off-line, half-duplex and full-duplex. When the terminal is in the *off-line* mode, the keyboard communicates *only with the CRT screen.* Each keystroke is examined by the keyboard encoder chip, and the appropriate ASCII code is generated and processed by the terminal logic board (TLB), which determines whether the code represents a displayable character. If it does, it passes a signal to the video display board which displays the appropriate dot pattern on the CRT screen. If the ASCII code is a command-type (nondisplayable) character, the terminal logic board either waits for the next character (of an escape- or control-sequence) or—if the code is a self-contained command, such as carriage return, form feed, etc.—it carries out that instruction.

When the terminal is *on-line* and in the *half-duplex* mode, the TLB is like a Y-connector or a fork in the road. It examines the ASCII codes in the manner described above and *simultaneously sends them to the video display and to the host computer for processing* (either the onboard computer in the Z/H-89 or to a remote computer.)

When the terminal is on-line and in the *full-duplex* mode, the TLB passes the ASCII code only to the host computer, which determines whether it is displayable or not. If it is, the code is "echoed" (transmitted) back to the terminal and passed to the video board for display; if it is a commmand code, the computer either waits for the next code or acts on the initial command, as required. *In full-duplex mode the character is not displayed until it is received by the host computer, recognized and echoed back to the terminal.*





There are several places in a typical computer where permanent instructions (firmware) are used to perform elementary functions. This type of program is electronically embedded in a read-only memory (ROM), and cannot be readily changed. For example, ROM-based firmware is used on the terminal logic board to tell the video section how to form each character displayed on the screen.

instructions required for producing each displayable character on the screen; this firmware is contained in the *character-generator ROM.* When the terminal microprocessor decides that a keyboard signal is calling for a displayable character, it signals the character-generator ROM to produce it. In addition to the Roman alphabet used in the U.S., the U.K. and Commonwealth countries, special character-generator ROMs are available for French and German, with appropriate diacritical marks; and there are Arabic and other alphabets available as well.

Displayable characters are formed from a matrix of dots that represents all the possible parts of any printable character. This matrix is five dots wide and nine dots tall and is called, as you would suspect, a *5-by-9 dot matrix.* The top row of these dots is as high above the line as the tallest ascender (the portion of a letter that extends above the baseline), and the bottom row is as low as any descender (the portion that extends below the baseline); see accompanying illustration.

When the character-generator ROM receives instruction to form a character, it checks a "lookup table" in a reserved memory area to determine the precise array of dots needed to form the requisite character, and generates instructions to display this array. These instructions are then sent to the CRT-controller.

THE CRT-CONTROLLER

The heart of the video display is actually the *CRT-controller.* When the controller understands that a character is to be displayed, it will write the instructions in a temporary storage area called *video RAM.* Video RAM operates in a similar fashion to main RAM but is much smaller—there is room for 2048 characters—and it is dedicated strictly to the video display. As characters are placed in the video RAM, they are recognized and reproduced on the CRT screen display.

These are not all of the components and functions that occur inside the Z/H-19 terminal, but they are the primary events aimed at transmitting "keycodes" throughout the terminal and from the terminal section to the computer. We will continue to examine computer communications in the next section, but before we move on we are going to look more closely at these codes.

ASCII AND OTHER CODES

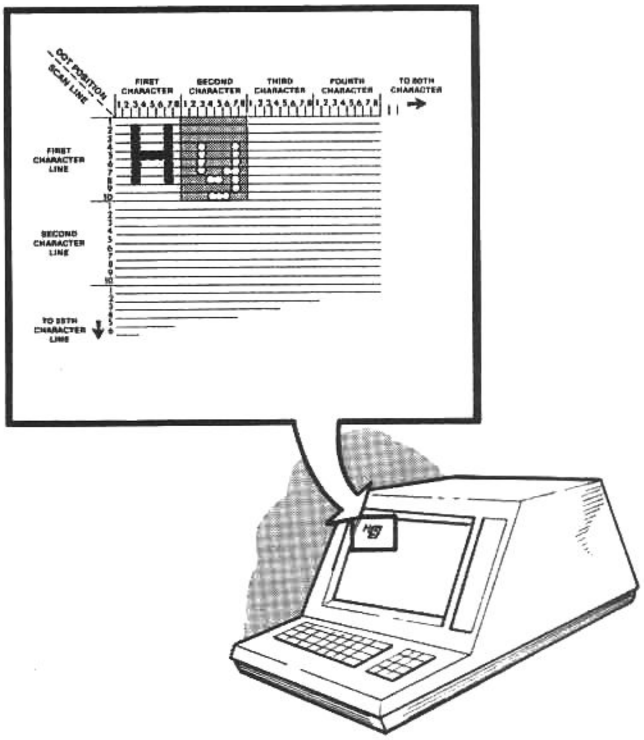
The alphabetic and numerical characters of language would have no meaning to the computer if they were not coded in a recognizable binary form. This code is known as the American Standard Code for Information Interchange (abbreviated *ASCII*, and pronounced "ASK-ee").

ASCII coding was designed specifically for computer communication. Each character is represented by an 8-bit byte. Each bit can be high or low. Thus for the first bit, there are two possible combinations; for the first and second bit together, there are four possible combinations; for the first, second and third, together, there are eight. For positions 1-4 there are 16; for 1-5 there are 32; for 1-6 there are 64; and for 1-7 there are 128 combinations.

(A mathematically-inclined person will notice that the progress of these numbers is logarithmic, based on powers in the base 2: 2^2, 2^3 . . . 2^7.)

Because the 128 combinations available from the first seven bits is more than adequate to represent the letters and numerals in most languages, the ASCII code makes use of just the first seven bits. The eighth bit is generally (though not always) used for an error-checking routine.

The ASCII code is important because it is almost a universal standard. Any terminal which produces ASCII



HOW CHARACTERS ARE FORMED ON THE SCREEN

Each character on the CRT display is formed from tiny dots of light arranged in an 8x10 matrix. This is an enlarged image of an upper-case "H" formed in *reverse video* (black dots against a white/green background) and a lower-case "y" formed in *normal video* (white/green dots against a black background). Note in the lower-case "y" that this matrix reproduces letters with *descenders*, the portion of a letter that drops below the baseline. This permits display of upper- and lower-case letters, something not found on less-sophisticated displays that are limited to upper-case letters, or which set all characters above the baseline.

codes can be connected (somehow) to any computer which understands them. This is an oversimplification, because there are technical limitations, but in general it is so.

There is an important exception to the universal nature of ASCII encoding: a code called *EBCDIC.* This encoding is used by IBM on virtually all of their keyboards (although, significantly, not on the IBM Personal Computer). Fortunately, if conversion is necessay on the machine level, it is not difficult to make a firmware chip that contains a lookup table to convert one code into the other.

THE KEYS AND WHAT THEY DO

The keyboard of Z/H-19 and Z/H-89 has 84 keys. They can be divided into four logical groupings. The central array of 47 keys produce *displayable characters of the alphabet, numbers, and symbols* similar to those found on a typewriter. Clustered on either end and below this grouping are 16 keys that produce *non-displayable command-type instructions* (such as the SPACE BAR, CARRIAGE RETURN, SHIFT, BACK SPACE, etc.). The top row of the keyboard contains nine software-programmable *special-function keys.* The function of these keys is defined (or ignored) by whatever software is controlling the computer or terminal at any one time. The last 12 keys are clustered separately in a *numerical keypad* on the right side of the keyboard.

ESCAPE SEQUENCES

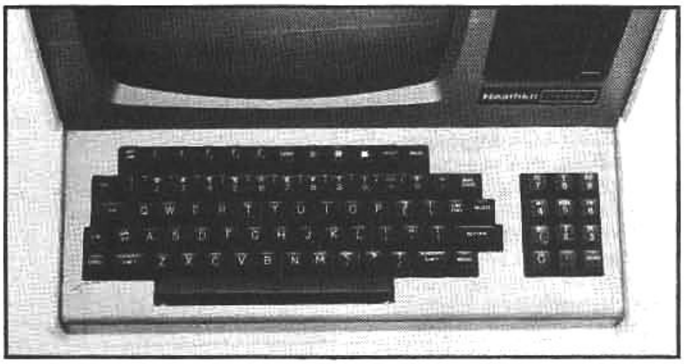
Compared with a standard typewriter, the most notable feature of a terminal keyboard is the number of extra keys that don't represent normal letters, numbers or symbols. These keys usually send "commands" to the computer: instead of producing a displayable character, they transmit codes to tell the computer to do something. In the following discussion we will look at the general nature of these command keys.

The *ESCAPE* key—labeled ESC—is on the left end of the second row of keys. Escape is an ASCII code, like a letter or a number, but it is one which is not, generally, displayed. It is depressed in combination with another key, to produce an *escape-sequence.*

The terminal firmware expects to receive an escape code followed by one or more additional codes (characters); it keeps a list (or, more precisely, a look-up table) of all the possible escape-sequences. If the escape-sequence is not recognize, the terminal will ignore the escape code and print whatever character follows it. If it does recognize the escape-sequence, it will execute the activity which the look-up table indicates. What this activity will be is defined in the terminal firmware, or may be determined by the particular software which is controlling the computer at that time.

For example, in the Z/H-19 the ESC-p sequence (i.e., the ESC key, followed by a lower-case p) tells the terminal to change the display from normal to reverse-video. The ESC-q sequence exits the reverse-video mode and returns the terminal to normal white-on-black display. *The escape-sequences will only be recognized, however, if the software communicating the terminal is designed to accept them or if the terminal is in the off-line mode* (as described earlier).

The escape-sequences and their corresponding activities (when the terminal is in the off-line mode) are unique to the Z/H-19 terminal and related Z/H-89 and Z/H-90 computers. The interpretation of these escape-sequences is predetermined by the built-in firmware, and they are known as the Heath or ZDS (Zenith Data Systems) escape-sequences. When the terminal is used as part of a time-shared computer system, with a remote host computer, it is more common that that computer will have its own look-up table of escape-sequences. These will probably be those of the American National Standards Institute (ANSI), and called ANSI escape-sequences, or *they may be defined by the software to produce entirely different results.*

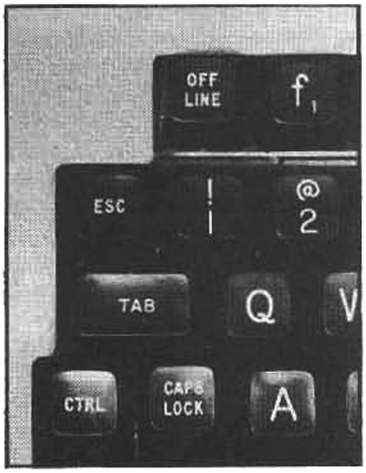


The Z/H-19 keyboard is based on a standard typewriter layout, but special keys are included to transmit commands to the computer software. The special-function keys are on the top row, ESC and CTRL are at the left, and the numeric keypad is to the right. The CAPS LOCK key shifts and holds the *letters* to upper-case, but the number, symbol and command keys are not shifted. This is a convenience feature for mixing upper-case letters and numbers on the same line (often required when writing programs).

The Z/H-89 or -90 user will seldom (if ever) need to alter the escape-sequence mode, but there is a small switch on the Terminal Logic Board which permits the terminal section to be used with ANSI escape-sequences. This mode can also be switched with an escape-sequence (by the software, if desired). Detailed information on these special configurations and a listing of Heath/ZDS escape-sequences is available in Zenith/Heath technical literature.

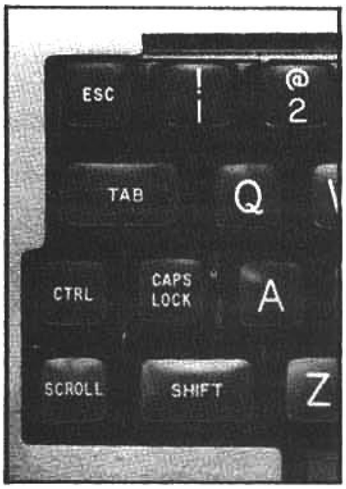
CONTROL SEQUENCES

The *CONTROL* key (labeled CTRL) performs a unique function. When terminals were first developed—based on the model of the teletypewriter—engineers realized that there were not enough keys to handle every necessary data processing function. Rather than build in an unwieldy



The ESCAPE key normally does *not* produce a letter or symbol, it is used to send *commands* to the terminal or computer. Pressing ESC and then pressing one of the letter keys, generates an *escape-sequence.* These sequences can be used within the terminal to control the display (move cursor up or down one line, delete or insert one line, enter or exit reverse video, etc.), or they can be used at the computer level to perform any function defined by the software.

The CONTROL key normally does *not* produce a letter or symbol, it is used to send *commands* to the computer software. Holding down CTRL while pressing one or more additional keys, produces a *control-sequence*. When these sequences are received by the software controlling the computer or terminal, the software may cause some specific machine task to be performed.



number of special-purpose keys, they chose to add just one that could "control" or alter the standard keys to perform special functions. In practice, the CTRL key is held down while pressing another key, and the conjunction produces a non-displayable ASCII control-sequence. *The function of this code is determined by the software in use at the time.*

The use of control-sequences is a feature that adds versatility to computers. For example, a software writer has only to specify that CTRL-e will move the cursor up one line; it doesn't matter then what brand of terminal is used—the results will be the same (as long as the terminal generates the correct ASCII CTRL-e codes).

The very popular WordStar word processing program uses a vast repertoire of control-sequences to initiate nearly any text-handling tasks the user can imagine. This allows it to be used on virtually any microcomputer that has a standard keyboard (and uses the CP/M operating system), and is one reason why it is so widely used.

In the Z/H-19 certain control-sequences are already programmed into the terminal logic ROM, and may be used when the terminal is in the off-line mode. For instance, a CTRL-I produces the same effect as the TAB key, while CTRL-M duplicates the function of the RETURN key. A complete listing of these control-sequences can be found in the operations manual for the Heath H-89 computer.

SPECIAL-FUNCTION KEYS

The *SPECIAL-FUNCTION* keys on the top row of the keyboard add considerable power to the terminal keyboard. The five keys to the left are labeled: *f1, f2, f3, f4 and f5.* The three keys to the right are identified with colored squares: *blue, red and white.* Between these two groupings is a single special-function key labeled *ERASE.* These labels have no special significance other than to identify the keys.

The software programmer can define these keys to perform any function desired. In effect, each of these keys transmits an escape-sequence code to the processor. The software directs the processor to perform specific tasks whenever one of the special-function keys is pressed. However, if the software is not designed to use these keys, they will do nothing.

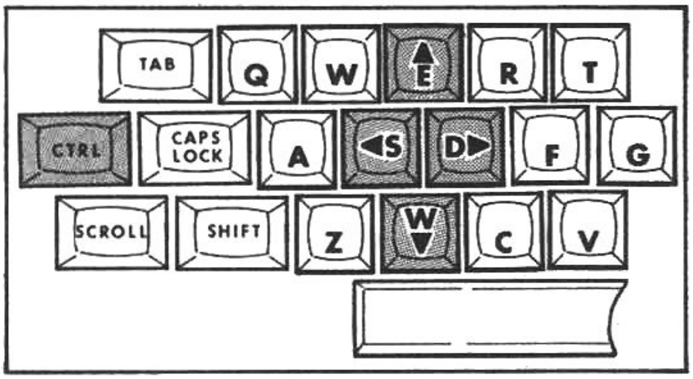
For example, the F1 key generates the ESC-S code (equivalent to depressing the ESCAPE key followed by a capital S). When the word processing software called Magic Wand was adapted for use with the Z/H-19 terminal and Z/H-89 computers, it was modified slightly so that an ESC-S—or the f_1 key—would scroll displayed text down one line.

Though the ERASE key can be defined to do almost any task the software designer wishes, the most logical use is to erase text. Many word processing programs use this key to erase all the text from the cursor position to the end of the line or to the bottom right side of the display.

One unusual definition of this key is found in the Video Scribe word processing program (see chapter 6). In this program, ERASE is an "OOPS" key—any previous manipulation of text is immediately reversed, a very useful feature when a complex text manipulation doesn't work out exactly as the user planned!

OTHER COMMAND KEYS

At the upper left of the keyboard is the *OFF-LINE* key. For normal communications with the computer this key is latched in the up position. This is the on-line position, and all communications generated by the keyboard and terminal section are then sent to the computer CPU. When this key is depressed and latched in the down position, the keyboard and terminal section are taken off-line and isolated from the computer CPU.



This in an example of how control-sequences are implemented in the popular WordStar word-processing program. The user moves the cursor around the screen by holding down the CTRL key and pressing one of the letter keys: CTRL-X to go down one line, CTRL-E to go up one line, CTRL-S to move the cursor one character to the left, CTRL-D to move it one character to the right. These are just four of over 100 control-sequences used in WordStar.

In rare instances, the terminal may be taken off-line to type a command or produce something on the terminal display that the user does not wish to be transmitted to the computer CPU. For instance, if the user wishes to change from the normal display mode (white characters on a black background) to a reverse display (black characters on white). This can be done by transmitting an ESC-p sequence to the terminal processor. However, if the terminal is on-line, the escape sequence will be sent to the main CPU and will be ignored (unless the software controlling the main CPU recognizes this specific escape-sequence; and if it does, it may produce a different result). If the off-line key is latched down, the escape-sequence will be sent only to the terminal processing unit. The programming of the terminal processor will recognize the ESC-p and change the display to reverse video. (Incidentally, if the reader tries this, an ESC-q will return the terminal to normal video.)

The *RESET* key works only when the right-hand SHIFT key is also held down. *It is restricted in this way to avoid an* *unintentional catastrophe!* The SHIFT-RESET combination is used to—essentially—stop all computer operation and restart from the most elementary level of control. In computer jargon this is known as a *system reset.* When a system reset is commanded (with a SHIFT-RESET), control of the computer is immediately returned to the system monitor program and the H-prompt appears in the upper-left corner of the screen (almost as if the power switch was turned off and back on). At this point, if the user wishes to resume using the computer in a normal manner, a command must be issued to the monitor to reboot the operating system into RAM.

Never command a system reset while working on any data or text: any data in RAM will be almost impossible to retrieve. Remember, any newly created data in RAM must be written to a disk file before it becomes a permanent record. A system reset is used only on those rare occasions when the user wants to stop and re-start the entire system from scratch.

BREAK is often used in time-sharing applications. It signals the host computer that the user wants to interrupt whatever is going on. Unlike a system reset, it does not "break" the connection or the current operating mode of the computer/terminal. The host computer will have its own software for interpreting the break command.

BACK SPACE, like the same key found on a typewriter, moves the cursor backward (i.e., left) one character. When used with most operating systems and software, it will "erase" the character to the left of the cursor. When the cursor reaches the left margin, the back space key will not move it; however, several word processing packages will allow a back space to take the cursor to the right-hand end of the previous line.

TAB is the "tabulation" key, familiar to every typist. Some computer applications will not accept the signal from the tab key unless the number of tab spaces has been pre-set. Some word processors provide the user with a menu from which a choice of tab settings can be made. Some programming languages, such as Pascal, rely strongly on tabbed commands, and nearly every package of software has some purpose for the tab function.

LINE FEED moves the cursor down one line. If the cursor is already on the bottom line, pressing the line feed key causes the text or data on the screen to move up one line, as if the cursor had moved down. The text that was previously on the top line of the screen will move up (i.e. off the screen) one line to accommodate the new material.

DELETE is a definable key, like the special-function keys. Typically, however, it is assigned a function consistent with its name: most word processors accept the delete key as identical to the back space key, however, some programs assign it a unique function. Magic Wand, for example, uses it to delete one entire word at a time.

CAPS LOCK is very much like shift lock on a typewriter. In many software packages (HDOS & CP/M included), the caps lock shifts only the alphabet keys into capital (upper-case) mode; number keys and symbol keys are not shifted to the alternate characters (the standard shift key must still be depressed to shift these keys to the alternate characters or symbols, see below).

RETURN moves the cursor to the first (far left) character position of the line it is currently on. Under word processing or text editing software, return may also force a line feed, just as it does on a typewriter. Return is usually not displayed, but it can be displayed if the software defines it as a displayable character.

SCROLL is used by some software as a stop-and-go switch for displaying text continuously. Magnolia Microsystem's version of CP/M uses the scroll key to interrupt the display of data on the screen produced with a TYPE command to the operating system (see chapter 5). The combination SHIFT-SCROLL clears the screen and displays the next 24 lines of the file.

SHIFT is used, as on a typewriter, to generate capital (upper-case) letters and the alternate characters or symbols that are inscribed on the keys. On the Z/H-89 and Z/H-90, the numerical pad can be shifted to the alternate functions (cursor-control arrows, insert and delete function keys) with an escape-sequence, or a program may automatically produce this shifted function for the user.

REPEAT does nothing by itself. When it is held down at the same time that another key is depressed, it will repeat the action at the rate of about 15 times per second. REPEAT-a generates: aaaaaaaaa. It is often used with the cursor-control arrows to speed the movement of the cursor around the screen.

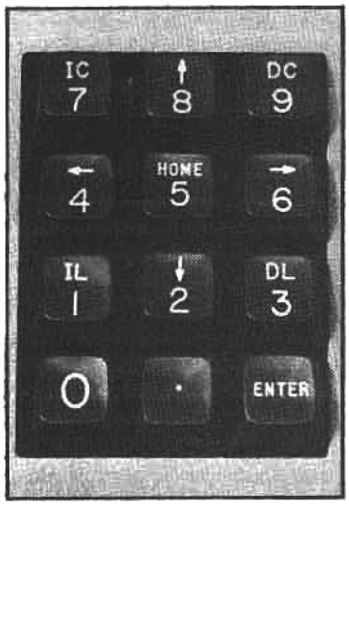
THE NUMERIC KEYPAD

The numeric keypad on the right side of the keyboard is provided for convenient data entry; many data processing tasks are speeded by a standard 12-key numeric keypad.

The numeric keys can also serve as function keys for text editing: controlling the movement of the cursor, inserting and deleting characters and/or lines. This is achieved by shifting the keypad into the alternate mode, either by software (such as Text Processor or Magic Wand, for word processing) or by holding down the shift key. In addition, the sequence ESC-t will place the keypad into the alternate-function mode, and ESC-u will return to the standard functions. When the keypad is already in the shifted mode, the shift key works in reverse—generating the numbers, rather than functions.

When supported by the appropriate software, the alternate (shifted) functions of the keypad will generally produce the following:

ARROWS (i.e., the cursor-control arrows) are self-explanatory; they move the cursor up, down and sideways, one line or character at a time. If they are used with the repeat key, they will move the cursor quickly. If the cursor reaches the right-hand margin and the right arrow key is depressed again, the cursor will jump to the left end of the next line down and continue moving right. The reverse holds true if the cursor is moved continuously to the left: it will jump up to the right end of the next line above and continue moving left. If the cursor is moving down, and reaches the bottom line of the display, the text will jump up



The Zenith/Heath terminal and computer numeric keypad can be shifted to an alternate mode by the hardware (a switch on the TLB) or by the software (e.g., some word processors—Magic Wand, PIE, Text Processor—shift the keypad automatically). In the shifted mode, the keys generate unique Heath/ZDS escape-sequences.

(as if the cursor were still moving down) and the screen will display next line of text. Likewise, if it is moving up, it will force the displayed text down to reveal the previous line of text.

HOME brings the cursor to the left-hand margin, although some word processing programs extend that function: pressing HOME may take the cursor to the upper-left corner of the screen. Likewise, pressing home a second time in some programs may take the cursor down to the left margin of the lowest line currently displayed on the screen, or to the lower-left corner if the screen is filled with text.

IC (insert character) is a text-editing key that permits the user to insert one or more characters at the place where the cursor is located. If there is text already occupying that spot (hence the need for insertion), the new characters will appear in the cursor position and force the rest of the text to move to the right. In some word processing software, words forced past the right margin, by inserting new text in a line, will be automatically moved or "wrapped" down to the beginning of the next line. Some programs will remain in the insert-character mode until shifted out by a second depression of the IC key.

DC (delete character) is a text-editing key that removes whatever character is at the cursor position. Depending on the software, when text is removed, the rest of the line or paragraph may reformat to close the gap. When the DC key is used to remove a character, no space remains in the text after a character is deleted. (SPACE, remember, is a character too, and can also be deleted.) The DC mode, unlike IC, must be engaged each time by pressing the DC key—that is, whether the cursor moves or remains in the same place, DC will delete a character whenever the DC key is pressed.

IL (insert line) is used to insert more than a few characters at a time—most often a sentence or two. IL cuts into the text like a pair of scissors, giving the user a blank line to fill. If desired, the user can continue to add blank lines to provide room for a paragraph or chart, etc.

DL (delete line) is like DC in operation—except it will delete an entire line each time it is pressed. Under some software, the entire line on which the cursor is positioned may be deleted, others programs will delete only the part of the line after the cursor position. Because it could wreak havoc if pressed accidentally, the DL key is often protected from careless fingers: some software causes it to be activated only if it is pressed twice. In such cases, after it has been pressed twice, it generally has to be pressed only once for each additional line to be deleted.

The keys labeled *0*, *(.)* and *ENTER* are unaffected by the shift and caps lock keys, but they can be shifted to the alternate mode by the proper escape-sequences. When used as normal ASCII characters, the zero and the decimal point keys on the keypad are identical to the zero and period on the alphanumeric keyboard, and ENTER is the same as RETURN—they may be used interchangeably. However, under software control, there may be enhanced functions associated with these keys.

At the back of the Z/H-19 terminal and the Z/H-89 computer are electrical plugs with 25 pins in them. This is an industry-standard RS-232C serial communications interface, providing a connection to the outside world. In effect, the RS-232C interface permits the Z/H-19 terminal to transmit or "talk" to a remote computer. The subject of data communication will be explored in the next section.

COMPUTER COMMUNICATIONS

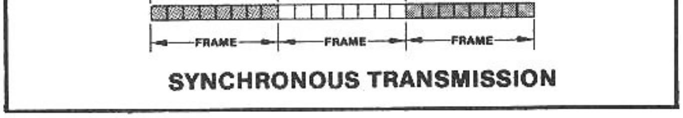
There is a significant component of the terminal logic board that we have not discussed yet: the *asynchronous communications element* (ACE). This tongue-twisting name is best understood if you read it backwards. It is an "element" (a major one) in the process of "communicating" with the host computer, and it is "asynchronous." But there are two methods used for computer communications; let's take a look at them.

SYNCHRONOUS COMMUNICATIONS

When a host computer communicates with a terminal, the data is sent back and forth as a continuous stream of 8-bit bytes. To keep things straight, the receiving machine must keep track of the incoming bits, continually divide them into groups of eight and successfully reform the bytes into useful data. There is little room for error in this process. One misplaced bit will destroy the integrity of the entire data stream.

Synchronous communication is carefully divided into uniform blocks of time, called *frames*, which must be kept filled. If there is data, it is transmitted at the rate of eight bits per frame. If there is no data, then a synchronization character is sent instead—also eight bits long. Once a connection is established and a synchronization character has been received, the receiving device will continue to accept incoming data until it detects a special character used to identify the end of a transmission.

Synchronous transmission works best between machines, as when a large bank polls its remote, stand-alone teller machines, to collect their transaction records. The communication link between the machines is established, data is transferred in a rapid, uninterrupted flow and the



Data is transmitted in a stream of 8-bit bytes. Each byte consists of seven *data bits* (representing the character of data) and one *parity bit*—a high or low bit used for error-checking. *Synchronous transmission* is formed into segments called *frames*. A frame containing a data character is followed by a frame with a *synchronization character*. This produces a stream that looks like this: data-pulse-data-pulse. The stream is monitored at the receiving end and the data characters (bytes) are pulled out of the stream. The parity bit is always used. In *asynchronous transmission* a start bit is added in front of each data byte and two stop bits are added at the end; a parity bit is not always used, but a position for it is available.

Synchronous transmission produces a steady stream of frames; when data is not available, synchronization characters are used to fill the stream. It is an efficient way to send a large quantity of data, in a one-way stream, to a remote receiver. Asynchronous transmission, however, is better for sporadic, two-way streams of data, as when a user at a terminal communicates with a computer. It is less expensive to implement (usually) and more suited to interactive human/computer communications.

link is broken as soon as the transfer is complete. In a completely automated mode, the machines can transfer data in a synchronous manner because it is a very efficient method of communications.

However, when the flow of data is sporadic, as when people must interact with computers, things are somewhat less predictable. For example, at an airline ticket counter no one can predict when a clerk will need to check a reservation for a customer, but in a synchronous mode the link between the clerk's terminal and the host computer would have to be kept busy, all day and night, with synchronization characters, waiting for the occasional moment when data will be sent. This is a very inefficient use of computer time, but there is an alternative.

ASYNCHRONOUS COMMUNICATIONS

Asynchronous communication is like an ordinary telephone call: it arrives at an unpredicted hour, each person speaks for a random number of seconds at a time, and one person can hang up before the other does. Between a host computer and a terminal, therefore, asynchronous communication is more like human conversation than the constant-flow synchronous link.

With asynchronous communication, there are certain *protocols* used to match the irregularities of the human interaction with a computer system. One protocol must be established to tell the computer that you have begun sending information. This is done with the *start bit.* Before any character (byte) is transmitted, it is "announced" by the presence of the start bit; and when the character has been transmitted, the end is signified by a *stop bit*. The circuit may remain idle for any length of time, since the transmission of data is not dependent on fitting it into preexisting frames. The frames of asynchronous transmission are defined by the start and stop bits, and the receiving device accommodates itself to this "start-stop" reference.

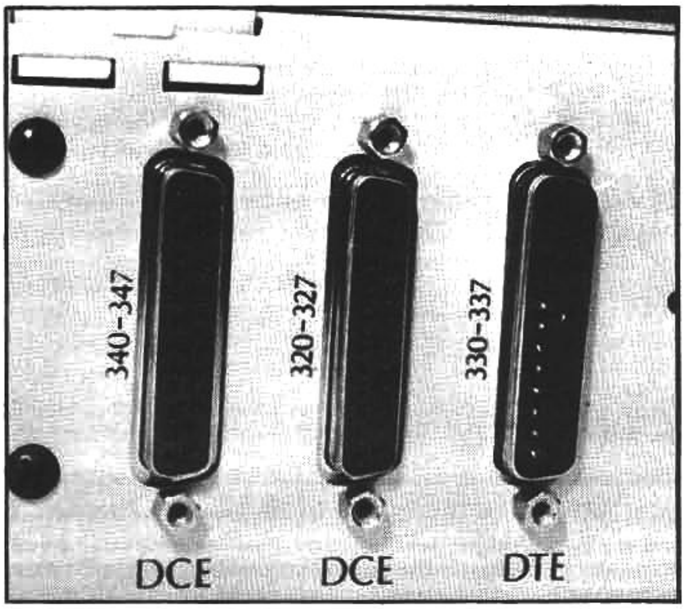
Each type of transmission has advantages and disadvantages. Synchronous communication is expensive, somewhat delicate, and limited in its range. Synchronous communication is also very efficient; once the synchronizing character has been recognized, more data can be squeezed into the stream, since only the data need be transmitted. Asynchronous communication requires that extra data be transmitted to do the "housekeeping" between bytes, making it somewhat less efficient. However, asynchronous communication is less expensive, because the circuitry for transmission and reception is less sophisticated.

Because of the irregular movement of data transmissions between the Z80 terminal processor of the Z/H-19 (or the terminal section of the Z/H-89) and the CPU, the processor communicates with the outside world asynchronously through the ACE. The ACE supplies the start and stop bits for each byte of data in these transmissions.

SERIAL AND PARALLEL

There are two ways to let cattle out of a corral: open the big gate and let them all charge through at once, or open just a small chute and send them through one at a time. There are also two ways to send data from a computer terminal to a computer: *parallel* (many at a time) or *serial* (one at a time).

SERIAL TRANSMISSION



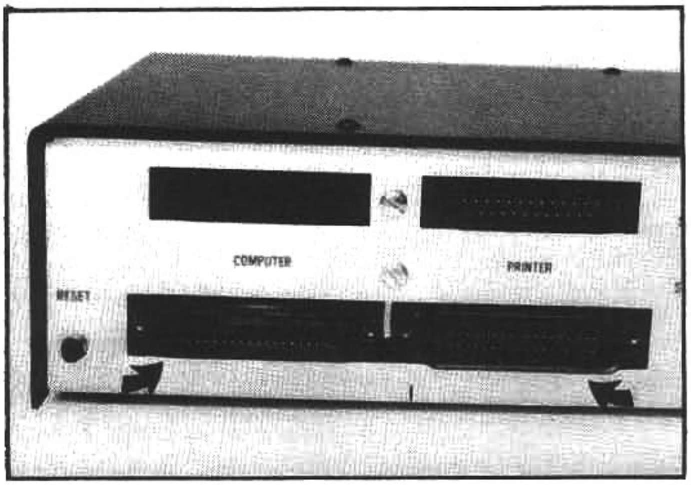
A *serial port* transmits data one bit at a time—in a series. There are two types of RS-232C serial ports in the back of Zenith/Heath computers. One is called a data terminal equipment (DTE) port, and is normally used for two-way communication (such as with a modem); the other is a data communication equipment (DCE) port and is used for one-way transmission of data to a listing device (such as with a printer). All late-model Z/H-89s have three ports: one DTE and two DCE—allowing the simultaneous connection of a modem and two serial printers.

Lining up bits of data and sending them, in effect, one at a time through a wire, is a simple way of making sure that they all get to their destinations. No matter how restrictive the wire may be (a restrictive wire—sometimes referred to as "thin"—will only carry limited amounts of data at one time), the data will eventually get to its destination. This is serial transmission.

Telephone lines are very thin; in a house or office they are made of two wires (a complete circuit) wound around each other in a double helix for strength and to reduce electromagnetic interference. In the telephone industry, they are known as a "twisted-wire pair." Sending data over a telephone line requires serial transmission because the wires have a limited capacity. A chain is only as strong as its weakest link, and that link in a telephone network—despite satellite relays, microwave towers, and thick undersea cables—is only as "wide" as a single twisted-wire pair.

The Z/H-19 terminal and the Z/H-89 and Z/H-100 series computers communicate externally by serial signals transmitted through the multi-pin DB-25 plug connector(s) on the back of the machine (there are parallel communication boards available for the Z/H-89, but serial transmission is standard). This is often called an RS-232C serial port, and it is one of the few standards (with some variations) in the microcomputer industry that is observed by most major manufacturers.

To establish a communication link between two elements (e.g. to connect the Z/H-89 to a serial line printer—see chapter 7 for details of printer interfacing) a multi-wire cable with DB-25 connectors on each end is used to interconnect the RS-323C ports of the two communicating elements. Both elements must be designed for serial communications and both elements must be synchronized to send and receive data at the same speed (see below, baud rate).



A *parallel port* (arrows) simultaneously transmits all eight bits of a byte on eight parallel wires. Parallel communication is usually much faster than serial communication, with data rates up to 3000 characters per second (equivalent to 30,000 baud). Parallel communication is supported on the Z/H-89 with the Z/H-89-11 multi-mode parallel interface upgrade (and appropriate software). Note the difference in the appearance of the parallel-port (lower) and the serial-port (upper) connectors. This configuration for a parallel port is common, but not universal.

PARALLEL TRANSMISSION

The alternative to sending one bit at a time over a single wire is to transmit eight bits—one entire byte—simultaneously, over eight parallel wires. In this way an 8-bit byte, is transmitted and received during each communications cycle. This is parallel transmission.

Serial interfaces must contain a buffer, or memory, to hold each of the first seven bits until the eighth has been received; then they can all be passed along as a complete byte, making up one data character. Parallel traffic is much faster, does not need a buffer, and thus is widely used for "dumping" data from one machine to another.

Parallel transmission is limited by distance. All the wires must be the same length, and because of electrical and magnetic "noise," the wires usually can not exceed a few hundred feet. Parallel transmission is impossible over standard telephone wires, however, it is often done over channels that are "wider" than a twisted-wire pair. Using special hardware at both ends, parallel transmission can be sent along what are called broad-band circuits, such as coaxial cables, microwave or other radio links.

All Z/H-90s and Z/H-89s shipped after November, 1981, have three RS-232C serial ports; they can be modified to accommodate a parallel port also. The Z/H-100 has two serial and one parallel port. The physical size and shape of a parallel port plug is different from that of a serial plug, and the two are not in any way compatible.

THE BAUD RATE

Two communicating elements must agree on the speed with which they will exchange information. If not, data may be lost when one element loses track of what is going on. In the computer world, the rate of data exchange is measured in the number of transmitted bits per second, and is commonly called the *baud rate*.

Some serial communications elements take care of this automatically. The signal from the transmitting element (e.g. a terminal) will arrive carrying a certain number of initial data bits. If the software in the receiving element (e.g. a host computer) is able to communicate at several different baud rates, it can be configured to interpret and lock onto the incoming data baud rate and reply at that speed. Otherwise, the two must be pre-set at the same baud rate before transmission is attempted. The latter is most often the case when using Zenith/Heath computers for communications.

The slowest rate at which the Z/H-19 and Z/H-89 can communicate serially is 110 baud (bits/sec); the fastest is 19200. The Z/H-100 will operate from 45.5 to 38,400 baud. Generally, the higher rates of transmission are limited to communications with printers, hardware spoolers and similar localized elements (see chapter 7). Microcomputer data communications to remote elements is also possible over telephone lines, but the baud rate of such transmissions is usually lower.

An external device—called a modem (see below)—can be used to establish computer data communications over telephone wires. The modem must be compatible with the desired rate of data transfer. In ordinary telephone networks, the rate of data transmission is usually very slow—300 baud is the most common. Higher baud rates can be as just as reliable, but they place a greater demand on the system; and faster communication requires use of a more rugged, expensive modem. Nonetheless, a high-rate modem uses the telephone link for correspondingly fewer minutes and the extra cost may be recovered in the form of lower telephone and timesharing charges.

In the Z/H-19 and Z/H-89 the terminal baud rate (the rate of communications between the terminal section and the main CPU) is set manually. A switch (S401) on the terminal logic board can be adjusted to produce baud rates of 110, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200 and 9600 (an undocumented 19,200 baud is also possible but not recommended). The baud rate may also be selected through software control; however, after a SHIFT-RESET the computer/terminal will return to the baud rate selected by switch 401.

COMMUNICATION WITH OTHER COMPUTERS

A Z/H-19 or a Z/H-89 (operating under software that emulates terminal communications) can only understand keystrokes, display characters and transmit or receive data, but when it is connected to a computer, it has access to the entire power of that computer, no matter how big or small. A terminal can easily be connected with remote computers that have been designed to accept telephone data communications. In such a case, communication is made over great distances through a telephone circuit, in much the same way that voice or telex communications are made. Additional hardware is required—and when using a Z/H-89 or Z/H-100 computer, some inexpensive software is also necessary—but the rewards are usually great; extending one's reach is a fundamental human trait.

MODEMS

A *modem* lets you send and receive computer data over telephone lines. It is so called because it *mo*dulates and *dem*odulates sound waves in a manner that computers can interpret. A modem is generally connected to the terminal/computer through the RS-232C serial port with a cable similar to the one used to send data to a printer. Modems are available in two different forms: acoustic coupled and direct coupled.

ACOUSTIC COUPLING

When you use a telephone, you talk into one end of the handset and listen through the other. A modem is built to do the same thing—to talk into the microphone end and hear through the speaker end.

An acoustic modem produces audible sounds that the microphone in the handset can receive and pass along. This sound is like a high-frequency buzz, but well within the range of frequencies that the telephone is capable of transmitting. It is an acoustic representation of the data being transmitted by the computer. A tone with a certain frequency represents the presence of a high bit and a tone with a different frequency represents a low bit. By transmitting these frequencies, the modem is sending over the telephone, bit by bit, a representation of each byte of computer data.

DIRECT COUPLING

The acoustic-coupled modem is fairly inexpensive, but not considered entirely reliable as a conveyor of information. Whenever data is converted from digital (electrical bits) to analog (audible signal) form, some noise is introduced, and an acoustic coupler can pick up environmental sounds and transmit them through the telephone line, along with the data. This often results in scrambled and inaccurate data transmission.

An alternative is a direct-coupled modem. In this device there is no microphone and no speaker. The computer data is converted to an analog signal in the same manner, but there is no speaker, and no sound is produced that the user can hear. The telephone handset is not used because a direct-coupled modem is like a telephone unit itself: it plugs directly into a modular telephone jack.

Direct-coupled modems are more expensive than acoustic-coupled modems, but they are more nearly noise-free. They often are capable of carrying data faster—at a higher baud rate—and may have optional features. Some models can even answer the telephone automatically and respond (under software direction) if the caller is another computer. Some modems can be connected to clock-like timers and set to initiate calls, for example, to send data to a remote computer late at night when long-distance rates are lowest. They can also be set up to share the telephone circuit with ordinary voice traffic. One manufacturer calls this kind of product a "Smartmodem."

The Z/H-19 terminal can be coupled to a modem and used to communicate with a remote computer, however, the communication will be limited to the visual display on the video screen: The user will type commands on the screen and the computer will respond with data that is displayed on the terminal screen. However, it is also possible to buy software that will allow the Z/H-89 computer to emulate a terminal. (The Z/H-89 has all of the terminal hardware built in, but this hardware is dedicated to communicating with the onboard CPU; the emulation software will, in a manner of speaking, bypass the onboard CPU and allow the terminal section to communicate with the remote computer.)

Using the Z/H-89 as a "smart terminal" has a big advantage. Though the software forces the terminal section to communicate with the external computer, the rest of the Z/H-89 hardware is still available for the software to use. Specifically, the disk drive can be used to record information as it is received from the remote computer. Therefore, with the proper software (discussed in chapter 6) and a Z/H-89 it is possible to contact a remote computer system and "download" even very large files of information from the remote system and store them on a disk.

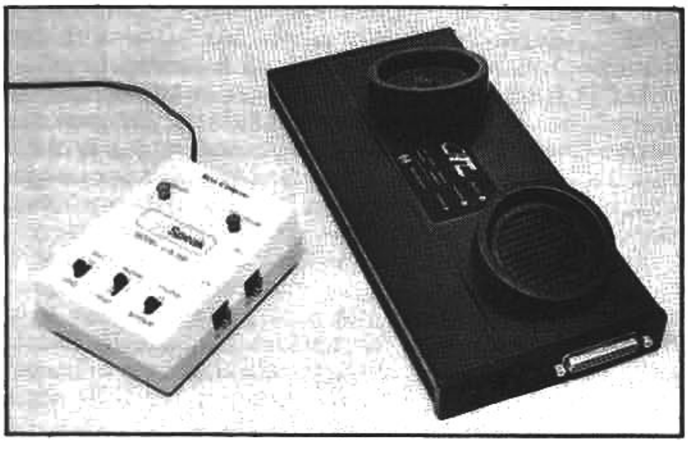
Some consumer-oriented remote computers (see below) also have large software *databases* (a database is computer jargon for a large collection of data) that can be examined by a computer user with a modem. The user can download these programs and add them to his or her software collection.

If the user wants to use only one database, such as a commodity-trading service, he will not need complicated or expensive hardware or software. If, however, he or she wants to communicate with a variety of computers, terminals, and data-bases, or to send text files back and forth in a large network, then the software should be more sophisticated. The specific software may also vary, depending on the baud rate of the terminal and the modem.

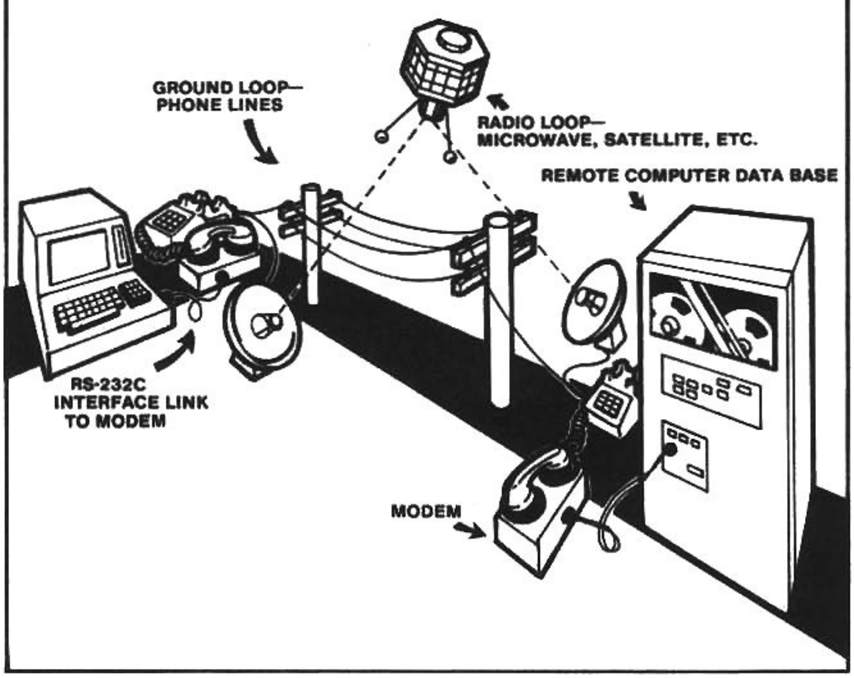
REMOTE DATABASES

Any Zenith/Heath computer user with a modem can dial into several nationwide information networks. All that is required is a computer, an operating system, and a program that will convert the computer into a "smart' terminal (REACH and CPS are such programs, see chapter 6).

According to Cheryl Rhodes, who writes for DataCast Magazine (for CP/M users), "REACH turns a Z-89 computer into a transmitting/receiving terminal..., The network thinks the Z-89 is just another 'dumb' terminal, but the REACH software can actually 'listen' and capture on disk all information sent and received. I stored the results



An *acoustic-coupled* modem (right) has flexible cups to accept the telephone handset. Incoming and outgoing signals are acoustically transferred between the telephone handset and the modem. A *direct-coupled* modem is connected directly to the telephone line with standard telephone modular jacks.



ACCESS TO A REMOTE DATA BASE

Large telephone-accessible subscription databases—such as Dialog, The Source etc.—are maintained in a mainframe computer (or a series of interconnected mainframes) which can store huge quantities of data. Using a small computer or terminal and a modem, subscribers can call the database mainframe to search through the indexed data or download data (copy the data into the user's computer). The subscriber simply calls a telephone number linked to the database mainframe and establishes a communications link through his modem. As the subscriber types commands on his keyboard, the ASCII codes are converted by the terminal into a digital signal; which is transmitted by the RS-232C serial port into a modem; where it is converted to asynchronous analog signals. These signals are transmitted by telephone land-line or microwave-satellite links to a modem at the database mainframe, where they are converted back into digital signals—identical to the ones that left the terminal. When the request reaches the database mainframe, the big CPU searches through a series of information files stored in disk "packs" (multiple hard-disk drives) to find the requested information. These packs are updated regularly by personnel who maintain the database. When the requested data file is located, it is copied into the mainframe RAM, and as the user requests specific information, it is transmitted back over the previously-established communication path.

of several searches on one single-density, single-sided 8-inch floppy disk. REACH...can also 'spool' the information to a printer... Perhaps the most useful feature of REACH is the ability to send text from a disk file as if I had typed it. I can save my information search strategy in a disk file rather than pay the network to store it on their disk file."

The Heath Users Group operates a bulletin board (HUGBB) which any Zenith/Heath user can read by dialing into a network called MicroNet (see page 73 for additional information on CompuServe's MicroNet). Terry Jensen, who runs HUGBB from Heath's headquarters in St. Joseph, Michigan, says users dial into MicroNet and can access the Heath Users Group and a number of other Special Interest Groups (SIGs). A menu will guide the user to the options for reading bulletins, scanning message headlines, and leaving or retrieving individual messages.

To participate fully in the message services, a user must be a member of the Heath Users Group, but non-members may examine a short version of the menu, and leave messages for the system operator (known on the network as "SYSOP"). Non-members are also encouraged to read option "M" (membership information) before leaving a message or exiting from the bulletin board.

Additional information about joining MicroNet can be obtained from local Heath Computer Centers and from Zenith Data Systems dealers.

There is also a HUG bulletin board on The Source. The Source is probably the largest microcomputer network in the U.S. It offers not only message/bulletin board service (several of them, for different computer user groups) but also games, educational and computational software, and other programs.

Based in Virginia near Washington, D.C., The Source was in financial trouble only a few years ago, during the early days of the microcomputer boom. It was rescued by none other than the Readers' Digest: the giant publishing company bought The Source and kept it afloat. Though it has not yet offered any part of the Digest's enormous inventory of printed materials over the network, there is speculation that eventually The Source may become a pioneer in "electronic publishing."

It costs $100 (a one-time fee) to join The Source, plus—as all networks have—there is a fee per minute or hour that the user is connected to the system, with rates determined by the time of day. Generally, local time is applied, and the rates are highest during local business hours, with proportionally lower rates during the evenings and nights. Like telephone calls, connect-time rates favor night owls and insomniacs.

Additional information about The Source is available from practically every microcomputer retailer, or from advertisements in personal computer magazines.

DIALOG is a scientific and technical research service operated by Lockheed, the military aircraft manufacturer, in California. DIALOG holds databases on agriculture and food sciences, business and economics, energy and environment, government publications, humanities and the arts, law, medicine, physical sciences, social sciences, and engineering. These are assembled by computer-trained librarians who comb through conference papers, abstracts, professional journals, newspaper and magazine articles in every field.

DIALOG operates seminars for new users, and offers some free "search time" as an incentive to make use of the system. As with any large database, finding something in DIALOG depends on the user's familiarity with the material. This is not a place to go browsing because DIALOG contains only the abstracts or bibliographical references. After a source is discovered, the user can request that a professionally reproduced hard copy of the referenced document be sent from DIALOG's document-supply service. Search time is billed at different rates, depending on the specific part of the database used.

Up to now, we have described the functioning of a terminal and the important communications link between a terminal and a computer. But the real power of a microcomputer lies in the abilities of the CPU microprocessor to run a vast array of software. In the following three chapters we will closely examine the CPU hardware and compatible software for the Z/H-89 and 100 microcomputers.

THE ZENITH ZT-1

The Zenith ZT-1 is a terminal designed primarily for remote communications with data networks. It is of a unique design and does not share any major hardware with the Z/H-19 terminal.

The ZT-1 is a specialized terminal, designed with a built-in "smart" modem. And though it is not a computer system, it can be programmed to perform up to 26 specific terminal-like tasks, and it will retain these instructions in a permanent memory. For instance, it can be programmed to automatically (at any predetermined time) establish a telephone hookup, dial a remote computer telephone network, enter a password, and begin to send or retrieve information, all with no assistance from a user.

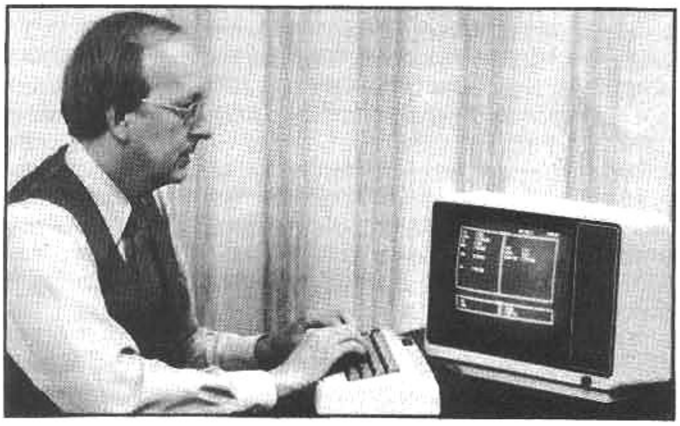
Firmware within the ZT-1 displays a list of user-selectable tasks for the user to view. These tasks can be coordinated to work with the special requirements of a remote database. The software uses English-language phrases and instructions, and responds to special keys like HELP (display the menu of instructions) and QUIT (end this session). "A terminal like this is more dependent on internal programming than on hardware," said a Zenith executive when the ZT-1 was unveiled, in March, 1982.

The ZT-1 also includes a parallel port for a printer. This allows data that is retrieved by the ZT-1 to be transmitted to the printer and output as hardcopy, even while the user is absent. The CRT screen is a standard Zenith ZVM-121 green-phosphor monitor—widely used with computers that have no built-in video display.

This terminal is an outstanding choice for an individual who is interested only in remote communications: it is an inexpensive and very powerful terminal. Users may also encounter the ZT-1 under other brand names: Zenith is selling this smart terminal to companies that want to establish private intra-company communications networks and databases. Commercial "information-providers," such as the Videotext network, may also make ZT-1s available to their clients on a rental or purchase basis.

ELECTRONIC MAIL

Automated computer-to-computer communications makes it possible to transmit "electronic mail" during the late evening hours, when the cost of telephone traffic is the lowest. This interesting subject can only be treated briefly here, but to give the reader some idea of the future potential for remote computer communications, we will look at the innovative system established by Zenith to transmit electronic mail to its many authorized dealers and distributors.



The Zenith ZT-1 is a sophisticated communications terminal with many features that make it ideal for remote "electronic mail" message transfer. Because of extensive built-in firmware, the ZT-1 can be "programmed" to perform automated tasks selected by the user from a menu display.

The system, called ZGRAM, began in 1982. It permits a dealer to use a Z/H-89 or Z/H-90 for conventional word and data processing and as a sophisticated communications center. Using the proprietary ZGRAM software, the Zenith computer gains access to Telemail, a data communication network service of GTE. Messages are prewritten using Magic Wand, a word processing program (see chapter 6). These messages are then encrypted before transmission (for security) and addressed to a specific recipient who is also a member of the Telemail network.

At night, the unattended computer automatically dials the main Telemail computer at a predetermined time, and transmits each encrypted message to the "address" for which it is intended, and the main computer stores these messages under each specified recipient's address. This message may be addressed to one recipient, to several recipients sharing certain characteristics (such as geographical proximity), or to all Telemail users. The Telemail computer then returns to the calling-computer any messages that may be waiting in its address space. These return messages are stored on a floppy disk and, in the morning, the user examines the floppy disk to "read the electronic mail." Messages can later be printed out for a permanent record.

Alternative approaches to electronic mail include the concept of "computer conferencing," which requires a large "host" computer. One such system, called EIES, permits many users to share the same message file. For example, a group of researchers in different parts of the country can collaborate to write an article which incorporates all of their respective findings. They work at their own terminals, but the text and messages they send to the master file remain for all their colleagues to see.

Another use for this capability is called "Delphi" polling, named for the Greek oracle. When several people are trying to reach a consensus on a difficult issue, group dynamics and strong leadership may interfere with some members' impartiality. To overcome the effects of these pressures, group members enter their responses to the issue individually, at remote computer terminals. The host computer, or its operator, tallies the responses and releases the findings to all the group members simultaneously.

CHAPTER 4 THE 8-BIT/16-BIT Z/H-100

Designed from the desktop up as a powerful, high-speed business computer, the Z/H-100 series offers the best of existing 8-bit technology and the ability to upgrade with 16-bit software. The Z/HF-120 (pictured here) is an all-in-one microcomputer with two main microprocessors (8-bit and 16-bit), 128 Kbytes of RAM, one or two high-density 5.25-inch disk drives, and a black & white display. An external color display can be attached with a minor upgrade.

THE 8-BIT/16-BIT Z/H-100 COMPUTER SYSTEM

The Z/H-89 system was designed in 1977 for what was then predominantly a hobby market. However, the new Z/H-100 system has been designed from the desktop up for business applications. Zenith's self-proclaimed target markets for this new system are: (1) small business applications—accounting, database management, word processing, etc.; (2) large businesses interested in using the Z/H-100 as a "smart" terminal for a multi-user mainframe computer system; (3) and computer "packagers" who want to customize Z/H-100s for special applications.

The purpose of this chapter is to acquaint the user with the features of the Z/H-100 system. At times we will make comparisons with the Z/H-89 system. However, the Z/H-100 was designed with a different set of criteria and the benefit of more mature microcomputer technology. As a result, it is a more capable and more expensive machine. The comparisons have little validity, therefore, except to help the prospective user understand the extended potential of the Z/H-100.

The Z/H-100 is irrefutably a more powerful, faster, and more "expandable" system than the Z/H-89, so a potential buyer may be tempted to assume the Z/H-89 is obsolete. Not at all true! Spokesmen for Zenith and Heath vigorously claim that they will continue to manufacture and support the Z/H-89 system in the "forseeable future." They also indicate that the pricing of the Z/H-89 will become "increasingly competitive." A reasonable interpretation of these two statements would lead one to believe that the Z/H-89 is going to be sold for at least a few more years and that the price is going to decline significantly.

There is currently an abundance of software and hardware accessories available for the Z/H-89, and it is unquestionably a "proven machine." This is ample reason to carefully consider the capabilities of the original all-in-one. For many day-to-day applications it is, and will continue to be, a very useful, efficient, low-cost personal computer.

SYSTEM CONFIGURATIONS

The Z/H-89 system was designed as an "all-in-one microcomputer," and the Z/H-100 retains this self-contained philosophy. There are two basic system configurations. A model designated as the Z/HF-110—and often referred to as the "low-profile" model—is a two-module design: with the keyboard, CPU, floppy disk and support elements in one enclosure and a separate display (monitor). This computer has full color display capabilities. A second basic system configuration is designated as the Z/HF-120—and is often called the "all-in-one." A stand-alone system with all of the elements incorporated in one enclosure, it has a monochrome (non-color) display, and without modification, it does not have the capability to produce color displays.

One might assume from appearances alone that the two models are dissimilar, but internally they are almost identical. The non-color Z/HF-120 can be easily upgraded to gain the color graphics capability of the Z/HF-110. It only requires the addition of a few integrated circuits (ICs) to the main printed-circuit board (the sockets are already provided). With this modification, and a separate color monitor, the all-in-one will perform exactly like the color

Z/HF-110. Conversely, in a black-and-white mode, the low-profile model functions similarly to the all-in-one.

The stimulus for producing two different models came from the increasing—but somewhat specialized—demand for high-quality color graphics. There remains little doubt that color graphics is useful in certain instances, but some applications will gain only modest benefit from color displays. For example, word processing and accounting applications have little need for brightly colored displays. A user might find them distracting or tiresome—though accounting programs, spreadsheets, and even word processors and spelling checkers that will use color displays are currently under development. To reduce the cost of systems used primarily for business applications, Zenith decided to produce an all-in-one model with a standard green-on-black display (during long periods of viewing, a green display can be less tiresome than a white-on-black display).

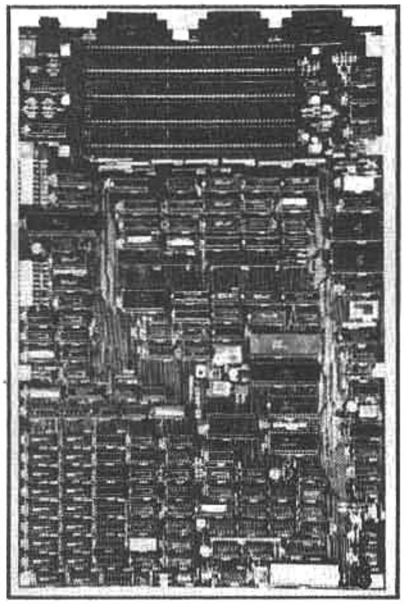
The low-profile Z/H-110 color computer incorporates the latest color-graphics technology available in a microcomputer. It uses a remote color monitor to reproduce high-resolution color graphics, with up to 144,000 points of light (pixels), displayed in any one of eight colors. This capability (with the proper software) will allow the Z-110 to perform advanced functions, including full-color reproduction of intricate three-dimensional structures (a technique suited to computer-aided design applications).

When Zenith introduced the Z/H-100 system, their engineers also discussed plans to offer an additional hybrid model—tentatively designated the Z/HF-130. It will have the low-profile color hardware in the all-in-one enclosure. This combination should appeal to users who want the color capability but do not desire a separate color monitor. In effect, it will be an upgraded and more expensive (because of the additional color hardware) all-in-one.

In the forseeable future, Zenith will offer both color and monochrome systems, even though they will have many overlapping capabilities, and as we shall soon see, the ZDS engineers have opened the door for virtually unlimited expansion of both models.

THE CPU AND MEMORY

When the Zenith design group considered the criteria for a new computer, one of the highest priorities was to increase the capacity of the CPU microprocessor. This could readily be achieved by switching from the Z80 processor used in the Z/H-89 to a faster and more powerful 16-bit processor, but the final decision was not a simple one.

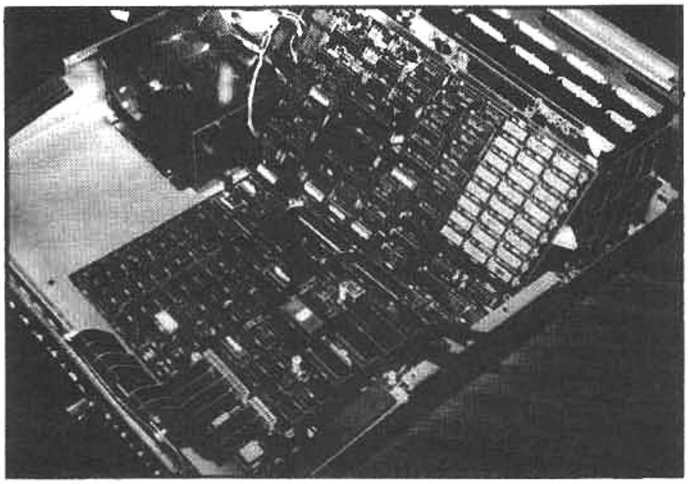


The main circuit board is "home" for the 8085 and 8088 processors, the keyboard processor, several ROMs (containing system instructions), and considerable support circuitry. Both the main board and the video board will be supplied fully assembled and tested in all kit versions of the Heath H-100. This greatly reduces the complexity of the kit and ensures the kit-builder a high degree of success (Heath has a proven history of generous support for kit builders).

Let's go back for a second and look at the 8-bit Z80. The term *8-bit* refers to the length of each byte of data processed during one machine cycle. In other words, the 8-bit byte is the most elementary *meaningful* instruction or data element that an 8-bit processor (like the Z80) can recognize. But when the Z80 is storing and retrieving data, it assigns each data byte a 16-bit (two-byte) memory address. This address is a "pointer" to indicate the location of each and every byte of data stored in RAM memory. Consequently, the maximum size of RAM is limited to 65,535 memory locations—as this is the maximum number of locations that can be represented by the two-byte Z80 binary address. (For details of binary numbers and 8-bit technology see chapter 8.)

Some of the restrictions created by this memory limitation are: (1) programs cannot be larger than about 50 Kbytes; (2) the size of a document that is being edited in memory is often limited to about 20 Kbytes or less (about the size of the longest chapter in this book); (3) the speed of processing is slowed because complex programs must be broken into segments and "chained" in and out of memory (for example: often there is not enough room for all programs in an accounts receivable package to reside in memory simultaneously, so the program is broken into utility modules and a master module selects and loads these utility modules into memory as needed). On the positive side, however, is the availability of existing software for the Z80: there are hundreds of reasonably inexpensive programs available for this processor.

On the other hand, a 16-bit processor is more powerful and faster because it can "process" more bits per cycle of operation. As a result, it has increased data-handling efficiency and it can utilize larger binary numbers to address many more memory locations. However, even though the 16-bit processors have been around for quite some time (relatively), they were not practical in the microcomputer—because of high cost—until 1981. And, though these processors are undeniably powerful, when the Z-100 was designed, there wasn't much 16-bit software available for the microcomputer market. (Unfortunately, existing 8-bit software can not be used on 16-bit processors unless the programs are rewritten in 16-bit code.)



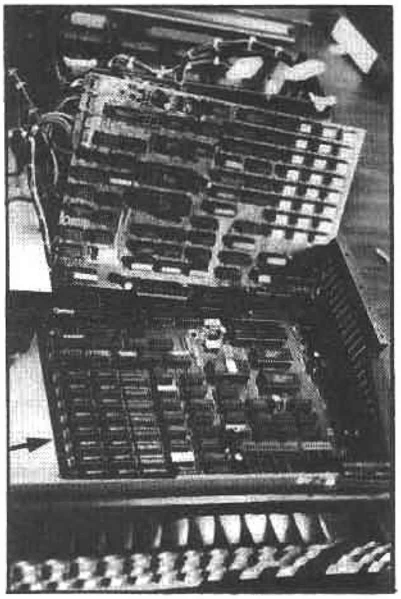
Internally, the Z/HF-110 and Z/H-120 are virtually identical, though the 110 lacks the necessary hardware to support a color monitor. Two "coprocessors" (indicated by arrows) are used: an Intel 8085 handles the 8-bit processing, ensuring compatibility with CP/M-80 and its enormous software base; while an Intel 8088 allows the Z/H-100 to run powerful 16-bit software under either the CP/M-86 or MS-DOS 16-bit operating systems. The high-density main board (lying flat on the bottom of the cabinet), video logic board (above the main board, leaning toward the rear), and the S-100 expansion interface (at the rear of the main circuit board) were all arranged by Zenith engineers to put the maximum computing power in a compact package.

As Zenith engineers considered the new design, the situation was clear: 8-bit processors are limited by the constraints of memory addressing and throughput (how fast a job can be completed), but benefit from a large pool of available software; 16-bit processors are faster and capable of addressing much more memory, but there is limited software currently available. The solution: build a system with two main processors; one that allows the user to run most (if not all) of the current 8-bit software, and a second processor that can run the new 16-bit microcomputer software as it becomes available. *In effect, build a hybrid—providing an effective transition between the existing 8-bit technology and the more powerful 16-bit technology of the future.*

However, we wish to make one thing clear: although the Z/H-100 is a hybrid, it is not a "kludge." The Z/H-100 is a significant advance over the Z/H-89 in some important aspects. For 8-bit compatibility, the Z/H-100 uses an Intel 8085 microprocessor that will run software written for the 8080 processor (the predecessor of the Z80). For 16-bit operation, the Z/H-100 uses an Intel 8088 microprocessor that will execute program instructions more than seven-times faster and can address 16-times more memory than the Z80 in the Z/H-89.

THE 8-BIT INTEL 8085

"Compatibility" was unquestionably the overriding concern throughout the Z/H-100 design. A key example is the selection of the Intel 8085 to handle 8-bit processing. It is compatible with programs written for the 8-bit Intel 8080 (and it is electrically compatible with Intel 8088 16-bit co-processor, see below). *This will allow a significant portion of the existing 8-bit 8080-coded software to run on the Z/H-100.*



The standard Z/H-100 is equipped with 128 Kbytes of RAM. An additional 64 Kbytes may be added by plugging memory chips into available sockets on the main circuit board. Still more user RAM can be gained by plugging S-100 memory cards into the expansion buss—up to a maximum of 768 Kbytes. However, plugging new 256-Kbit RAM chips (instead of the standard 64-Kbit chips) into the 24 DIP sockets on the main board allows expansion to 768 Kbytes of onboard memory *without* using the S-100 expansion buss.

(Note: most of the assembly- and machine-language programs available for the Z80-equipped Z/H-89 are actually written with the Intel 8080 instruction set. For the most part, these programs were intended to run on both the Heath H-8, which used an 8080, and the Z/H-89, which used a Z80. As stated earlier, the Z80 is a more capable processor and it is designed to be fully compatible with the 8080 instruction set—the Z80 was selected for the Z/H-89 because of this fact. However, *the full Z80 instruction set is more extensive than the 8080 set, and programs written for the extended Z80 instructions will not run on the 8080.* Consequently, existing 8080-coded programs should be compatible with the 8085 used in the Z/H-100, but not all Z80-coded programs will be compatible!)

Compatibility, however, is far from the whole story. The 8-bit side of the Z/H-100 is substantially faster than the Z/H-89. The speed comes from several improvements: (1) At 5 MHZ (mega-hertz) the master system clock (which controls the rate at which data is "processed") is more than twice as fast as the Z/H-89 clock rate (2 MHZ), allowing the 8085 to execute 5 instructions for every 2 in the Z/H-89; (2) The 8-bit operating system uses an enhanced "disk deblocking" routine (de-blocking refers to the way blocks of data are read from and written to a disk storage device), providing faster data storage and retrieval; (3) The 8085 uses additional instructions, beyond those for the 8080, that are more efficient—when a program is written using these instructions, execution time is reduced. *The sum of all these enhancements on the 8-bit side of the Z/H-100 allows as much as a three-fold performance improvement over the Z/H-89 system.*

THE 16-BIT INTEL 8088

On the 16-bit side, compatibility is once again the keyword. An Intel 16-bit 8088 processor was selected to gain overall compatibility: the 16-bit 8088 and the 8-bit 8085 are electrically compatible, and, therefore, they can be interfaced readily to the same support components. But the story doesn't end here.

Software is also important. Tom Dornback, Vice President of Software Development at Zenith, insisted on using the 8088 processor because it is—first of all—an excellent 16-bit processor. However, in private discussions he admits there was a second consideration: it is the same processor used in the IBM Personal Computer. At first it may seem strange that he would favor the same processor used by a competitor; but the strengths and limitations of a computer are determined to a great extent by the availability of high-quality software, and with hardware and software compatibility, Zenith and IBM users will be able to share an extensive and diversified software pool!

The 16-bit 8088 takes the Z/H-100 one step closer to the power and computational capabilities of the mainframe computer. The 8088 uses a sophisticated data-handling architecture that speeds program execution, and it has the ability to address huge amounts of memory. This means that word processing, accounting, data management, and other programs written for the new processor, will be very fast and powerful. The 8088, with 16-bit software designed to take advantage of these features, will operate about *seven times faster* than a Z/H-89.

RANDOM-ACCESS MEMORY

By utilizing the 8088 processor, which has a 20-bit memory address, the Z/H-100 can access *up to one million memory locations*! The standard system is currently supplied with 128 Kbytes of RAM memory (twice the maximum in the Z/H-89). There are extra sockets on the main processor board to allow expansion to 192 Kbytes (using 64-Kbit chips). By installing additional memory cards in the expansion buss (see system expansion, below), the RAM can be extended to a maximum of 768 Kbytes. However, higher density memory chips are under development, and when these 256-Kbit chips reach the market, the Z/H-100 is designed to allow the user to install a *full 768 Kbytes of system RAM on the main processor board, without using any of the expansion slots.* According to a Zenith/Heath spokesman, there are plans to offer a Z/H-100 model configured in this fashion as soon as the chips are available.

GRAPHICS

The Z/H-100 series computers are truly leaders in high-resolution color graphics—far surpassing the graphic capabilities of the Z/H-89. In fact, the Z/H-100 does not have a non-graphics mode; the normal display of alphanumeric characters (letters and numbers) are generated by *bit-mapped graphics.* Bit-mapping allows the programmer to address each and every one of 144,000 points of light, called *pixels*, on the screen. When alphanumeric or graphics characters are displayed in a normal mode, they are formed from an 8 x 9 pixel block, and in the color mode each of the pixels can be switched to one of eight possible colors: red, green, blue, white, black, cyan, magenta, and yellow.

The Z/H-100 is a "software definable" machine not only in the variety of operating systems and applications programs that can be run, but also because many internal functions of the system are controlled by programs and data tables recorded in ROM chips. This is also true of the graphics. Normally, the system displays 640 pixels per line on 225 lines per screen, but by re-programming the graphics ROM (a 6845 IC), the system display can be "interlaced" (double-density scan line) to gain a high-resolution mode of 640 pixels by 500 lines (black & white, with eight levels of gray).

This high-resolution mode will permit the display of pictures that are "TV-like" in appearance and indistinguishable from normal television displays of still pictures. These computer-generated pictures can be stored on disk and recalled as desired. They could also be intermixed with conventional graphics displays to extend the interactive nature of the computer into areas not previously possible with a microcomputer.

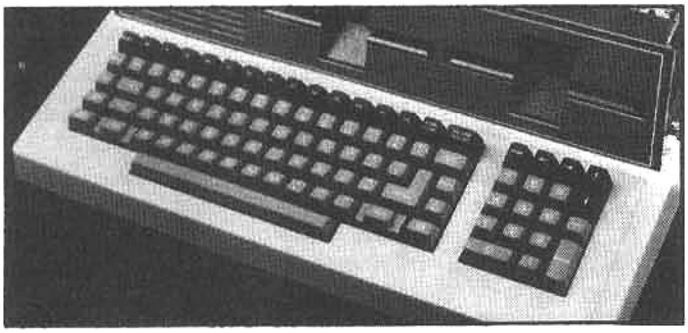
The graphics capability of the Z/H-100 will be useful in many industrial, scientific and educational applications; however, Zenith and Heath have indicated that they have no immediate plans to support this option because of the complex and special nature of the programming required. Nonetheless, the hardware capability is there, and independent suppliers are sure to develop many useful graphic applications. We suspect that hobbyists will also pursue these capabilities, and eventually graphics software and accessory hardware will be available for reasonable prices.

THE KEYBOARD

Along with the display screen, the keyboard completes the vital communication link with the user. A good keyboard—one that is well designed, smooth and comfortable—will reduce operator frustration and speed overall throughput. Toward this end, the Z/H-100 system has incorporated several advances over the Z/H-89 system. The new keyboard uses a sculpted key design that fits the fingertips, and each key has a smooth, stick-free action. All of the keys produce an auto-repeat signal when they are held down for longer than one second. The keyboard also has a FAST-REPEAT key that will increase the speed of the repeat mode (for extremely fast cursor movement, etc).

The keyboard layout is conventional, with no unusual or distracting key placement. The 13 keys in the top row are designated as special-function keys, and can be used by the applications software to ease and speed user control. There is a separate 18-key numeric keypad with cursor-control, decimal point, minus (-), and ENTER keys. In addition, the main keyboard has a HELP key, which most of the new Zenith/Heath software packages will support. When the user presses the HELP key, a variety of helpful information will be displayed (the specific contents will depend on the program in use at the time). The keyboard also incorporates a power-on indicator and an enlarged RETURN key.

The keyboard uses its own processor (an 8035) and, like the rest of the Z/H-100, is software controlled (by instructions in ROM). The standard programming produces three modes of operation. The first is the standard



The keyboard is logical, functional, and very powerful. The design centers around a conventional typewriter layout, and the auxiliary command keys will not hinder a touch-typist. Each key produces an auto-repeat if held down, and a fast-repeat key can step up the repeat rate to speed cursor movement. Thirteen special-function keys are included on the top row of the main keyboard and an auxiliary 18-key numeric keypad eases data entry and cursor positioning. The keys are nicely sculpted and have a smooth, positive action.

The main keyboard has people-sensitive features. The enlarged RETURN key—common on top-of-the-line electric typewriters—will be appreciated by fast typists; note the large SHIFT keys and the elimination of any auxiliary keys between the central array and the SHIFT keys (some computer keyboard designs place auxiliary keys inside of the SHIFT keys, a source of constant irritation for operators familar with conventional typewriter keyboards). A HELP key (arrow) is also included; pressing it will cause some software to display helpful "menus" or other information.

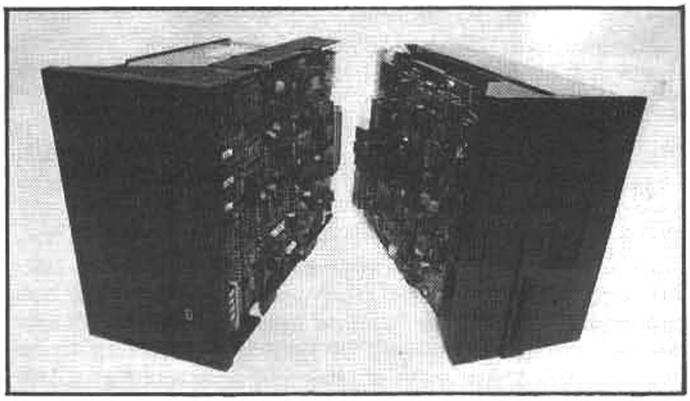
ASCII mode, in which each key—including all the function and numeric keys—produces a unique 8-bit code. This code generation simplifies programming and ensures that programs will be able to interpret or address each key. The second mode is a Z/H-19 compatibility mode (once again, note that word compatibility). This mode assures that software written for the Z/H-19 terminal and Z/H-89 computer will work on the Z/H-100. The last mode is an "up-down" mode. In this mode each key produces an 8-bit code when depressed and a different 8-bit code when released. This special mode will find its application in games and educational programming.

Because of the software control of the keyboard, the keyboard ROM may be programmed to redefine any key to perform any desired function. This feature may be useful to system "packagers" if they wish to reconfigure the Z/H-100 keyboard for special requirements).

MASS STORAGE

Both the all-in-one and the low-profile models are equipped with a single 5.25-inch floppy-disk drive. As an option, a second drive can be installed in the system enclosure. The standard drive is a double-side, double-density unit, with a 48tpi (track-per-inch) configuration.

The numeric portion of the auxiliary keypad is laid out the same as a conventional 10-key adding-machine keypad. The ENTER key is supported by many software packages and is enlarged for easier use. CURSOR keys and a HOME key complete the functional layout.

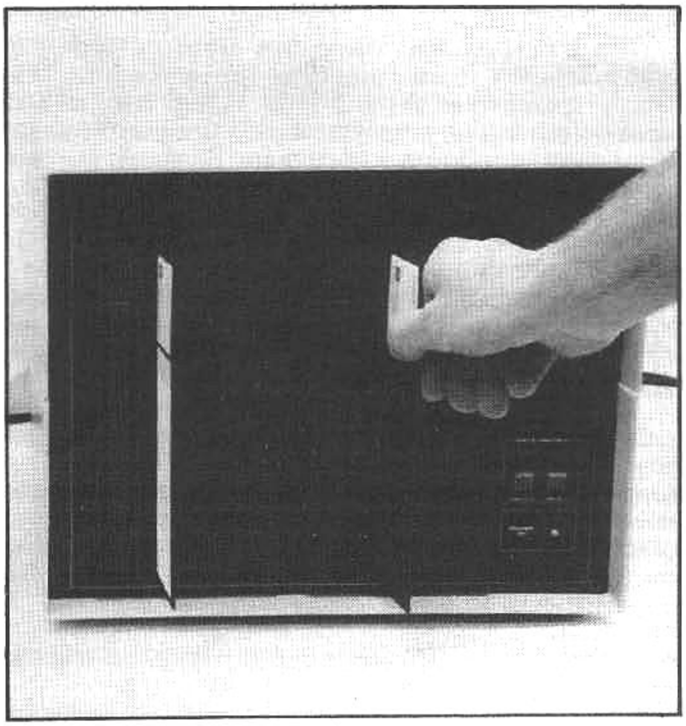


The all-in-one and the low-profile support one or two onboard high-density 5.25-inch floppy-disk drives. A user may wish to replace one of the onboard floppy drives with a 5.25-inch hard-disk (left), providing 5 megabytes (5,000,000 bytes) of fast-access storage. With one 320-Kbyte floppy and one 5-Mbyte hard-disk mounted in the cabinet, a Z/H-100 has more than 52 times the storage capacity of a Z/H-89 with a standard disk drive (hard-sector).

This format will store up to 320 Kbytes, about three times the storage provided by each disk in the standard single-side, single-density Z/H-89 disk system.

The electronics that control the floppy disk drive(s) are mounted on a circuit card that plugs into the expansion buss provided in the Z/H-100 (see system expansion, below). This disk-controller card will support four 5.25-inch drives and four 8-inch drives, in any combination of densities (single or double) or sides (single or double). Since the main enclosure will only hold two 5.25-inch drives, all additional units must be mounted in accessory cabinets.

Zenith and Heath offer a hard-disk drive that will mount



The disk-controller card supports Zenith/Heath 5.25- and 8-inch disk systems and most specialty systems, like this dual 8-inch Remex package from Data Compass. The controller uses the standard 50-pin SASI (pronounced "sass-ee") buss that is another standard in the microcomputer industry.

within the Z/H-100 cabinet (in place of one of the floppy drives). The hard disk requires a controller card that plugs into one of the expansion slots. In addition, the hard-disk drive contains a data-separator circuit card that is powered by the Z/H-100 power supply. A power connector for the data separator is built into all Z/H-100 computers, making the hard-disk installation simple—just plug it in and turn it on. The hard disk will add 5 Mbytes of extremely-fast-access, onboard mass storage. Combined with 320 Kbytes of floppy storage, the hard-disk/floppy system will give the Z/H-100 system more than 16 times the storage that can be mounted within the cabinet of a fully upgraded Z/H-89.

This is an especially significant capability. Mass storage is important in many business applications. Virtually all database management software needs a lot of space to store component programs and utilities, and they can quickly fill the remaining storage with the several databases that are usually required (records of customer addresses, etc.). Accounting programs also use a lot of storage to accumulate the amount of data associated with financial records. Advanced word processors are also large, complex programs, and in an office environment they are used to produce and store large numbers of text files (letters, contracts, manuscripts etc.). The more an individual or business uses a computer, the more important disk storage becomes. If the system can not be easily upgraded as the storage requirements increase, it will become a liability.

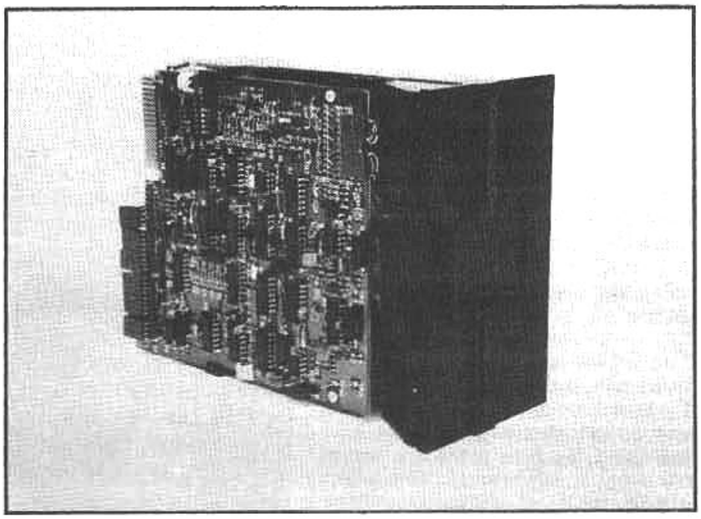
The Z/H-89 is easily upgraded (see chapter 7), but the expandability of the Z/H-100 is remarkable. The Z/H-100 system was designed for applications that require a lot of computer power, and without question, a user who foresees the need for substantial storage expansion should consider the Z/H-100.

INPUT/OUPUT

A computer system must communicate with the "outside" world. Printers, modems, and other input/output devices provide essential functions and enhancements for the computer. In this respect, the Z/H-100 design is outstanding. The standard system includes two RS-232C serial ports that are capable of both synchronous and asynchronous communications (the synchronous mode is compatible with IBM-system protocol) at baud rates from 110 to 38,400 bits per second. A parallel port configured for a Centronics printer is also standard. All of the hardware to drive these communication elements is located on the main circuit board. As a result, none of the expansion slots is required to use the standard ports.

For applications requiring additional serial ports (such as a multiple-user environment), Zenith/Heath will offer a multi-port serial board to fit all Z/H-100 models. This board is designed to plug into one of the expansion slots, and will permit as many as eight additional serial devices to be connected to the Z/H-100. To support this expanded capability, pre-punched knockouts are provided on the rear panel to accommodate up to eight auxiliary DB-25 (RS-232C) connectors.

There is also a light-pen port on the rear panel. It is supported by hardware on the main circuit board. Though Zenith/Heath have no immediate plans to support light-pen operation, this capability will provide interesting potential for independent programmers. Software to support this I/O device could include creative graphics (drawing pictures), education (interactive communication with young students), word processing (rapid cursor movement, block definition, etc.), and much more. The light pen is designed to have sufficient resolution to address *each* of the 144,000 pixels on the display screen.

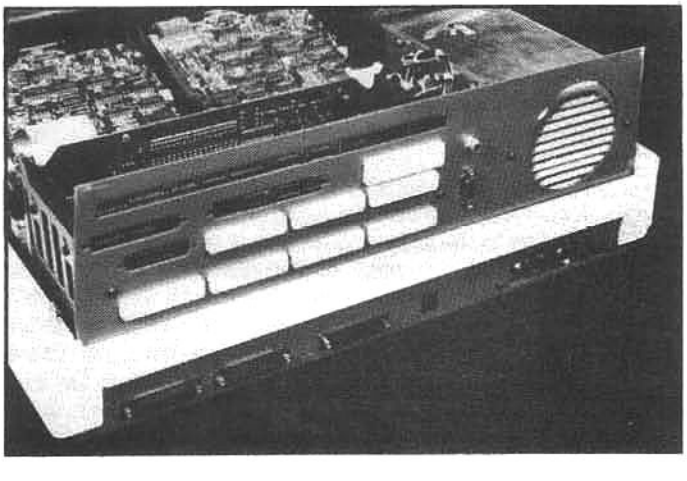


The 5.25-inch disk drives used in the Z/H-100 are Tandon soft-sector, double-sided, double-density units, that will store 320 Kbytes on each disk.

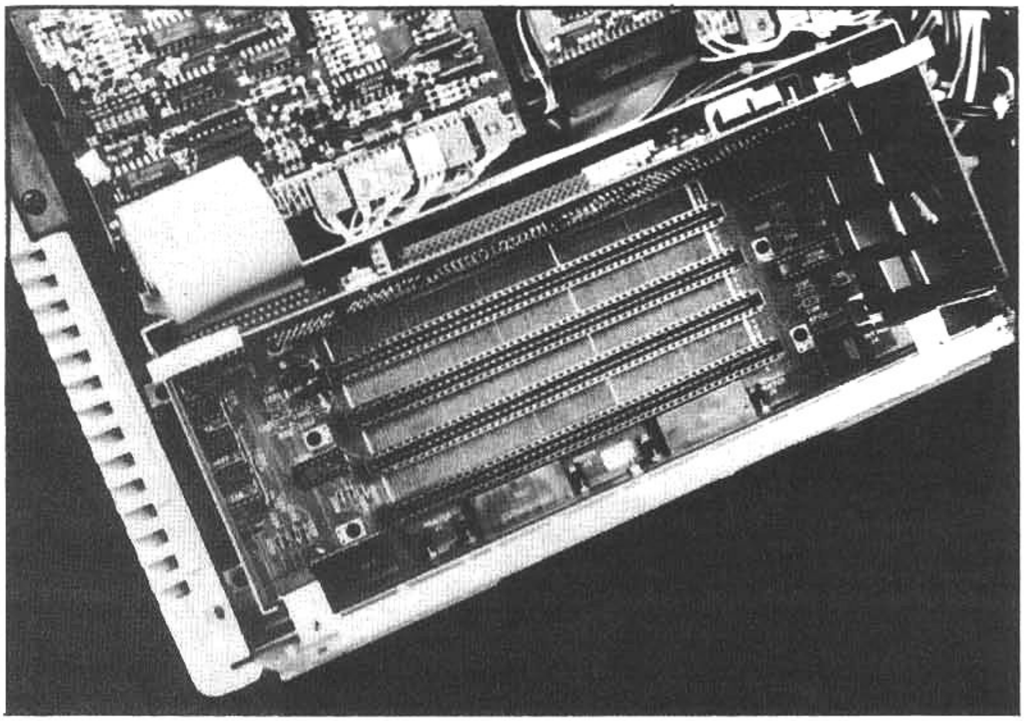
SYSTEM EXPANSION

The success of the Z/H-89 was, in some repects, a stroke of luck. It was first offered as a modest hobby computer (cassette tape storage, limited memory, HDOS, etc.), but when the market for a compact, full-feature computer came into exisitence (almost overnight) the engineers at Zenith and Heath expended considerable effort and ingenuity to "patch" the Z/H-89 to handle more RAM, better disk drives, CP/M, etc.

The memory of these troublesome expansion problems must linger in the minds of Zenith/Heath engineers, because it is safe to say that the Z/H-100 system will

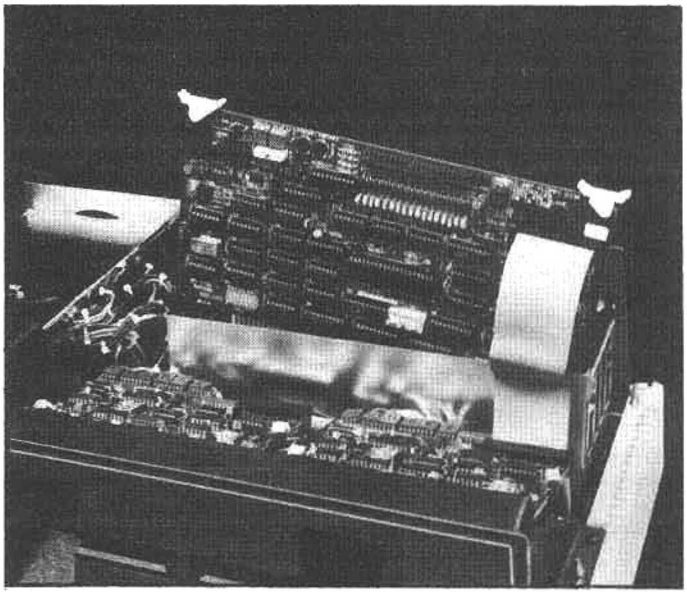


The Z/H-100 series computers can be connected to peripheral devices through expansion ports on the rear panel. The standard system is supplied with two RS-232 serial ports (one for a printer and one for a modem), a Centronics-compatible parallel (printer) port, and a light-pen port. Eight additional punch-outs are provided for RS-232C (input/output) serial ports. The Z/H-100 can be upgraded to support multiple users with auxiliary terminals connected to these expansion ports. The remaining slots are for external mass storage, special instruments, etc.



Perhaps the most unusual feature of the Z/H-100 series is the S-100 expansion buss. The S-100 buss has evolved into a recognized standard in the microcomputer industry (one of the few). Though the Z/H-100 is not a true S-100 computer (because the main components are *not* mounted on S-100 cards), the *expansion* buss allows the Z/H-100 computers to accept a wide variety of auxiliary S-100 cards.

The S-100 expansion buss has five slots, but one of them is used for the disk-controller card—required to link the disk drives with the CPU. So four slots are open for system expansion. The disk-controller card will support four 5.25-inch disks and four 8-inch disks, but extra drives themselves—beyond the two onboard—will require an external cabinet. If a hard-disk drive is installed, one of the S-100 buss sockets must be used for the hard-disk controller card.



readily accept high-density memory, laser-scanned optical disk storage, and any other future technology. In fact, the engineers included a feature that will allow the user almost unlimited upward expansion: they opted to incorporate an *S-100 expansion buss* into the system.

In simple terms, a *buss* can be defined as a connection between two or more computer elements. In some computer designs, a buss is used to connect several circuit boards that are mounted in a series of slip-in slots. Each slot supports one board, and a built-in, multi-terminal, female connector strip at the bottom of the slot makes contact with male connectors on the circuit boards. The slip-in circuit boards can accommodate a variety of hardware (e.g., RAM boards, video boards, voice-recognition, interfaces to scientific test equipment, specialized mass storage, etc.). No modification is required within the system hardware to accommodate these boards: the circuit board is simply slipped into the buss and the system is ready to go (provided, of course, the user can acquire or currently has software that will recognize and support the capabilities of the board).

There are different types of busses used in the computer industry; however, the S-100 is one of the most widely used standard busses. Early microcomputer designs were not integrated onto one or two high-density circuit boards (as are most contemporary designs). Instead, they consisted of several modular boards. To connect them all together into a workable system, the boards were typically mounted in a centralized buss structure. Many early microcomputers (including the pioneer Altair microcomputer, one of the first "usable" micros) were based on a standard that eventually became known as the S-100 buss. These computers were often known generically as "S-100 buss computers." In the ensuing years literally hundreds of expansion and upgrade boards were designed to fit the S-100 buss, and it is still used in many multiple-board microcomputers.

The S-100 expansion buss chosen for the Z/H-100 may be the most innovative feature incorporated into the system. (In fact, Zenith designer Barry Watzman's insistence on incorporating an S-100 buss in the system was one of the primary factors behind the Z/H-100 designation.) The buss allows any user to readily upgrade, customize, and configure a system to suit virtually any requirement. In practice, the basic Z/H-100 has so much onboard power (in the main circuitry) that most users will find little immediate need for expansion, but the potential is there, and specialty manufacturers are certain to develop many special-application boards for the Z/H-100.

To make certain that users can plug in existing S-100 boards, the buss structure meets the accepted IEEE-696 manufacturing specifications for S-100 circuit cards. However, to avoid confusion, it must be noted that this is an auxiliary buss—the primary hardware of the Z/H-100 is contained on the main circuit board, separate from the expansion buss. In other words, the Z/H-100 is *not* a so-called S-100 computer. *The Z/H-100 is a stand-alone computer that can function entirely without the buss; but for expansion purposes, all access that can be performed with a true S-100 buss computer, including direct memory access (DMA) to the system RAM and the video RAM, can* *be accomplished with the expansion buss.*

Zenith and Heath will offer a hard-disk interface and a multi-port serial interface for the expansion buss. A sound generator board with digital-to-analog and analog-to-digital conversion circuits is also planned. This card will, no doubt, find an eager market with those users involved in education or interactive computer games. Beyond these Z/H cards, independent suppliers offer a multitude of cards, including:

Memory expansion, with up to 256 Kbytes per card
Non-volatile CMOS memory
Multi-purpose serial and parallel I/O
Erasable/Programmable ROM programmer boards
System clock/calendars
Modem data communications card
Home controllers (to turn on/off lights, etc.)
Various floppy/hard-disk (fixed and removable platter) controllers
Nine-track and/or reel-to-reel tape storage interface
Prototyping boards

How does the S-100 buss affect a buyer's decision between the Z/H-89 and the Z/H-100? The Z/H-89 is by no means a limited system; however, the S-100 buss gives the Z/H-100 unlimited expansion potential. Nonetheless, the Z/H-89 is ideal for applications that do not need the expansion capability.

SOFTWARE FOR THE Z/H-100

When Intel introduced the 8080 microprocessor in the mid-'70s, the microcomputer was born almost overnight. The evolution of microcomputer software was "somewhat slower." Even today, the micro-chip giants (Intel, Zilog, Signetics, Motorola, etc.) are producing ever-more-sophisticated components faster than programmers—who must spend thousands of hours developing specialized code for each chip—can produce the appropriate software. But the software shortfall is much diminished today, especially for the 8085 and the 8088—the processors used in the Z/H-100 system.

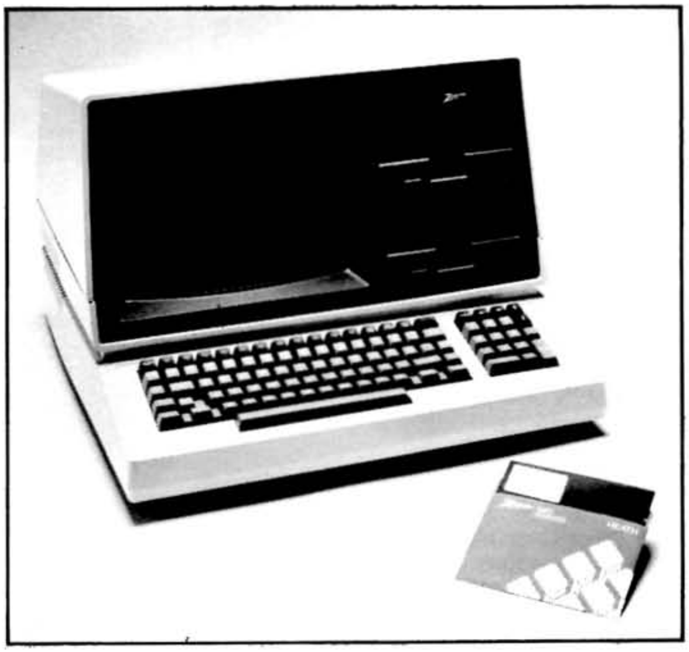
OPERATING SYSTEMS

The 8085 will run software developed for the 8-bit 8080 microprocessor. This has assured the Z/H-100 system continuing compatibility with many existing operating systems and applications programs.

Each Z/H-100 will be shipped with CP/M-85. This 8-bit operating system (really CP/M 2.2) supports one of the widest software followings in the computer industry, with thousands of programs able to run under its control (for a description of operating systems, see chapter 5).

The Basic-85 interpreter—a MicroSoft Basic-language interpreter for CP/M-85—is also supplied with the Z/H-100. This interpreter is identical to the Basic-80 interpreter (available for the Z/H-89), except the Basic-85 "looks" at the operating environment to verify it is running on a Z/H-100, while Basic-80 is designed to look for a Z/H-89 system.

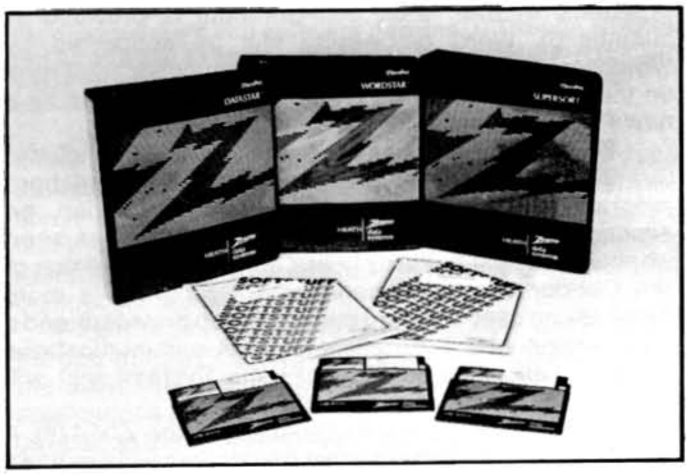
Zenith and Heath will continue to support HDOS—a popular operating system among Heath computer users—by supplying an updated version for the Z/H-100. The new HDOS version 3.0 release will be fully compatible with the Z/H-100. (And, by utilizing 3.0, Z/H-89 users will be able—for the first time—to use the Z/H-67 Winchester hard disk/floppy disk upgrade: see chapter 7.)



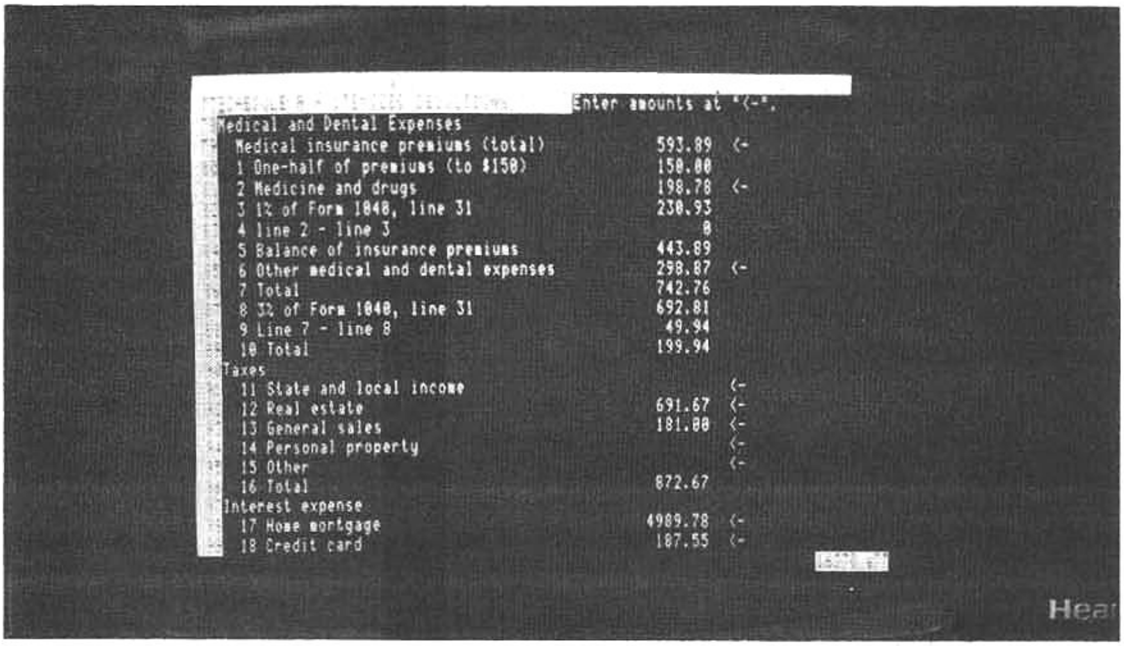
All Z/H-100s are supplied with Z-DOS, an operating system that is compatible with the wide variety of software developed for the IBM Personal Computer, and CP/M-85, a slightly modified version of the standard CP/M-80 (CP/M-80 is identical to CP/M 2.2 used on the Z/H-89). In addition, CP/M-86 will be available as an inexpensive option. That super-fast 16-bit operating system is optimized for the Z/H-100 hardware, resulting in disk I/O speeds up to 300% faster than Z-DOS.

The standard operating system for the 16-bit side of the Z/H-100 will be Z-DOS. This operating system is identical to MS-DOS (the operating system used on the IBM Personal Computer), and suggests virtually 100% compatibility with the IBM PC software.

A 16-bit Basic interpreter, called Z-Basic, will be supplied with the Z/H-100 system. (This interpreter is virtually identical to the Basic-language interpreter supplied with the IBM PC). Z-Basic also supports the capabilities of the Z/H-110 with many powerful commands that can draw circles, lines, squares, or ovals on the



The dual-processor design of the Z/H-100 series ensures that the system will have extremely broad "usability." Conventional 8-bit CP/M-80 programs (8080 code) can be used with the system, as can a number of enhanced 16-bit software packages. For example, an 8-bit version of the popular WordStar word processing program (supporting the cursor-arrow and special-function keys) will be offered by Zenith, and an all-new 16-bit version of WordStar (version 3.4) will provide advanced word processing features not previously implemented on a microcomputer.



```
                                          Enter amounts at "<-".
Medical and Dental Expenses
  Medical insurance premiums (total)              593.89  <-
  1 One-half of premiums (to $150)                150.00
  2 Medicine and drugs                            198.78  <-
  3 1% of Form 1040, line 31                      230.93
  4 line 2 - line 3                                    0
  5 Balance of insurance premiums                 443.89
  6 Other medical and dental expenses             298.87  <-
  7 Total                                         742.76
  8 3% of Form 1040, line 31                      692.81
  9 Line 7 - line 8                                49.94
  10 Total                                        199.94
Taxes
  11 State and local income                               <-
  12 Real estate                                  691.67  <-
  13 General sales                                181.00  <-
  14 Personal property                                    <-
  15 Other                                                <-
  16 Total                                        872.67
Interest expense
  17 Home mortgage                               4989.78  <-
  18 Credit card                                  187.55  <-
```

The Z/H-100 does not lack software support. High-quality business programs—spreadsheet analysis (a sample IRS 1040 worksheet is shown here), word processing, accounting, database management, inventory control, mail-list maintenance, and more—are already available. Feverish activity in software hatcheries ensures that the future will bring more and better programs for the popular microprocessors in the Z/H-100 series.

color display and tint these shapes with any of eight colors.

As an option, Z/H-100 owners will also be able to purchase CP/M-86, a "high-performance" operating system for the 8088. CP/M-86 has been "souped-up" with disk de-blocking techniques that increase data transfer speeds up to 300%. Many programs will be available for CP/M-86, and in applications where IBM compatibility is not a factor, it will provide a substantial performance improvement over Z-DOS.

APPLICATIONS SOFTWARE

The Z/H-100 system will not only benefit from the software compatibility provided by the 8-bit 8085 and the IBM-compatibility provided by the 16-bit 8088, but also from a number of advanced programs available from Zenith and Heath (a detailed look at some of these programs, in their 8-bit configuration, is provided in chapter 6). Word processing will be supported by WordStar version 3.4, a 16-bit "word cruncher" that runs on the 8088. Accounting chores will be supported by a new Peachtree series, also 16-bit for the 8088.

Business programs for the Z/H-100 include: SuperCalc, a color version of the popular spreadsheet program for Zenith/Heath computers; Multi-Plan, an expanded business forecasting program that has been warmly accepted by many users; an expanded version of the Condor database management program (a more detailed and user-oriented manual is also provided); and a new version of the excellent modem communications program CPS (Computerized Phone System) that will extend the "reach" of the Z/H-100.

With all of the hardware capability in the Z/H-100, it seems certain that aftermarket developers will continue to produce a diversity of useful specialty software. In short, Zenith has taken several important steps to ensure that there will not be a software shortfall for the Z/H-100.

AND A HEATH H-100 KIT

One of the most amazing facts in the computer industry is rarely discussed in the trade press or seriously considered by a computer user: *You can build your own computer from a kit, and you can be sure that it will work when the project is completed!*

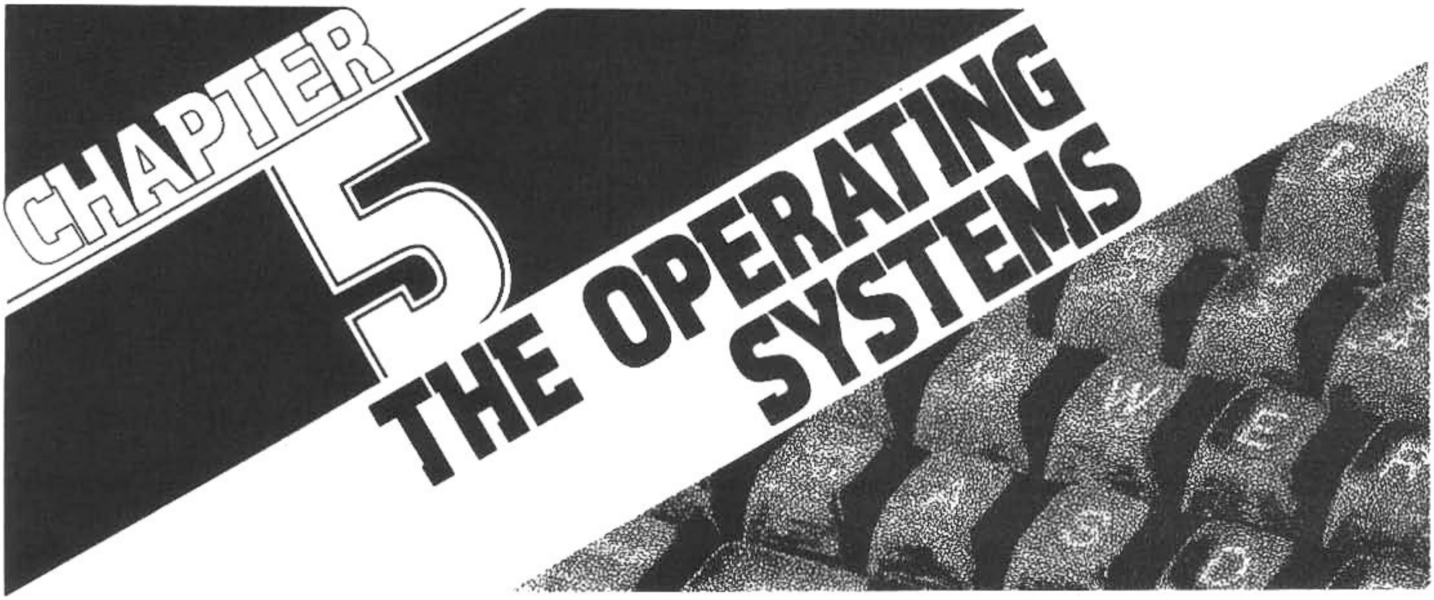
Who in their right mind would attempt to build a kit computer? Thousands of individuals with little or no electronics experience, that's who! Amazing? Perhaps, but when you take note of the superb assembly and operating manuals that accompany the H-100—the Heath kit version of the Z-100—and you further consider the many Heath Electronic Centers around the country that can provide on-the-spot support, the idea is believable.

The H-100, the H-89 computer, and the H-25 printer, like all electronic kits sold by Heath, can be readily assembled by anyone with a modest mechanical background and the ability to read and follow instructions precisely. Moreover, *a person with absolutely no aptitude can tackle the job with a high probability of success.* The cornerstone of Heath success is a commitment that states, in effect, "a Heath kit-builder will not fail," and they back this commitment completely. Heath has a long and well-documented history to prove it.

The job is really not as difficult as it sounds: *At least one-half to two-thirds of the primary assembly is already completed by Heath before the kit is sold.*

Compared to the H-89 kit, the H-100 will require more time to complete (some details of an H-89 assembly are shown in chapter 10). In the H-89 assembly, the builder must put together a small power-supply board and the video board, and "bolt" the primary system assemblies into the enclosure. The H-100 kit requires the builder to assemble the video-raster and disk-controller boards, and mount the primary components in the enclosure. In both kits the important computer boards—the terminal-logic and CPU boards—are assembled and tested at the factory, taking much of the "tricky" work out of the kit. This almost assures the scrupulous builder of immediate success.

But what is the benefit of all this work? A dedicated user can learn a lot about computer operation from the excellent assembly manuals. But for someone that is interested only in the bare operational essentials, there is one other thing to consider: building a kit computer can save roughly 30% of the cost. Take a look at chapter 10 and judge for yourself if it's worth the effort.



CHAPTER 5 THE OPERATING SYSTEMS

HDOS AND CP/M: THE OPERATING SYSTEMS

A *disk operating system* is the most fundamental microcomputer program. It performs the elementary machine-oriented functions. For example, in a disk-based system, such as the Z/H-89 and Z/H-100, the operating system permits the user to change disks in the disk drive, read the file directories of disks and load applications programs from the disk into RAM. The operating system is different from an applications program: the operating system is largely concerned with general computer operation, it is not directly responsible for specific tasks (e.g., word processing, accounting, data management, etc.). Although it is not an accurate description, an operating system can be described rather simplistically as a "program loader."

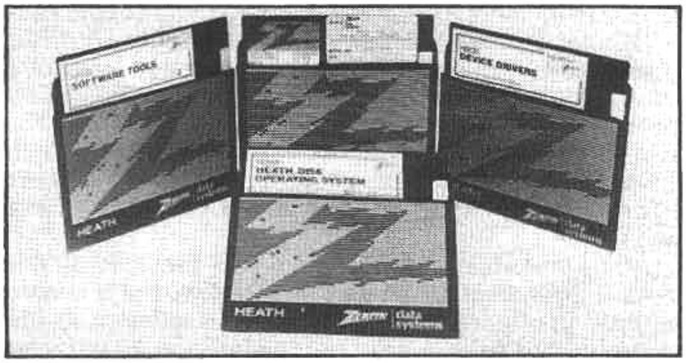
Computers began to need elaborate operating systems when disk drives were developed. In the beginning, each manufacturer had a slightly different way of storing data on a floppy disk, but computer and disk drive designers eventually agreed to standardize certain functions. These standards developed into disk operating systems, or more simply "operating systems." (The two terms are not quite identical, but are comfortably synonymous for our discussion.)

Two primary operating systems are available for Zenith/Heath computers: HDOS and CP/M. HDOS (pronounced "H-doss") is an acronym for Heath Disk Operating System and CP/M is an acronym for the Control Program for Microprocessors.

THE CHAIN OF COMMAND

The difference between the operating system and an applications program becomes increasingly obvious as the user gains experience. WordStar and Electronic Typing, for example, are applications programs (specifically, the application of word processing), but the computer must have a functioning operating system in memory before either of these programs can be successfully executed.

Applications programs generally do not include instructions for general disk-based tasks, such as reading from or writing to a specific track on the rotating floppy disk. These tasks are essential to every program, but to repeat them inside every applications program would make programming unnecessarily tedious. Each applications program would also be very long and cumbersome. Programmers have opted, instead, to group the most common general functions into the operating system and free the applications programs to concentrate on specific tasking.



The Heath Disk Operating System (HDOS) was originally designed to fulfill the needs of early hobby-oriented H-8 and H-89 users, but much credit must be given to Gordon Letwin, the chief writer, because HDOS has remained a powerful, user-friendly, very-reliable operating system. Heath makes the internal code of HDOS available to users and programmers (the code for many other DOSs is considered proprietary) and Heath provides a wealth of valuable information for programmers and software designers. Though HDOS is not as popular as some other operating systems, it is a very capable product that Zenith and Heath have supported with updates and new releases.

When the user is working with an applications program, the operating system is still in RAM, but it is transparent to the user. The user is primarily communicating with the applications program, however the applications program can call on the operating system, if need be, to accomplish functions that are normally the responsibility of the operating system. The user will not be aware of this interaction because the instructions are sent from the applications program, through the CPU, directly to the operating system.

UTILITY PROGRAMS

Some system-oriented tasks may be accomplished with *utility programs*. These programs may be considered part of the operating system, because they perform essential system functions, but they are executed independently. For instance, in most operating systems utility programs are used to format new disks so they will accept data; copy files from one disk to another; list on the screen a directory of disk files; erase or delete unwanted files from the disk; etc.

Some utilities can be executed by a user command but others may be linked directly to the operating system. The beginner is often not aware when a utility is being executed by the operating system, because the operation is transparent to the user. But the user should be aware that *some of the programs on a system disk may be utilities that are essential for the operating system to function properly.* Any one of these utilities may be called up and executed from within the operating system, and once the specific task is completed, the utility returns control to the operating system. This sequence may take place with a single user command and no further user interaction.

There is a variety of programs that can be generally categorized as utilities. The Zenith/Heath version of CP/M includes a utility that enables the user to, in effect, modify or configure the operating system to accept new peripheral devices, such as printers and disks. Another type of utility can examine a disk for files that may have been accidentally deleted from the directory listing, and allow the user to restore them to active service. There are also utilities that display on the screen all of the active disk drives in a system, with their respective capacities, sizes and assigned letters or numbers. The list could go on and on.

Many important utility programs are provided on the disks supplied with the operating system, but it is also possible to acquire additional utility-type software. As the user becomes more familiar with system operation, these accessory utilities can be helpful to "fine tune" the system to the user's specific needs.

HDOS

"HDOS was designed to be maximally flexible, upwardly expandable, and user-modifiable," says its chief designer, Gordon Letwin. "I didn't think I could design a system that would be the ultimate operating system, because I knew that microcomputers were just starting out—it was then 1977—and I felt they shouldn't be restricted in what they could do by the limitations of the software."

At that time, he says, Heath wanted an operating system that would control the disk drive in its yet-to-be-introduced new computer, the H-89. HDOS was developed by Heath with a terminal connected to a DEC (Digital Equipment Corporation) computer. It made use of the usual key strokes and control-sequences, but HDOS provided for additional special-function keys that could be used by an applications program to implement complex operations with a single keystroke. HDOS was one of the first successful disk-operating systems, and it is still extremely popular with many Zenith and Heath customers.

WORKING WITH HDOS

When a computer is first switched on, *the first order of business is almost always the bootstrap operation.* The expression "bootstrap" implies doing something without help—as in "lifting oneself by ones own bootstraps." This term, though somewhat inaccurate, is widely used to describe the process through which a computer loads an operating system into RAM. After the power is on, a disk containing HDOS must be placed in the disk drive, and the user types the letter **B**. The computer responds with **oot** to form the word **Boot**—short for bootstrap—indicating that it is ready to load the operating system. (This form, where the computer completes a word after the user types the first one or two letters is a kind of computer "shorthand" implemented in some software.)

When the user presses the **RETURN** key, the CRT displays the word: **ACTION? (BOOT).** The command inside the parentheses will be executed when the **RETURN** key is pressed again. This is called the "default option"—the command or reponse that will be performed automatically, unless it is overridden by a different user response. When the Boot command is executed, the computer boots (loads) a duplicate of the operating system from the disk to a location in RAM that is set aside specifically to hold the DOS.

During the boot operation the computer announces the following:

```
SYSTEM HAS 48K OF RAM
HDOS Version 2.0 (or other versions)
Issue #50.06.01
Date (DD-MMM-YY)
```

The date must now be typed on the keyboard *in the exact form shown* (e.g., 15-FEB-83). The computer will then store this date for further reference after the user touches the **RETURN** key. Then the CRT will display this:

```
Volume 000, Mounted on SY0
Label: HDOS 2.0 Issue #50.06.01

>
```

Here, the > symbol is a "prompt" to indicate to the user that the operating system is ready to accept the next command. At this point the user can communicate directly with the operating system, and must issue another command to load an applications program or perform any operating-system task. (If desired, there are ways to command the computer to proceed automatically after the boot process and load a specific applications program from the disk. Further information on this technique is available in the manuals that accompany operating system software.)

CAT—THE HDOS FILE CATALOG

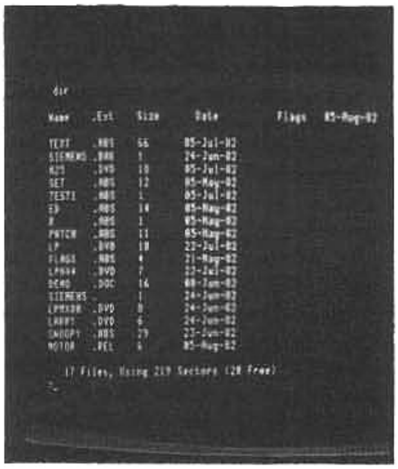
When the > system prompt is displayed, there are several operating system tasks that the user can command. For instance, the user may wish to see a listing (or catalog) of all files on the system disk (the disk currently in the disk drive, from which the operating system was booted).

The user types **CAT** (short for "catalog") followed by a **RETURN**, and the operating system will display a catalog—also called the directory—of disk files on the screen. Systems files (primary HDOS and utility files) are "suppressed" and will not be displayed unless the user types **CAT/S**. When this command is given, *all* the files residing on the disk will be displayed. If there are more than 24 files on the disk, the display will scroll up and some of the listing will disappear at the top of the screen (the screen can normally display only 24 lines of text at one time). If the user wishes to stop the scrolling to read what is currently displayed on the screen, simply hold down the **CTRL** (CONTROL) key and simultaneously press the **S** key. This generates a control-S sequence and stops all terminal activity (and computer activity) until a control-Q sequence is typed.

HDOS also permits the user to see the catalog of files by typing **DIR** (short for "directory") and pressing **RETURN**. **CAT** was the original HDOS directory command, but after CP/M became popular, HDOS was modified to accommodate the **DIR** command—the same one CP/M uses—for the convenience of users who may have both operating systems.

FILE DESIGNATIONS

HDOS file designations have a three-part structure: first, *a prefix* that directs the computer to the specific disk drive containing the floppy disk with the file recorded on it (or to be written on it); second, *a filename*, which may consist of up to eight characters; and third, *an extension*, which may consist of up to three characters to further describe the file. For example, the file designated **SY0:EDIT.ABS** is on the disk in drive SY0:, its name is **EDIT** and **.ABS** is the file extension.



```
dir

Name    .Ext  Size   Date        Flags  05-Aug-82

TEXT    .ABS  56     05-Jul-82
SIEMENS .BAS  1      24-Jun-82
H25     .DVD  18     05-Jul-82
SET     .ABS  12     05-May-82
TEST1   .ABS  1      05-Jul-82
ED      .ABS  14     05-May-82
X       .ABS  1      05-May-82
PATCH   .ABS  11     05-May-82
LP      .DVD  18     22-Jul-82
FLAGS   .ABS  4      21-May-82
LPH44   .DVD  7      22-Jul-82
DEMO    .DOC  16     08-Jun-82
SIEMENS .     1      24-Jun-82
LPMX80  .DVD  8      24-Jun-82
LARRY   .DVD  6      24-Jun-82
SNOOPY  .ABS  29     23-Jun-82
MOTOR   .REL  6      05-Aug-82

  17 Files, Using 219 Sectors (28 Free)
>_
```

The disk is used to *permanently* store files of software and data. When the user is at the operating system level the catalog (CAT) or directory (DIR) command can be used to view a listing of files currently on the disk. Note that the size of the file is displayed in sectors (a segment of disk space), along with a date indicating when the file was last accessed.

Each portion of the file designation is important. The *prefix* addresses a file to a specific disk drive. In a system with more than one 5.25-inch drive, each drive is given a specifier. The primary (boot) drive is called SY0:, and auxiliary drives are numbered in sequence: SY1:, SY2:, SY3:, etc. (SY is short for "system").

If no drive specifier is given, HDOS assumes that drive SY0: is the default option (a file designation without a drive specifier will be presumed to have an SY0: specifier). *In systems with more than one disk drive, any command to manipulate a file must include a drive specifier to indicate where the operating system will find or place the file* (though SY0: is still a default). For example, if EDIT.ABS was on the disk in drive SY2:, the file would have to be designated as **SY2:EDIT.ABS**. And don't forget the colon (:), it must be present to indicate that the prefix is a drive specifier.

The *filename* (e.g., EDIT) can be as short as one character or as long as eight, but the first character must be a letter of the alphabet and not a number; the other seven characters may be either letters or numbers, as desired. The user may select and change filenames as he desires (though some operating system and auto-linked files cannot be changed).

The file *extension* may be one, two or three letters or numbers long, or they can be eliminated entirely, if desired. Extensions are useful to indicate a general category for a file, or they may be used to indicate different versions of data files that may have the same filename.

The extension begins with a dot (.)—usually spoken as "dot" rather than "period" or "point" (thus, "edit dot A-B-S" would be a spoken reference to EDIT.ABS). In this case, the .ABS extension indicates that the file is written in *absolute binary source*—i.e., the program instructions are directly executable by the microprocessor. (Occasionally, programs written in absolute binary source are referred to as *machine-language programs*). Other common extensions are: .BAS for programs written in a version of the high-level language called Basic; .SYS for programs that are part of the HDOS operating system; .TXT for data files that contain text material; and .DOC for text files that explain (or document) programs.

File designations may seem cryptic to the beginner, but they will quickly become familiar. Software designers usually select unique and descriptive names for their programs. However, a utility that allows the user to change any filename and extension is provided in the operating system. It is possible, then, for the user to change filenames and extensions (though it is generally better to stick with some of the accepted conventions).

For example, suppose the user obtains word processing software that is named **WORDPROC.ABS**. If the user finds typing this complete designation tiresome, he can shorten the filename to **WP** or any other unique name that he prefers. Now, when the user sends the command **WP** to the operating system, the word processing program will be loaded into RAM and executed.

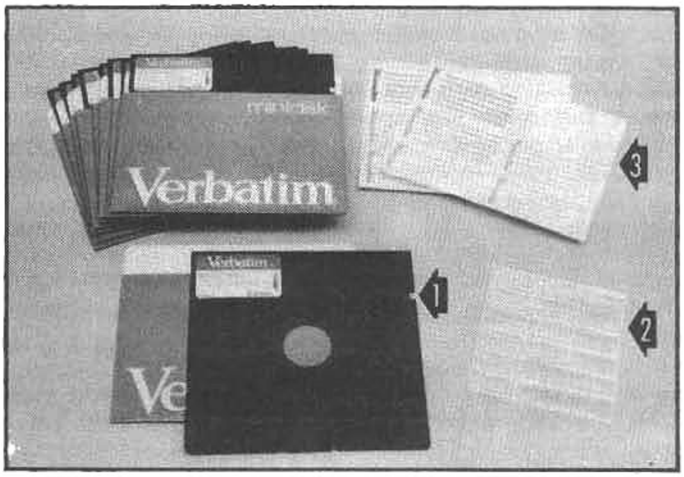
Note that the .ABS extension has been dropped altogether. This is acceptable if the user doesn't need to know or care that this is an absolute binary source file. It must be remembered, however, that the operating system assumes the .ABS extension is a default for directly executable programs, if the user assigns a different extension to such a program, it *must* be included in the execution command.

Choosing meaningful filenames will make computer files easier to remember and locate. The guidelines for naming files (as explained above) must be observed whenever the user wishes to rename a file, but the user is generally free to name files as desired—*except for interlinked operating system files, which must always retain their original filenames* (otherwise the operating system may not function properly).

ABOUT DISKS

So far, we have discussed one general type of disk: a *system disk*. A system disk is a "bootable" disk, meaning that it contains several system files which, together, make up the HDOS operating system. Another general type of disk is the *data disk*. This is a "non-bootable disk." It is physically identical to a system disk but does not contain the operating system files (therefore, it cannot be used to boot the operating system into RAM), but since the system files are not on the disk, there is more free space to store files. A data disk is used primarily in a system with more than one disk drive. In this instance, a system disk must be used in drive SY0: to boot the operating system into RAM and data disks can be used in the auxiliary drives to store programs and data files.

When disks are purchased they are generally "blank" and are unusable. To make a blank disk usable, the user must record data tracks on the disk surface with a system utility program (see below, INIT). After this process is completed, files can be recorded to the disk and retrieved



Before a blank 5.25-inch disk can be used to store programs or data files, it must be initialized (HDOS) or formatted (CP/M). These procedures put information on the disk that is required before the disk drive (and disk-controller) can write data on the disk surface in an orderly manner. All 5.25-inch disks can be protected from accidental data loss by covering the notch (1) with a write-protect tab (2). When the adhesive tab is folded and placed over the notch, data can be read from the disk, but it is impossible to write or erase data on the disk. The tag can be easily removed when the protection is not needed. Adhesive disk labels (3) can be used to record the identifying number, disk name and other useful information on each disk.

by the operating system (see below, PIP). But at this point the disk is only usable as a data disk. To make it into a bootable system disk, a copy of the operating system must be transferred to the disk (see below, SYSGEN).

INIT—DISK INITIALIZATION

Imagine a new suburban housing tract. Before the developer can start building houses, a crew of workers must lay out and name the streets and assign each house a number. In essence, either *formatting* a CP/M disk or *initializing* an HDOS disk is an analogous operation. The computer records on the blank disk information that will be used as a guide by the operating system to place or locate new data or files on that disk.

It is generally necessary to format or initialize a new disk only once, though used disks can be reformatted or reinitialized if the user, for some reason, desires to construct a "new" disk from scratch. It must be remembered, however, that during this process any data or file that was previously recorded on the disk will be destroyed. It is also important to know that a "CP/M formatted" disk cannot normally be used by the computer when it is under the control of the HDOS operating system, and vice versa. Also, the process consumes a few thousand bytes of a disk's potential storage capacity: as much as 10 Kbytes in what would otherwise be a 102-Kbyte disk.

Once a disk is initialized or formatted, it is ready to accept any files that the user may wish to copy onto it. Since it does not contain the operating system files, it is not bootable (though it can be made bootable by coping the necessary system files onto it, see below). However, if the user has a multi-disk system it can now be used as a data disk.

To use the INIT utility to initalize a disk with a Z/H-89 system that has *only one disk drive*, proceed as follows:

1) Boot an HDOS system disk as described above.

2) When the > prompt appears, type **INIT** and touch **RETURN.**

3) The system asks: **PROCEED (YES/NO) ?** where NO is the default option. To proceed, type **YES** and touch **RETURN.**

4) On instructions from the program, remove the system disk from the disk drive and touch **RETURN.**

5) The screen will display: **Device (SY0:)?** This offers the user the default option disk drive (i.e., drive SY0:). Touch **RETURN** to accept the default.

6) On instructions from the computer, insert a blank disk into SY0: and press **RETURN.**

7) For a few moments, the drive will spin the disk. If the disk has been previously initialized, the screen will display the existing label and number. If not, it will display none. Answer the inquiry or press **RETURN**, depending on the computer prompt.

8) The display will ask for a new disk number; it must be between one and 255 (HDOS will not accept any number outside this range). Select a number, using either the keypad or the number keys on the main keyboard, and touch **RETURN.**

9) The display will then ask for a label (a general name) for the disk. For practice, type **SAMPLE**, and then touch **RETURN**. In normal work, the disk may be labeled with a more useful name, such as ACCOUNTS PAYABLE or LETTERS.

10) The disk drive will spin the disk inside and initialize it, a process which takes approximately 30 seconds. HDOS permits the user to make use of partially damaged disks by asking if any sectors on the disk are damaged. (See chapter 8 for details about disk sectors). Assuming, however, that the new disk has no bad sectors (which is almost always the case), when the CRT asks: **Sector?** touch **RETURN.**

11) The drive will spin the disk for a few more seconds and the initialization process will be complete. Remove the initialized disk. The HDOS program will automatically return to the point at which another disk may be loaded into the drive for initialization (step six, above), or a system disk must be inserted in drive SY0:, and the boot process must be completed before the user can go on to a different task.

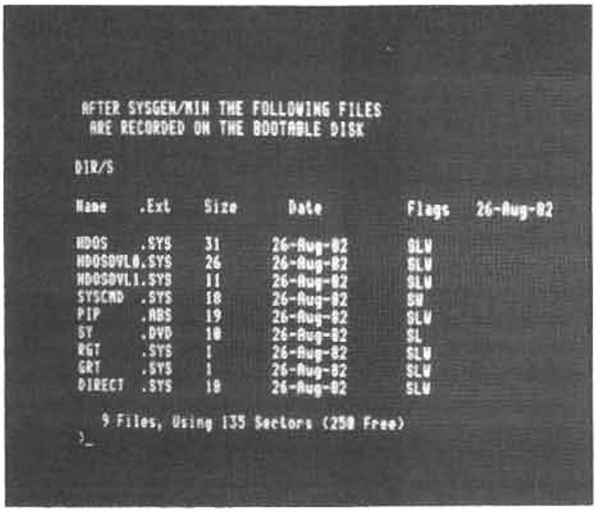
Some users initialize all new disks immediately; that is, they open a box of disks, execute the INIT program, and consecutively initalize all of the disks. The only drawback to this technique is that one must predetermine the number and label of all the disks. However, utilities are available to rename and renumber disks after they have been initialized.

SYSGEN—SYSTEM GENERATION

System generation is a technique used to transfer operating system files from one disk to another. This is accomplished with the utility called SYSGEN. The process will also transfer non-system files, although there are separate utility programs that do this more efficiently.

It is helpful to keep the operating system on extra disks, as a precaution against damage or accidental erasure. It is also convenient to have HDOS utilities such as ONECOPY

SYSGEN is the utility program that is generally used to produce a "bootable" system disk. Normally, SYSGEN will transfer the entire operating system and some important system utility programs to the blank disk. If a user wishes to transfer the minimum bootable system (minus the utility programs) he may use the "/MIN" feature of SYSGEN. This is the directory produced after a minimum system generation. ***All of these files must be present on the disk or the boot operation cannot be successfully completed.***



```
AFTER SYSGEN/MIN THE FOLLOWING FILES
 ARE RECORDED ON THE BOOTABLE DISK

DIR/S

Name    .Ext   Size    Date         Flags  26-Aug-82

HDOS    .SYS   31      26-Aug-82    SLW
HDOSOVL0.SYS   26      26-Aug-82    SLW
HDOSOVL1.SYS   11      26-Aug-82    SLW
SYSCMD  .SYS   18      26-Aug-82    SW
PIP     .ABS   19      26-Aug-82    SLW
SY      .DVD   10      26-Aug-82    SL
RGT     .SYS   1       26-Aug-82    SLW
GRT     .SYS   1       26-Aug-82    SLW
DIRECT  .SYS   18      26-Aug-82    SLW

   9 Files, Using 135 Sectors (250 Free)
>_
```

on the current disk (see below) so that text or data files can be copied without having to replace the disk with a system disk.

In the following SYSGEN example, only one disk drive is needed:

1) After booting an HDOS disk and receiving the > prompt, type **SYSGEN** and touch **RETURN**.

2) The program will ask: **Destination Device (SY0:)?** where the default option is drive SY0:. Touch **RETURN** to select drive SY0:.

3) At the direction of the program, displayed on the screen, remove the HDOS disk from SY0: and touch **RETURN**.

4) Re-insert the HDOS disk and touch **RETURN**. (Steps three and four allow the use of a system disk that is different from the disk used to load the SYSGEN program.)

5) The program will read a portion of the HDOS system disk and tell the user when to remove it, and to substitute a *blank initialized disk*. It is not necessary to touch **RETURN** after inserting the new disk. The program will write the HDOS system files onto the new disk, called the *destination* disk, having read these files from the *source* disk in step four.

6) Step five will repeat itself several times, so it can read all of the various HDOS system files from the source disk. When the SYSGEN program is complete, it will display: **13 Files Copied**.

7) The system can now be booted with the new disk. Ensure it is in drive SY0: and touch **RETURN**.

8) All HDOS disks, when they are *first* SYSGEN'ed, need to record the baud rate at which the terminal has been set (see chapter 8 for details, but disregard it here). No prompt will appear, but after a few seconds the disk will stop spinning. Touch the **SPACE** bar a few times, this records the baud rate on the disk, then the normal boot routine will begin, and when completed, the > prompt will appear.

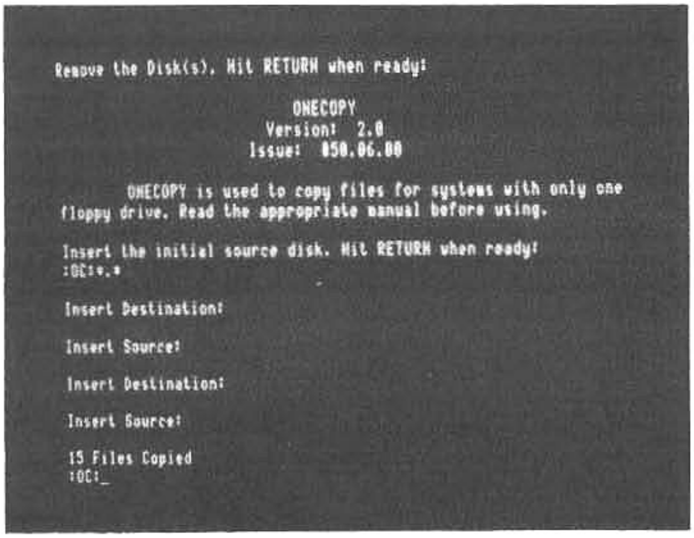
9) To verify that files have been transferred to the new disk, type **CAT/S** and touch **RETURN**; this displays the extended catalog, including system files.

There are two additional ways to use SYSGEN. By following steps one through six, but typing **SYSGEN/MIN** at step one, the computer will transfer only the minimum number of files necessary to make the new disk bootable. By following steps one through six, but typing **SYSGEN *.***, the computer will transfer *all* files from the source disk to the destination disk.

COPYING FILES

For systems with one disk drive, the ONECOPY utility is used to make copies of files onto another disk. The steps resemble the SYSGEN steps in that the *files are read from a source disk, held temporarily in the main memory, and written onto a destination disk when it is inserted in the disk drive*. The source disk need not be a bootable disk, and ONECOPY can not be used to transfer system files.

ONECOPY is an HDOS utility program that will allow users of systems with only a single disk drive to copy one (or more) files from one floppy disk to another. When ONECOPY is called up, it will give some simple instructions and display an "OC:" prompt. At this prompt the user types the name of the file(s) that he wishes to copy. ONECOPY will then load the file(s) from the source disk into computer RAM, prompt the user to install the destination disk, and write the file from RAM to the disk. Several disk swaps may be necessary to complete the transfer of a large file (or several small files).



```
Remove the Disk(s). Hit RETURN when ready!

                    ONECOPY
                  Version: 2.0
                 Issue: 050.06.00

        ONECOPY is used to copy files for systems with only one
floppy drive. Read the appropriate manual before using.

Insert the initial source disk. Hit RETURN when ready!
:OC:*.*

Insert Destination!

Insert Source!

Insert Destination!

Insert Source!

15 Files Copied
:OC:_
```

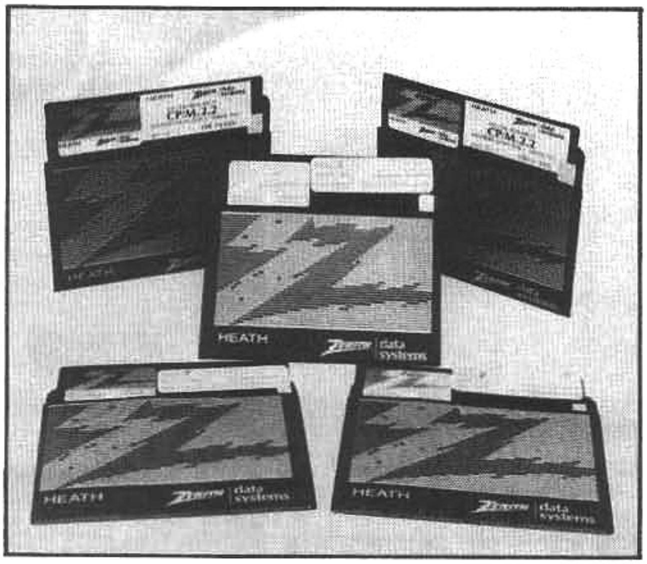
After the command **ONECOPY** is typed and the user follows the instructions on the display, the program prompt **:OC:** appears. At this prompt the user types the name of the file that is to be copied. It is possible to copy all files from the source to the destination by typing *.*— when manipulating multiple HDOS files the * symbol can be used as a "wildcard" designator to save typing effort (details about wildcard file designators are included in the HDOS manual).

In many cases the usable storage space in RAM will be smaller than the amount of file material to be copied from one disk to the other. When this happens the copy transfer will be accomplished in multiple steps. During this process it will be necessary to switch back and forth between the source and the destination disks several times as blocks of copy are transferred to the available RAM space and then onto the destination disk (ONECOPY will prompt when this is to be done). In addition to copying files, ONECOPY will display a catalog of the current disk by typing a **/L** at the **:OC:** prompt. Also, the source disk may be changed by typing **/MOU**—at which point ONECOPY will instruct the user to remove the current source disk and install another.

MULTIPLE-DRIVE HDOS SYSTEMS

Heath, Zenith and several independent manufacturers

CP/M is a very popular microcomputer disk operating system that can support many different kinds of peripheral equipment (disk drives, printers, etc.). Though CP/M is not perfect, it is extremely flexible because is uses a separate BIOS (Basic Input-Output System) module in the operating system. That is, whenever the DOS must communicate with peripheral hardware, it first consults the BIOS to determine the "configuration" of the system (hardware). As the user adds or upgrades peripheral equipment, the BIOS can be altered to reflect changes. The Zenith/Heath version of CP/M has a special BIOS that "fits" CP/M to the Z/H-89, but no other modifications were required to adapt CP/M; as a result, most of the applications programs written for standard CP/M can be readily used with the Zenith/Heath version.



Learning all of the CP/M commands and utilities is not an easy task. However, Digital Research and Zenith/Heath supply a handy reference card to help the novice keep things straight. Since all commands (and many variations of them) must be structured according to specific conventions, the reference card is a handy way to check details that may be confusing.

offer accessory disk drives that are compatible with the Z/H-89, Z/H-90 and Z/H-100 computers. Some of the manufacturers suppysoftware modifications that allow up to four additional drives, in addition to the onboard drive. Instruction for configuring add-on disk drives and other components of a computer system are given in chapter 7. All of the HDOS programs described here will also work on multi-drive systems, if the steps described are modified slightly.

For example, in SYSGEN'ing disks, it is not necessary to remove the source disk from the drive (SY0:) if the destination disk is mounted in a second, accessory drive (SY1:). In this case, at step two, when the screen display asks for the destination drive and offers *(SY0:)* as the default option, type *SY1:* and then touch *RETURN*. This will tell the system to copy the SYSGEN files to the disk in drive SY1:.

If you have a single-drive system, the ONECOPY program is the only way to copy files from one disk to another (hence the name), but if the user has a multi-drive system it is far easier to use the program called PIP to transfer files between disks.

PIP—THE MOST IMPORTANT UTILITY

HDOS would not function properly without the utility called *PIP*—an acronym for Peripheral Interchange Program. *It is the utility HDOS uses to transfer data from one place in the computer to another.*

When a user types **CAT** to look at the directory of a disk, HDOS actually calls upon the PIP program to transfer a copy of the directory from the disk to the terminal. Instead of the normal **CAT** directory command, try typing **PIP TT:=/L** at the system prompt. The effect is the same because the **PIP** portion of the command invokes the PIP transfer utility, the **TT:** portion tells PIP to transfer the data to the terminal, and **/L** is an abbreviated file designation for the catalog (directory) listing.

Note the structure of the command and the use of the (**=**) symbol. In all cases PIP commands must conform to this structure. Specifically, note that the source (usually a filename) listed after the = is transferred to the destination device listed before the = symbol. In effect this structure says "to this destination, send this source."

PIP is used mainly to copy files from one disk drive to another. The filename—with extension—and the drive specifier must always be included. A PIP command is structured in this way: **PIP DESTINATION DRIVE:FILENAME.EXTENSION=SOURCE DRIVE: FILENAME.EXTENSION**. For example, to use PIP to transfer the file EDIT.ABS from drive SY0: to drive SY1:, type: **PIP SY1:EDIT.ABS=SY0:EDIT.ABS** and touch **RETURN**.

If the user wishes to change the name of the file while making the transfer, it is possible to enter a new destination filename. Using the example above, **PIP SY1:NEWEDIT.ABS=SY0:EDIT.ABS** would produce a copy of EDIT.ABS (from SY0:) on SY1:, but it would now be named NEWEDIT.ABS. During the transfer it is often helpful to use the HDOS wildcard designators. Again using the example above, **PIP SY1:*.*=SY0:EDIT.ABS** would produce a copy of EDIT.ABS (from SY0:) on SY1: and would retain the same name because of the use of wildcards in the *destination* filename.

To the beginner this may sound like doubletalk, but with a little practice the methods (and shortcuts) for using PIP quickly become second nature. We present these elementary examples to show the basic operations of PIP, and we recommend that every user study the detailed explanation of PIP in the HDOS manual.

ADDITIONAL HDOS COMMANDS

Besides INIT, SYSGEN, ONECOPY, CAT (DIR) and PIP, there are at least a dozen system commands that HDOS understands. The following is a partial listing; the HDOS manual explains all system commands in detail.

HELP—Prints a list of helpful instructions

BYE—Dismounts all disks for re-boot

DATE—Display or (followed by new date) set date

STATUS—Display the status of files on a disk (default SY0:)

DELETE FILENAME—Delete a file called FILENAME

RENAME NEWNAME=OLDNAME—Rename a file to NEWNAME from OLDNAME.

RUN FILENAME—Run a program called FILENAME

TYPE FILENAME—Type contents of file called FILENAME on the screen

MOUNT SY(X:)—Mount disk on drive "X"

DISMOUNT SY(X:)—Dismount disk on drive "X"

CP/M

While he was working for a computer manufacturer, Gary Kildall developed the CP/M operating system as a way of linking the then-new 8080 microprocessor with the also-new floppy disk. His employer wasn't interested in the idea, so Kildall bought the rights and started his own company with them. Today, Digital Research can claim to have sold more copies of CP/M than any other operating system for 8080/Z80-based computers. CP/M is—de facto—an industry standard.

Like HDOS, CP/M was conceived as a way to link the three important microcomputer elements—the terminal, the CPU and the disk drive—together. By some standards, CP/M is not considered particularly efficient, but by sheer numbers, there seems little doubt that CP/M will remain the dominant operating system for 8-bit microcomputers for many years to come.

WORKING WITH CP/M

CP/M works best with systems that have more than one disk drive. CP/M boots very much the same way HDOS does, except that the user has fewer steps to perform. Simply insert a CP/M disk into a disk drive (most people use the onboard drive in the Z/H-89 or Z/H-90 as the boot drive). When the **H:** prompt appears, type the letter **B** and touch **RETURN**. The system completes the word **Boot** by adding the **oot**, and in a few seconds CP/M is "up and running."

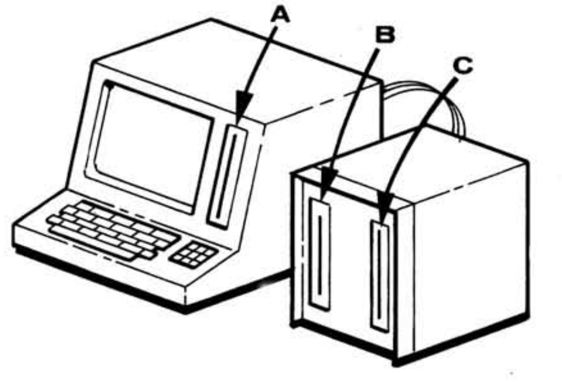
CP/M indicates its readiness to perform by displaying a system prompt that looks like this: **A>**. It means that the booted disk is mounted in drive A: and that control of the computer is centered in the disk mounted in drive A:—in effect, it is the default drive.

If a bootable disk is mounted in another drive, and the user wishes to transfer system control to that disk (e.g., to run one or several programs residing on a disk in drive B: without having to always include the drive specifier), type the letter corresponding to that disk drive, followed by a : and touch **RETURN** (e.g. **B: RETURN**). After a moment, CP/M will respond by issuing a new prompt: **B>**. Drive B: will then be the default drive until the user either designates another default (by typing **A:** or another valid drive specifier) or reboots the computer. The computer will assume all files reside on the default drive unless instructed by the user to look elsewhere (e.g. **C: PROGRAM**).

Disk initialization under CP/M is called formatting, and

PIP—HOW IT WORKS

Copying a file named "Letter" on disk B to disk C



COMPUTER TYPES	USER TYPES	COMMENTS
A> →		CP/M default disk is A
A>	PIP →	User runs PIP to "teach" computer how to copy files
* →		PIP gives Asterisk prompt; means "ready to proceed"
*	C:=B:LETTER →	Copy "letter" from disk B into RAM, when finished, write "letter" from RAM onto disk in drive C (also add "letter" to directory on disk C)
* →		Ready to copy something else
*	Control-C →	Forget PIP and goto default CP/M disk (A), relearn CP/M—load it into RAM
A> →		CP/M is ready to accept next command

The peripheral interchange program (PIP) is an integral utility of CP/M and HDOS that transfers files from one source to another. That is one of the most important functions within the DOS.

At the CP/M system prompt (in this case the A> indicates drive A: is the default) the user types *PIP*. The computer loads the PIP utility from the disk into RAM, and displays an asterisk (*) on the screen to indicate PIP is ready to begin the copy routine.

The user enters the transfer command in the following format: the destination drive, an equals sign, the source drive, and the filename *(e.g., C:=B:LETTER)* and presses *RETURN*. The computer checks the directory on the disk in drive C:, to make certain the destination disk can accept a copy of the file (LETTER), then a check is made of the directory on the disk in drive B: to ensure that the source filename is valid. After these tests, the computer copies the file from the disk in drive B: into RAM; and when completed, it writes the file from RAM onto the disk in drive C:. Finally, the new filename and the size of the file are recorded in the directory on the disk in drive C:.

After the transfer is complete, the screen displays another asterisk to show that PIP is ready for another task. If the user (not wishing to copy any more files) touches RETURN or CTRL-C, the computer will essentially "forget" the PIP routine, and go to the CP/M system disk (in drive A:) and "relearn" CP/M by reloading it into RAM. When the A> system prompt is displayed, CP/M is again in control and ready to accept another command.

If the system has only a single disk drive, PIP will work (much the same way as the HDOS ONECOPY program does). All of the above steps are followed, except that drive B: and C: should be thought of as *disks B: and C:*. PIP will prompt the user to *PLACE DISK B IN DRIVE A, AND PRESS RETURN.* When the source disk is again needed, PIP will request *PLACE DISK C IN DRIVE A, AND PRESS RETURN.*

is accomplished with a utility program called FORMAT. It is necessary to format any 5.25-disk before it can be used with CP/M. To format a disk, follow these steps:

1) Boot a CP/M disk (which includes the FORMAT program). When the **A>** prompt appears, type **FORMAT** and touch **RETURN**.

2) The program will ask the user whether or not to proceed with the FORMAT program and ask for a (Y/N) response. To proceed, type **Y** (for yes) and touch **RETURN**.

3) The program will ask the user to identify the destination disk. If the disk you wish to format is in drive B:, type **B** and touch **RETURN**.

4) Make sure the blank disk is in drive B: and touch **RETURN**, as prompted by the screen display.

5) Formatting takes approximately 60 seconds; when one disk is done, the program will display: **INSERT A NEW DISK AND PRESS RETURN**.

6) Since it is easy to format a whole box of new disks at once, remove the formatted disk and replace it with a fresh disk. Repeat steps four, five and six until all disks have been formatted.

7) Some versions of CP/M (particularly the Zenith/Heath version) ask the user: **Do you have more disks to format? (Y/N)**. If that prompt appears, you may end the FORMAT program by typing **N**. In other versions FORMAT can be stopped by just pressing **RETURN**. At this point, CP/M will return with the **A>** system prompt.

CP/M comes with an extensive documentation manual that is widely regarded as being inadequately organized for novice users. There are several excellent introductory texts which do a better job of explaining CP/M than the system manual does, and any new user is advised to read at least one of them. The information presented here is only a brief overview of a few major CP/M system operations.

COPYING CP/M AND COPYING FILES

At all times, the operating system must know what computer hardware it is expected to control. Under CP/M, there is a "system image" that the operating system keeps "in mind" while it is working. The system image includes the capacity of the computer memory (RAM) and the size and configuration of the disk drives (whether 5.25-inch or 8-inch, etc.).

One of the features that makes CP/M so flexible (usable on many different computers) is the ease with which the system image can be changed. The system image is created through a process known as "configuring CP/M." This is accomplished with a utility program called CONFIGUR. After the program is called, the user simply answers questions and fills in blanks as the program displays various configuration options on the screen. At the end of the process, the system image is formed and stored in the operating system. (Detailed directions for the use of CONFIGUR are outlined in the Zenith/Heath addendum to the CP/M manual).

Once CP/M is set to your system configuration, *it is wise to make backup copies for daily use.* Keep the original in a safe place and use it only as a source for copies; use the copies for day-to-day work with the computer. This is very important, since disks can be damaged or lost.

Creating a bootable system disk is similar to the process discussed in the HDOS section.

To copy a CP/M disk (i.e., to make a bootable copy), follow these steps:

1) Boot a CP/M disk (which contains a copy of SYSGEN) in drive A:, and place a blank, formatted disk in drive B:. At the **A>** prompt, type **SYSGEN**.

2) The program will issue a sign-on message, and then request the source drive. At this request respond **A**.

3) After a few seconds, the program will request the destination drive. Press **B**. This writes the boot information to the boot tracks on drive B.

4) When this operation is complete, SYSGEN will ask: **COPY BIOS.SYS (Y/N)?** At this prompt, reply **Y** to write the CP/M system image to the disk in drive B:.

5) This completes the operation. Press **SHIFT-RESET** to re-boot.

The PIP utility under CP/M operates in virtually the same manner as that for HDOS and may be used to transfer files selectively from one disk to another. The syntax for CP/M PIP commands is: **PIP DESTINATION DRIVE:=SOURCE DRIVE:FILENAME**. For example, to transfer a file called EDIT.COM to drive B: from A:, type: **PIP B:=A:EDIT.COM** and touch **RETURN**. Note, however, one significant difference between the HDOS and CP/M copy commands: in the CP/M command, *the filename does not have to be included in the destination address if the user wishes the destination file to retain the same name as the source.*

CP/M FILE DESIGNATIONS

The CP/M manual and ancillary references explain file designations, filenames, drive specifiers and extensions in great detail, though the designation structure is somewhat similar to that of HDOS. When copying, it is possible to *simultaneously transfer several files which have similar names or extensions.* The * is a wildcard designator for filenames or extensions. Thus, ***.COM** will designate all files with a **.COM** extension (all CP/M utility and application files have the .COM extension, the equivalent of .ABS in HDOS). In a similar fashion, **REPORT.*** will designate every file named **REPORT**, regardless of its extension. A combination is also possible. The ***.*** (star-dot-star) is a wildcard designator for any file with any extension (in effect, *all* of the addressable files on the disk).

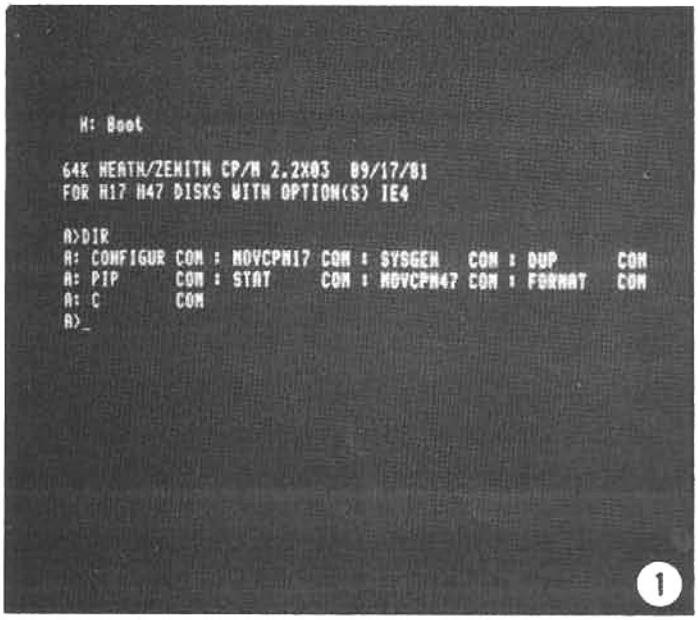
Another CP/M wildcard file designator is **?**. It is used when several files *share a few common letters in their names.* For example, if the user has written nine letters, each filed consecutively (e.g. LETTER1, LETTER2 . . . LETTER9) and wishes to copy all of them, it is possible to type:

PIP B:=A:LETTER?

and every file that fits the filename (with wildcard) will be copied. Obviously, then, CP/M does not allow an "?" or "*" as part of the ordinary filename.

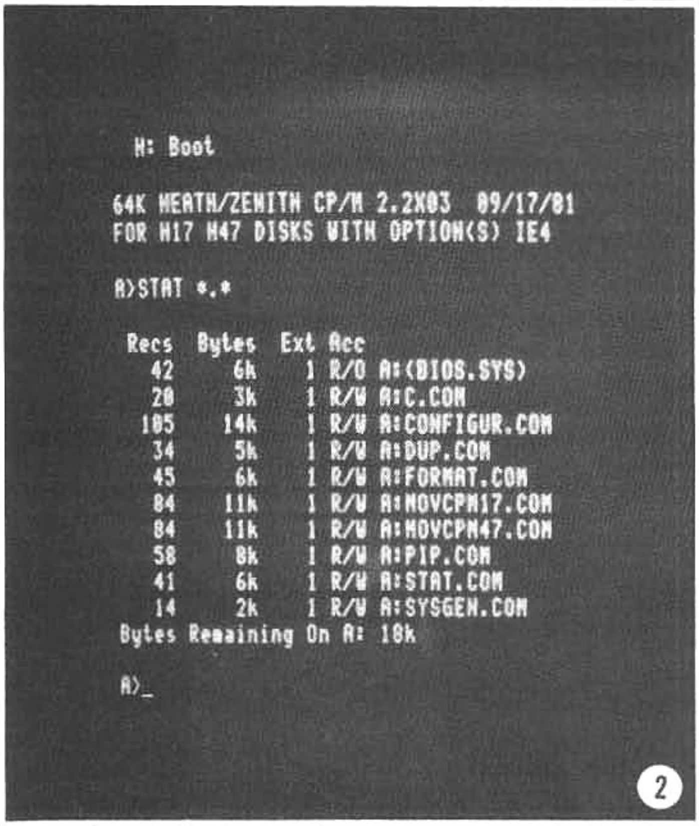
The DIR command is used to examine the directory of a disk. Typing **DIR** and touching **RETURN** will produce the directory of the currently-used drive: i.e., if the **A>** prompt appears, drive A: will be accessed. To examine the directory of another drive, type **DIR (drive):** (e.g. **DIR B:** will display the directory from drive B:).

The STAT command is used instead of DIR when more information is required; **DIR** merely lists the files. To list all disk files with STAT, type: **STAT*.***. STAT places them in alphabetical order (by the first letter or letters of each file's name), gives the file size in terms of the number of *records* (a disk-management reference that is not of great value to the average user) and also in terms of the number of kilobytes (Kbytes or groups of 1024 bytes) each file occupies. This indication of size is extremely valuable. The user can examine the size listing to see if a file may be too large to fit on a disk with limited free space.



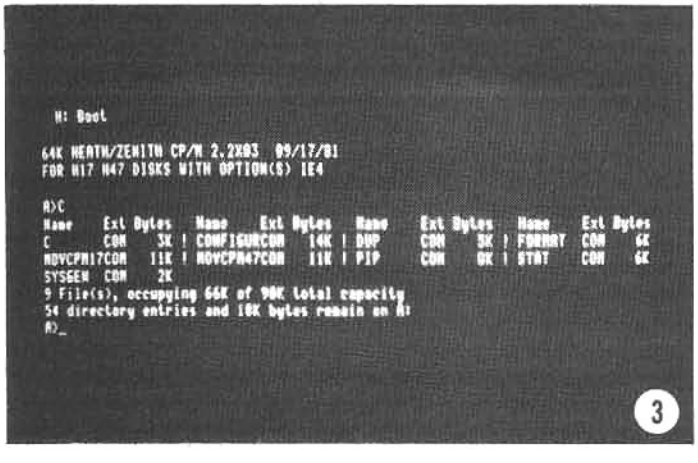
```
  H: Boot

64K HEATH/ZENITH CP/M 2.2X03  09/17/81
FOR H17 H47 DISKS WITH OPTION(S) IE4

A>DIR
A: CONFIGUR COM : MOVCPM17 COM : SYSGEN   COM : DUP      COM
A: PIP      COM : STAT     COM : MOVCPM47 COM : FORMAT   COM
A: C        COM
A>_
```

1



```
  H: Boot

64K HEATH/ZENITH CP/M 2.2X03  09/17/81
FOR H17 H47 DISKS WITH OPTION(S) IE4

A>STAT *.*

 Recs  Bytes  Ext Acc
   42     6k    1 R/O A:(BIOS.SYS)
   20     3k    1 R/W A:C.COM
  105    14k    1 R/W A:CONFIGUR.COM
   34     5k    1 R/W A:DUP.COM
   45     6k    1 R/W A:FORMAT.COM
   84    11k    1 R/W A:MOVCPM17.COM
   84    11k    1 R/W A:MOVCPM47.COM
   58     8k    1 R/W A:PIP.COM
   41     6k    1 R/W A:STAT.COM
   14     2k    1 R/W A:SYSGEN.COM
Bytes Remaining On A: 10k

A>_
```

2



```
 H: Boot

64K HEATH/ZENITH CP/M 2.2X03  09/17/81
FOR H17 H47 DISKS WITH OPTION(S) IE4

A>C
Name    Ext Bytes   Name    Ext Bytes   Name    Ext Bytes   Name    Ext Bytes
C       COM    3K ! CONFIGURCOM   14K ! DUP     COM    5K ! FORMAT  COM    6K
MOVCPM17COM   11K ! MOVCPM47COM   11K ! PIP     COM    8K ! STAT    COM    6K
SYSGEN  COM    2K
9 File(s), occupying 66K of 90K total capacity
54 directory entries and 10K bytes remain on A:
A>_
```

3

CP/M has a built-in utility that displays the files recorded on a disk. Typing *DIR* will generate an abbreviated directory listing (photo 1). No information on the size of each file or the remaining space on the disk is displayed. The "status" of those files can be obtained by using the utility called STAT. Typing *STAT* *.* will produce an alphabetized listing of all files on the disk, along with their individual file sizes (photo 2). The remaining free space on the disk is also given. Many independent programmers have also developed "custom" directory listing programs. A disk directory listing produced by one of these programs is shown in photo 3.

For example, some word processing programs routinely maintain backup copies of every file to prevent accidental data loss. The backup files are listed in the directory with the extension .BAK, and can fill up a disk very quickly. For instance, let's assume the user has a 12-Kbyte document called REPORT already existent on the disk, but there is less than 12 Kbytes of free space left on the disk. A *copy* of REPORT can be read from the disk and loaded into RAM by the word processor for editing, but the *original* copy will be left on the disk. (This is a precaution against data loss: if there is an accidental destruction of the RAM files—perhaps due to a power failure—the user still has a version of the file resident on the disk, even if it isn't the very latest.) When the user wishes to write the edited version of REPORT back to the disk, there will not be enough room for it along with the original version, which has remained on the disk with a .BAK extension.

In this instance, the user would be in a jam. He would have to lose the updated RAM-file copy of REPORT and start from scratch with a disk that has more available free space. However, there is one exception to this. If the word processor is designed to link with the CP/M ERASE command, it would be possible to erase the backup file from the disk, freeing up enough space to write the RAM file to the disk. This is not possible with all word processors, therefore, before editing (loading) a file, the wise user should check STAT to ensure that ample free space exists (on a crowded disk) to accept the updated version when it must be written back to the disk.

Besides ERA, other CP/M commands include REN—to rename a file; TYPE—to type the contents of a file onto a screen or a printer (depending on previously-set conditions); and SAVE—to store a block of data, text, or a program from RAM as a disk file.

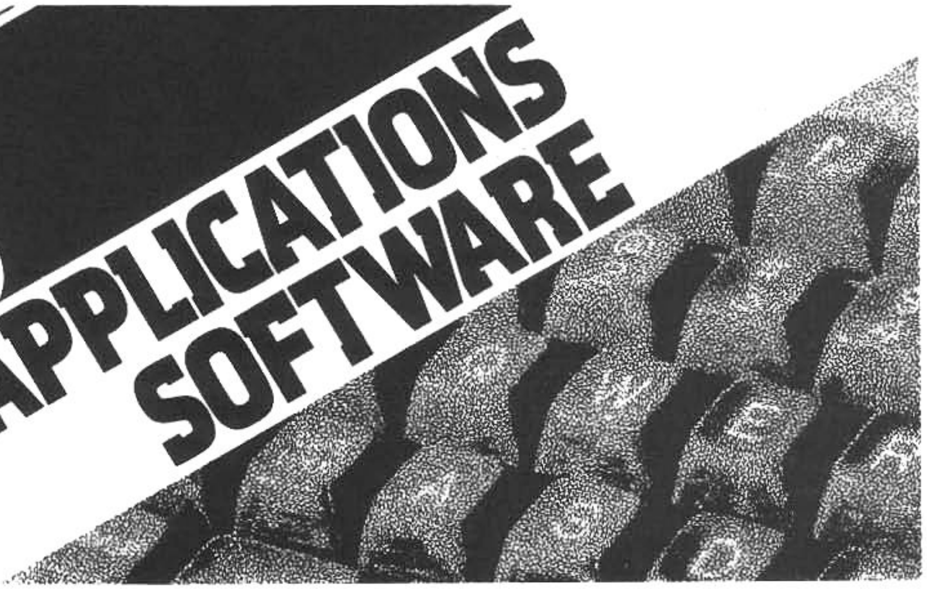
HDOS AND CP/M COMPARED

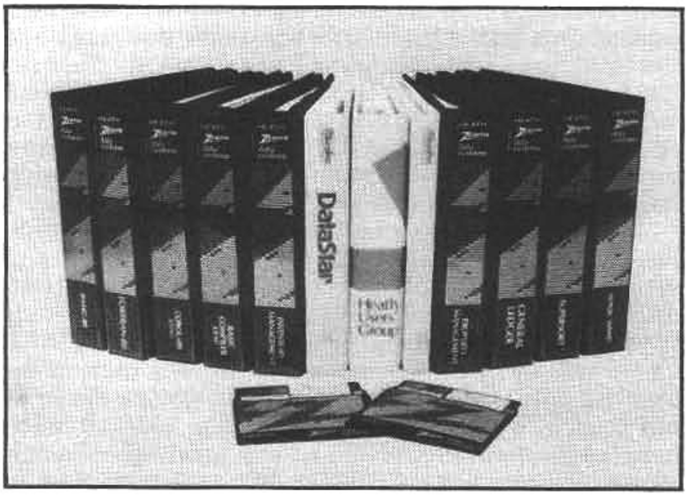
It is not fair to compare these two operating systems feature by feature. Although HDOS and CP/M are approximately the same "age," having been created only a few years apart, they are not comparable by criteria of value to an average user. More applications programs run under CP/M than under HDOS, though people who use them both have said that HDOS is more "user-friendly" and forgiving of mistakes after certain kinds of errors are made.

HDOS, however, will only run on Zenith/Heath computers, while CP/M has been modified to run on virtually *every* microcomputer in some form or other. This means that applications programs written to run under CP/M outsell those for HDOS by an enormous margin. For many users, the variety of available applications software is a strong reason for using CP/M.

In the next chapter, we will look at some of the most popular HDOS and CP/M applications programs.

CHAPTER 6 APPLICATIONS SOFTWARE





The Zenith/Heath user can choose from a wide array of operating systems, computer languages, applications software, communications packages, countless system utilities, and more.

Local computer clubs and users' groups can provide valuable help for enthusiasts. At club meetings members discuss mutual problems, trade software, and in general have a good time sharing a common interest. The national Heath Users' Group (HUG) sponsors yearly meetings that draw enthusiasts from all over the U.S.

SOFTWARE FOR MANY APPLICATIONS

Once an operating system has been booted into system RAM, a Zenith/Heath computer is ready to run (execute) an applications program: the software that actually does something "useful" for the user. This applications program may be one of the many pre-written "packaged" programs (available from a variety of sources) or a special program that the advanced user has developed to answer a particular need.

The information in this chapter provides a general overview of software for Zenith/Heath Z/H-89 and Z/H-90 computers. Although some of the specific programs discussed will *not* operate on the Z/H-100 series, they are representative of the packages that will soon be released (many with enhanced features) for the Z/H-100 computers.

It is impossible to include an evaluation of every available program, but we will explore major *types* of applications software, and we will review some of the popular programs in each category. However, programs vary greatly in quality and cost, and we caution all readers to *carefully examine sales literature and all available documentation before buying any software.*

Some widely advertised programs are not compatible with Zenith/Heath computers (VisiCalc is a good example) because they are not designed to work with the microprocessor used in Zenith/Heath systems. But, similar programs (in this case, SuperCalc or Zencalc) are designed for Z/H systems, and they will perform at least as well, if not better in some ways, than the better-known software.

Much of the software discussed here is designed for small business applications. A microcomputer can bring powerful capabilities to any business, and Zenith/Heath computers can be readily employed to satisfy the requirements of nearly any small business. Zenith is obviously responding to these needs by including two programs discussed here—Basic-80 and SuperCalc (both for CP/M)—with each new Zenith computer. This provides the user with a package of hardware and software suitable for spreadsheet projections (SuperCalc) and the support of accounting programs or user-oriented special-applications programs (Basic-80).

On the other hand, many computer users, and especially hobbyists, find little appeal in business software, and prefer to explore the intricacies of programming languages or recreational computer games. There is wide variety of programs—available for both CP/M and HDOS operating systems—to satisfy these interests.

The Heath Users' Group (see appendix), The CP/M Users' Group (see appendix), local CP/M or Heath computer clubs and other local computer clubs are excellent sources for new software—some sources may even supply software free of charge or make programs available for just the cost of a blank distribution disk. These programs may be useful, but not all of them are recognized by Zenith/Heath as being fully-tested and free of bugs. If such software does not run correctly, the user cannot assume that Zenith/Heath will fix the problem. There are, however, many experienced users and skilled independent programmers who monitor the HUG-sponsored bulletin board on the CompuServe telephone network. These users are usually willing to give advice for debugging minor program problems.

The professionally-written software discussed here are examples of some of the best programs available for Zenith/Heath computers. A novice user can be assured of their quality, and though no software is guaranteed completely free of quirks, these programs are at least *nearly* error-free. Most of them are available from authorized Zenith or Heath dealers. However, Heath dealers, the Heath mail-order catalog, and the Heath User's Group are the primary sources for HDOS programs (programs written for HDOS are generally more popular with Heath computer users). Some of the programs are available only from independent suppliers and, whenever possible, we have listed the source so readers can obtain additional information.

SPREADSHEET PROGRAMS

The *spreadsheet* program is directed at the very heart of business management. For example, by placing all of a company's monthly sales records in a row, above a columnar listing of all sales personnel and their individual contributions to the total, a sales manager can see how productive each salesperson was during the year, and what percent of the company's business was directly attributable to each individual.

The problem with doing this type of analysis by hand is that, even with the help of a calculator, each computation requires many minutes of record analysis; in a large company, the manager will spend hours at the task. Also, if any additions or changes are made, they affect many of the other figures. Even with a soft pencil and a good eraser, the spreadsheet quickly becomes a mess.

In the early 1980s, a computer program converted the paper spreadsheet into a screen display, and performed the calculations at each *cell* (where a row and column meet). This time-saver, called Visicalc, became a best-seller, but the original was not written for CP/M or HDOS and could not be used with Zenith/Heath computers. Independent programmers have now developed spreadsheet simulators that are compatible, and a leading example is SuperCalc.

SUPERCALC

To run SuperCalc in a single-drive system, boot a CP/M disk, then remove it and insert the disk on which SuperCalc resides. With 8-inch or double-density 5.25-inch drives, it is possible to include SuperCalc on the same disk as CP/M and switching disks is not necessary. After CP/M is loaded, SuperCalc is invoked by typing **SC** and touching **RETURN**.

At any time the user can get help—by touching the **?** key—and a list of possible commands appropriate to the user's current place in the program will appear. Initially, when no data has been entered, the help menu will give instructions for creating a spreadsheet. Pressing any key takes the user out of the help menu and back to the "spreadsheet" screen.

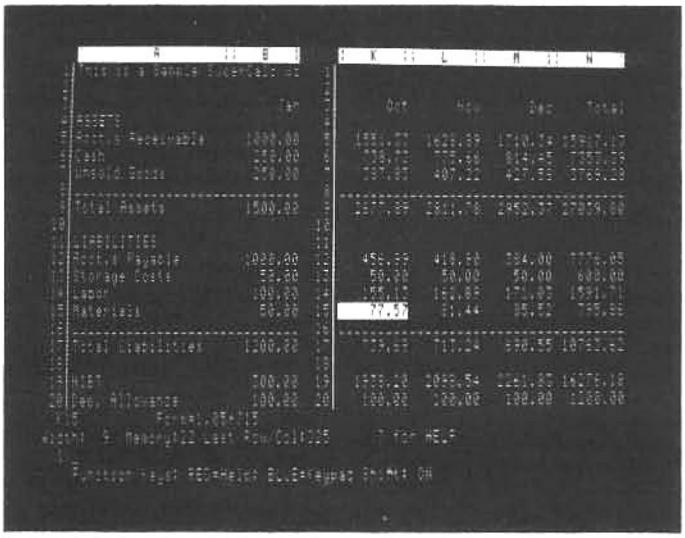
BRKEVN—A SUPERCALC SAMPLE PROGRAM

Breakeven analysis is a sample program that comes with SuperCalc to demonstrate the capabilities of a spreadsheet analysis. The cost of producing an item can be broken down by category and compared against demand, price fluctuations, quantity discounts, returns, and so on. These comparisons will help predict how many items must be sold at a given price or cost before the company breaks even on the products sold. To run the demonstration the user types **BRKEVN** and SuperCalc will display a matrix comparing production costs (down the left column) with factors affecting sales (along the top row).

One of the most popular uses for the small computer is the analysis and projection of business or personal expenses, profits, and growth. An "electronic spreadsheet" puts sophisticated and powerful monetary planning tools in the hands of every microcomputer use.

Each "cell" is identified by the intersection of a column (letters A thru Z, AA thru ZZ, and BA thru BK, for a total of 63) and row (to a maximum of 254). The cell in the upper left is A1, while the extreme bottom right would be BK254. In practice, it is rarely necessary to use so many cells, and computer memory limitations generally prevent the use of all cells in a single spreadsheet. The width of the columns can be changed by the user, so long titles or numbers with many decimal places can be accommodated. The cursor is a reverse-video box that fills the cell in which it is located; its size will vary according to the width of each column.

At each cell in the BRKEVN demonstration, a calculation has been made according to a pre-defined formula, but these are easily modified by the user. The "factor" for each calculation can be a constant (such as "10 per cent") or a formula (for example, "multiply the number found in cell A6 by the number found in cell T47 and display the product here"). Virtually anything that can be run on a programmable calculator can be emulated in SuperCalc.



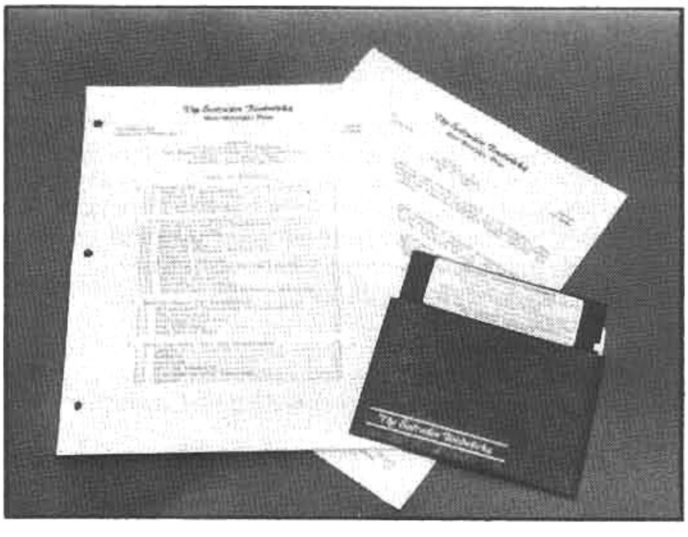
```
         A         ||   B   |   |  K  ||  L  ||  M  ||  N  |
 1|This is a sample SuperCalc ws   1|
 3|                      Jan      3|    Oct      Nov      Dec    Total
 4|ASSETS                          4|
 5|Acct.s Receivable  1000.00      5| [illegible] 1628.39 [illegible] [illegible]
 6|Cash                250.00      6| [illegible]  775.[illegible]  814.45 7357.[illegible]
 7|Unsold Goods        250.00      7|  387.83   407.22   427.58  3769.28
 8|--------------------------      8|------------------------------------
 9|Total Assets       1500.00      9| [illegible] [illegible] 2952.37 27859.[illegible]
10|                               10|
11|LIABILITIES                    11|
12|Acct.s Payable     1000.00     12|  456.[illegible]  418.[illegible]  384.00  7776.05
13|Storage Costs        50.00     13|   50.00    50.00    50.00   600.00
14|Labor              100.00     14|  155.[illegible]  162.8[illegible]  171.03  1591.71
15|Materials           50.00     15|   77.57    81.44    85.52   795.[illegible]
16|--------------------------     16|------------------------------------
17|Total Liabilities  1200.00     17| [illegible]  717.24   690.55 10763.[illegible]
18|                               18|
19|NIBT               300.00      19| [illegible] 2093.54 [illegible] 16276.18
20|Dep. Allowance     100.00      20|  100.00   100.00   100.00  1200.00
 K15        Format:$8*15
Width: 9 Memory:22 Last Row/Col:N25      ? for HELP
 1>
   Function Keys: RED=Help BLUE=Keypad Shift: ON
```

The SuperCalc spreadsheet can be used to analyze many types of data. In this example the user has created a business ledger with assets and liabilities for a 12-month period. The data for February through September have been moved off the screen, so the user can view the cost of materials for the month of October.

THE SUPERCALC WORKSHEET

The SuperCalc worksheet occupies 20 of the 24 lines displayed on the screen. Line 21 tells the user which cell the cursor is currently occupying, what's in that cell, and whether the information there is a formula, text, or data. Line 22 describes the width of the cell, how much memory (in Kbytes) the program is currently using, and the current extent of the matrix—the column and row of the bottom-right cell. Line 23 displays user entered commands and/or data. Line 24 provides labels for the red (HELP) and blue (KEYPAD SHIFT) special-function keys.

The cursor (current cell) is moved with the keypad arrow keys, one row or column at a time for each press of the key. For long-distance moves, however, it is possible to key in the column and row of a desired cell and the cursor will jump there directly.



ZenCalc, from the Software Toolworks, is nearly as powerful as SuperCalc, and sells for under $100. This inexpensive program uses the special-function and arrow keys to speed data input. It is supplied with a detailed manual and sample spreadsheets.

Completed spreadsheet displays may be saved on disk for future reference or for additional evaluation. To save a display, the user types **SAVE** and enters the desired filename with (optionally) the extension **.CAL** to show that it is a SuperCalc file. Thus, to save a sales chart called "SALES" on disk drive B:, the user would type **SAVE B:SALES.CAL** To retrieve it, type **LOAD** followed by the disk and filename: **LOAD B:SALES.CAL**

Some advanced features of SuperCalc are the ability to "lock" titles in place while the columns are scrolled; to create display "windows" (for example, to keep totals or sub-totals in view), and to insert new or "blank" columns anywhere in the worksheet. The user can also ask SuperCalc to display the formulas defining each cell, instead of the data contained in them. The order in which the formulas are applied can also be modified (e.g., to have calculations in the rows computed first, instead of those in the columns).

REQUIREMENTS: SuperCalc requires at least 48K of RAM to operate properly. It is distributed on 5.25-inch disks; both hard- and soft-sectored (see chapter 8), and on 8-inch disks. It is available from Zenith and Heath.

ZENCALC

This spreadsheet package supports up to 255 lines and 52 columns, and contains most of the features of SuperCalc. ZenCalc is also designed to take full advantage of all the cursor arrow and special-function keys of the Z/H-89. Adding or deleting rows or columns is performed much like the popular word processor PIE (discussed in the next section); making ZenCalc a powerful "math-processing" program.

The latest revision of ZenCalc—2.0—includes a number of enhancements over earlier versions; more functions are permitted and additional format options for displaying numbers and column (or row) labels are included. A wide variety of "pre-programmed" math functions are available, and all maintain 13 decimal digits of precision. In addition to the usual "add, subtract, multiply, divide" are:

exp(x)—Compute e (2.718281828) to the x
ln(x)—Compute natural log of x
log(x)—Compute log to base 10 of x
sin(x)—Compute sine of x
cos(x)—Compute cosine of x
atan(x)—Compute arctangent of x
abs(x)—Compute absolute value of x
int(x)—Compute integer of x
sqrt(x)—Compute square root of x
sum(range)—Compute sum of rows/columns over a range
avg(range)—Compute average over a range
lsr(x,y)—Compute lesser of x and y
gtr(x,y)—Compute greater of x and y
min(range)—Compute the minimum of rows/columns
max(range)—Compute the maximum of rows/columns

Each of these functions can be performed on a fixed number or on the value of another cell. This is not a complete list of the functions that ZenCalc provides, but it is clear that there are few computational constraints.

Zencalc is accompanied by a 56-page instruction manual that includes a glossary, a command summary, and a section titled: "Doing your income taxes with ZenCalc." In addition, seven tutorial worksheets are provided on disk, including a personal monthly budget with cash-flow analysis and a real-estate investment spreadsheet.

Plus, five sample data files are included that act as templates for the IRS forms 1040, schedule A, and the tax computation schedules.

Electronic spreadsheets are perhaps the most versatile business programs available for microcomputers, and the value of a spreadsheet is only limited by the needs and imagination of the user. ZenCalc is an excellent spreadsheet calculator and should be considered by anyone in need of business or scientific number-crunching capability.

REQUIREMENTS: ZenCalc requires CP/M or HDOS and at least 48K of RAM, though 64K is recommended to take full advantage of this program. A printer is required if you wish to generate a hard-copy record of the spreadsheet. ZenCalc is distributed on 5.25-inch hard-sector disks for both HDOS and CP/M, and on 8-inch disks for CP/M. It is available from the Software Toolworks (see appendix).

WORD PROCESSING PROGRAMS

Word processing is often cited by users as the primary reason for purchasing a microcomputer. In fact, word processing has, perhaps, the best chance of finding at least some application in every computer installation. Writing letters, contracts, invoices, reports and resumes is considerably easier using a word processor than a conventional typewriter because corrections and changes are very easy to make. Some of the features that word processors provide are:

1) Text is displayed on the computer screen *before* it is printed, so the user can modify, rewrite, insert, delete, and rearrange any part of the material before committing it to paper.

2) Return addresses, "boilerplate" sections of contracts, standardized salutations and closings, and other repetitive material need be written only once, stored on disk, and then called up to drop into new text, without retyping.

3) The word processor can "reformat" material (e.g., make a wide paragraph narrow, etc.) without retyping anything.

4) Incorrect material can be changed automatically—at every occurrence of the error, if required—at the touch of a key, and the tedious tasks of indexing, page numbering and footnoting can all be performed by the computer.

Not all word-oriented programs are true word processors: some word programs are categorized as *text editors* or *line editors*. The differences between these types of word programs are not distinct, and the terms can cause considerable confusion for the novice, but for our purposes we make the following general definitions:

A *word processor* is a program designed primarily for *the creation of ordinary text documents* (e.g., memos, letters and manuscripts). This type of program offers a full range of text-handling features, including text formatting (see below). A *text editor* usually (but not always) offers fewer advanced features, and generally does not include a full formatting capability. Text editors are often designed to *write and edit computer programs*. Advanced text editors (like PIE, see below) can approach the power of word processors, but most of them are not well suited for writing or editing complex text.

Some advanced text editors and virtually all word processors are *cursor-oriented*. The screen cursor can be positioned anywhere within the text, and words can be dynamically inserted, deleted or moved at will, using the cursor as a "pointer" to indicate where the words will be inserted, deleted, etc. Editors with this capability are sometimes refered to as "screen" editors. Some editors, however, are not cursor-oriented. Instead, they are line-oriented and are generally called *line editors*. In this type of editor each line of text is assigned a sequential number (as is common in computer programs), and the user can only rewrite one line (or a group of lines) at a time by issuing structured commands to the editor. This is a cumbersome and much slower method of editing.

Versatile software can turn a Zenith/Heath microcomputer into a powerful word processor. Such a system is often cheaper than a dedicated "electronic typewriter," yet it can do much more.

FORMATTING

Formatting commands are used to rearrange text so that it may be printed in an orderly manner—within preset margins, with page numbers, with page-header titles, etc. For instance, as a result of extensive copy editing, a piece of text may be dissected into many short and long lines. If this text is then sent to a printer in literal form, the printed copy will also have an uneven appearance, with a very jagged right margin (in a literal format the computer will consider every blank space on the screen as a valid character and send corresponding blank spaces to the printer). To give this text a right margin that appears smooth and even, the words must be rearranged from line to line to fill the blank spaces. If the software has a formatting capability, this can be done swiftly with a few commands.

At other times, the user may want to move the entire block of text to the right or left on the page (shifting both margins an equal amount) or the user may want to make the block of text wider or narrower. All of this, and much more, can be accomplished with formatting software, and *flexible formatting is generally considered mandatory in advanced word processing applications.*

Formatting can be accomplished in one of two ways. Some word processing software will allow the user to *dynamically format* the text—the words will be rearranged to fill the empty line space and immediately displayed on the screen in reformatted form. Another type of software, a *text formatter*, "processes" the source text from a disk file and creates a new, reformatted text file, which is then transmitted to the disk or to a printer. In this case, the user does not see the formatting take place on the screen because the program does the formatting in RAM without presenting a direct video display of the activity. The user can only review the results by calling up the new file with the editing software to see what it looks like in reformatted form or, if the formatted file was transmitted directly to a printer, he can review the printout.

Either type of formatting software can be used to produce professional-looking text printouts (and some software will do both types of formatting), but dynamic formatting is preferred by many writers because the results can be reviewed immediately. It should be noted, however, that word processing software which incorporates dynamic formatting is generally more expensive.

Text editors that do not have built-in formatting features are usually smaller (they take up less disk and RAM space), and they are less complicated than full-feature word processors (because the user has fewer commands to learn). These editors are primarily intended for writing program listings and complex formatting is generally not required for this type of work. But the value of some text editors can be extended by combining them with stand-alone text formatters (see below, PIE and TEXT). In this way a text editor can also be used to produce letters, memos or simple text material (though it still won't be as powerful as a full-featured word processor).

ELECTRONIC TYPING

Zenith's word processing program, called Electronic Typing, runs under HDOS. It is considered easy to learn and use, and comes with a self-teaching tutorial. Electronic Typing is a simple word processing program that takes the drudgery out of typing and introduces many text-handling features to the office. These include: automatic carriage return ("wrapping" text to the next line without hitting the return key); inserting and deleting words, sentences or paragraphs; automatic reformatting when the page length or margin settings are changed; movement of blocks of text anywhere in a document; and the ability to print documents on a letter-quality printer.

Electronic Typing stores text on disk in a manner that is not readable by most other word processing or formatting programs. So, if the text must be compatible with other equipment or software (e.g., if it will be typeset from the original disk file, or transmitted by telephone to other systems, etc.), one of the "literal-file" word processing packages—discussed later in this chapter—will be a better choice.

REQUIREMENTS: Electronic Typing runs only under HDOS and requires 48K RAM; it will run with one, two, or three disk drives, and comes on a 5.25-inch hard-sector disk. The price is $195.00. A letter-quality printer is recommended but not necessary. Available from Zenith or Heath.

MAGIC WAND

Magic Wand is a command-driven word processing program: a powerful method of text manipulation. Magic Wand, as supplied for the Z/H-89 and -90, makes use of the special-function keys for forward and backward text scrolling, and shifts the numeric keypad to allow cursor positioning—left, right, up, and down—and inserting or deleting text. In this respect, Magic Wand is somewhat easier to learn and use than word processors that use multiple-key control-sequences, because many Magic Wand functions can be invoked with a single keystroke.

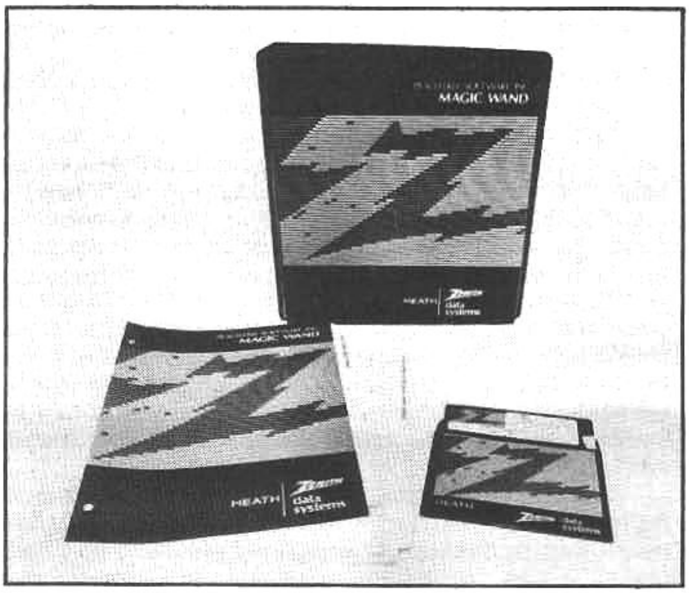
Magic Wand will respond to over 100 commands; some of the more often used are: character and line insert or delete; move to the top or bottom of the current screen or file; move (copy) or delete blocks of text; search for a word or letter; display the disk directory; insert part or all of another file into the current file; print text in almost any conceivable form; use variables and conditional commands to vary the form of the printout when certain conditions are met. Several commands have been added to the latest version (1.1) that allow easier handling of files that are larger than the memory can hold, writing blocks of text to disk, concatenating files, counting the number of words currently in text, proportional-space and bi-directional printing, printing mailing labels or envelopes, and more.

Zenith's Electronic Typing software (also called Autoscribe) is designed for office workers who lack computer experience. It is a self-prompting program that guides the user through the creation, editing, printing and storage of documents. Though popular when it was first introduced, more sophisticated word processing programs are now available.

Magic Wand is a powerful word processor, and it makes efficient use of disk space, but it does not produce a literal text file on the disk (see below). However, a Magic Wand file can be reformatted and written to a separate disk file instead of (or simultaneous with) output to a printer. This will produce a literal file—though it will only be literal if Magic Wand is instructed to not use bi-directional or other special-feature printing.

Although very versatile, Magic Wand—by its very nature—does have some drawbacks. For example, the text displayed on the screen (unformatted) does not appear as it will on a formatted printout, and the page breaks are not shown on the screen until the printing function is invoked to send the file to a printer or to display it on the screen. Magic Wand commands take the form of symbols embedded in the text, which are prefaced by a *flag character* (in this case, a backslash) to indicate that they are commands. The specific commands are unique to Magic Wand and are "transparent" during normal output, that is, when the file is sent to the printer, the embedded commands are executed by the computer but the command symbols do not appear on the printout.

A good step-by-step instruction manual and a handy quick-reference card for text-manipulation keyboard commands are included with Magic Wand. The manual guides the user through the numerous commands and their variations, and though detailed, it does not include general information to help the beginner understand what



Some word processors use the computer screen to emulate paper in a typewriter: the screen is a literal representation of the word file. But this can limit the power of a word processor. Magic Wand, for example, does not use a "what you see is what you get" approach, but relies on embedded commands to manipulate—format—the text when it is transmitted to a printer. This approach may be too complex for simple letter writing, but is very helpful in professional applications that handle extensive documents.

the computer is doing while the program is functioning. This targets the manual toward the experienced computer user, and encourages a "push the button and don't worry about what it does" attitude—an approach that can lead to a lot of frustration for the neophyte.

REQUIREMENTS: Magic Wand requires CP/M, at least one disk drive, and 48K RAM, although having 64K RAM enables the user to write longer documents. It is supplied on a hard- or soft-sector 5.25-inch or standard 8-inch disk. Available from Zenith or Heath. The price is $295.00.

LITERAL-FILE WORD PROCESSORS

PIE

PIE is a "full-feature," cursor-oriented text editor that has many of the features of a full-blown word processor, and though limited in capability (it does not have a built-in formatter), it is powerful and suitable for many general text-editing tasks. It is a cost-effective piece of software (it does a lot for little money), and for all practical purposes it can be considered suitable for either manuscript or program editing.

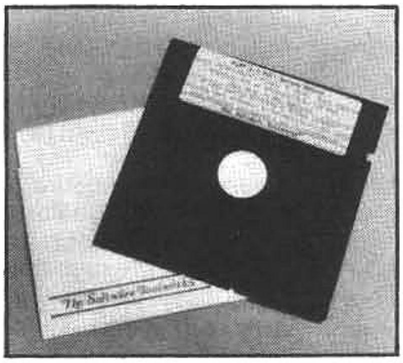
With PIE the user can create and edit text with several powerful commands—most of which can be implemented with a single keystroke. When the file is stored on the disk, it is literal (without embedded command codes) and can be transferred to another computer or sent directly to a printer where it will appear in exactly the same form as it was displayed on the computer screen.

PIE is very easy to learn, provides a good array of built-in editing commands, occupies very little disk or memory space and is designed to work with the special keyboard features of the H/Z-89. But if you must have an editor that can handle files larger than the free space in memory (for a 48K system, a file larger than about 120 sectors or 30Kbytes), or one that has built-in text formatting (rejoins lines after editing, alters margins, automatic pagination, etc.), or advanced features like merging or dissecting files, then you may want to consider one of the more powerful word processing programs.

Some of the features of PIE are: character and line insert and delete; move and copy lines; "insert mode" to insert words within displayed text; forward and reverse scrolling of text on the screen; move cursor quickly forward (tab) and backward (reverse tab); jump to beginning or end of file; automatic carriage return at end of line (word wrap); and search for a word or phrase. In addition, there is a versatile "macro" feature that allows the user to "program" a series of functions, which PIE will then perform in rapid succession (e.g., search-and-replace a misspelled or incorrect word, etc.).

Most cursor movement and text scrolling commands, and some advanced PIE commands are invoked by the special-function keys and the numeric keypad on the Z/H-89. This is not a tremendous advantage, but it does make learning and using PIE easy.

Even though PIE does not have flexible formatting features, it can be substantially expanded by using it in conjunction with a suitable formatting program. (Unlike Electronic Typing, Magic Wand, Text Processor and WordStar, PIE does not have a built-in formatter.) A stand-alone formatting program that nicely complements PIE is called "TEXT." After the document is created with PIE and written to the disk (where it becomes a "source file"), TEXT is invoked and directed to make a copy of the source file, reformat it according to prescribed commands, and send the formatted file to a destination device—either a printer or a separate disk file. (The source file remains in its original form on the disk.) By writing the reformatted file to a disk file, it is possible to review the results before the file is sent to the printer. TEXT has a wide variety of commands, and is capable of performing most advanced formatting functions. Together PIE and TEXT provide effective and relatively inexpensive editing and formatting capabilities similar to a full-function word processor.

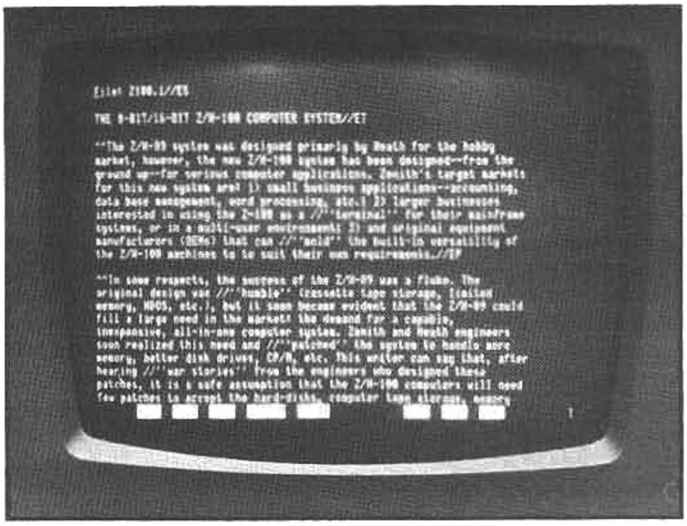


PIE is a powerful literal-file "text editor." For letters, program writing or other tasks that do not require extensive rewriting and reformatting, this inexpensive editor is an ideal choice. PIE is designed to use the special features of the Z/H-89 keyboard, providing single-key commands for many common functions.

REQUIREMENTS: PIE and TEXT are available for either HDOS or CP/M, and are provided on hard-sector 5.25-inch disks. Memory requirement for PIE is 32K, however, 48K or more is recommended. This software is available from Zenith/Heath as a "Full Screen Editor" and "Text Formatter," or as "PIE" and "TEXT" from The Software Toolworks (see appendix) for about $50.00 each.

TEXT PROCESSOR

All of the features of PIE, plus a good deal more, are incorporated in Text Processor. This editor uses more than 50 commands to edit and format text on the screen,



```
File 2100.1//ES

THE 8-BIT/16-BIT Z/H-100 COMPUTER SYSTEM//ET

``The Z/H-89 system was designed primarily by Heath for the hobby
market, however, the new Z/H-100 system has been designed--from the
ground up--for serious computer applications. Zenith's target markets
for this new system are: 1) small business applications--accounting,
data base management, word processing, etc.; 2) larger businesses
interested in using the Z-100 as a //``terminal'' for their mainframe
systems, or in a multi-user environment; 3) and original equipment
manufacturers (OEMs) that can //``mold'' the built-in versatility of
the Z/H-100 machines to to suit their own requirements.//EP

``In some respects, the success of the Z/H-89 was a fluke. The
original design was //``humble'' (cassette tape storage, limited
memory, HDOS, etc.), but it soon became evident that the Z/H-89 could
fill a large need in the market: the demand for a capable,
inexpensive, all-in-one computer system. Zenith and Heath engineers
soon realized this need and //``patched'' the system to handle more
memory, better disk drives, CP/M, etc. This writer can say that, after
hearing //``war stories'' from the engineers who designed these
patches, it is a safe assumption that the Z/H-100 computers will need
few patches to accept the hard-disks, computer tape storage, memory
```

For less than $50 Text Processor provides remarkably powerful editing capabilities. It has flexible formatting, search-and-replace functions, the ability to handle files larger than memory, and a feature that allows material from another file to be added to the current working file. The cursor-arrow and special-function keys of the Z/H-19 and Z/H-89 are employed.

yet it is almost as easy to use as PIE. This additional power doesn't come without a trade off, however, as Text Processor is more than three times as large as PIE (PIE is 22 sectors, Text Processor is 74). But, unlike PIE, Text Processor is able to work with files larger than free RAM space, by first reading into memory only a part of the source file, and then, at the user command, this first portion is written back to disk and the remaining portion is loaded into RAM (and displayed on the screen) for editing.

Some additional functions that can be performed with single keystroke sequences are: split and merge lines; reformat text to any margin, paragraph by paragraph; search and search/replace functions (capable of recognizing spaces and carriage returns); insert another file from disk into the current file; automatically center titles; display helpful information on the screen when desired (sometimes easier than going to the manual); and a very handy "OOPS" key to undo the last command.

REQUIREMENTS: Text Processor is available for both HDOS and CP/M, and is available on hard-sector 5.25-inch disks. The minimum memory requirement is 32K. It can be purchased from many Heath Electronic Centers or from Newline Software (see appendix) for under $50.00.

WORDSTAR

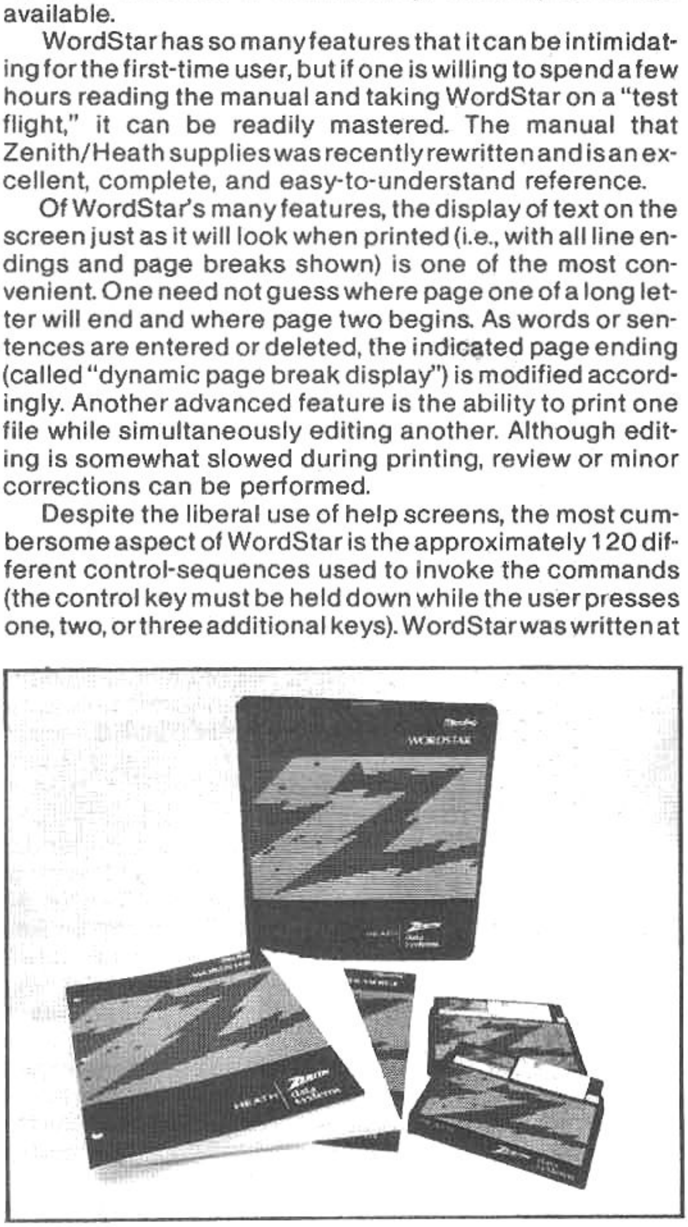
WordStar is a versatile and powerful word processor, and it is possibly the most widely used word processing software for microcomputers. The preface to the users' manual describes WordStar as a visual experience, similar to that of an aircraft cockpit. This description is quite appropriate because various menus and helpful displays guide the user through the operation of WordStar, and the user is put in *complete* control of the text or document. Text may be entered, edited, and extensively reformatted; paragraphs can be copied, moved or deleted; and text may be sent to a printer or disk file, or added to another disk file. In addition, disk directories can be listed from within the program, and other programs can be executed from within WordStar and upon completion, command will automatically be returned to WordStar. This comprehensive word processor does all of this and, if you can believe it, much more!

The first-time user of WordStar is assisted by "help menus" that are displayed on the screen along with the text file. These displays guide the user through the many functions available within the program, and the user may select the "level" of help he desires. "Level three" provides the most extensive help display (and, consequently, displays the least text on the screen), while "level zero" provides no help screens, allowing the greatest amount of text to be displayed. This programmable help feature ensures that the novice can make full use of the functions. Selecting from the menus is like ordering food in a restaurant—but the experienced user, like a steady patron, will not be bogged down by unnecessary prompts from the "waiter." WordStar is user-friendly, and help is always available.

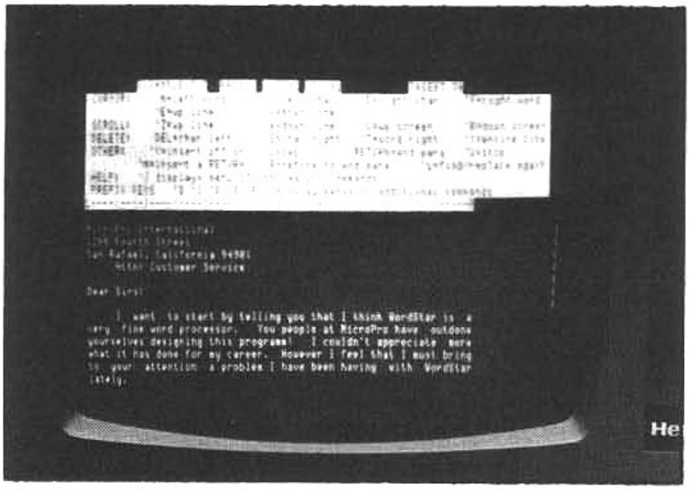
WordStar has so many features that it can be intimidating for the first-time user, but if one is willing to spend a few hours reading the manual and taking WordStar on a "test flight," it can be readily mastered. The manual that Zenith/Heath supplies was recently rewritten and is an excellent, complete, and easy-to-understand reference.

Of WordStar's many features, the display of text on the screen just as it will look when printed (i.e., with all line endings and page breaks shown) is one of the most convenient. One need not guess where page one of a long letter will end and where page two begins. As words or sentences are entered or deleted, the indicated page ending (called "dynamic page break display") is modified accordingly. Another advanced feature is the ability to print one file while simultaneously editing another. Although editing is somewhat slowed during printing, review or minor corrections can be performed.

Despite the liberal use of help screens, the most cumbersome aspect of WordStar is the approximately 120 different control-sequences used to invoke the commands (the control key must be held down while the user presses one, two, or three additional keys). WordStar was written at

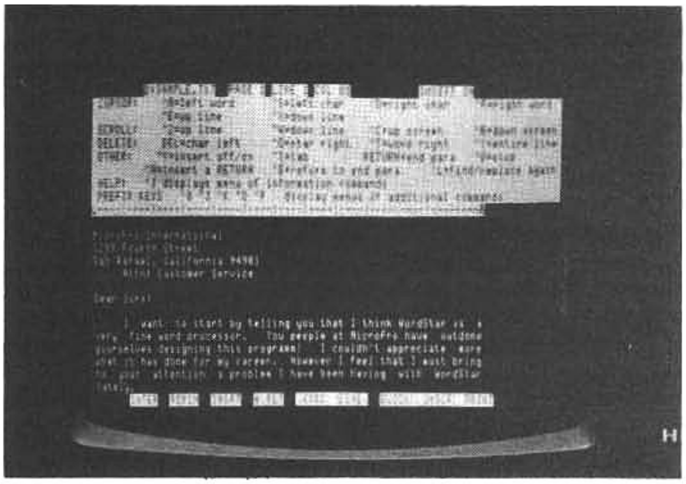


WordStar is one of the most powerful and complex word processing programs available today; and the step-by-step instruction manual produced by MicroPro Intl. (the program authors) and Zenith is an excellent and thorough reference. It provides helpful information on basic computer functioning, working with the operating system, creating new files, producing backup files, and other important working procedures.



```
CURSOR:   ^A=left word   ^S=left char   ^D=right char   ^F=right word
          ^E=up line     ^X=down line
SCROLL:   ^Z=up line     ^W=down line   ^C=up screen    ^R=down screen
DELETE:   DEL=char left  ^G=char right  ^T=word right   ^Y=entire line
OTHER:    ^V=insert off/on  ^I=tab   RETURN=end para   ^U=stop
          ^N=insert a RETURN  ^B=reform to end para   ^L=find/replace again
HELP:     ^J display menu of information commands
PREFIX KEYS:  ^K ^O ^Q ^P  display menus of additional commands

MicroPro International
1299 Fourth Street
San Rafael, California 94901
     Attn: Customer Service

Dear Sirs:

     I  want  to start by telling you that I think WordStar is  a
very  fine word processor.   You people at MicroPro have  outdone
yourselves designing this program!   I couldn't appreciate  more
what it has done for my career.  However I feel that I must bring
to  your  attention  a problem I have been having  with  WordStar
lately.
```

WordStar uses a portion of the screen to display functional information for the user. Because this extensive program has over 120 text commands, it can be confusing for the beginner, but the "help" screens make familiarization much easier. WordStar is a very professional program, and is widely considered to be not only powerful, but also one of the most reliable software products available.



```
CURSOR:   ^A=left word   ^S=left char   ^D=right char   ^F=right word
          ^E=up line     ^X=down line
SCROLL:   ^Z=up line     ^W=down line   ^C=up screen    ^R=down screen
DELETE:   DEL=char left  ^G=char right  ^T=word right   ^Y=entire line
OTHER:    ^V=insert off/on  ^I=tab   RETURN=end para   ^U=stop
          ^N=insert a RETURN  ^B=reform to end para   ^L=find/replace again
HELP:     ^J display menu of information commands
PREFIX KEYS:  ^K ^O ^Q ^P  display menus of additional commands

MicroPro International
1299 Fourth Street
San Rafael, California 94901
     Attn: Customer Service

Dear Sirs:

     I  want  to start by telling you that I think WordStar is  a
very  fine word processor.   You people at MicroPro have  outdone
yourselves designing this program!   I couldn't appreciate  more
what it has done for my career.  However I feel that I must bring
to  your  attention  a problem I have been having  with  WordStar
lately.
```

The universal design of WordStar uses control-sequences to initiate text-handling commands, and it does not, therefore, support the cursor-arrow and special-function keys on the Z/H-89. However, a modification program called 89/Star "patches" WordStar to support these useful keys. This simplifies the familiarization process for beginners, and veteran users who are accustomed to the powerful Z/H keyboard will enjoy the added convenience.

a time (late 1970s) when few microcomputers had special-function keys, so it makes use of the one key which all microcomputer keyboards have—the CTRL key: as a result the cursor movement and special-function keys of the Zenith/Heath keyboard are not supported. Some people find this objectionable, but once the mental adjustment is made, most of the control-sequences become second-nature.

An additional technical note: WordStar is listed here under literal-file word processors because it has two editing modes. The "non-document mode" produces a literal file, but in this mode the program is limited in power (paragraphs cannot be reformatted, etc.). The "document mode" brings all the power of WordStar to bear, but the resultant file is not literal (the line endings are carriage returns with the high-order bit set). However, some users may like to know that a WordStar file with non-literal carriage returns is easily converted to a literal file by using a little-known option in the CP/M PIP utility. If the file is copied using the "z" flag (explained in the CP/M manual), the high-order bit is stripped and the file can be edited or manipulated by other literal-file software.

WordStar is part of an extended family of powerful MicroPro software products that includes: MailMerge, a file-merging program that can produce "personalized" form letters; SpellStar, a program to help eliminate misspelled words; DataStar, a data-entry and database program that will handle invoices, shipping and receiving forms, address lists, etc.; and SuperSort, a sorting and merging program that allows the conversion of files to a different record type, changing record lengths, merging up to 32 files at 560 records per minute. Some of these MicroPro products are discussed later in this chapter.

REQUIREMENTS: WordStar is available from Zenith and Heath or from MicroPro International (see appendix). It requires CP/M and a minimum of 48K RAM. WordStar and all compatible MicroPro programs are distributed on hard- or soft-sector 5.25-inch or standard 8-inch disks.

WORDSTAR UPGRADE FOR Z/H SYSTEMS

As mentioned, WordStar uses numerous control-key sequences. This may not be bothersome for experienced users (particularly those who have become very familiar with and primarily use only WordStar), but the new user, or the individual who uses other editing software on Zenith/Heath computers, may find the control-key method cumbersome, especially for the movement of the cursor.

This limitation can be overcome with "89-Star," a set of unique programs that readily customize WordStar (and the overlay files) to support the cursor-arrow and special-function keys on the Z/H-89 (or any CP/M system using a Z/H-19 terminal). These programs are "self-installing patches." All the user has to do is copy the appropriate files to a disk containing WordStar (multiple disk drives make the job easier). Then type: **SUBMIT WSPATCH**, and after a few minutes type **SUBMIT MNPATCH**. When both program patches are complete, the customized WordStar program is ready to run.

The modified WordStar will now acknowledge 30 commands that make use of the special keys on powerful Zenith/Heath keyboard. Cursor movement and screen scrolling is performed by the numeric keypad (shifted). And the number of commands sent by the keypad is increased by pressing a "quick prefix" key—the (.) on the keypad—that extends keypad control of the cursor (sending it to the top or bottom of the file, etc.) and invokes search-and-replace functions. The special-function keys are programmed (and labeled on the screen—see photo) to perform eight functions. They are: interrupt the current command, find-and-replace again, insert mode, "hard" carriage return, erase word, erase line, block prefix, onscreen prefix, and print prefix. In addition, the carriage return key on the main keyboard is reprogrammed to call "help."

89/Star is part of a series of software products that include: EP/Star, an Epson print driver for WordStar; Graph-It, a graphing system that will plot data from SuperCalc, Condor, dBaseII, MBasic, or other programs; and a Flight Simulator, that is said to be a "real flight simulator" not a game (much harder to learn, but it does not have a canned "flight plan").

REQUIREMENTS: 89/Star is designed to modify

WordStar version 2.2 or higher, and runs under CP/M version 2.2.03 (the 2.2.02 version will not work). It is distributed on a hard-sector 5.25-inch disk, and the cost is $19.00 ($2.00 for the manual only) from Patrick McNally (see appendix).

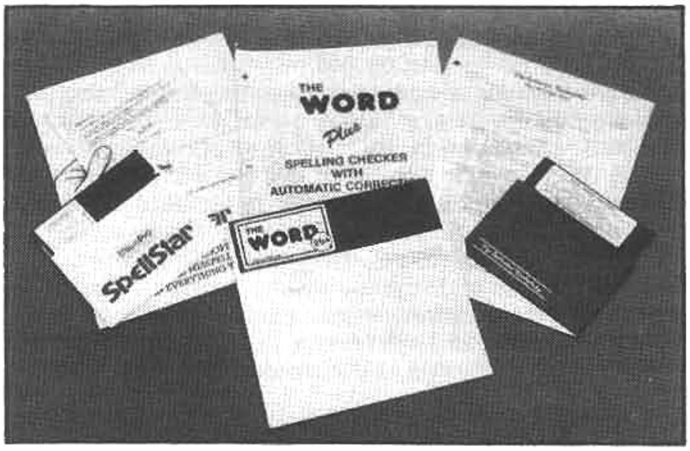
SPELLING/PROOFREADING PROGRAMS

Computers are extremely suited to jobs that are repetitive and require a high degree of accuracy. A task that fits this description perfectly is proofreading a document for spelling errors, a chore that will bring out the "human" element in even the best editor. Well, computer proofreading programs are not perfect either, but they're close!

A computer "proofreads" a document by comparing every word in the text with a file of correctly spelled "dictionary words." After the comparison process, all words from the text that are not matched with words in the dictionary are displayed for the user to review and, as necessary, correct. (Obviously, not every word in the document will be found in the dictionary; for example, proper nouns, technical terms and unique words will probably not be matched in the comparison, even if they are correctly spelled.)

Depending on the design of the particular proofreading program, the unmatched words will either be marked in the document or a list of unmatched words will be displayed for review. In the former case, a word processing program can be used to locate the marked words (in context) and correct them or, in the latter case, the program may provide some sort of internal correction feature to replace or selectively mark unmatched words that are in fact misspelled (for later location and correction by a separate word processing program). And with some programs, if an unmatched word is found to be correctly spelled but it does not have an equivalent in the dictionary, it can be automatically added to the main dictionary.

In this section we will review four spelling (proofreading) programs: SpellGuard, SpellStar, Spell, and Word Plus. Each program is reliable and of professional quality, however, they function somewhat differently. Since the cost-effectiveness of these programs is realized by minimizing the amount of time the user must interact with the software, your specific application should be considered carefully before you make your choice. For example, if you are using MicroPro's version of WordStar, you may wish to consider SpellStar, as it can be invoked without leaving WordStar (the version of WordStar sold by Zenith/Heath does not support this feature).



After a document is created with word processing software, "proof reading" software can be used to check for spelling errors. The spelling programs shown above represent three different approaches: SpellStar is designed specifically to work with WordStar; Word Plus is extremely comprehensive for general use; and Spell is an inexpensive, but capable, program.

If complete accuracy is a primary requirement and disk storage space is not a restriction, a spelling program that utilizes a literal dictionary is the best choice: some programs "compact" the dictionary by using a pattern of commonly used suffixes, prefixes, and roots, instead of literally comparing every word in the document with every word in the dictionary—occasionally, though rarely, the non-literal method can be faulty. However, the reader must realize that *no proofreading program will find a correctly spelled word that is incorrectly used within the text.* For example, if the word "from" is used in place of the word "form," the mistake will not be identified by any proofreading program. Only human eyes can detect this kind of transmutation. Spelling programs are a useful aid for secretaries, editors, and all those who deal with words, but they have definite limitations!

SPELLGUARD

SpellGuard is a fast and accurate menu-driven proofreading program. It is accompanied by a detailed and understandable operating manual that is well suited to a professional environment.

The standard dictionary that accompanies SpellGuard is a literal 20,000-word file that is compressed into 53 Kbytes. A larger, optional dictionary containing 30,000 to 40,000 words and specialized dictionaries for the medical and legal professions are under development, and a British spelling version (occupying 62 Kbytes) will also be available soon.

Once the filename of the document to be proofread is entered, SpellGuard checks all words against the dictionary. Words that do not match are listed one by one (or, optionally, all at once), permitting the operator to decide whether each word is to be ignored, marked for correction in the document, or added to the dictionary. If the user is working with a large document (10-20 pages) and the original 20,000 word dictionary, the program may generate a long list of unmatched words. However, a typical list can be diagnosed in just a few minutes (or seconds if the user is a competent speller) and dissipated with commands that only require a single keystroke. With additional use, the dictionary may be expanded by the user (the size of the dictionary is limited only by capacity of the disk), and the resultant list of suspect words will generally be smaller.

When misspelled words are identified, the user marks the errant words in the text. SpellGuard accomplishes this by replacing the last character of the misspelled word with a flag symbol. After proofreading and flagging is completed, a word processing program is used to locate all of the flag symbols (using the search mode of the word processor). The user must then delete the flag symbol and restore the last character of the word, and correct the spelling error. Replacing the last character adds some time to the correction process, but does not disturb the format of the document file (no characters are added).

SpellGuard handles hyphenates by displaying both segments of the group and indicating if the hyphen occurs within the line (compound word) or at the end of a line (hard-hyphenated word). This allows verification not only of spelling, but also hyphenation. However, there is one drawback: SpellGuard will display *all* hyphenates in the file. This will slow the proofreading process considerably, especially in a technical manuscript with many compound words, unless these hyphenated word-pairs have been added to the dictionary.

This is a very professional software package. If the user needs a proofreading program with extensive documentation—especially important if it will be used by non-computer-trained personnel—SpellGuard is a good choice.

REQUIREMENTS: SpellGuard requires CP/M and is available on a double-density 5.25-inch or a standard 8-inch disk. At least 32K of RAM and one drive is required, however, a second drive will allow much easier operation. It is available from Sorcim (see appendix) for $295.00.

SPELLSTAR

This is a CP/M spelling checker that is designed to work within the MicroPro version of WordStar, and is executed by choosing the "S" option from the WordStar main menu (the "S" option and the ability to run this program are not supported by the version of WordStar sold by Zenith/Heath). SpellStar is a menu-driven program using a 20,000 word, compressed, literal dictionary. Supplemental dictionaries can be created, and SpellStar will proofread a file using the main and one supplemental dictionary on each "pass." SpellStar allows additions to the main and any supplemental dictionaries, with the only limitation on size being the amount of free space on the disk.

The literal dictionary incorporated in SpellStar means that this spelling checker does not "build" words from suffixes, prefixes or roots. Thus, all marked words are misspelled, typographical errors, or not in the dictionary. The user can have confidence that *ALL* the words not located in the dictionaries (main and supplemental) will be flagged.

Perhaps the most powerful feature of this spelling checker is the use of an enhanced WordStar edit mode for review of suspect words—using a WordStar overlay file. The review process allows the user to: bypass the word, ignore the word, add the word to main dictionary, add the word to the supplemental dictionary, or "fix" (correct) the word using the edit mode.

The bypass option holds the word in abeyance, so that after the initial review is complete, words that must be looked up in a dictionary can be checked at one time. The ignore option will cause SpellStar to remember the word (in a temporary memory that will store the latest 20 words) and not stop at any other occurrence in the file. When words are added to the main or supplemental dictionary, they can also be placed in the temporary memory, guarding against duplicate additions to the dictionary. However, if a word is misspelled and is "fixed" with the last option, it must be corrected at each occurrence throughout the file—no global correction is performed; although the search-and-replace function of WordStar may be invoked for this purpose.

SpellStar is "married" to WordStar because it must be invoked from within WordStar and it uses the editing features of WordStar to correct the document. But it is an excellent marriage. WordStar is a capable editor that gives SpellStar the most powerful editing features available *within* any proofreading program (although other spelling programs are designed to delegate the correction process to an external editor, i.e., WordStar or any other editor).

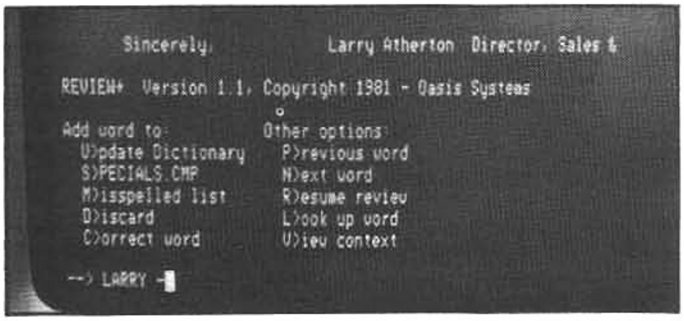
REQUIREMENTS: SpellStar requires CP/M and a non-Zenith/Heath version of WordStar. It is available from MicroPro (see appendix). All Zenith/Heath disk formats are supported.

WORD PLUS

The Word Plus is a revised version of a spelling program—originally developed by Oasis Systems—called The Word. The early version rivaled the very best proofreading programs on the market, but the designers did something that is unusual in the world of computer programming: they encouraged users to comment on the original version of their program, and incorporated the best of these suggestions in a completely revised program. As a result, The Word Plus, is almost in a class by itself!

The Word Plus utilizes a literal 45,000 word dictionary that has been compressed to occupy only 160 Kbytes. This large dictionary greatly reduces the number of words that are flagged as suspect. The dictionary can be expanded by selecting the "update" function during review of the flagged words. This does not add words to the original dictionary, instead, words are added to an "up-

SpellStar was developed by the same company (MicroPro) that created the WordStar, and is designed to proofread documents created by WordStar. This relationship is quite permanent, because SpellStar is invoked from within WordStar and will not function otherwise.



```
      Sincerely,           Larry Atherton  Director, Sales &

REVIEW+  Version 1.1, Copyright 1981 - Oasis Systems
                        o
Add word to:           Other options:
  U>pdate Dictionary     P>revious word
  S>PECIALS.CMP          N>ext word
  M>isspelled list       R>esume review
  D>iscard               L>ook up word
  C>orrect word          V>iew context

 --> LARRY -█
```

Word Plus is possibly the most extensive spelling software currently available for microcomputers. This CP/M program allows the user to locate, view in context, and correct "in place" or mark (for later correction) suspect words that are not found in the 45,000-word internal dictionary. The screen indicates that the word "LARRY" (lower part of screen) was not found in the dictionary. The user then selected the "view" option to display the line in which the suspect word was located (top line). The suspect word may now be discarded, or added to the dictionary (an appropriate choice if the word "LARRY" is often used in letters, contracts, etc).

date dictionary," to which the proofreading program refers after it searches the main dictionary. This allows not only the use of different update dictionaries for specific applications, but also the incorporation of new versions of the main dictionary (as they are made available by Oasis Systems), without losing all of the special words that have been added by the user.

The "review" portion of the program displays each word in the suspect list one at a time, permitting the user to make several decisions about its disposal. If the word is identified as correctly spelled, it may be added to the update dictionary or discarded. If it is misspelled, an error flag—normally an (*) symbol, but the user may choose a different flags symbol if desired—is placed before the misspelled word in the document file, permitting fast location by word processing software. The error mark does not *replace* any characters in the document, rather it is *added* to the file.

However, if the reader is uncertain about the spelling or usage, a powerful feature of Word Plus will help. With an additional command the program will display a line of text from the document file with the suspect word centered, providing an "in context" view. If you are still uncertain about the word, another command will tell Word Plus to look up and display the correct spelling. If the program can find a close match, a list of these "possibles" will be displayed. If the suspect is now determined to be misspelled, the user can insert an error flag (for later reference by a word processor) or another powerful function may be invoked: "in context correction." This command will cause Word Plus to automatically correct the misspelled word with spelling chosen either from the main dictionary or as entered by the user. It's hard to imagine any useful improvements that could be added to this superb program, but there are even *more* features included with Word Plus.

Three interesting utility programs are included in The Word Plus package. The first is called "Dicsort;" a high-speed alphabetizing program that will sort a disk file consisting of a list of words (in the format of one word on each line). Word Plus requires the update dictionary (the list of words added to the main dictionary) to be alphabetized, and this is automatically done by "review" when new words are added to the dictionary from the list of suspect words. However, if the user wishes to create a different update list, or if the update dictionary is modified (by using a word processor or text editor), the list must be processed by Dicsort before the proofreader can properly access the file. This is an extremely useful feature because it permits the user to go directly into the update dictionary to add words or to correct misspelled words that may have been inadvertently added to the dictionary (with many spelling programs, words can only be added to the dictionary from a list of suspect words created after a file is proofread).

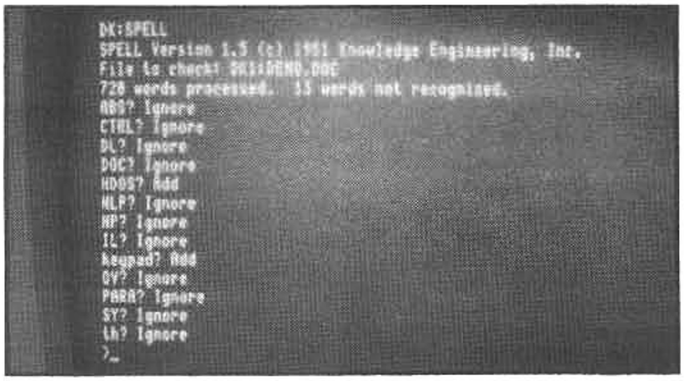
Another utility provided with Word Plus is "WC." This is a program that will count every word in a text file. Publishers and copy editors will find that the ease and lightning speed of this program may make "estimating" the words in a manuscript a thing of the past.

Another utility is available to insert discriminatory *soft-hyphens* in a text file. Soft-hyphens indicate where legal breakpoints exist between the syllables of a word: this is not the same as a *hard-hyphen*, which must be used between compound words or with certain prefixes. When the "soft-hyphenated" text file is formatted by a word processor (or typesetting equipment) capable of recognizing soft-hyphens, a multi-syllable word that falls at the end of a line will be broken apart *at an indicated soft-hyphen* and a hard-hyphen is inserted at the end of the line. By introducing as many soft-hyphens as possible in a text file, the formatted (or typeset) text will have a smoother right margin. (Some word processors, such as Magic Wand and WordStar, also incorporate a soft-hyphen feature.)

REQUIREMENTS: The Word Plus runs under CP/M and is available on a standard 8-inch disk (a soft-sector 5.25-inch version is supposedly under development). 32 Kbytes of RAM are required, and for maximum utilization, a word processing program and a second disk drive are recommended. The Word Plus is available from Oasis Systems (see appendix) for $150.00.

SPELL

If the user has a single-disk system, or the proofreading is primarily limited to letters and short manuscripts, this inexpensive spelling checker might be the answer. Spell is available for both HDOS and CP/M, and will operate in a system with only one 5.25-inch single-density disk. The minimum memory requirement is 48 Kbytes, but a larger dictionary file is included for users that have a full 64 Kbytes of memory available.



```
DK:SPELL
SPELL Version 1.5 (c) 1981 Knowledge Engineering, Inc.
File to check: SY1:DEMO.DOC
728 words processed.  33 words not recognized.
ABS? Ignore
CTRL? Ignore
DL? Ignore
DOC? Ignore
HDOS? Add
HLP? Ignore
HP? Ignore
IL? Ignore
keypad? Add
OV? Ignore
PARA? Ignore
SY? Ignore
th? Ignore
?_
```

Spell is an excellent and inexpensive general-purpose spelling checker. Every word in the document is compared to a 50,000-word dictionary and unmatched (suspect) words are presented in sequence, allowing the user to select: IGNORE (a correctly spelled suspect word that happens to not be in the dictionary), ADD (to include the suspect word in the dictionary), or MARK (to replace the last letter in the suspect word with an error character so it can be located later and corrected with an editing program).

Despite the modest appearance and price of Spell, there is little tradeoff in performance. Spell uses a non-literal 20,000 dictionary that incorporates a prefix and suffix table to build thousands of additional word combinations. The result is a dictionary with an effective size of over 50,000 words. Spell reads this dictionary file into RAM, and since the access time of system RAM is very fast, Spell can look-up about 2000 words per minute (in a Z/H-89 or -90).

As suspect words are located, they are presented one-at-a-time for disposition. The user can choose to add the word to the dictionary, add a portion of the word to the root table (e.g., add "store" as a root in "restore"), ignore the word, or mark the word in the text file (Spell replaces the last letter in the word with an error flag).

If the word is added to the main dictionary, Spell uses a unique technique to expand the dictionary without increasing the size of the file (it cannot grow larger or it will not fit in system RAM). A method of "graceful degradation" is employed to compress additional words into the same space. The result of this compression is the increased possibility that Spell will let error words slip by without detection. For example if 3000 additional words are added to the dictionary, Spell will miss about two error words for each 1000 located, obviously adding only a few thousand words (the average user will probably add less than 2000) has only a slight effect on the reliability of Spell.

Since Spell uses a prefix and suffix table to construct words, it will occasionally allow misspelled words to pass undetected. For instance, the word "bater" might be accepted (correct spelling is "batter"), as a simple suffix rule allows the addition of "er" to a legitimate root (in this case "bat"). However, the correct algorithm—also included in Spell—is to double the final consonant before the addition of the suffix. Nonetheless, there are certain times when this technique recognizes a misspelled word as legitimate.

During execution, the main Spell program, the dictionary, and the list of suspect words are all resident in RAM; and the RAM can get crowded. As a result, a long document may fill the suspect word list and cause Spell to terminate the proofreading phase. After the suspect words are disposed of (and presumably some are added to the dictionary), the checking phase must be re-run from the start. This will normally not be a problem with short documents (like letters), but with long text files, Spell may be cumbersome to use.

Another minor drawback of Spell is the "one-shot" presentation of suspect words. Unfortunately, once the user has selected a disposition for a word on the review list, it is not possible to change the option (i.e., if the user accidentally hits the wrong command key while reviewing words, there's no way to undo the command).

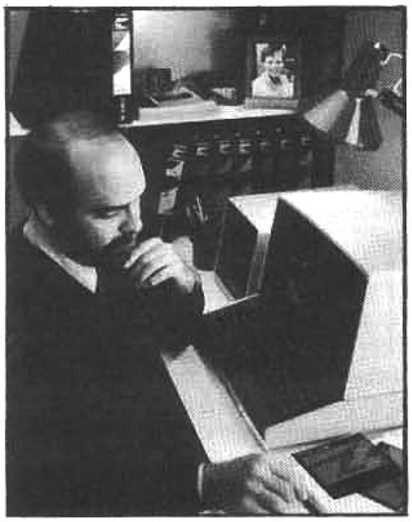
REQUIREMENTS: Spell is available for both HDOS and CP/M. A minimum memory of 48 Kbytes is required, but 64 Kbytes will improve performance. It is distributed on a hard-sector 5.25-inch disk for HDOS and CP/M or an 8-inch disk (for CP/M only). Spell can be purchased from The Software Toolworks (see appendix) for $49.95.

BUSINESS ACCOUNTING SOFTWARE

Large businesses have been computerized for years, but with the development and popularity of microcomputers, even the smallest business can benefit from automated general ledger, accounts receivable and payable, inventory control, etc.

Most of the following applications programs are written in the high-level computer language known as Basic (see below: programming languages). Unlike word processing or most database management programs (which are usually written in directly-executable assembly-language or machine-code) this type of software requires that a suitable "interpreter" program reside in RAM before the applications program will execute. (In effect, the program communicates with the computer CPU through the interpreter program.) However, the language interpreter is not commonly supplied with the accounting programs (except for the Peachtree series), but must be purchased separately.

Experience has shown that, although computerized accounting can save the small business an enormous amount of time, manual record-keeping should be continued for at least two months after the implementation of any business accounting program. It is very tempting to immediately set aside the stacks of paper and ledger-books; but, as with anything that is new and complex, *there is always a chance that something can go wrong.* This is especially true during the familiarization period, and this is the time when the user is least capable of correcting what may be only a minor problem, or is most likely to increase the damage by trying to improperly "save" the situation.



Computers were originally developed to perform fast and accurate numerical calculations; as a result, business accounting became one of the most useful applications for early, expensive computer systems. Today, full-feature microcomputers, like those from Zenith/Heath, can provide any small business with efficient, accurate and inexpensive accounting assistance.

Most problems spring up during the first few weeks that a new system or new software is in operation. During this period a glitch can leave the user totally in the dark, unless manual backup records are available to set things straight. Fortunately it isn't common, but some errors can have a wide-ranging catastrophic effect on important computer files, perhaps wiping out every trace of who-owes-who-what; this is a nightmare that most businesses probably don't need! In the long run, *it is wise to be patient and make certain that all operators are thoroughly familiar with the computer and the software before important records are dedicated totally to the computer (although up-to-date "hard-copy" printouts of accounting records should always be maintained).*

SOFTSTUFF GENERAL LEDGER II

This accounting package (written by Clark Systems, Corp.) is directed at the professional person and small-business owner. The limitation on the number of transactions that can be processed, or the number of accounts that can be maintained, is set only by memory capacity and the number of disk drives. For most small businesses, all

As a result of efforts to encourage software development by independent programmers, Heath's "Softstuff" department offers many useful and inexpensive programs for the hobbyist or general user. These user-written programs are tested (though only on a limited basis) and are provided with helpful documentation.

records for an entire fiscal year can be kept on one soft-sectored disk. Larger businesses may have to use multiple disks—each dedicated to a shorter fiscal period (perhaps a quarter). At the end of the fiscal period, the program will "close the books."

General Ledger II is manipulated by two main menus; one controls the creation of new accounts and the printing of financial statements, and the other guides the user through the routine tasks of data entry and the selection of reports to be printed. Most keyboard input is made with the entry of single letters or numbers, and the program is equipped with several automatic error-detection and correction features. As a safety feature, only authorized personnel can utilize the ledger, because the program is locked to anyone unaware of the system password. This can prevent the loss (either intentional or accidental) of important financial and tax records.

The manual that accompanies General Ledger II is written for a person who is experienced in computer operation and knows how to use CP/M to copy files, "sysgen" disks, etc. Further, the manual assumes the user is already familiar with general-ledger accounting practices. Little help is provided for those that may be implementing a general ledger for the first time. Numerous sample printouts are provided to help the user understand the general structure of the ledger.

The program automatically allocates debits and credits, so the user need not remember to enter a minus sign before debit account entries. Accounts, headings, subtotals, and totals may be added or changed after the ledger is set up, to allow for initial oversights or business expansion.

REQUIREMENTS: General Ledger II is available from Zenith and Heath, and will run under CP/M or HDOS (depending on the version that is ordered). It requires the MBasic interpreter (called Basic-80 under CP/M), a minimum of 48K RAM, and a 132 column printer. It is distributed on hard- or soft-sector 5.25-inch disks.

SOFTSTUFF INVENTORY

This program is a companion to the General Ledger II discussed above and is aimed at the same small-business market. Most businesses that have a need for an inventory program will find this program an excellent starting point. The program will store up to 300 inventory items on a system with a single disk drive (H-37), and if two drives are used (one for programs and one for data) 1000 part numbers can be manipulated. In addition, the program will support multiple data disks, allowing an almost unlimited number of inventory items.

Softstuff Inventory will allow up to 12 letters and numbers in the inventory part numbers. Spaces are also allowed, as long as they are not used as the first character. Upper and lower case letters are read as separate characters, and the user must be careful to enter them correctly. The program requires that all part numbers be the same length (up to 12 maximum), so shorter part numbers must be entered with leading zeros.

The program generates three data printouts: an inventory status report, listing current quantities, prices, and usage; an abbreviated version of the inventory status that highlights inventory items that need reordering; and an abbreviated report showing part numbers and prices (in effect, a catalog of products and prices).

The manual is short but somewhat more understandable than the general ledger documentation, although in all fairness, the inventory program is inherently more simple than the ledger. Once the program is mastered, only an occasional reference to the manual is required.

REQUIREMENTS: Softstuff Inventory runs under CP/M or HDOS (depending on the version that is ordered), and requires the MBasic interpreter (called Basic-80 under CP/M). It requires a minimum of 48K RAM and a 132 column printer, and is distributed on a hard- or soft-sector 5.25-inch disk.

PEACHTREE SERIES 5

Peachtree business programs are designed to provide a turn-key system of accounting and data-storage modules that will work together to exchange information or as independent programs. These programs are very user-friendly and the operator need not have any knowledge of computers or operating systems. Once a program disk is inserted and the boot sequence begun, the user invokes all functions through the menus displayed from within each program, including disk formatting and file backup. In order to accomplish this, each program disk is supplied with a fully-functioning system, including CP/M, utilities and Basic-80 (where required).

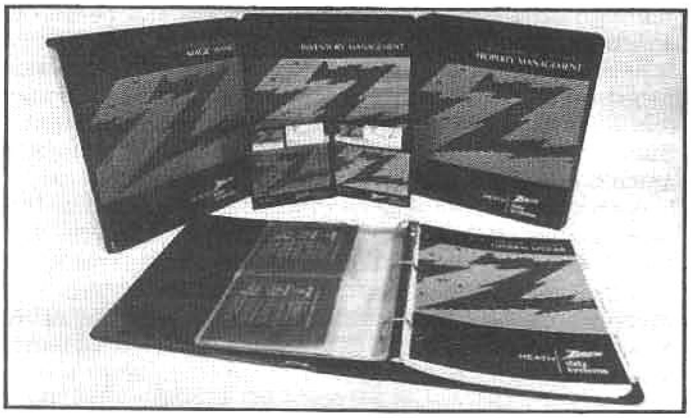
Data integrity is a vital part of any professional accounting system, and a computer accounting system must have reliable backup and security procedures to minimize the chance of losing non-recoverable accounting data. In this respect, the Peachtree series has several outstanding features.

Security is provided by a password lockout, and most of the programs use multiple passwords—the correct password must be entered before the program will begin to operate and an additional password is required to enter portions of the program that will alter account amounts or delete transactions. The controller and operator passwords are established during program set-up and initialization. Further security is provided by validation of each keyboard entry before new data is passed to the system files. Most programs contain mandatory file backup procedures to prevent data loss in the event of software or hardware failure. As a last resort, if all copies of a system file are damaged and the data cannot be accessed through normal procedures, a program called "File Fix" can be used to recover as much data from the file as possible.

All of the programs in the Peachtree series can accommodate large numbers of transactions, accounts, part numbers, etc. For example, if 8-inch disks are used, the general ledger will maintain up to 500 accounts with over 10,000 transactions in the holding file, and the inventory program will maintain over 5500 items. The Peachtree programs will certainly satisfy the requirements of most small-size businesses.

The extensive documentation takes the user step-by-step through familiarization. The manuals effectively explain computer terminology and provide excellent guidance for initializing the programs. And the programs are supplied with dummy data files that can be used to try out the examples presented in the manuals. The user may add to these files, delete entries, print reports, etc. and—in effect—become familiar with all functions of the software *before* entering real business data. This makes it easy to learn how the system works, and it gives realistic practice without the risk of losing actual records.

A short description of each program in the series follows. This synopsis can only provide the prospective user with a cursory feeling for the package. Each program includes many features that we do not have the space to ex-



Zenith and Heath support their computers with software suitable for professional applications. The Peachtree-developed accounting packages are flexible in design, very thorough, and are totally integrated with Zenith/Heath hardware to produce a "turn-key" package (the disks are complete with operating system, MBasic interpreter, source code, initialized data files, etc.). Comprehensive, easy-to-read manuals accompany all professional software, and a telephone "hotline" is provided to answer specific customer queries.

plore.

The Peachtree "General Ledger" includes programs for creation, maintenance and updating of files, and a report-generator for income statements and balance sheets, departmental reports, transaction reports, and depreciation schedules. With soft-sector 5.25-inch disk, the General Ledger System will handle approximately 300 accounts and 6500 transactions in the holding file. With 8-inch disks, it will handle up to 500 accounts and over 10,000 transactions.

"Accounts Receivable" produces invoices and monthly statements, and also maintains files of customers' names, addresses, telephone numbers, order numbers, balance due and other status information. It will post journal entries to the General Ledger, as needed.

"Accounts Payable" keeps track of current and late accounts, cash flow, and vouchers, and can also be set up to print checks automatically on the system printer. Like all programs in this series, operations are invoked through menus, and it can interact with the other programs (in this case the General Ledger).

"The Inventory Management" package maintains details of part numbers, descriptions, vendor and re-order data, item activity, costs, pricing and sales data. A 5.25-inch soft-sectored disk will handle approximately 3500 inventory items; an 8-inch disk approximately 5,500.

"Sales Invoicing" can be used with the Accounts Receivable and Inventory Management programs to access and update both receivables and inventory during the invoice printing procedure. Its many uses include verification of a customer's credit limits, or ensuring an adequate supply of a product before a sale is completed.

"Property Management" is a multi-property accounting system with three modules: file maintenance, transaction processing, and report generation. Charges, collections adjustments, and other changes are entered from a menu of options. Property management ledger entries can be passed along to the General Ledger, if desired.

"Multiple General Ledger" is a specialized program for CPAs and companies that maintain ledger accounts for several business. This program will allow an unlimited number of general ledgers, with a maximum of three per disk. This program, like the Property Management software, is an income-generation system and is priced higher than the other packages in the series

REQUIREMENTS: All of the Peachtree programs are provided with CP/M and Microsoft Basic (where necessary). Z/H-89s must be equipped with 64K RAM and at least two double-sided, soft-sector, 5.25-inch disk drives (H-37-2). A printer is virtually a necessity, and it must be able to print 132-column wide output, as most reports are generated in this format. The cost of the packages vary from $295.00 to $995.00.

DATABASE MANAGEMENT SYSTEMS (DBMS)

A database is a collection of data that is organized along predictable lines. This data can then be retrieved or sorted to isolate certain portions of the database that may have specific characteristics in common. A familiar example is a telephone directory: all of the customers form a general database. This database includes the names, addresses and telephone numbers of all telephone subscribers. In an ordinary telephone directory, this database is sorted alphabetically by subscriber name. But there are other specialized telephone directories that are sorted differently. Cross-reference directories are organized by consecutive house numbers on a street, with the streets themselves in alphabetical or numerical order. Newspaper reporters may use these to call neighbors at the scene of a fire, to get their eyewitness reports. Other directories have every telephone number in the city listed numerically; sales people rely on these to call everyone in a certain neighborhood, irrespective of their names or addresses.

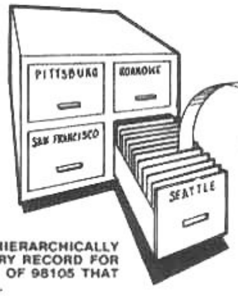
Very few businesses have a database as large as the telephone company, but nearly every business has some sort of database that it uses regularly. It may be the records of employees, a list of suppliers, dealers or customers, or something as simple as a list of files and documents. If this information can be selectively sorted to isolate all records that have a specific trait in common, the information becomes more useful to the business. For example, there is limited value in a list of employees arranged in the chronological order in which they were hired, unless the only required information is "who has seniority." It would be far more valuable to be able to extract from this list a ranking of all employees by salary, or a grouping by department, age, sex, or social security number.

A database management system is designed to allow the user to review a large data source and isolate segments that have common characteristics. The data must be structured in such a way that it can be retrieved in many different ways, but once the basic structure of the data is established, the database can be sorted or probed and *data can be selectively retrieved to supply information that relates one data item to another, perhaps revealing facts that would otherwise be unavailable.*

Every business can profit from valuable database information, and the potential applications for the professional person or small business are many and varied. A few of the most obvious uses are: specialized accounting (receivables, payables, ledger), customer information files, inventory and backorder files, employee records, salesperson contact and followup files. Many database systems also allow existing data files (no matter what the current format) to be reorganized into a specialized database that can be probed to gain unique information. This makes it possible, for example, to convert an existing computerized accounts-receivable file into a database

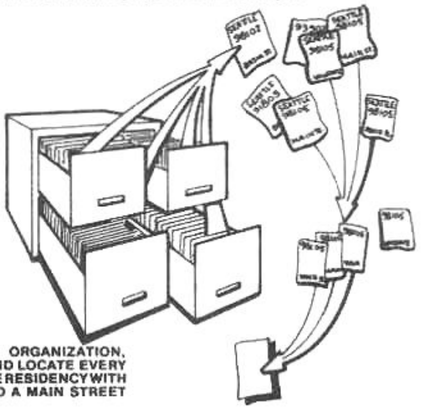
DATA BASE MANAGEMENT

HIERARCHICAL DATA ORGANIZATION



WITH THE DATA ALREADY HIERARCHICALLY ORGANIZED, LOCATE EVERY RECORD FOR SEATTLE WITH A ZIPCODE OF 98105 THAT RESIDES ON MAIN STREET.

RELATIONAL DATA ORGANIZATION



WITH NO INITIAL DATA ORGANIZATION, SEARCH ALL RECORDS AND LOCATE EVERY OCCURRENCE OF SEATTLE RESIDENCY WITH A ZIPCODE OF 98105 AND A MAIN STREET ADDRESS.

There are two common types of database management systems: An *organized* (hierarchical) database is composed of data that is entered and stored according to a predetermined structure. With this type of DBMS, data can only be retrieved in ways that conform generally to the manner in which the data was entered. In a *relational* database the data is not organized in a strict fashion, but the software allows data to be "selectively" withdrawn in a variety of ways. The following examples further illustrate the differences between organized and relational databases.

It is easiest to imagine the data in an organized (structured) database as being contained inside a complex series of file folders that are nested one within the other. The data is retrieved by "digging into the files." For example, to find everyone in Seattle, who has a 98105 zip code, and lives on Main Street, an organized database will go to the "file drawer" marked Seattle, locate the file marked 98105, and extract every index card with the heading Main Street. Note the hierarchical structure of the data storage, if the data is not put into the file according to this specific structure, it cannot easily be recovered with a conventional sorting procedure.

A relational database search is more complex because the desired information is not highly structured. Using the same example, imagine that the index cards are randomly stored in the file drawers. To perform a sort according to the same criteria, a relational database will first search the entire file (all drawers) for any Seattle residents. When it finds a Seattle resident, it will look at the rest of the data on the card, and if the zip code is 98105, it will look further to see if the resident lives on Main Street. If all the criterion are met, the record is pulled from the file.

Sorting a relational database may not seem as efficient—compared to the hierarchical sort—but a relational database is tremendously versatile. Data does not have to be entered according to pre-selected criteria—as it must in a hierarchical database—and the data can be sorted in nearly any manner. For example, using the same database as above, the relational system can easily find any records in the file that have the name Smith; however, a structured hierarchical database organized by city (as in the sample) cannot be readily sorted by name (to find the Smiths).

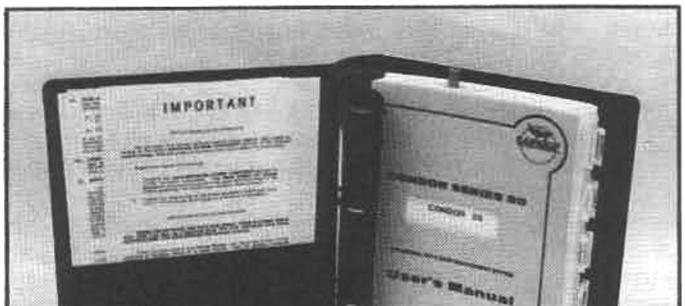
that can be sorted to find all receivables in a particular state or region that are 30 days (or later) past due. This information could then be supplied to a local salesperson or representative to help the company collect past-due payments. Without much effort, most business people can think of similar ways to make a database management system beneficial to them.

The programs discussed in the following section display database items as they "relate" to one another—as a telephone user's interest in the British economy may relate to the frequency of his calls made to England, or an employee's interest in retirement plans may relate to her age. These programs are called "relational" databases.

CONDOR SERIES 20 DBMS

If you associate the term "database" with programs that only sort names and addresses or print mailing labels, you will have to re-define the term before you can fully understand the Condor DBMS and other relational database systems. Through the use of more than 50 commands, the Condor DBMS user can manipulate data in almost any conceivable fashion. The power of Condor (or dBaseII, discussed next) doesn't come without a tradeoff, however: the user must be intimately familiar with each command and how it affects the file records and data output. This is not an impossible task, but for the first-time user this is a major undertaking.

As explained in previous chapters, an operating system (like HDOS or CP/M) allows the user to manipulate files, direct the computer to run various programs, and in general, control the overall operation of the computer. The Condor DBMS uses a "higher-order operating system," which is in many respects patterned after a conventional operating system. For example, when the Condor system is running, the computer issues the prompt **A>>** (similar to the CP/M prompt **A>**), identifying which disk drive is the default unit (the default drive is the disk that the computer searches to find programs and files, unless specifically instructed to look elsewhere). The default drive may be switched to drive B: by typing **B:** (just as with CP/M) and the DBMS will display **B>>**. This feature of Condor simplifies command syntax, for if the database file



The Condor database management system uses language-like commands—similar to high-level language statements—to create, sort and retrieve information from a database. The Condor "language" is very flexible, allowing wide-ranging access to the database, and a new step-by-step manual is under development to help the beginner learn to use this unique language.

is on drive B:, the drive label need not be specified each time the database is referenced (e.g., **DATABASE.DAT** is easier to enter than **B:DATABASE.DAT**).

Most of the commands in the Condor repertoire direct the system to create, access, modify, or extract information from a database file. A series of commands, arranged by the user in a logical sequence (in effect, a list of program commands), can direct the computer to perform extensive functions: for example, to search a customer data file, locate all customers residing in California, build a new database with this information, sort the new database by zip code, and finally print mailing labels for all customers in the database that have invoices exceeding $100.00. This hypothetical example illustrates the broad range of capabilities that are possible in a flexible and sophisticated database program.

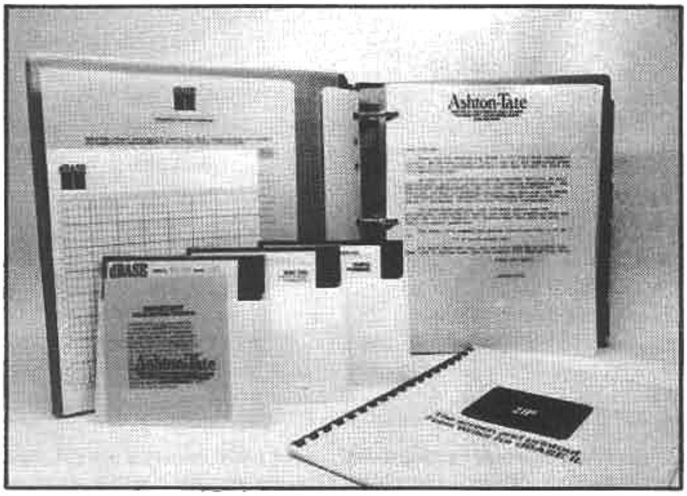
As the user becomes familiar with the command structure of the Condor system and develops useful command sequences, these commands can be stored in a file (command file) so that they can be rerun without re-keyboarding the entire series. The command **RUN filename** tells the DBMS to locate a command file (in this instance called "filename") and execute the instructions listed in this file. Command files may be created with most word processing programs (as long as they are of the literal-file type). In addition, the system operator can set up the Condor command files so they can be called from a screen menu display, which allows inexperienced office personnel to execute complex manipulations by simply selecting options on the menu.

Each record in a Condor database file may include five types of data: (1) *alphabetic*—a (or A) through z (or Z), space, single quote, period, and hyphen; (2) *alphanumeric*—all of the above plus the numbers 0 through 9 and special symbols, i.e. (*), (/), etc.; (3) *numeric*—up to 10 digits with values between +/-2148373647; (4) *dollar*—up to 8 digits plus two decimal places between the values of +/-21474836.47; (5) *calendar date*—a date represented as mm/dd/yy. The Condor system compresses numeric data when writing files to disk (this is the reason for the strange maximum and minimum numeric values). It will store a ten digit number in only 4 bytes, or a calender date in only 3 bytes. This feature can substantially reduce file size if a large portion of the database is comprised of numeric data.

In the Condor system each *data file* can contain up to 32,767 *records*. Each record may contain up to 127 *fields* (or categories), but the total *record size* (sum of all fields) must not exceed 1024 characters (bytes). Each record is entered in a *format* (a desired appearance on the computer screen) that is defined by the user at the time the database file is opened, and it may be changed at any time. The format dictates how the data in each record is presented to the user on the screen; it may represent an invoice, or a mailing label, or take any imaginable form.

The Condor Series 20 DBMS is a very powerful package, but it has one weak point: the documentation. The current manual is too brief, lacks details, omits important background information, and it does not illustrate sample responses to each command. This means that the new user will probably have to spend considerable time learning to use the system. A new manual is supposedly under development, and this may alleviate some of the shortcomings of the existing documentation.

REQUIREMENTS: Condor DBMS requires 64K RAM and two double-sided, double-density 5.25-inch disk drives or a single double-density 8-inch drive. A printer is required for data printout, mailing label generation, etc. The Condor Series 20 is supplied by Zenith and Heath on soft-sector 5.25-inch or 8-inch disks.



DBaseII is also based on a "high-order language" concept, however, the dBaseII language commands are more discrete. The command sequences are, therefore, longer and more complex, but the user has precise control over database manipulations. A comprehensive manual makes learning the powerful dBaseII language a relatively straightforward task.

DBASE-II

Dbase-II is very similar to the Condor package, because it also uses a unique form of computer "language" to perform tasks, but dBase-II operates in a much different manner. The Condor system uses commands that manipulate the entire database (usually); a number of commands produce "result" databases that contain the needed information. DBase-II on the other hand, uses a very concise, Basic-like language. Database records can be manipulated individually, and many features that are incorporated in a full-scale computer language are available in dBase-II.

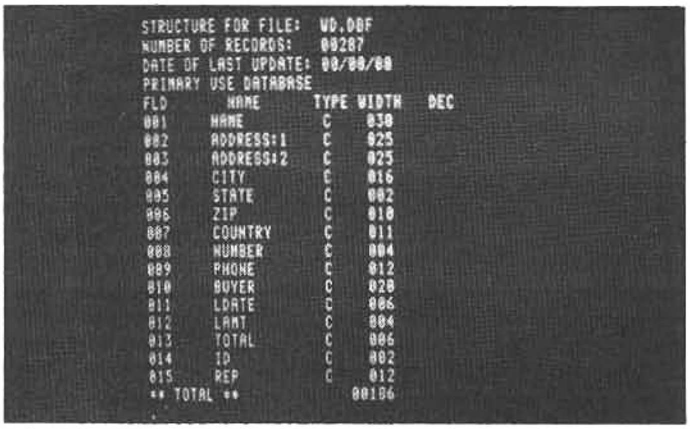
These language differences can be illustrated by comparing a typical data extraction from each of the systems. Assume that an identical database of names, addresses, and occupations is manipulated by each of the DBMSs. To print a columnar listing, with headings, the Condor system requires the command:

```
PRINT DATA BY NAME ADDRESS CITY STATE ZIP OCCUPATION
```

To accomplish the same listing, dBase-II must be given the following commands:

```
SET PRINT ON
?'  NAME   ADDRESS   CITY STATE  ZIP OCCUPATION'
?
DO WHILE .NOT. EOF
DISPLAY OFF NAME ADDRESS CITY STATE ZIP OCCUPATION
SKIP
ENDDO
SET PRINT OFF
```

This listing requires a more complex dBase-II command (because, among other things, the Condor PRINT command is pre-programmed to produce a columnar listing); however, if you would like to reproduce on pre-printed forms the name and address data (without the occupation field) from our example database, the Condor system



```
STRUCTURE FOR FILE:  WD.DBF
NUMBER OF RECORDS:   00287
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD      NAME        TYPE WIDTH   DEC
001     NAME          C    030
002     ADDRESS:1     C    025
003     ADDRESS:2     C    025
004     CITY          C    016
005     STATE         C    002
006     ZIP           C    010
007     COUNTRY       C    011
008     NUMBER        C    004
009     PHONE         C    012
010     BUYER         C    020
011     LDATE         C    006
012     LAMT          C    004
013     TOTAL         C    006
014     ID            C    002
015     REP           C    012
** TOTAL **              00186
```

A dBaseII database file is divided into groups of related information called *records*. The contiguate data entries (bytes) in each record must be contained within *fields* of designated lengths. However, the dBaseII user is relatively free to predetermine the number and size of each field when the database is initially formed. In addition, after the data file is formed, fields may be added, deleted or modified, and the record format may be readily emulated in a new data file to create an identical field structure.

would require generating a new screen format file, then a new "result" database. In addition, the pre-printed forms could not be larger than 80 columns wide and 24 lines long, since format files are limited to the computer screen size. DBase-II has none of these restrictions. Forms printing is only slightly more complex than columnar listing. For example:

```
SET FORMAT TO PRINT
DO WHILE .NOT. EOF
@ 3,25 GET NAME
@ 4,25 GET ADDRESS
@ 5,25 GET CITY
@ 5,45 GET STATE
@ 5,48 GET ZIP
SKIP
ENDDO
SET FORMAT TO SCREEN
```

This command will print information on lines 3 thru 5, and in columns 25 thru 48 on the printer. By just changing the line and column specifiers in the "@" commands, it is possible to print anywhere on the form.

Taken as a whole, dBase-II language is very powerful, precise, and versatile, but most applications require command programs with several steps. However, even this drawback can be circumvented in some instances by a special command feature called REPORT. This command will generate a columnar listing just about as easily as the Condor system, plus the user can select right and left margins, page length and page width—all with one command.

And dBase-II has the additional ability to use key- or sort-files for rapid data access. DBase-II will either search through the database records sequentially to find a selected item, or it can construct a special sort-file—based on data from the records (e.g., name, or city, or occupation, etc.)—that provides a "key-link" to the records in the main database file. Specific data is then located by first searching the sort-file (which is usually smaller than the main database) and then jumping directly to the record in the main database to retrieve the desired data. With large databases this technique can significantly decrease the time required to find specific information (using sort files, most records can be located in less than three seconds, even for database files several thousand records long). DBase-II will perform both types of searches—sequential or sort-file—depending on the specific requirements.

The extra effort required to learn the complexities of dBase-II is partially offset by the extensive documentation and numerous programming examples provided (both in the operator's manual and on a "sample program" disk). The manual is divided into three sections; a first-time-through approach, a detailed look at all the commands, and 80 pages of program examples. In addition, over 30 pages of additional programming examples (a checkbook balancer, an accounts payable program, and a mailing system with form letter capabilities) are provided on the sample program disk. The disk examples are complete with ready-to-run sample databases, command files, etc.

Each dBase-II data file can contain up to 65,535 records. Each record may contain up to 32 fields, but the total record size cannot exceed 1000 characters (bytes).

DBase-II is sold in the form of two system disks: one sealed in a plastic jacket, containing the full program, and the other packaged in a paper sleeve, containing a demonstration program with a maximum capacity of 15 records per data file. The customer may examine the limited system first, and if not satisfied, return the unopened full-functioning disk for a refund. According to the publisher of dBase-II, only three percent of its sales were returned in 1981.

REQUIREMENTS: Although dBase-II will run with 48K RAM, a full 64K is suggested. Two disk drives are required for all but the smallest database files; a printer is required for data printout, mailing labels, etc. The software is available from Ashton-Tate (see appendix) on hard- or soft-sector 5.25-inch or standard 8-inch disk formats.

HIGH-LEVEL PROGRAMMING LANGUAGES

This is not a book about computer programming. The subject is too broad to cover in the space available, and there are already many books that discuss various microcomputer programming languages in detail. The background information included here is presented only to help novice users grasp the general distinctions between the various high-level programming languages available for Zenith/Heath computers.

The individual instructions in a *high-level program* are not actually written in the cryptic syntax of machine code. Instead, the instructions are usually made up from conventional words or phrases that are given very specific definitions within the overall structure of the language (e.g., words like "READ," "WRITE," "GOTO" and "PRINT"). When a high-level program is executed by the user, each instruction is processed by a program called an *interpreter* or *compiler*. These programs transform the word-like instructions into machine-language instructions that the computer can understand.

Programs written in high-level languages that use an interpreter execute more slowly than programs written with directly-executable machine-language instructions, but it is much easier to write a high-level language program. After a few hours study a beginner can write simple high-level programs, and with continued practice, it is possible for the diligent student to write and debug complex programs. However, it takes many months (often years) of study to reach the same level of proficiency with a machine language.

The most popular high-level language is called Basic. (Within this general class there now exists numerous derivations that share the overall characteristics of the original Basic, but each of these offshoots have their own peculiarities.) The original Basic was conceived at Dartmouth College in 1960 as a simple programming method for people who needed to use the large and complex computers of those days, but didn't have time to learn intricate machine-language programming. It was an immediate success, and it has proven ideal for many applications.

In recent years many other high-level languages have been developed. The most notable examples are Fortran, Cobol and Pascal. These programming languages share some of the principles of Basic, but each has its own command definitions and structure. Some of the advanced languages even approach the power of operating systems, controlling the total computer environment.

In the following sections we will discuss the characteristics of the major high-level languages available for Zenith/Heath computers. The reader is encouraged to research this subject in much greater detail before committing any resources to a specific language.

BASIC-80

In 1976 two young software writers developed a workable microcomputer version of Basic, and they established Microsoft, Inc., a software company that would become one of the major forces in microcomputer programming. Their original high-level language is called Microsoft Basic, or just MBasic. It has grown into one of the most widely used interpreter languages in the industry, and Microsoft Basic-80 (a CP/M version of MBasic) is supplied with all new Z-89 and Z-90 computers.

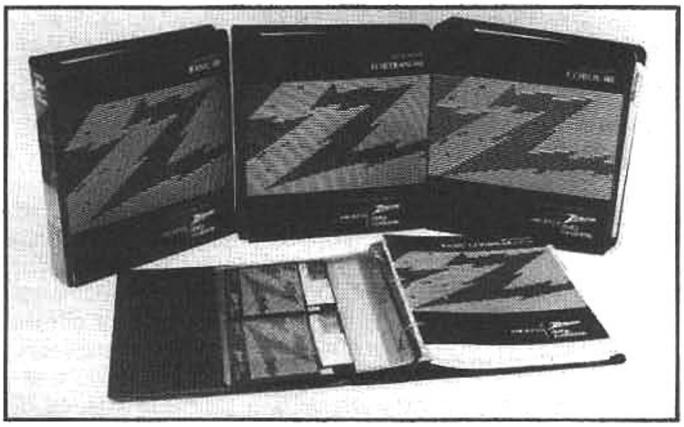
MBasic and Basic-80 are "interactive" languages—that is, they allow program writing, error detection, editing, and execution from within the program (this is an advantage because the user does not have to keep jumping back and forth between a program editor and the Basic interpreter while writing a new program). Basic-80 is a conventional interpreter language: during execution the program statements are interpreted one by one into machine-language equivalent instructions. The machine instructions are then executed by the Z80 microprocessor. This intermediate "interpretation" slows down the rate of execution.

Basic programming is relatively easy to learn. The program statements (instructions) are written in words and symbols that are virtually self-explanatory. For example, a program that will add up all the numbers from 1 to 100 looks like this:

```
10 R=0
20 FOR X=1 TO 100
30 R=R+X
40 NEXT X
50 PRINT R
60 END
```

Statement 10 sets the variable "R" (for Result) to zero. Statement 20 sets up a repeating loop with statement 40 (a FOR/NEXT loop) so that a 100 time repetition of the statements between the "FOR" and "NEXT" will occur. Statement 30 performs the math; it adds the value of "X" (which increments by one from 1 to 100) to the previous value of "R" and assigns this new value to "R." After the addition is complete, statement 50 will print the value of "R" on the terminal screen.

This is not a very useful program, but it demonstrates the logical statements used in MBasic. More than a hundred statements similar to these allow MBasic to control program and data flow, and when organized in the proper sequence, MBasic can function as a general ledger, handle accounts receivables, or put into practice whatever the mind of the programmer desires.



High-level computer "languages" consist of structured English-like words or phrases that the user can arrange into program *statements* to accomplish specific tasks. These statements are easier for the user to understand than the detailed, cryptic syntax of *assembly-language* or *machine-language instructions*, but before a computer can perform tasks defined in a high-level program, the individual statements in the program must be converted—by either an *interpreter* or *compiler* program—into a series of discrete *instructions* that the CPU can understand. Interpreter or compiler programs for most popular languages—Basic, Fortran, and Cobol—are available for Zenith/Heath computers.

Some of the more advanced features of MBasic are: "PEEK" and "POKE," for reading and writing information directly to specific memory locations; "IF THEN" a statement to modify program flow if certain conditions are met; "RND(X)" generates a random number; and an error-trapping routine called "ON ERROR GOTO"—and a great deal more. In addition, numerous mathematical functions are available, including algebra, trigonometry, and floating-point arithmetic.

MBasic has a built-in line editor that will insert, delete, or change any character anywhere within a program line without having to rewrite it. The editor is invoked by typing "EDIT" and the desired line number.

REQUIREMENTS: Basic-80 (MBasic), Version 5.2, requires CP/M and at least 48K of RAM. It is distributed on hard- or soft-sector 5.25-inch disks or standard 8-inch disk. An HDOS implementation (Version 4.82) is also available on hard-sector 5.25-inch disk. However, MBasic for HDOS does not support all of the commands available in the CP/M version. Available from Zenith or Heath

BASIC-80 COMPILER

The Basic-80 *compiler* produces a directly-executable machine-language program from a source program written in MBasic. Once the MBasic source is converted (compiled) to a machine-language program, it executes exactly like any other machine-language program (i.e., each statement does not have to be interpreted), and the overall speed of the program is faster than a Basic source program executed through an interpreter. The disadvantage in using a compiler, particularly to a novice programmer, is that the "interactive," error-displaying and "friendly" services of the interpreter are absent.

To gain the advantages of a fast-executing compiled program while still being able to debug and modify with an interpreter, programmers often develop programs with an interpreting Basic; then, when the program is functioning as desired, they use a compiler to generate a machine-language version of the program.

REQUIREMENTS: Versions of the Basic-80 compiler are available for both CP/M and HDOS. A minimum of 48K RAM and at least two disk drives are required. It is supplied on hard- or soft-sector 5.25-inch disk or standard 8-inch disk. Available from Zenith or Heath.

CBASIC-2 INTERPRETER/COMPILER

CBasic-2 is nearly as popular as MBasic. It was developed in the mid-1970's by a Navy submariner, and for technical reasons, it is very popular among professional software writers and publishers who use CP/M. It differs from MBasic in that it is not an interactive language. CBasic-2 allows a free-form statement construction, requiring line numbers only on GOTO, GOSUB, ON, or IF statements. In addition, CBasic-2 utilizes a slightly different approach in the use of variable names, internal program comments, and other functions. It is considered by some programmers to be a more sophisticated language than MBasic, and it is reportedly more accurate in mathematics involving lengthy decimal numbers.

CBasic-2 operates as both a compiler and an interpreter. The compiler portion converts an ASCII statement file (generated by an editor or word processing program) to a machine-language intermediate disk file; the second program, a "run-time" monitor, directly executes the intermediate instructions. A third program is supplied with CBasic-2 that generates a cross-referenced listing of all the variables used in the source code program.

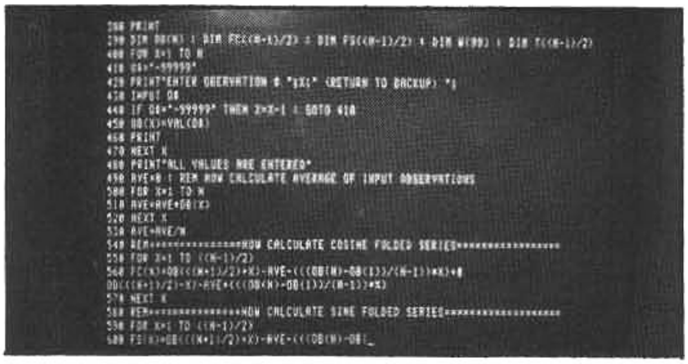
REQUIREMENTS: CBasic version 2.0 runs with CP/M and at least 48K of RAM. Program development can be done with one disk drive, however, two or more drives are suggested. It is supplied on hard- or soft-sector 5.25-inch disk or standard 8-inch disk. Available from Zenith or Heath.

PEARL-III

Pearl-III is a program generator that can be very helpful for serious programmers, especially those who develop software for custom applications. It creates applications programs in CBasic-2 by interacting with the programmer through menus. After the data type and length, etc., is accepted by Pearl-III, it can generate programs to produce custom reports, perform computations, sort or update files (as might be done in a general ledger when transactions files are used to update the master file).

Although Pearl-III can shorten a large programming project by literally hundreds of hours, it will not (usually) produce code that is "ready-to-run." The user should have a firm grasp of CBasic-2 programming because it is often necessary to hunt through lines of programming code to locate bugs or make modifications.

REQUIREMENTS: Pearl-III requires CP/M and CBasic-2. A full 64K of RAM and Z/H-37, Z/H-47 or Z/H-67 disk drives (at least two drives) are a necessity. It is distributed on soft-sector 5.25-inch or standard 8-inch disk. Available from Zenith or Heath.



```
380 PRINT
390 DIM OB(N) : DIM FC((N-1)/2) : DIM FS((N-1)/2) : DIM W(99) : DIM T((N-1)/2)
400 FOR X=1 TO N
410 OB$="-99999"
420 PRINT"ENTER OBERVATION # ";X;" (RETURN TO BACKUP) ";
430 INPUT OB$
440 IF OB$="-99999" THEN X=X-1 : GOTO 410
450 OB(X)=VAL(OB$)
460 PRINT
470 NEXT X
480 PRINT"ALL VALUES ARE ENTERED"
490 AVE=0 : REM NOW CALCULATE AVERAGE OF INPUT OBSERVATIONS
500 FOR X=1 TO N
510 AVE=AVE+OB(X)
520 NEXT X
530 AVE=AVE/N
540 REM****************NOW CALCULATE COSINE FOLDED SERIES*****************
550 FOR X=1 TO ((N-1)/2)
560 FC(X)=OB(((N+1)/2)+X)-AVE-(((OB(N)-OB(1))/(N-1))*X)+
OB(((N+1)/2)-X)-AVE+(((OB(N)-OB(1))/(N-1))*X)
570 NEXT X
580 REM****************NOW CALCULATE SINE FOLDED SERIES******************
590 FOR X=1 TO ((N-1)/2)
600 FS(X)=OB(((N+1)/2)+X)-AVE-(((OB(N)-OB(_
```

Basic is the most popular computer language because it was designed to be easily understood and readily learned. It breaks down elementary computer tasks into simple, logical program steps. This is a portion of a Basic program that analyzes stock market price data and selects investment possibilities.

FORTRAN-80

Fortran (an acronym for *For*mula-*tran*slator) is one of the oldest high-level languages, and was developed by scientists for use on large mainframe computers. It is still enormously popular for research and technology because of its ability to handle complex formulas. The version available for Zenith/Heath computers was developed by Microsoft and uses a compiler to generate machine-executable instructions.

A program file of instructions is written in ASCII by a word processor or text editor, and then compiled into a *relocatable module*. This module file is then translated by another compiler-type program, called a *linking loader*, into an executable machine-language program. This somewhat complex series of steps says a lot about Fortran-80: this is not a language for the first-time computer user to jump into. However, the familiarization procedure can be made easier by using the tape-cassette learning tutorials offered by Zenith/Heath. (There are instructional courses in this series that also cover MBasic, Cobol, CP/M, and assembly language, and they are all excellent).

Since Fortran programs are "fitted" to a particular computer system when compiled, the Zenith/Heath version of Fortran-80 is compatible with many Fortran application programs. This fitting process is performed by the compiler and other special programs that are included with the Fortran-80 package. The exact function of these special-purpose programs is beyond the scope of this book; additional information can be obtained from local Zenith or Heath dealers.

REQUIREMENTS: Fortran-80 version 3.4 needs a minimum of 48K RAM and two disk drives. This CP/M version is supplied on hard- or soft-sector 5.25-inch or standard 8-inch disk. An HDOS version (Fortran 3.35) is supplied on hard-sector 5.25-inch disks.

COBOL-80

Another Microsoft product, *Cobol-80*, is a microcomputer version of the widely used Cobol business language. Common business applications include general accounting and database management. Cobol was one of the first languages to handle disk files, and is generally considered "user-friendly" for clerical people who have had little experience with computers.

Cobol is unique in its division of program functions into four groups: identification, configuration, data, and procedures. This highly-structured language is less hardware-oriented and more "job-oriented." That is, each division addresses a particular part of accomplishing a task. Cobol is not as easy to learn as Basic, but is probably more widely used in large businesses than any other language.

REQUIREMENTS: Cobol-80 is available for both CP/M and HDOS. A minimum of 48K RAM is required and two disk drives are recommended. It is supplied on hard- or soft-sector 5.25-inch or standard 8-inch disk.

UCSD P-SYSTEM (PASCAL)

This language is generally known as *Pascal* (named for the 17th century French mathematician and mystic). It is popular with many programmers, and incorporates a kind of operating system, working with *pseudo-code* (called p-code).

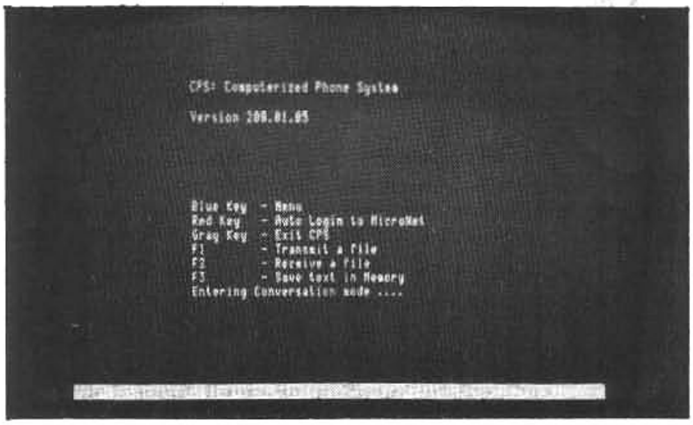
The developers of the p-System, Softech Microsystems, of San Diego, California, say that programs written with it will run on virtually any other microcomputer. Programs are translated automatically into a made-up language for an "imaginary" CPU chip, then a fast translating program converts that language into the one appropriate to the CPU which is actually being used. Thus, no matter what kind of machine a program was developed on, it can run on any other computer that is also set up to accept the p-System.

REQUIREMENTS: A Z/H-89 with 48K RAM; the p-System is distributed on a hard-sector 5.25-inch disk (only supports hard-sector media) or a standard 8-inch disk. Two disk drives are recommended.

COMMUNICATIONS SOFTWARE

A Zenith/Heath computer is capable of gathering information from over 1000 mainframe computers located all over the world. These remote computer systems offer their services to anyone who is willing to sign up and pay a charge, usually on a per-minute basis, for information that is transmitted to them over the telephone. Billions of bytes of information, everything from computer games to sophisticated applied-science data, can be placed at the user's fingertips through the use of a program that turns a conventional microcomputer into a powerful communications terminal.

A Zenith/Heath computer, operating under the control of communications software, has several advantages over a simple terminal (like the Z/H-19). The most significant advantage is the built-in disk drive, which can be used to permanently store incoming data. Furthermore, because the terminal section of the computer is software programmable, the user can select software that provides the most efficient data transfer from the remote computer. With communications software, a Zenith/Heath computer becomes a powerful communications system.



```
CPS: Computerized Phone System

Version 200.01.05



Blue Key  - Menu
Red Key   - Auto Login to MicroNet
Gray Key  - Exit CPS
F1        - Transmit a File
F2        - Receive a File
F3        - Save text in Memory
Entering Conversation mode ....
```

Microcomputers can readily communicate over vast distances with conventional telecommunications methods. Software to support computer communications is widely available. One of the most powerful packages is Computerized Phone System (CPS), which supports file transfers, output to printer or floppy disk, various baud rates, and automatic log-on to remote systems.

COMPUTERIZED PHONE SYSTEM

This program for Z/H-89 and -90 computers is usually referred to as CPS. It is best suited to two specific applications. First, CPS is designed to provide easy log-on to the Compuserve/MicroNet computer system; *MicroNet* is a portion of the *Compuserve* information network designed for microcomputer users. With CPS, the remote Z/H-89 can transfer messages and programs to and from Compuserve and other database systems. MicroNet can provide the user with free software (some of which is remarkably good) by *downloading* a program from the main system library to the disk system of the user's computer. A second, important application for CPS is communication with another Z/H-89 (also operating under CPS). *This interconnection of two CPS-driven systems provides full error-checking and insures 100% accuracy of all transmitted data.*

MICRONET COMMUNICATIONS AND FILE TRANSFER

CPS has an "auto log-on" feature for MicroNet. All the user has to do is dial the appropriate phone number, activate the modem, and push the red special-function key—the rest is automatic. CPS will send the unique number and password (issued to the user by CompuServe) in the correct sequence and display the sign-on message from MicroNet as it is received. If the ID number or password is changed, CPS can be easily reprogrammed to accommodate the changes.

As information is received from an outside system, data can be saved on disk. CPS will store in main memory (RAM) all data that is received after the f3 special-function key is pressed. When all of the desired data has been received, the user presses the f4 key, and CPS will request a filename and write the file from RAM to the disk. The Communication link is not broken during the disk-write opera-

Compuserve and The Source are two of the largest remote computer database systems available to microcomputer users. Compuserve has a personal-computing area called MicroNet, a shop-at-home service called Comp-U-Star, newspaper summaries, weather information, travel services, stock-market information, and thousands of "pages" of general information.

tion and the normal display of data will resume as soon as the write is completed. However, there is a limit to the amount of information that can be stored in RAM before it must be written to disk; for a 48K system this is about 30,000 characters, and for a 64K system about 45,000 (or about 117 and 175 disk sectors, respectively).

File transmissions to a remote system are also easy with CPS. Three methods of transfer are supported: transmission to MicroNet (with protocol defined by MicroNet); send file with no protocol; send file to another Z/H-89 running CPS with full handshaking (discussed in next section).

The first choice will send files to MicroNet without the need to manually enter **R FILTRN** (required by MicroNet) to initiate the transfer. After each line is successfully sent, a **(.)** will be displayed on the screen, a **(/)** will be displayed if a line must be re-transmitted. If a re-transmission is necessary, CPS will handle it automatically. The second choice sends a file with no protocol. This means that CPS will send information without verifying that the receiving system is getting all of the data. A non-protocol transmission method is used for communication with most remote databases and computers, since most systems, unfortunately, do not support any method of error checking and re-transmission if an error is detected.

CPS-TO-CPS COMMUNICATIONS

If a Zenith/Heath user needs to transmit data with complete accuracy, CPS is the answer. This software will send an ASCII file to a remote system, *which must also be running CPS*, with guaranteed integrity. If an error occurs during transmission, CPS will resend the data until the receiving system verifies to the originating system that the received data exactly matches the original file data.

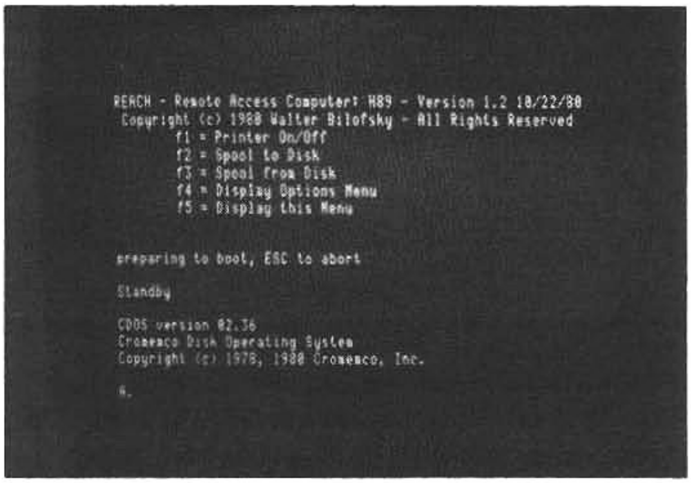
This is very important when the nature of the data (e.g., accounting records, critical text files, program listings, etc.) cannot tolerate errors that may be generated by random "noise" in the communications link. For example, the publishers of this book regularly use CPS-to-CPS communications to send large volumes of text from the editorial office to their typesetting facility located 60 miles across Los Angeles. These transmissions have totaled several million characters, and not one error has been traced to communications problems.

REQUIREMENTS: CPS is available from Zenith or Heath on hard-sector 5.25-inch disk for both HDOS and CP/M, and on soft-sector 5.25-inch disk for CP/M. The current price is under $50.00.

REACH—REMOTE TERMINAL EMULATOR

Communication with a remote computer is becoming a common business tool. Most major companies utilize large computers for accounting, word processing, or database management routines, and most of these systems use the telephone network to communicate with outlying offices. As a result, authorized employees can use a Z/H-89 and a program called Reach to emulate a terminal for communicating with the company computer. Once the telephone link has been established (as prescribed by company operating procedures), Reach can transmit, receive and manipulate data within the remote system, and it can direct incoming data to a printer that has been connected to the Z/H-89.

Reach is a versatile terminal emulator. It can be modified (patched) to transmit and receive data with almost any type of telephone-linked computer system. Transmission speeds up to 9600 baud, various protocol techniques, and numerous printer configurations are supported. Reach is a powerful tool, but it requires some knowledge of computer data transmission techniques. The documentation is not written for a beginner.



```
REACH - Remote Access Computer: H89 - Version 1.2 10/22/80
 Copyright (c) 1980 Walter Bilofsky - All Rights Reserved
        f1 = Printer On/Off
        f2 = Spool to Disk
        f3 = Spool from Disk
        f4 = Display Options Menu
        f5 = Display this Menu


preparing to boot, ESC to abort

Standby

CDOS version 02.36
Cromemco Disk Operating System
Copyright (c) 1978, 1980 Cromemco, Inc.

A.
```

Another popular communications program for Zenith/Heath computers is called Reach. Here a Z/H-89 running Reach is being used as an "intelligent" terminal to transmit data to a Cromemco business computer.

REQUIREMENTS: Reach is available for HDOS or CP/M on a 5.25-inch hard-sector disk from The Software Toolworks (see appendix) for $19.95.

PROGRAMMING TOOLS

Most of the following utility programs are available from Zenith and Heath. These special "tools" make programming easier or the operation of a program more convenient.

SID/SZID is an advanced version of the CP/M utility called DDT (Dynamic Debugging Tool) that has all the features of DDT and more. SID/SZID allows the use of names at the source code level, instead of addresses, in dumping and modifying memory, setting breakpoints, and making patches to a program in assembly language. It can also produce histograms showing the frequency of a group of subroutines, and it must be used with a linkage editor—the MAC Macro Assembler from Digital Research or Microsoft's Link-80. Available from Zenith or Heath on hard- or soft-sectored 5.25-inch disks or standard 8-inch disk, and requires CP/M with at least 48K of RAM.

Super Zap is an extremely powerful "disk dump" utility that can examine and modify the contents of a disk file. It is used primarily to modify (patch) machine-language program files, however, text files can also be examined. Each sector of the program file is displayed on the screen (in either hex, octal or ASCII format), and the user can modify the program bytes with an editing function that operates very much like an interactive word processor. The cursor can be moved throughout each sector using the cursor keys, and the bytes are changed by typing a new value over the existing byte. The keyboard function keys allow stepping sector-by-sector forward or backward through the file. Super Zap can also remove flags from files, print the contents of a file or a sector to a listing device (printer), list a disk catalog, search a file or entire disk for a hex/octal/ASCII string, or save a sector to a file. Available for both HDOS and CP/M from The Software Toolworks (see appendix) for $24.95.

MAC Macro Assembler (produced by Digital Research) converts assembly-language programs (under CP/M) to machine-language files. It offers conditional assembly operators, title and page pseudo-operands, and Intel-compatible macro definitions. Supplied on the distribution disk, also, are macro libraries useful for performing disk I/O and other tasks. Available from Zenith or Heath on hard- or soft-sector 5.25-inch disks or standard 8-inch disk; requires CP/M and at least 48K of RAM.

Macro-80 is an assembler that claims to handle over 1000 lines of code per minute. Nesting of macros is limited only by the size of memory. Link-80, a program within it, will function with relocatable libraries or modules produced by Macro-80 or other languages and compilers. Available from Zenith or Heath on hard- or soft-sector 5.25-inch disks or standard 8-inch disk; requires CP/M and 14K of RAM.

Despool is a multi-tasking printer driver that lists files on a line printer while the user is free to use the computer for other operations. By typing CTRL-F, when Despool is loaded in memory, a prompt appears; entering the appropriate file name sends that file to the printer while the computer is "returned" to the user. Available from Zenith or Heath on hard- or soft-sector 5.25-inch disks or standard 8-inch disk; requires CP/M and 48K of RAM.

SPECIAL-PURPOSE EDITORS

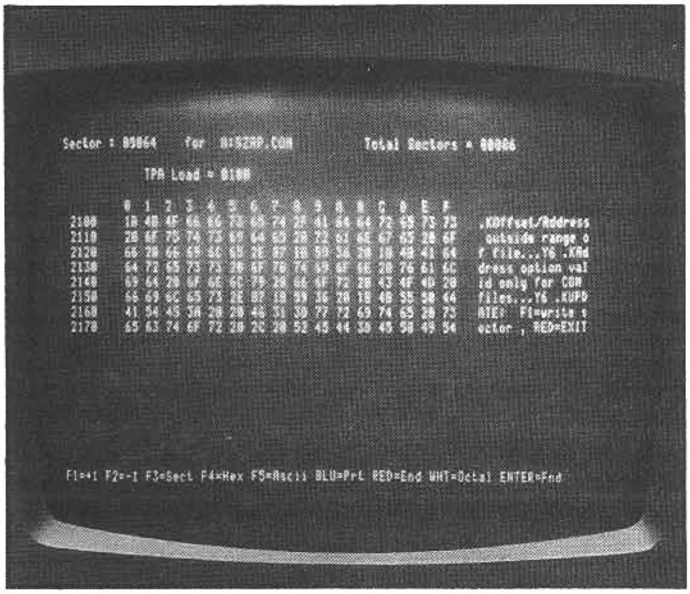
ED-A-SKETCH

Ed-A-Sketch is an editor that allows the creation and modification of graphic displays on the Z/H-89 or -90 computer (or a computer using the Z/H-19 terminal). The cursor-control keys allow rapid movement within any screen display, and the function keys permit displaying regular or reverse-video graphics characters, drawing vertical or horizontal lines, recovering from mistakes with an "oops" key, and saving a picture to disk. Saving or moving portions of an image (blocks) is also permitted. The most powerful feature of Ed-A-Sketch is the capability to save a graphic image in any of seven formats, including: picture format (picture can be copied (on screen) for display or re-edited; Benton Harbor format (picture is saved as a program listing in Heath's proprietary version of Basic, which they call Benton Harbor Basic); Microsoft Basic format (picture is saved as a program listing in MBASIC); Assembly Language format (picture is saved as an assembly language program); relative cursor positioning format (this selection modifies the above program listings to introduce positioning control of a portion of an image for placement anywhere on the screen, allowing the generation of moving images, etc). Ed-A-Sketch is available for both HDOS and CP/M on hard-sector 5.25-inch disk or standard 8-inch disk (for CP/M only) from The Software Toolworks (see appendix) for $29.95.

EDIT-19

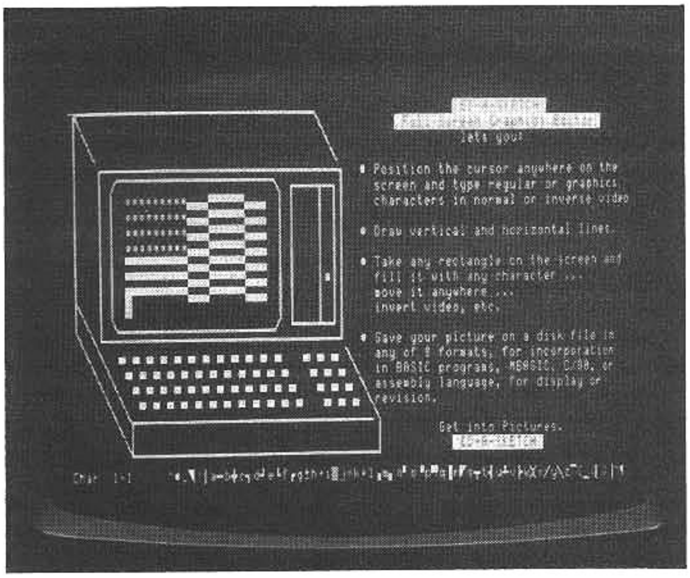
Edit-19 is much more than just an editor; it should really be called a "universal editor for HDOS." It is probably the most powerful general-purpose editor currently available for HDOS.

Edit-19 is a programmable program—similar to dBase-II and the Condor DBMS (discussed earlier), Edit-19 has its own language. The language evolved from the extensive instruction set (over 100 commands) contained in the program. These instructions can be written in a command file that is executed by Edit-19; in effect, this command file becomes a source program for the Edit-19 interpreter. The most common use for command files is to configure the editor to suit specific user requirements. Virtually every feature can be modified: the "functions" of the special-function keys, the margin settings, tab stop positions, etc. And a plethora of technical "states" can be configured: the number of drives to be included in a search for files, the condition of the terminal upon leaving Edit-19, whether to start the editing session in the screen or command mode,



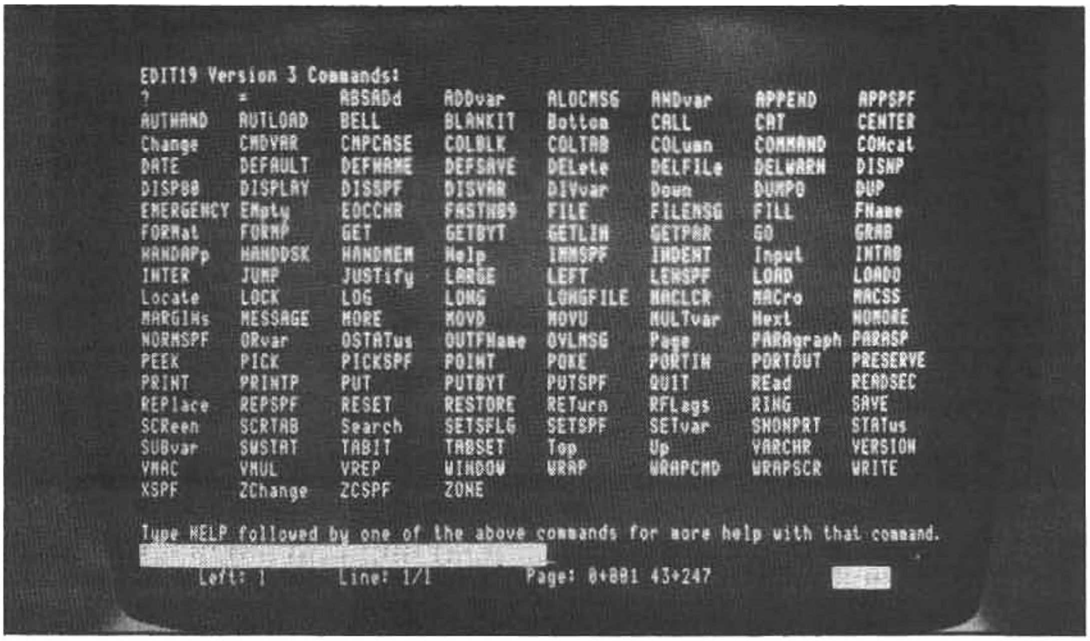
```
Sector : 00064    for  A:SZAP.COM          Total Sectors = 00006

          TPA Load = 0100

        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
2100    1B 4B 4F 66 66 73 65 74 2F 41 64 64 72 65 73 73   .KOffset/Address
2110    20 6F 75 74 73 69 64 65 20 72 61 6E 67 65 20 6F    outside range o
2120    66 20 66 69 6C 65 2E 87 1B 59 36 20 1B 4B 41 64   f file...Y6 .KAd
2130    64 72 65 73 73 20 6F 70 74 69 6F 6E 20 76 61 6C   dress option val
2140    69 64 20 6F 6E 6C 79 20 66 6F 72 20 43 4F 4D 20   id only for COM
2150    66 69 6C 65 73 2E 87 1B 59 36 20 1B 4B 55 50 44   files...Y6 .KUPD
2160    41 54 45 3A 20 20 46 31 3D 77 72 69 74 65 20 73   ATE:  F1=write s
2170    65 63 74 6F 72 20 2C 20 52 45 44 3D 45 58 49 54   ector , RED=EXIT

F1=+1 F2=-1 F3=Sect F4=Hex F5=Ascii BLU=Prt RED=End WHT=Octal ENTER=Fnd
```

Experienced programmers and advanced enthusiasts can sometimes modify programs to suit their own needs or desires by "patching" (altering) bytes within the program instructions. Super Zap, a utility program from the Software Toolworks, gives the sophisticated user access to the bytes stored on each disk sector and allows simple modification of individual bytes; it is available for both CP/M and HDOS.



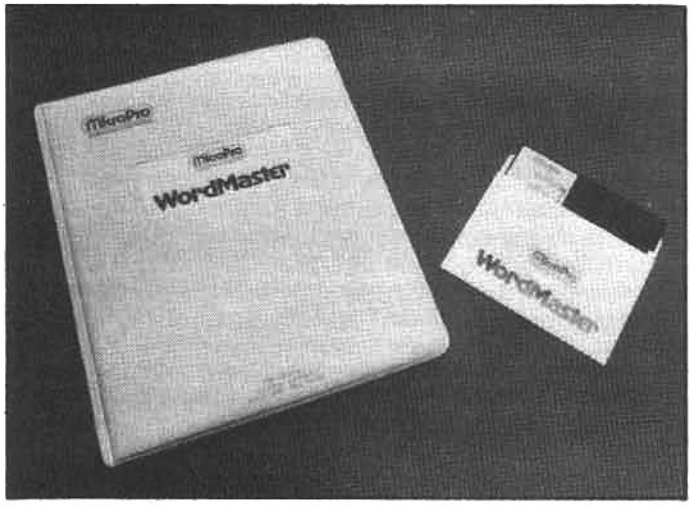
```
                ED-A-SKETCH
        Full-Screen Graphics Editor
                 lets you:

• Position the cursor anywhere on the
  screen and type regular or graphics
  characters in normal or inverse video

• Draw vertical and horizontal lines.

• Take any rectangle on the screen and
  fill it with any character ...
  move it anywhere ...
  invert video, etc.

• Save your picture on a disk file in
  any of 8 formats, for incorporation
  in BASIC programs, MBASIC, C/80, or
  assembly language, for display or
  revision.

           Get into Pictures.
              ED-A-SKETCH

Char  1-1
```

Ed-A-Sketch produces graphic images on the Z/H-89 screen in much the same way that a word processing programs creates and alters text. Simple commands allow the user to draw a picture on the screen with graphics and text characters, and the user can "save" the picture to disk as either a Basic-language listing or a displayable file (one that can be "typed" to the screen).



```
EDIT19 Version 3 Commands:
?         =         ABSADd    ADDvar    ALOCMSG   ANDvar    APPEND    APPSPF
AUTHAND   AUTLOAD   BELL      BLANKIT   Bottom    CALL      CAT       CENTER
Change    CMDVAR    CMPCASE   COLBLK    COLTAB    COLumn    COMMAND   CONcat
DATE      DEFAULT   DEFNAME   DEFSAVE   DELete    DELFILe   DELWARN   DISNP
DISP80    DISPLAY   DISSPF    DISVAR    DIVvar    Down      DUMP0     DUP
EMERGENCY EMpty     EOCCHR    FASTAB9   FILE      FILEMSG   FILL      FName
FORMat    FORMP     GET       GETBYT    GETLIN    GETPAR    GO        GRAB
HANDAPp   HANDDSK   HANDMEM   Help      IMMSPF    INDENT    Input     INTAB
INTER     JUMP      JUSTify   LARGE     LEFT      LENSPF    LOAD      LOAD0
Locate    LOCK      LOG       LONG      LONGFILE  MACLCR    MACro     MACSS
MARGINs   MESSAGE   MORE      MOVD      MOVU      MULTvar   Next      NOMORE
NORMSPF   ORvar     OSTATus   OUTFName  OVLMSG    Page      PARAgraph PARASP
PEEK      PICK      PICKSPF   POINT     POKE      PORTIN    PORTOUT   PRESERVE
PRINT     PRINTP    PUT       PUTBYT    PUTSPF    QUIT      REad      READSEC
REPlace   REPSPF    RESET     RESTORE   RETurn    RFLags    RING      SAVE
SCReen    SCRTAB    Search    SETSFLG   SETSPF    SETvar    SHOWPRT   STATus
SUBvar    SWSTAT    TABIT     TABSET    Top       Up        VARCHR    VERSION
VMAC      VMUL      VREP      WINDOW    WRAP      WRAPCMD   WRAPSCR   WRITE
XSPF      ZChange   ZCSPF     ZONE

Type HELP followed by one of the above commands for more help with that command.

        Left: 1        Line: 1/1            Page: 0+001 43+247
```

Edit-19 is probably the most comprehensive text editor available for HDOS. It has more than 100 text commands, and multiple commands can be incorporated into a command file, permitting complex tasks to be handled "automatically" by chained commands. Edit-19 is accompanied by well written operation and tutorial manuals.



WordMaster is a substantial enhancement over the cumbersome line editor provided with CP/M. It includes all of the conventional commands found in the CP/M editor, however, it has additional features and a convenient full-screen editing mode. WordMaster is a favorite among programmers because of its fast search-and-replace ability (especially helpful during the development of extensive programs).

and more.

Edit-19 is accompanied by an 85-page operating manual, that—although terse—provides a useful introduction and excellent details about each command. Also available is a tutorial manual and disk that provide a clear, in-depth look at the power of Edit-19. The tutorial is accompanied by several example files, and the user can run macros (command files) to learn their operation. One of the macros will prepare two columns of text from a normal text file. Another macro will construct boxes around comment paragraphs used in programming Basic, Fortran, etc.

This is, obviously, a complex editor. However, the careful selection of default values and the ease with which the user can attain help—several help screens are available and there is a help file that will display additional information about any command—all contribute to its user-friendly operation.

It is an excellent general-purpose editor for someone who does a variety of work under HDOS (i.e., creative writing and program editing). But, if conventional word processing is the only requirement, a standard word processor will probably be easier to use. However, Edit-19 *can* be used to augment a word processor. It can rapidly reformat an entire text file with one command, it has a lightning fast search-and-replace, and it contains many powerful features that may be useful to a professional writer or editor.

In summation, it is safe to say that many serious HDOS-users will consider Edit-19 a "must-have" tool. It was developed by Steven Robbins, a data-processing instructor in San Antonio, Texas, and it can be purchased from Mr. Robbins (see appendix) for $80.00. The tutorial disk and manual are an additional $20.00. There are discounts available for HUG groups (quantity purchases). Edit-19 is distributed on hard-sector 5.25-inch disks. Two disks drives (with high-density storage) and 56 Kbytes of RAM are recommended.

WORDMASTER

In the early days of microcomputing the only word-editing programs were of the "line-oriented" type. These editors are designed primarily to create and debug the text (instructions) of a computer program, and they do not operate on the interactive cursor-oriented principles common in advanced word processing software. They display the lines of a program or text file (usually one at a time) for verification and/or modification, and to change the text, the user must type a complex command to direct the editor to remove a certain portion of the line and replace it with new text. For word processing (and for many operations in program editing) this method is slow because the user must learn a complex, structured command syntax.

However, line editors have some powerful features. They typically have very fast search-and-replace functions because they are not required to display and scroll large blocks of text on the screen. And most line editors can make "conditional decisions" during global editing tasks. They are also small in size and, therefore, permit large files to be loaded into RAM and edited without splitting the file into smaller blocks.

When MicroPro designed WordMaster they took the

the powerful features of a line-oriented editor and effectively combined them with the interactive features of a cursor-oriented (screen) editor. This is an excellent editor for programmers working under CP/M. The ease-of-operation and speed of WordMaster permit rapid global searches, while still allowing easy single-character replacement. The version for Zenith/Heath computers also implements the following commands on the special-function keys:

f_1—duplicate of delete key
f_2—jump to beginning of file
f_3—jump to end of file
f_4—copy file to disk

And all of the special-function keys can be redesignated to perform any function that the user desires.

Another valuable feature of WordMaster is the ability to edit files larger than memory by using chained floppy-disk I/O to dynamically move forward or backward in the file. As the user moves through a large file, the section being edited is read into RAM, while the overflow is removed from RAM and written to a temporary file on disk. In addition, WordMaster will allow the user to read in other files (merge) and write out portions of the current file to disk (split up files).

WordMaster is a CP/M editor, written for 8080/8085 systems with 32 Kbytes of RAM or more. It is available in almost any format compatible with Zenith/Heath systems. For price and availability contact MicroPro International (see appendix).

GAMES AND OTHER ACTIVITIES

The total number of special-interest and recreational programs available for Zenith/Heath computers is unknown. Whatever the exact amount, it is large. The pool of CP/M software is practically bottomless, and more becomes available every day. Game programs tend to be inexpensive (and many can be obtained free-of-charge from public-domain software pools); so even if they turn out to be less than spectacular, they are usually worth a few hours of enjoyment. The few examples mentioned here are available from Zenith or Heath, and each should provide enjoyment for avid game-players.

Word Search Puzzle is a "locate-the-hidden-word" game. A grid is displayed on the screen hiding 12 to 32 words in 30 different subjects, including basketball, TV stars, mystery stories, etc. The Z/H-89 graphic capabilities are utilized.

Cassino is very similar to the dice game of the same name. The dice are displayed on the screen using computer graphics, and the user may select the dice to be "thrown" by pressing the special-function keys.

Munchkin is a game that resembles the popular arcade game. A moving, circle-shaped Munchkin works its way through a maze devouring the delectable dots and adding to the score. There are aliens that pursue our munching hero, but they, too, can be gobbled up if the Munchkin consumes an energy pellet just before the alien is encountered.

The 15 Puzzle looks a little like Rubik's Cube, but it is based on a puzzle made up of numbered tiles. A group of sixteen squares—fifteen contain numbers and one is blank—are moved, within constraints, to a preselected consecutive-number sequence.

Invasion is another arcade game look-alike. The aliens are descending from space, ever closer and ever faster. Take careful aim, because they shoot back!

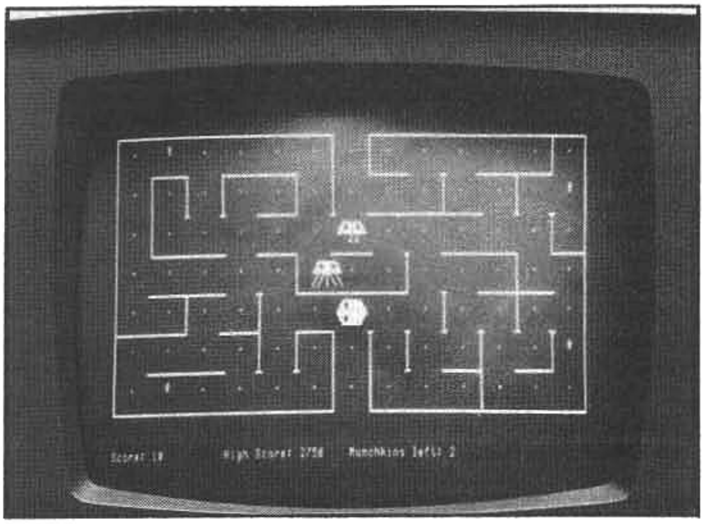
Air Traffic Controller "tests your control skills by putting you in the tower." It requires 32K RAM, one disk drive, and either HDOS or CP/M. "Run your own airport!"

Mychess is a chess-playing program that won a playoff at the 1980 West Coast Computer Faire. While not the most sophisticated chess player, Mychess will certainly challenge casual players. Several levels of difficulty are selectable, and range from beginner to "how to be intimidated by a computer." Mychess is available from Zenith or Heath, and also from The Software Toolworks (see appendix) for $34.95.

Computer Chef may be the program that saves some marriages from destruction by rampant microcomputer mania. This clever program provides a method of storing, retrieving, and scaling recipes for main dishes, desserts, soups, etc. Computer Chef comes with over 70 "family-tested" recipes, that include preparation and serving suggestions. Each of these recipes—and others that you may wish to add—can be recalled by title, ingredient, or keyword (a list of descriptive terms—i.e. sinfully rich, quick, outdoor, lunch, kids, freezes well, etc.). Once a recipe is selected, the ingredients may be scaled to any desired amount. For example: you may wish to make brownies but have only one egg, Computer Chef will tell you how much of the other ingredients to use. Or, if a recipe makes two dozen cookies and you need five dozen for your next party, Computer Chef will rescale all the ingredients for you. Recipes (scaled or not) can be printed out for easy use in the kitchen; the printed copy can then be discarded after it is covered with fudge, etc. Computer Chef can be purchased from The Software Toolworks (see appendix) for $29.95. A disk with an additional 100 recipes is available for $9.95.

RTTY Communications Processor is for ham radio operators who want to transmit computer-generated CW (Morse code), ASCII or Baudot code over the radio. The text of a message is typed on the keyboard of a Z/H-89 or -90, and the message appears on the screen as it is transmitted or received. It runs under HDOS (similar programs for CP/M are available from some sources).

Roots-89 is used to help organize family genealogical charts and records. It stores and retrieves facts about family members and computes your relationship to your ancestors. It requires 48K RAM, one disk drive, and CP/M.



Computer games are an intriguing diversion from "serious" work. This Zenith entertainment package includes five games, and may result in hours of "intense emotional interaction" with your computer.

CHAPTER 7 EXPANSION & UPGRADES

Zenith and Heath computer systems can be readily expanded with disk drive systems, printers, modems, extra terminals and a wide variety of specialty accessories. The Z/H-89 or the Z/H-100 can be configured to answer virtually any personal or business requirements.

AN OVERVIEW

One of the great virtues of Zenith/Heath computer systems is the ease with which they can be expanded. The range of expansion and accessory equipment available from Zenith/Heath and independent vendors is impressive. By upgrading or adding to the system hardware, the user can readily tailor the system to suit specific needs; but at the same time, nothing must be done to compromise dependability. In this chapter we will explore several popular system upgrades. They are not, by any means, *all* of the possible upgrades, but they are common examples of the most useful, cost-effective and reliable ones.

The most common types of system upgrades are: RAM expansion, mass storage (disk drive) expansion, and the addition of accessories to improve system capabilities or convenience. The value of these enhancements becomes more apparent as the user gains more experience and devises new ways to make a microcomputer useful.

For example, when the user commands the operating system to "run" a piece of application software, an exact duplicate of the program is transferred from the floppy disk (or other mass-storage device) to the system RAM, and the CPU begins executing the program instructions. As a consequence, *the amount of system RAM will determine, to some degree, which programs can be used.* If a program is too large, it may not fit in the available system RAM.

The size of system RAM is usually expressed (approximately) in the number of *Kbytes* (1024 bytes) of available space. The Z/H-89 is factory-equipped with 48 Kbytes of RAM; and the addition of 16 Kbytes of memory is a very common upgrade. This provides 64 Kbytes of memory—the maximum possible with conventional Z80 memory addressing—which permits the computer to handle larger programs and text files. (And it increases the speed of some computer operations.) In the following section on Memory Expansion we will look at the detailed procedures involved in the expansion of system RAM.

Many of the most popular upgrades involve greater mass storage and increased processing speed. Programs that repeatedly read and write disk data—database management is a good example—can operate notably faster with storage devices that handle data more efficiently. High-density 5.25- and 8-inch floppy-disk drives can reduce "search" time (particularly under CP/M), while providing more storage space for data files and programs. Hard disk drives give even faster access to data, and are able to store considerably more information. The absolute leader in speed, however, is really not a disk, but is a RAM expansion configured to act like a disk drive. This product, called a *disk emulator*, and other storage upgrades will be explored in the section on Disk Drive Expansion.

In the section on System Enhancements and Convenience Features we will look at some modifications to hardware and software that reduce operator fatigue and/or expand overall system capabilities. Independent programmers and hardware designers have developed "patches," plug-in boards, software tools, and other "convenience" items for Zenith/Heath computers. Many of these accessories reduce confusion and frustration, and give the user many tangible benefits.

MEMORY EXPANSION

When the H-89 was first introduced, it was sold with 16 Kbytes of RAM and enough empty sockets on the CPU board to accept an additional 32 Kbytes, for a total of 48 Kbytes of RAM. For at least the first year of production this was the memory limit; however, at that time, there were only a few elementary programs available, and the memory restriction was not a severe constraint. Today there are many sophisticated programs that need or can use a full 64 Kbytes of memory.

Users of Basic-language programs often feel the constraints of memory space. Like most Basic languages, Heath's Benton Harbor Basic or Microsoft Basic must have the language interpreter loaded in memory before a source program (that actually does the work) can be executed. These interpreters are large programs, and they do not leave much RAM space free for the source program. In addition, the Basic interpreter must set aside some of the remaining RAM space for variables, matrices, alphanumeric terms, etc. Though many Basic programs can be run with less memory, a full 64K allows the programmer to include more computational capability within each source program.

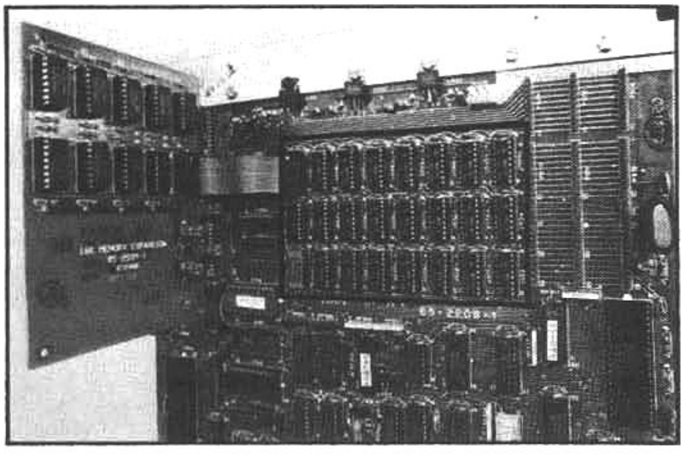
Systems that are used mainly for word processing can also benefit from larger RAM space, because larger files need not be broken up into small segments for editing or formatting. In fact, after the operating system is loaded, some of the more powerful word processors consume all but the last 16-Kbyte bank of memory. Without the full complement of memory some programs can only edit very small text files.

These are but two examples of applications that profit from a full 64K memory. There are many others, and as the user finds more sophisticated ways to utilize the Z/H-89, the advantage of more memory space becomes increasingly obvious. Fortunately, the installation of the 16-Kbyte memory expansion is relatively easy.

HOW TO INSTALL ADDITIONAL MEMORY

Early Heath computers were not designed to accept more than 48 Kbytes of memory; however, all late-model units will accept the Zenith/Heath 16-Kbyte RAM expansion with almost no modification. Early units are also expandable, but an upgrade kit called the H-88-7 is required (this is the same modification kit that allows the use of CP/M in addition to HDOS). Units that were sold with the cassette interface card will also require the H-88-7 modification before the computer will recognize the last 16-Kbyte bank of RAM. Once the H-88-7 modification is installed, the Z/H-89 will no longer support the cassette tape interface.

The installation of the 16-Kbyte upgrade requires the removal of the CPU circuit board (see accompanying illustrations), and an inspection should be made of the IC (integrated circuit) at location U562. This IC should be a 74S132 (there may be additional letters before and after this number). *If the IC is a 74LS132 (note the "L" before the "S"), the system will "crash" (malfunction) during the boot operation or while executing a program.* Most 16-Kbyte upgrade kits are supplied with a 74S132 (in case it is required), but there is no need to install it if the correct chip is already in place.



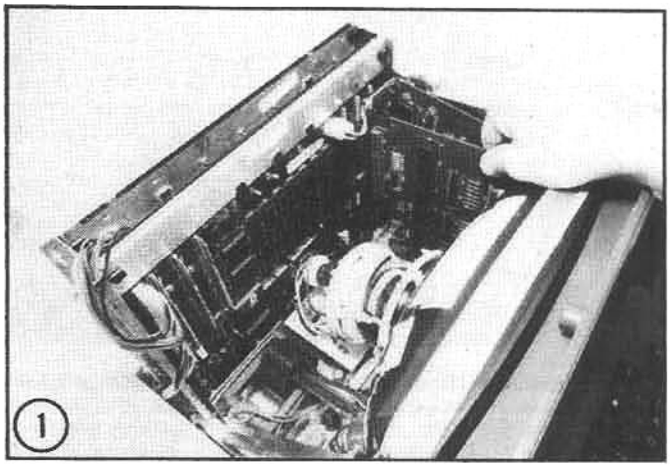
The Z/H-89 CPU board will accept up to 48 Kbytes of RAM (the 24 chips within the boxed area), and an additional 16 Kbytes can be added with a separate plug-in circuit board. This brings the total RAM memory to 64 Kbytes, the maximum RAM supported by conventional Z80 memory-addressing techniques.

The next installation step is the removal of the upper-left memory chip (U549) to make room for an adaptor plug that must be installed in the same socket. The 16-Kbyte upgrade will only work properly if the full 48 Kbytes of memory chips are already installed (i.e., all other memory sockets on the CPU board are filled); the 16-Kbyte expansion is designed strictly to *add memory to the existing 48 Kbytes on the CPU board.* Memory chips can be purchased from Zenith/Heath or from many independent suppliers (see advertisements in Byte and other microcomputer magazines). The Z/H-89 uses a standard memory chip called a 4116. Make sure you use chips that are rated at 400ns (nanoseconds) or faster (i.e., 4116-4).

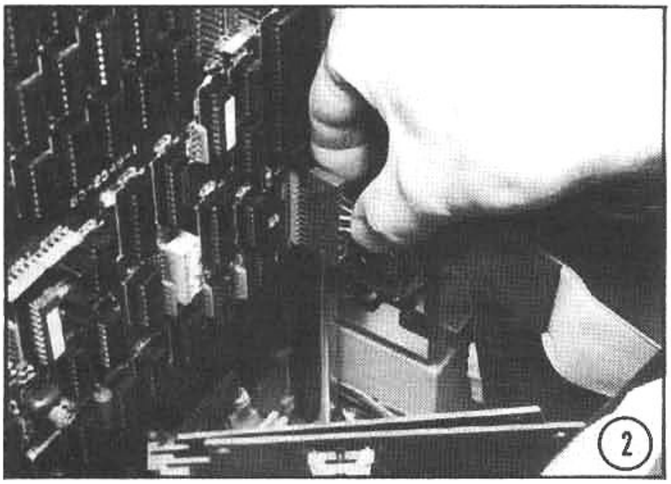
The expansion board should now be installed on the pins located on the left side of the CPU board. Make sure that the board is properly positioned on the pins; not offset too high or too low by one or more pins. The plug at the end of a short ribbon wire coming from the expansion board is plugged into the single empty socket at U549. Some earlier models also required a short jumper wire (see accompanying illustrations), however, all new computers have incorporated this jumper connection into the design of the CPU board.

The final requirement is to reconfigure the jumpers (small plastic clips) at JJ501 and JJ502 as shown in the illustrations. During the boot process these jumpers tell the CPU that a full 64K of memory is available. After the CPU board is reinstalled, the system may be powered up, and if the computer executes the built-in memory diagnostic without faults, the system may be put into service. During boot with HDOS, the computer will display **56K of RAM available** (not 64 Kbytes, because the bottom eight Kbytes is reserved for system use and only 56 Kbytes are available to the user). With CP/M you must run MOVCPM.COM and SYSGEN to generate a 64-Kbyte CP/M system that will use all of the memory now available (refer to the CP/M manual for more details).

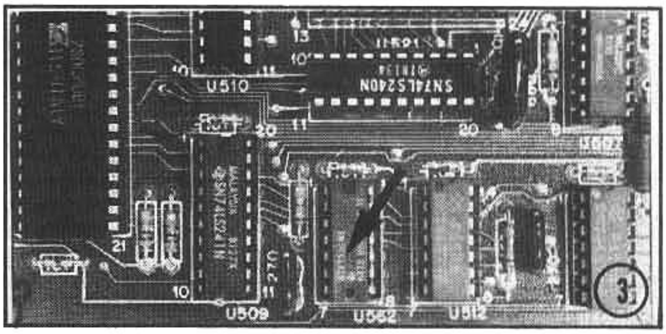
INSTALLING 16K RAM EXPANSION



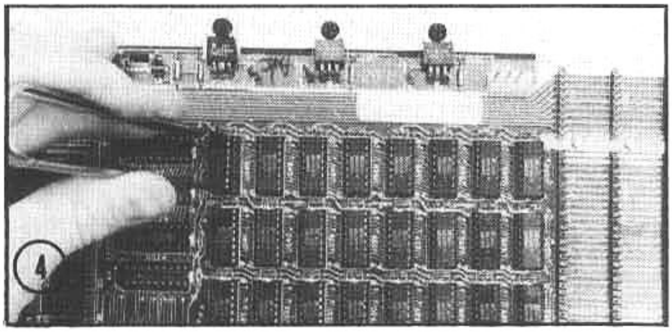
(1) The CPU board must be removed before the 16-Kbyte expansion board can be installed. This is a simple job that requires only disconnecting the power plugs and accessory circuit boards, and removing the two screws that support the CPU board.



(2) As the CPU board is lifted out of position, unplug the interconnecting cable to the TLB.

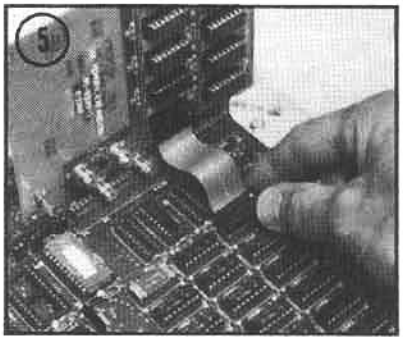


(3) The chip at location "U562" must be a 74S132, not a 74*LS*132. Replace this chip with the correct type—provided in the upgrade kit—if necessary.

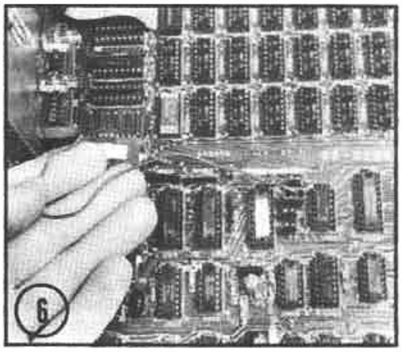


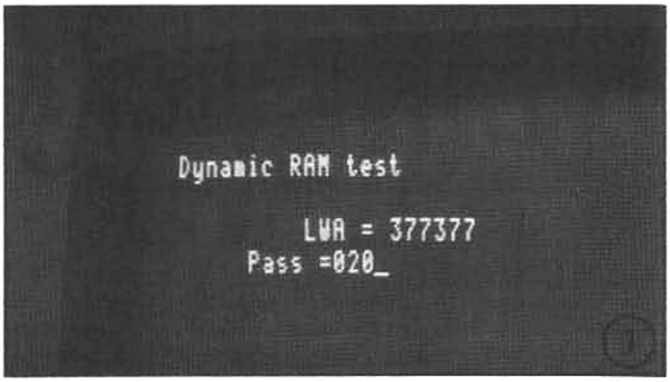
(4) Before the expansion board is installed, *the 24 available RAM sockets on the CPU board must already be filled with suitable memory chips.* The expansion board is only used to provide 16 Kbytes *in addition* to the 48 Kbytes contained on the CPU board. The last memory chip at location U549 is removed to make room for the ribbon interconnect cable from the expansion board. The removed memory chip is then transferred to the single empty socket on the expansion board. Position the chip as explained in the installation instructions.

(5) The expansion board can now be installed on the pins at P509 and P503. Plug the ribbon cable into the memory chip socket at U549.



(6) The jumpers at JJ501 and JJ502 should be configured as shown. If a connector is available at JJ503, the small jumper wire should be installed on the pins as shown. (Note: systems without pins at JJ503 do not require this jumper.)





```
Dynamic RAM test

        LWA = 377377
     Pass =020_
```

(7) After the CPU board is re-installed, the power and interconnect cable plugs, and the accessory circuit boards and support screws are replaced. When the power is switched on, the system should "beep" twice, and the user should immediately run the memory test (type *Go 7375*—at the H: prompt) to verify system operation. The screen should display LWA=377377; indicating that the full 64-Kbyte complement of RAM has been acknowledged.

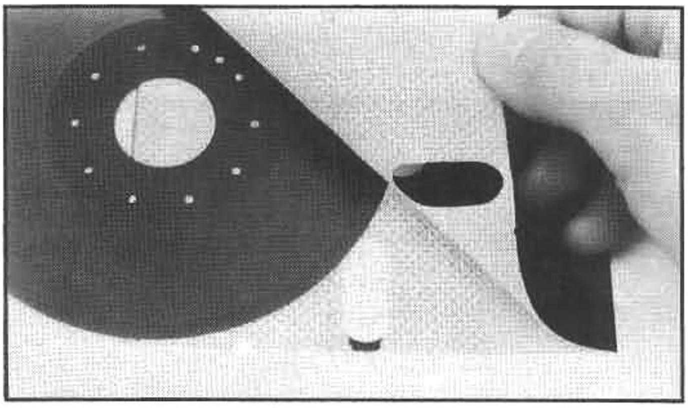
Memory circuits are extremely complex. Each chip contains more than 32,000 active elements; the entire computer system memory uses more than one million active circuits. But, thankfully, integrated circuits are very reliable. You may never have a memory failure in your system, but it pays to be careful; run the built-in memory diagnostic occasionally. A faulty memory chip can cause considerable frustration, but *the user can reduce the likelihood of program faults and/or lost data by regularly using the memory test routine*—Zenith/Heath built it into the system, but it's up to the user to take advantage of it.

DISK DRIVE UPGRADES

If there is one element of a computer system that is essential for performing practical applications, it is the floppy disk. The fast data access provided by floppy disk technology turns the microcomputer into a serious tool for business, professional, and educational use, but at times every Z/H-89 user feels restricted by having only one 5.25-inch disk drive. Many complex, applications programs occupy nearly all of the free space on a single-side, hard-sector disk. This leaves very little free space for storing/manipulating data or text files. And, copying files from one disk to another is a slow and tedious procedure with only one drive, requiring repeated removal and insertion of disks. For these reasons (and others), the expansion to multiple drives is one of the most common user upgrades.

Z/H-89 owners can choose from a wide variety of disk systems available from Zenith/Heath or independent vendors. The basic single-drive system can be upgraded with a separate enclosure that houses two 5.25-inch drives, and this, in turn, can be upgraded with a double-density controller card. Or expansion is possible with double-side, high-density 5.25-inch drives, or 8-inch floppy disk drives, or the ultimate in speed and storage (at this time) is available from hard disk drives that can easily be added to the Z/H-89 system. And these systems can be mixed, providing a wide range of storage possibilities.

We will take a step-by-step approach to storage expansions: starting with modest upgrades and proceeding toward the more expensive options. At each step we will attempt to explain what purposes are served best by that particular upgrade. Since application and software requirements vary greatly, it is seldom possible to say "this is exactly what you need" with any assurance, however, we hope to provide the reader with a preliminary understanding of the potential capabilities, cost, and installation procedures for each system.



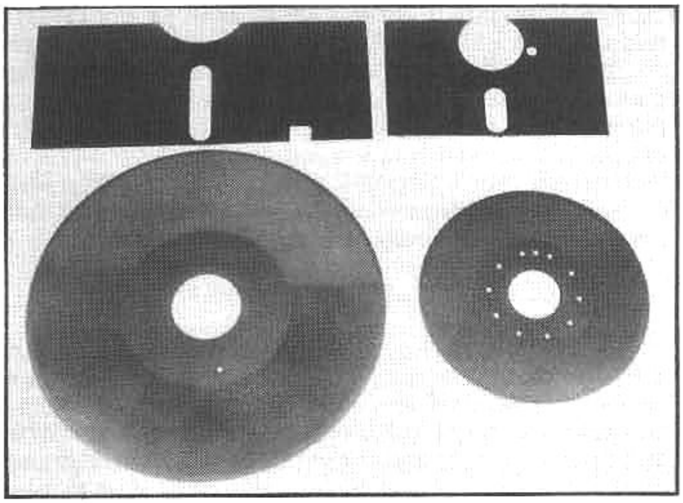
To organize the recording and retrieval of information, the disk-controller system structures the disk surface into pie-shaped segments called *sectors*. As information (programs or data) is deposited along the recording *tracks*, it is broken up into "blocks" defined by the sector limits. When the information is retrieved, the DOS goes to the appropriate sector(s) and track(s) to withdraw the blocks of information and reform them into a continuous program or data file. The sector limits are determined by the disk-controller electronics and the disk design. The recording disk has either a series of sector holes (far right: a *hard-sector* 5.25-inch disk) or a single index hole (right: a *soft-sector* 8-inch disk) to indicate the relative positions of the sectors to the disk controller.

Perhaps the most dramatic upgrade—from the user's viewpoint—is the addition of one or more disk drives to the system. Some high-capacity drives, like this 8-inch system from Data Compass, can expand storage capacity by a factor of 10, allowing a million or more bytes of program and data storage.

ADDING A SECOND DRIVE

If the system is a typical Z/H-89 with a single *hard-sector* drive (see accompanying illustrations for an explanation of hard-sector disk organization), the first expansion to consider is the addition of a second drive. The HS-17-1 (Heath part number) is an attractive accessory cabinet that will house one or two disk drives, and comes with a single drive installed. This system costs about $600.00 and will double the disk storage space, allowing larger (and more capable) accounting and word processing programs to be used, and making disk file copies much easier. *Many experts consider a storage subsystem with two 5.25-inch disk drives the minimum acceptable configuration for small-business applications.*

The hardware installation of the HS-17-1 is very simple and straightforward. (The software considerations will be



discussed later.) All late-model Z/H-89 computers have a plug on the rear panel that makes external drive installation about as simple as plugging it in. Earlier models can be modified to accept this plug-in feature by adding a backplate modification kit, part number H-88-6, that costs about $50.00.

Once the drive is plugged in, the "programming plug" must be configured. This plug looks something like an integrated circuit, and tells the computer how to "address" the drive. For example: it is important that the programming plug on the second drive in the accessory cabinet be configured to "hardware unit one," since the original drive mounted in the Z/H-89 is set for "hardware unit zero." The actual modification to this plug is illustrated in an accompanying photo.

After the programming plug is set and the second drive is properly connected to the main system, both the main system and the accessory cabinet (which contains a separate power supply for the second drive) must be turned on, and normal boot procedure can be completed. The second drive may now be accessed by telling HDOS to look at "SY1:"; for CP/M use the drive specifier "B:" (after running "CONFIGUR"—discussed later). And that's it...almost. There is one thing that must be remembered: *do not forget to turn on the accessory cabinet before you boot up the computer.* If there is a disk in the second drive and a boot is attempted before the accessory power is turned on, the computer will "corrupt" data recorded on the disk surface, making the disk unreadable and the data stored on it unrecoverable by normal methods.

The best way to avoid this problem is to use a multiple-outlet box that controls all the outlets with one switch. This will ensure that the computer and the accessory cabinet are always powered up simultaneously. An additional note of warning: mount the outlet box in a safe location. If the switch is accidentally bumped and the computer loses power at an inopportune moment, all data in RAM will be instantly lost; and if power is lost during a disk-write procedure, it is likely that the disk will be rendered unreadable!

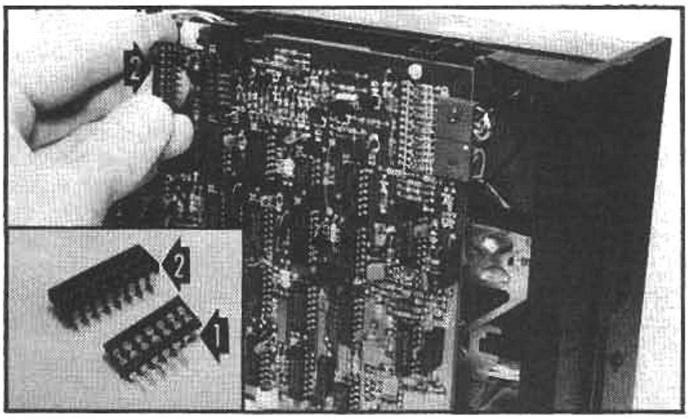
For additional convenience and flexibility, a third drive may also be added to the system rather easily. An H-17-1 disk-drive assembly is needed, the programming plug is configured for "hardware unit two," and it is installed in the remaining slot in the accessory cabinet (which is designed to hold two drives). Plugs are provided in the accessory cabinet for connecting the drive. The cost of adding the third drive is about $300.00.

Although most programs do not *require* three drives, *a third disk drive will increase the amount of online storage and considerably ease many file-handling tasks.* This is especially helpful if the system is used for editing very large text/data files or for developing complex programs.

Three drives of one type (in this case, of hard-sector format) are the maximum that HDOS or the Zenith/Heath implementation of CP/M will support. If more storage space and/or additional drives are required, high-density 5.25-inch drives, 8-inch floppies, or a hard disk may be used along with (or in place of) the existing drives, but no more than two *types* of drives may be used at once.

THE DOUBLE-DENSITY CONTROLLER

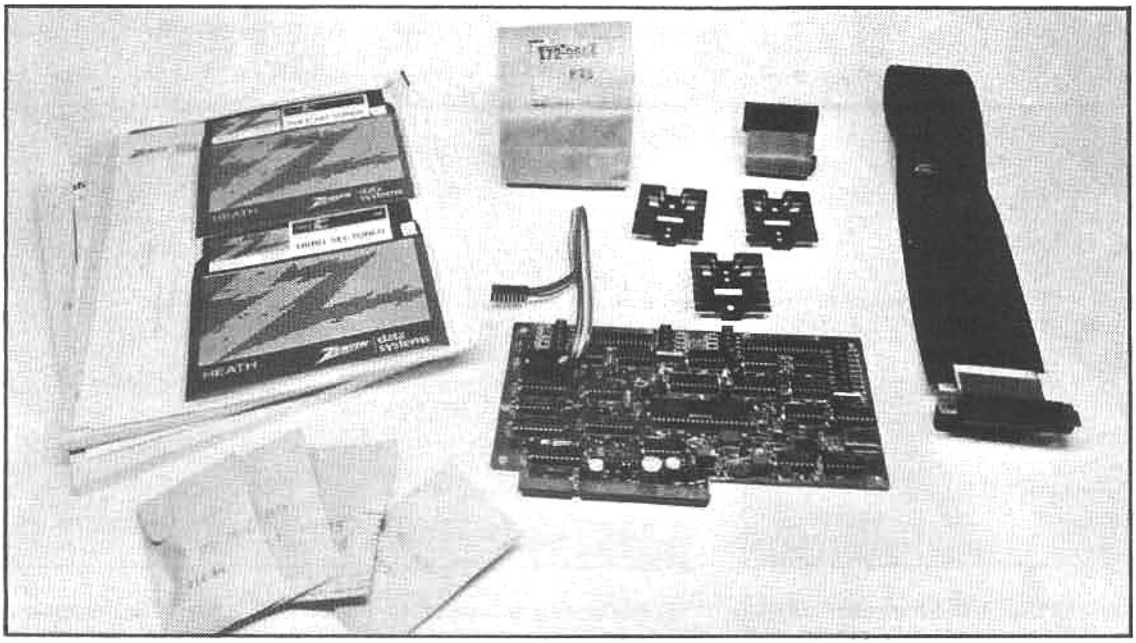
The next step up the Zenith/Heath storage ladder is the Z-89-37 "double-density" 5.25-inch disk-controller card. All Zenith Z-90s are equipped with this card; in effect, it converts standard hard-sector drives to the more-dense soft-sector format. Under control of this card, standard drives will store about 60% more data: 160 Kbytes, compared to the 100 Kbytes of a single-density, hard-sector disk.



Floppy disk drives usually have a simple "programming" device that the user must configure to tell the DOS whether the drive is the "first," or "second," or "third," etc., unit. (In CP/M the drives are formally addressed as A:, B:, C:, etc., and in HDOS the addresses are SY0:, SY1:, SY2:, etc.) The Siemens 5.25-inch drive has an insert plugged into a socket on the drive circuit board (arrow 1) that must be configured to establish the address. American Research & Engineering (see appendix) makes a low-profile, multi-pin switch (arrow 2) that fits nicely into the socket. With this switch installed, address programming is simple and can be easily reset if the user wishes to change the address at a later date.

This card also makes it possible to copy existing programs and data from hard-sector devices to soft-sector devices, and vice versa. For example: in a multi-drive installation, one drive may retain the hard-sector format and, using the Z-89-37 double-density controller card, the additional drives may be converted to the higher-density soft-sector format. After giving careful consideration to specific applications, the user may desire a different arrangement, however, it is generally advisable to select a configuration that will allow continued use of hard-sector software (which the user may already have), while gaining greater storage with the soft-sector controller. The configuration may be modified later, perhaps after all programs and data are converted to the soft-sector format, or as the user's requirements change.

A single-drive computer system is restricted and, at times, inconvenient for applications that require advanced programs and a lot of "data-crunching." However, extra drives and additional peripheral equipment can greatly extend the power and convenience of the system. This typical system utilizes two drives (one in the Z/H-89 cabinet and the second in an accessory cabinet—arrow) and a Z/H-25 printer.



The Z/H-37 is a sophisticated upgrade that uses *soft-sector* storage structure to increase the amount of data that can be recorded on a 5.25-inch disk. If used with standard drives (replacing the hard-sector controller), the storage is almost doubled, and if used with Tandon double-sided drives, the storage space is expanded by a factor of six.

The Z-89-37 double-density controller is installed alongside the standard H-88-1 hard-sector controller, but this is not as simple as it sounds. The installation of this card removes the Z80 "interrupt control logic" circuits from the CPU and relocates them on the controller board. A new monitor ROM is also needed. This means that the installation of the board requires: (1) the removal of five integrated circuits from the CPU board; (2) installing a special plug and connector; (3) reinstallation of three ICs; and (4) configuring several jumpers on the CPU and controller boards. In addition, if the computer being modified is not of the most recent design, the power supply will require modification; a rather tricky operation for those who are not familiar with electronic components.

Frankly, the instruction manual that comes with this kit is painfully brief. Without careful study, one can easily become confused. The major steps in this modification are shown in the accompanying illustrations, but the novice user may wish to have this modification performed by a dealer or service technician.

After the addition of the Z-89-37 controller, even more disk storage can be gained by replacing the H-17-1 standard single-side drive with the H-17-4 Tandon double-side drive. The Tandon drive doubles the storage capacity by writing on *both sides* of the disk, and doubles it again by recording on 80 tracks, instead of the standard 40. This "quad" capacity provides over 600 Kbytes of storage, permitting more than one million bytes (about the size of 10,000 average business letters) to be stored on two drives. The Tandon drives will also read and write in the single-side, single-density or single-side, double-density format for compatibility with H-17-1 drives. Each H-17-4 drive costs about $550.00 (although some discount suppliers sell them for $400.00 or less), a reasonable price compared to the substantial increase in storage space.

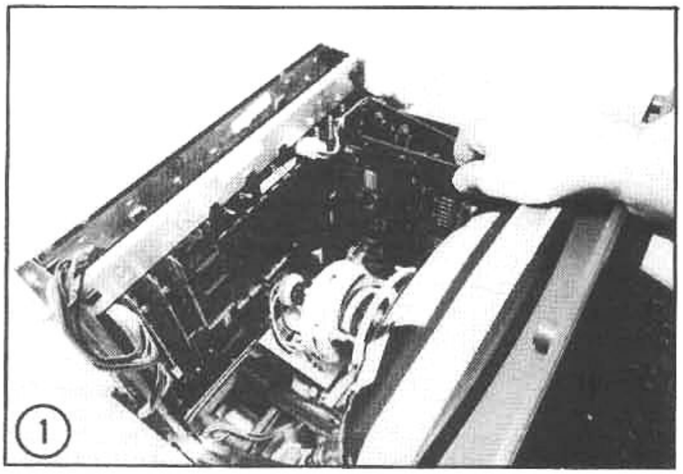
Quad capacity drives, however, are not without peculiarities. Because of their complexity, high-density drives (and the electronics that support them) operate on the fringe of the current technological twilight zone. Zenith and Heath have done an admirable job with the HS-37 package, but it must be pointed out that a design that pushes the physical limits of the disk area is more likely to suffer from reliability problems than a simpler design. When the recording surface is crowded with data, there is a greater likelihood of recording errors. In fact, the quad drives should only be used when they are installed in the accessory cabinet, as RFI "noise" inside the Z/H-89 can cause read/write problems. In addition, some users have voiced complaints that disks written by one high-density drive are unreadable by another drive, however, as a general rule, the HS-37 drives will provide the greatest overall reliability *if the user treats them with the respect and care that any complex piece of equipment deserves.*

Many read errors with high-density (or standard) drives may be traced to the quality of the disk media. It is very important that all disks used in a high-capacity drive be certified for double-side, double-density, 80-track operation: i.e., the manufacturers' specifications *must state* that the disk is certified for 80-track data formatting. In the long run, it is foolish to save a few dollars on cheap disks, especially for recording important data. (It is not our intent to endorse a specific product, but as a point of interest, we noted during the research for this book that many of system and accessory manufacturers we spoke with recommend and claim that they use Dysan brand disks exclusively.)

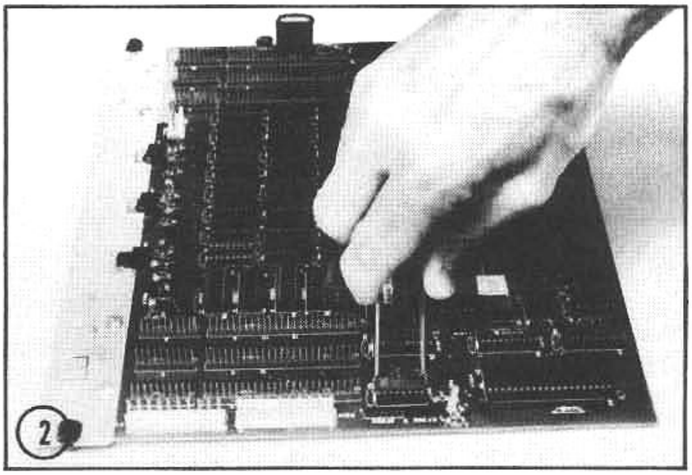
Beyond increased storage capability, high-density disks can access data as much as 50% faster when the CP/M operating system is used. However, under HDOS the access time is increased! Tests have shown that it takes up to three times longer for HDOS to load a program from the HS-37 because the controller and operating system take special measures when reading or writing information. This may not be an important consideration for some users, but it is important to realize that *the HS-37 system has been optimized for CP/M use*, and its operation with HDOS was considered secondary by the Zenith/Heath designers.

The additional storage provided by a higher-density 5.25-inch controller usually provides enough "elbow room" for most hobby or small-business users, however, this may not meet the *portability* demands of some applications. There is no universal 5.25-inch disk format and it is impossible to know for certain that a disk from one system will work in another system (though disks are generally interchangeable if both Zenith/Heath systems have drives of the same format). However, a standard format does exist for the 8-inch floppy disk: the IBM 3740

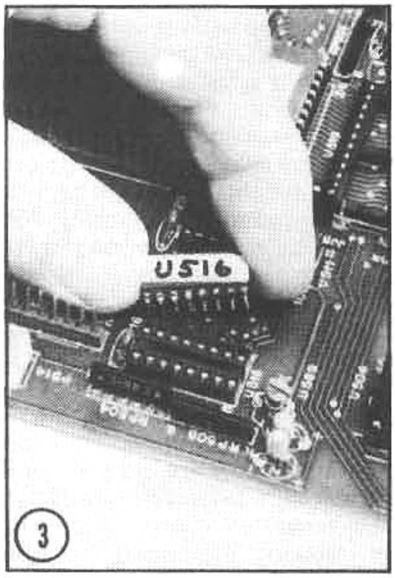
INSTALLING A Z/H-37 CONTROLLER



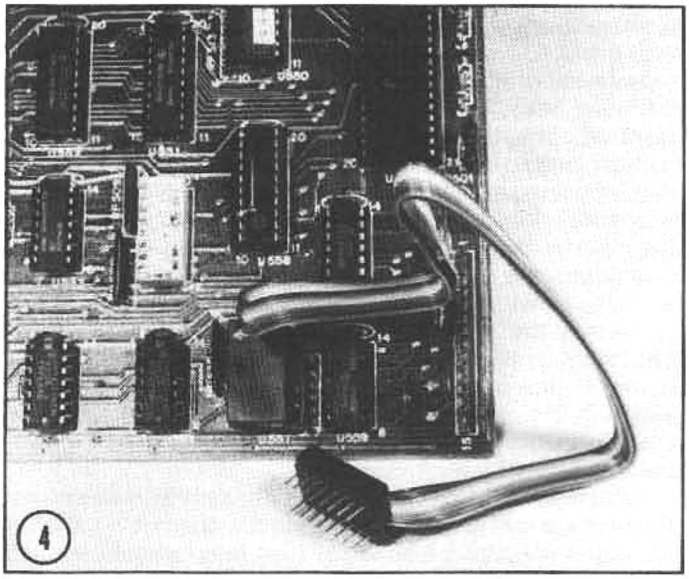
(1) The first step in installing the Z/H-37 double-density controller is to remove the CPU board (details of removing the CPU board are provided on page 80). Carefully remove the power, and interconnect cable plugs, and the accessory circuit boards.



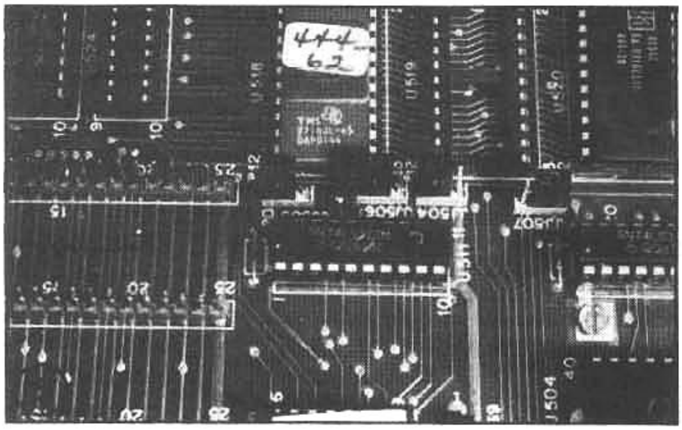
(2) Following the steps in the Z/H-37 installation manual, remove ICs at locations U516, U518, U550, U558, and U557; mark them as shown in illustration 3. Use the IC extraction tool provided in the installation kit and *carefully* work the ICs out of the sockets.



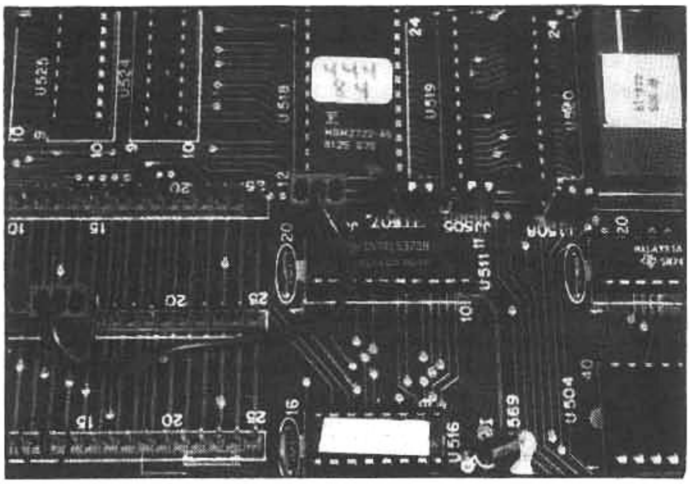
(3) As the ICs are removed, record the socket number on them with masking tape. (If you decide at some future date to remove the Z/H-37 upgrade, this will allow you to reinstall these ICs properly.)

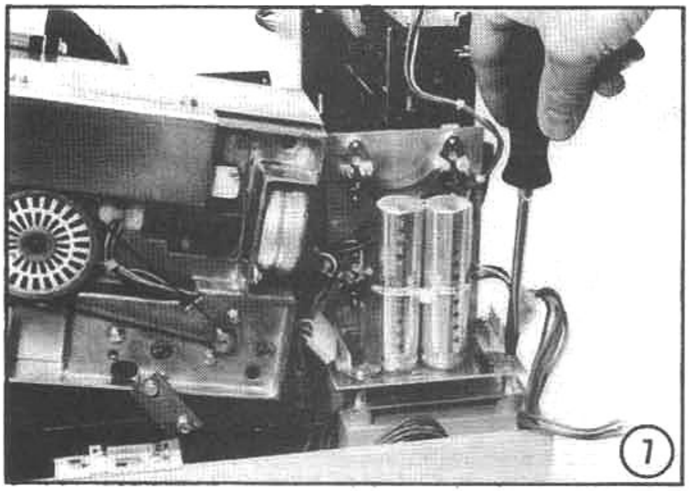


(4) Install the DIP interconnect cable and the special jumper plug at locations U557 and U558.

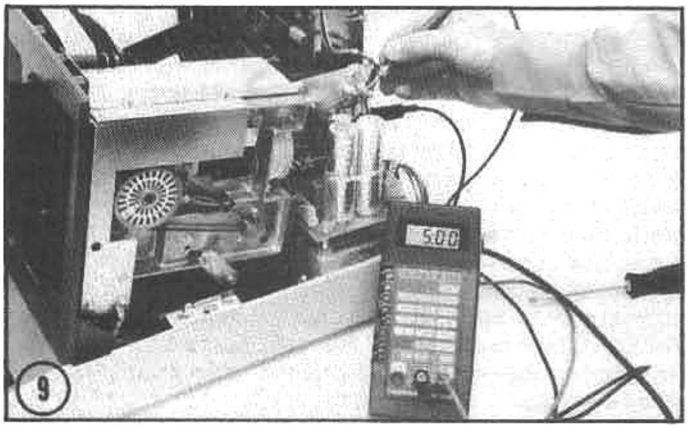


(5&6) This is a before (above) and after (right) sequence to guide you in configuring the plastic slip-on clips located next to U518. One clip is removed, and a short jumper wire is installed to P508 pin 14.

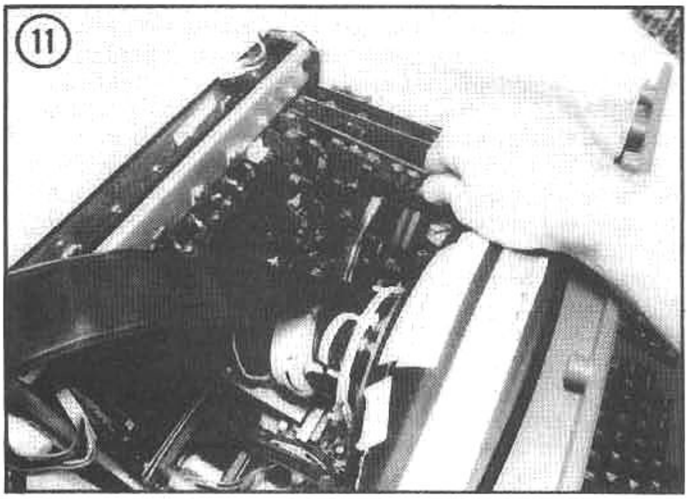




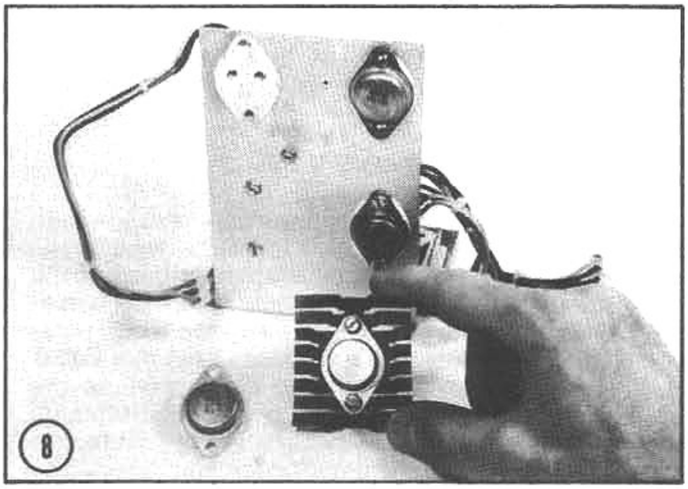
(7) Unplug the cable plugs on the power-supply circuit board. Remove the four screws that hold the power-supply board in position. Then, remove the four hex studs that hold the power-supply heat sink. Lift the power supply out.



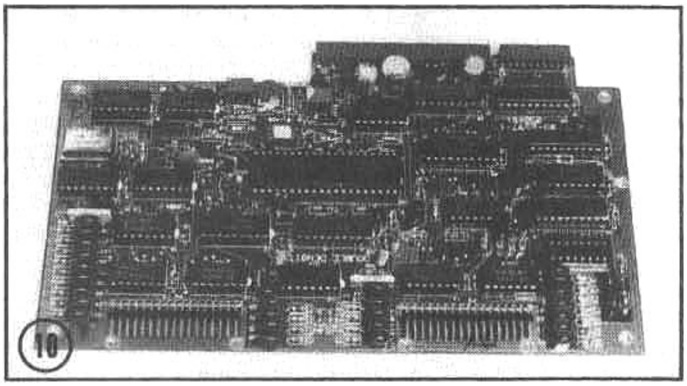
(9) After the heat sinks are installed, re-install the power supply and the plugs to the power-supply circuit board. Then turn on the computer and test the voltages at the power supply to ensure that they are at the proper level *before* reconnecting the power plug to the CPU board. Refer to the installation manual for details. After the voltages are tested, turn off the computer and re-install the CPU board and accessory boards (except the Z/H-37 board).



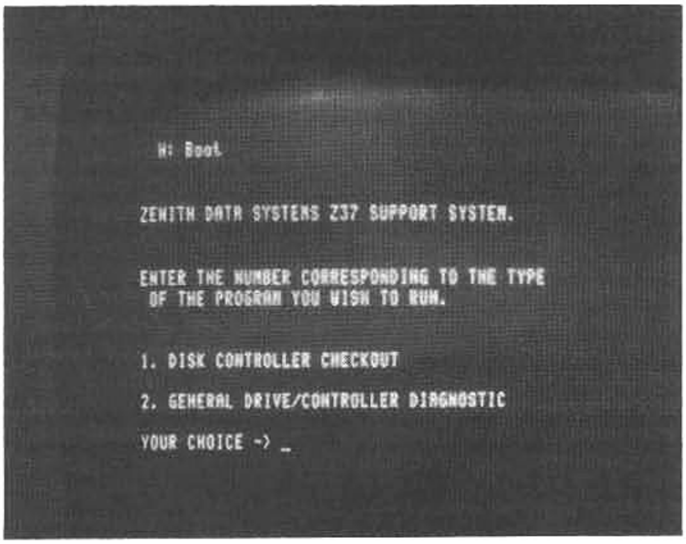
(11) Plug the drive cable(s) and the 16-wire DIP cable into the controller card, then install the card on CPU board pins P504 and P510.



(8) Because the Z/H-37 disk-controller card places extra demands on the power supply, heat sinks must be added under the regulators, and one regulator must be replaced with a heavier-duty model.



(10) Set the jumpers on the controller board as described in the installation manual. The Z/H-37 controller board shown here is configured for two outboard 96tpi disks (the Z/H-17-4 Tandon drives) and an onboard hard-sector drive (controlled by the standard hard-sector disk-controller board). No drives are hooked up to P3 on the Z/H-37 card.



```
H: Boot

ZENITH DATA SYSTEMS Z37 SUPPORT SYSTEM.

ENTER THE NUMBER CORRESPONDING TO THE TYPE
 OF THE PROGRAM YOU WISH TO RUN.

1. DISK CONTROLLER CHECKOUT

2. GENERAL DRIVE/CONTROLLER DIAGNOSTIC

YOUR CHOICE -> _
```

(12) Configure the programming plugs on the disk drives and ensure that all cables are properly routed and connected. Then, turn on the computer—it should "beep" twice—and run the diagnostic software supplied with the Z/H-37 to ensure that all is in working order.

single-side, single-density format. If data portability between systems is an important consideration, it may be better and more economical to opt for an 8-inch disk system, rather than spend money on high-density 5.25-inch formats.

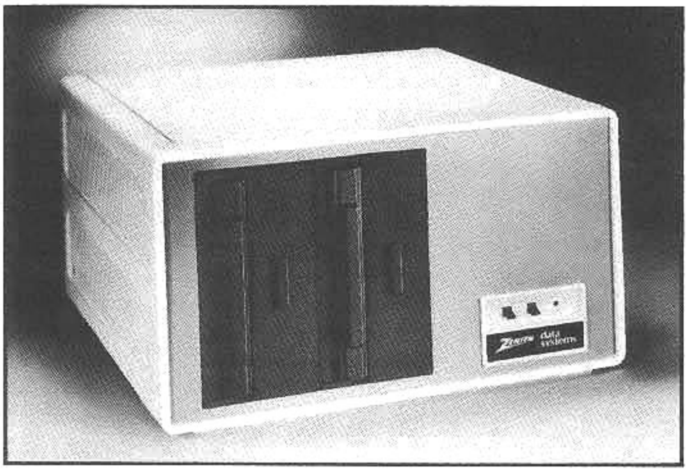
8-INCH FLOPPY DISKS

IBM developed the first floppy-disk storage unit several years ago and, as seems to be the case in the computer industry, the first successful design became the defacto "standard." The 8-inch, single-side, single-density (SS/SD) format has been offered on more computer systems than any other. If a system can read this standard, the user has access (without conversion) to the largest pool of software in the industry, and the IBM standard allows the greatest data portability from system to system—an important consideration for many users. The Z/H-47, Zenith/Heath's 8-inch drive system, will accept the IBM standard, and it will also record in a double-side, double-density format, storing *more than one million bytes on a single disk;* mixing formats is permitted.

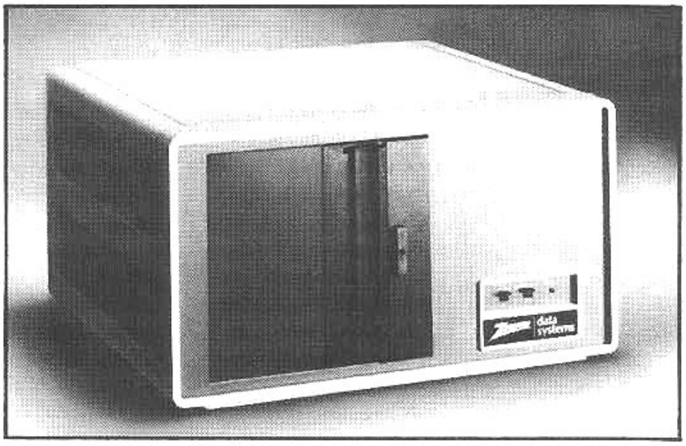
The 8-inch floppy disk is ideally suited to business applications. In SS/SD format an 8-inch disk will store about 1/4 Mbyte, usually ample room for most accounting and word processing applications. In DS/DD format only the very largest data files (unusually large!) will not fit on a single disk. The 8-inch floppy provides plenty of storage, high reliability, and fast data access; and any business manager considering the purchase of a Zenith/Heath computer for accounting, database management, or other professional work should also give thought to an 8-inch drive system.

The Z/H-47 uses Remex brand drives—an "intelligent" design that utilizes an onboard microprocessor. As a result, the system requires only a simple hardware interface to the Z/H-89. The Z/H-47 will house one or two drives, and if both are installed, one drive becomes the master unit (the one with the microprocessor) and the second drive a slave under control of the master.

Installing the Z/H-47 is relatively simple: plug in the interface card, flip one switch on the CPU board, plug the ribbon cable into the interface card, turn everything on and boot up! Software configuration is covered later in this chapter.



A system with 5.25-inch disk drives can accomplish a lot (especially with Z/H-37 Tandon drives), but dual 8-inch disk drives are more common in business applications. They are reliable, fast, and can store more than one million bytes per disk.



The Z/H-67 storage system combines a 1.2-Mbyte 8-inch floppy drive with an 11-Mbyte hard disk, resulting in a powerful, high-speed system. A business that needs large-scale storage or a professional data-processing service that services many small-business clients would use a hard disk system like this one.

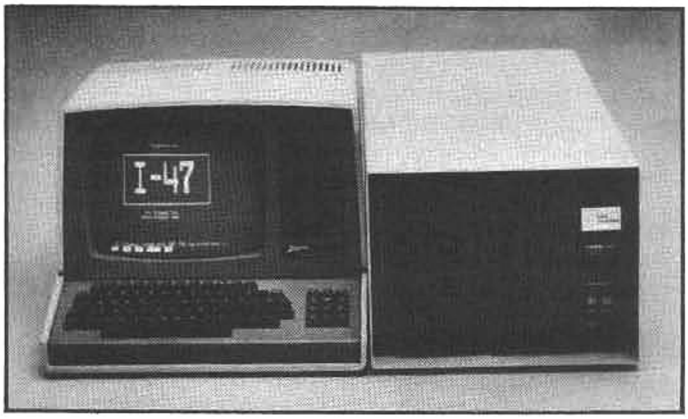
THE HARD DISK

All of the storage devices we have considered so far use floppy-disk technology. The Zenith/Heath Z/H-67 Winchester-type hard disk system does not use floppy media; instead, the storage surface is a non-removable, oxide-coated metal platter. The primary advantage of a hard disk system is the enormous storage capability—nearly 11 megabytes—but less obvious is the extremely quick data access—the platter rotates almost ten times faster than a floppy disk. The disadvantages are the high cost—currently about $6000, though industry observers predict that the cost of hard disks will decline as development continues—and the non-removable nature of the storage surface.

With the addition of the Z/H-67, the Z/H-89 takes on an entirely different character. The system becomes ideally suited for demanding business and professional applications. With suitable database software, thousands of customer records can be stored and any record may be retrieved almost instantly, and the time-consuming process of sorting records (by name, by city, etc.) can be completed in much less time. A hard disk also has enough power to suit professional data-processing needs. Attorneys, doctors, accounting services or business consultants can easily use the space and speed of a hard disk to produce statements, reports, and ledger information for many clients; and most—if not all—client information can reside on the same disk.

Since the hard disk can store a great deal of information, a malfunction can be catastrophic. Fortunately, Zenith/Heath provides a CP/M program to backup (copy) files from the hard disk to 8-inch floppies. The backup program will even copy files that are larger than the available space on one floppy by breaking files into sections and logging the floppy disks as "one-of-two," etc. The program will backup files as large as eight Mbytes—the largest hard disk file supported by CP/M. Electronics are very reliable, but failures do occur. Any hard disk user (and any conscientious computer user) should take great care to maintain backups for all important files!

The installation of the Z/H-67 is simple. It is controlled by an onboard microprocessor, and requires only an interface card for connection to the Z/H-89. To complete the installation: one switch must be reconfigured on the CPU



The Data Compass I-47 8-inch disk system is virtually identical to the Zenith/Heath Z/H-47. It uses Remex drives (as does the Z/H-47), a reliable power supply, and a cooling fan fitted with an air filter. It has 100% compatibility with Z/H hardware and software, but the I-47 is substantially less expensive than the Z/H-47.

card, the interface card must be installed, and some cables must be routed between the computer and the drive. After the software is configured, the system is ready to use.

DISK UPGRADES FROM OTHER SOURCES

DATA COMPASS DISK DRIVES

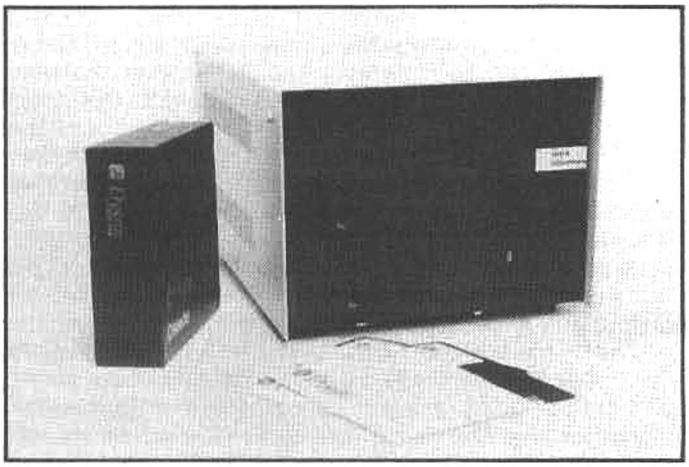
Data Compass, Inc., (see appendix) is a computer timesharing company that also sells paper products, computer accessories, and—of particular interest to Zenith/Heath users—floppy-disk drives. They became a distributor for Remex drives in 1979, and as a result of their ability to buy and sell in large quantities, they have an interesting alternative to the Zenith/Heath Z/H-47.

The Data Compass I-47—based on Remex 8-inch drives—is an assembled and ready-to-run disk system, but an interface card is not included, and all that is needed to complete the installation is a Zenith/Heath Z/H-89-47 interface card (which costs about $195). Because of the design of the Remex drives used in both the Zenith/Heath Z/H-47 and the Data Compass I-47, the two systems are virtually identical (there is 100% software compatibility), except that the I-47 costs about $1000.00 less than the Zenith/Heath equivalent.

The rugged steel cabinet of the Data Compass I-47 measures 13.75-inches wide, 23-inches deep, and 11.5-inches high. (It is about five inches narrower and five inches deeper that the Z/H cabinet.) The top is easily removed for service or reconfiguring the jumper-selectable defaults (drives can be individually configured for SY0: or SY1:, default density, etc.). The power supply includes an RFI-line filter to reduce radio interference, and a fused AC input. A cooling fan and air filter are mounted on the rear of the cabinet. The I-47 is shipped with Remex technical manuals (not supplied with the Z/H-47) that illustrate all the jumper defaults and master-controller programming techniques. The drive schematics are also supplied. The I-47 will operate with both HDOS and CP/M.

The I-47 has been such a success that Data Compass plans to expand their line of disk drive accessories for Zenith/Heath computer systems. They claim that the following products are under development:

The I-37—a Z/H-37 equivalent. With two 5.25-inch drives (using the Zenith/Heath Z-89-37 controller board) each disk drive will store 640 Kbytes



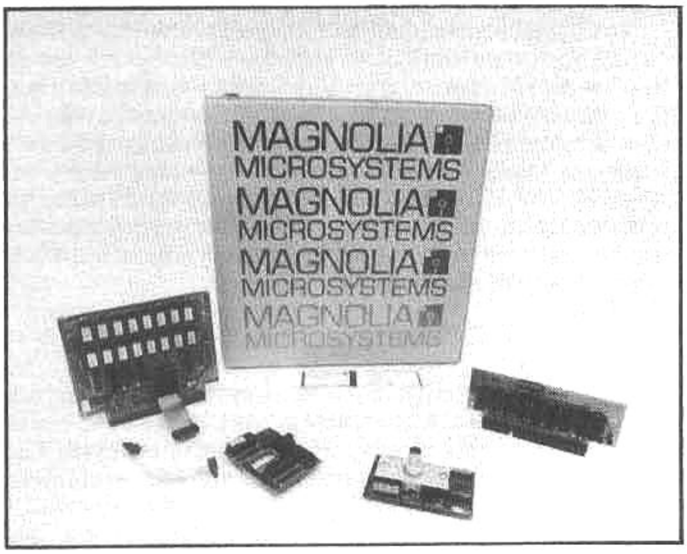
Data Compass also offers a system equivalent to the Z/H-37 high-density 5.25-inch drives. The I-37 houses two double-side, double-density drives and an internal power supply. A unique product is the I-37 (shown here) with one 5.25-inch floppy and one 5-Mbyte 5.25-inch hard disk.

The I-39—a unique product that uses one 5.25-inch floppy drive (of the I-37 design) and a 5.25-inch hard disk that will store 4.4 Mbytes (formatted). A controller card for this system will be available only from Data Compass.

The Data Compass systems are based on standard technology and hardware, and the cost-conscious user may find these alternatives interesting.

MAGNOLIA MICROSYSTEMS DISK DRIVES

Considered one of the premier sources of specialty products for Zenith/Heath computers, Magnolia Microsystems (see appendix) offers several disk-controller cards and disk drives that have been designed and manufactured to their own specifications. Their line of products includes: (1) a double-density disk controller card that operates four 5.25-inch *and* four 8-inch drives at the same time; (2) a controller card that can interface 8-



Magnolia Microsystems is one of the largest and best known independent suppliers of accessories for Zenith/Heath computers. Their products range from a 16-Kbyte memory expansion board to a professional business system that they can configure to any user's requirements.

inch Remex drives (Z/H-47 or I-47), a Corvus 5-, 10-, or 20-Mbyte hard disk, *and* contains three serial printer/modem ports; (3) a 128-Kbyte memory-expansion board that emulates a disk drive, but will read and write data 200 times faster than a floppy; and (4) a video-output board that duplicates the computer screen on a remote monitor.

The most popular card in Magnolia's line is their double-density controller for mixing 5.25- and 8-inch drives. This card (part number 77316) uses a proprietary technique to increase the number of I/O (input/output) ports on the Z/H-89, allowing the user to have up to eight disk drives at once, *in addition to* the three drives supported by the standard Zenith/Heath H-88-1 hard-sector controller. The Magnolia controller supports both single- and double-sided drives in both single- and double-density recording. The 96-tpi Tandon H-17-4 drives are also supported. The controller is shipped with cables for both the 5- and 8-inch drive, new monitor and I/O decoder ROMs (replacement integrated circuits), and Magnolia's modified version of CP/M.

For serious business and scientific applications, Magnolia Microsystems produces a number of hard disk storage systems that vary in price from $3195 to over $14,000. At the low end, a 5-Mbyte system (part number 5MCORV) comes complete with an interface card, case, power supply, and cables. Another product designed for the small- to medium-size business is the 10M+D8DSDD system. This setup has two Tandon "thinline" 8-inch floppy drives and a 10-Mbyte hard disk mounted in one cabinet. All controller and interface electronics are supplied, along with the cables and CP/M, for a price of $5995. Their high-end offering is a 51-Mbyte system with a 40-Mbyte fixed-platter hard disk and an 11-Mbyte removable-cartridge hard disk. The removable rigid disk can be transported, stored, exchanged and used for all of the functions that are common and (convenient) with floppy disks—yet it holds 11-Mbytes! It is a very powerful system, the cost is about $14,000.

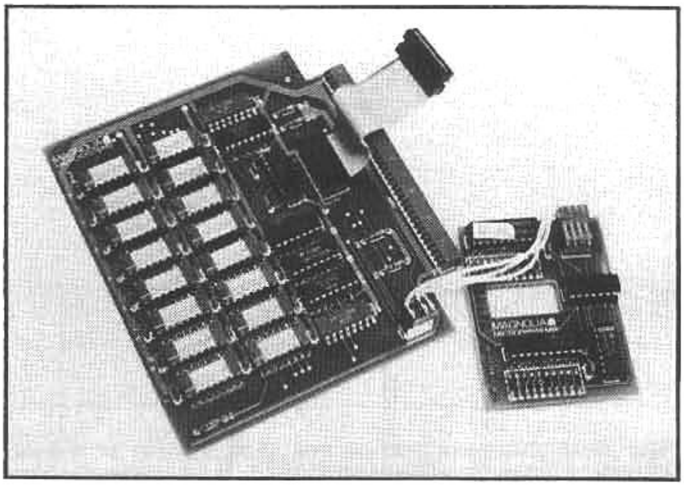
Another interesting product from Magnolia is their "invisible disk." It is a 128-Kbyte RAM board designed to function like a floppy-disk drive. It replaces the original 16K RAM expansion board and gives the Z/H-89 an upgrade from 48 to 64 Kbytes of RAM and an "invisible high-speed disk drive" with 112 Kbytes of storage.

The invisible disk is initialized upon system boot, then it can be accessed just like an additional floppy disk, except that the rate of data transfer is up to 200 times faster than a standard disk drive, but the information stored in the RAM disk is volatile (lost when the computer is powered down). Magnolia supports this board with their proprietary version of CP/M. The board will also run under MP/M (a Digital Research multi-user operating system); so that several users can share the CPU, with each user having the equivalent of a single-user CP/M system.

ULTIMETH—HDOS SUPPORT

Magnolia Microsystems makes systems and upgrade components that are designed to work with CP/M, but do not support HDOS directly. However, the Ultimeth Corporation (see appendix) offers HDOS software support for some Magnolia products.

Dean Gibson, Ultimeth's chief code-cruncher, has been a programmer and computer consultant for over 20 years, and he has developed a unique application for the Magnolia disk-emulator. In addition to providing HDOS disk operation similar to that produced by MMS's CP/M version (his HDOS version provides 111 Kbytes of storage), Gibson has produced software that uses the disk-emulator as a "cache memory" for programs or data that are normally stored on a disk but have to be repeatedly accessed by the user (or RAM programs).



Magnolia Microsystems offers a unique RAM card that adds a 128-Kbyte "logical disk" to the memory. Accompanying software tells the CPU to view this disk emulator as if it were a conventional disk drive, and the user accesses the memory just like a normal disk, but read and write speeds are more than 200 times faster.

The concept of cache memory is not unique, but it is relatively new in microcomputers. With certain applications—particularly program development—it can provide notable time savings because access time to information stored in the cache is reduced as much as 40%. With programs that repeatedly read and write data (e.g., database programs) this adds up to a considerable improvement.

In addition to supporting the disk-emulator card, Ultimeth also sells HDOS support software for Magnolia's disk-controller card and replacement monitor ROMs that provide increased HDOS processing speed and debugging capability.

OPERATING SYSTEM CONSIDERATIONS FOR DISK UPGRADES

One of the main functions of an operating system is recording and reading information on a floppy disk (or similar storage devices). However, data is recorded on a 5.25-inch disk much differently that on an 8-inch disk, and the recording patterns on a soft-sector disk are much different than those on a hard-sector disk. Within each of these disk formats there are additional variations in track configurations and recording densities. An operating system that could read/write to all of these formats would be huge, but most users need only a small portion of this capability. As an alternative, the designers of HDOS and CP/M have given the user a way to "configure" the operating system for the specific type of storage device in use, this makes the DOS flexible, yet fairly compact (consuming less disk space).

In the next section we will provide an overview of the procedures for configuring an operating system for specific hardware requirements.

BUILDING A BIOS FOR CP/M

CP/M handles the chores of disk input and output through a sub-program called *BIOS* (Basic Input/Output System). The BIOS is the portion of CP/M that is configured by the user to operate with specific storage devices. Zenith/Heath CP/M supports the use of up to two different types of disk drives at one time. The options presently supported are:

H-17 ONLY
H-37 ONLY
H-47 ONLY
H-67 ONLY
H-17 AND H-37
H-17 AND H-47
H-17 AND H-67
H-37 AND H-47
H-37 AND H-67
H-47 AND H-67

The process of modifying the BIOS is normally complex, but Zenith/Heath has removed most of the confusion by incorporating all of the steps into an automatic "chain of modification." The user runs a utility program called *MAKEBIOS*, and CP/M will—using a chaining utility called *SUBMIT*—execute a series of pre-programmed steps that complete the BIOS modification.

After the new BIOS is generated, it must be installed (written to the proper portion of the disk) with the help of two additional programs called *MOVCPM* and *SYSGEN*. This can be a confusing process and the user is advised to *carefully review the manual before attempting the procedure*. When the installation is complete, CP/M (with the new BIOS) will support the new hardware environment. And CP/M can be remodified at any time—using the MAKEBIOS utility—to support a different configuration.

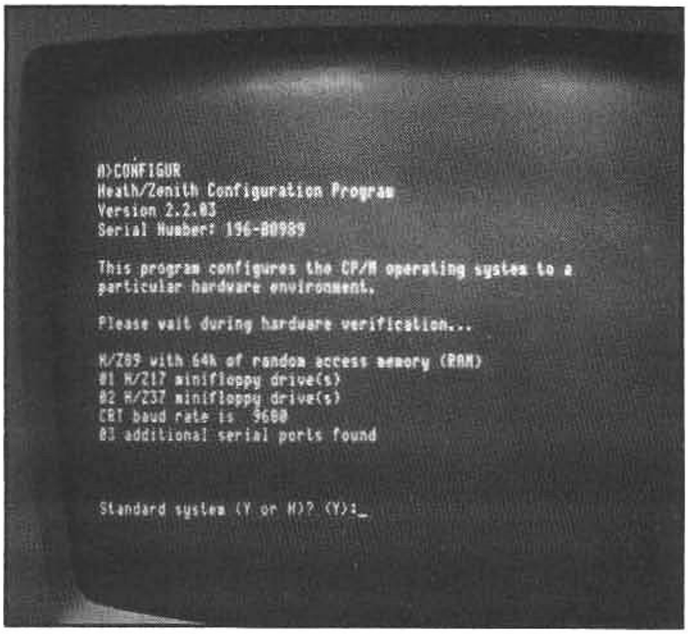
THE CONFIGURATION OF HDOS

HDOS 2.0 (issue 50.06.01) is very easy to configure for any combination of disk drives. The HDOS equivalent of the CP/M BIOS for disk-drive access is a special program called a *device driver*. There is a separate program (driver) for each type of disk drive, and all that is required to support the current combination of drives is to place copies of the appropriate drivers on the boot disk. HDOS will "look over" the disk during boot, and any legitimate drivers found on the disk will be incorporated into the operating system. It is that simple! However, the same combinations and limitations that apply to CP/M—no more than two *types* of disk drives may be used at one time—also apply to HDOS.

The modular concept of device drivers is an interesting feature of HDOS. This has encouraged some programmers to offer special drivers for unique applications or drivers that incorporate novel features and enhancements. There is also the likelihood that storage devices based on entirely new technologies will be available in the future (e.g., laser-scanned optical disks), and these devices can be readily supported by new HDOS device drivers.

CONVENIENCE ENHANCEMENTS

Some enhancements do not really boost the computing "power" of the system, instead they free the user from repetitive and time-consuming tasks. These "convenience" features include software modifications and utilities, customized hardware, and accessories that increase the practical value of the system.



```
A>CONFIGUR
Heath/Zenith Configuration Program
Version 2.2.03
Serial Number: 196-00989

This program configures the CP/M operating system to a
particular hardware environment.

Please wait during hardware verification...

H/Z89 with 64k of random access memory (RAM)
01 H/Z17 minifloppy drive(s)
02 H/Z37 minifloppy drive(s)
CRT baud rate is  9600
03 additional serial ports found


Standard system (Y or N)? <Y>:_
```

A utility program called CONFIGURE modifies CP/M for a specific hardware environment. Here the program has examined the system and determined the number of disk drives, serial ports, etc. This function is automatic and requires very little user intervention.

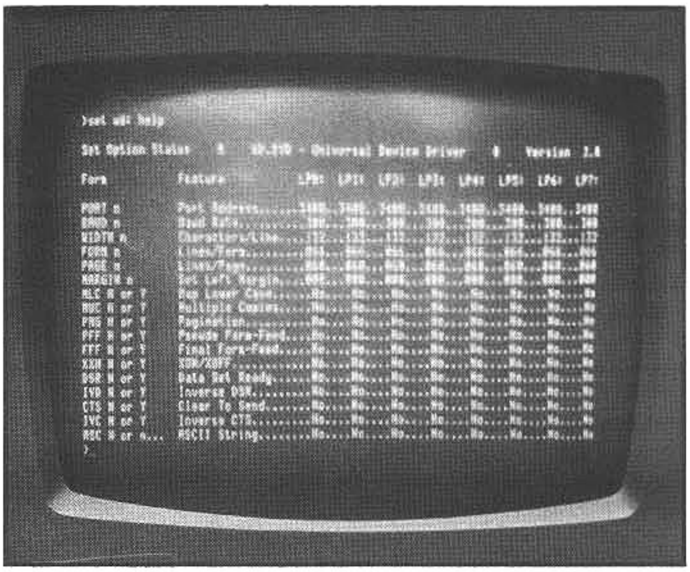
HDOS ENHANCEMENTS

Software enhancements can add "friendliness" to the computer. One of the most useful of these modifications is a utility program called SYSMOD. Developed by Jim Teixeira, of the SoftShop (see appendix), SYSMOD modifies HDOS by simplifying existing commands and adding some new features. It is extremely easy to use: the user transfers a copy of SYSMOD to a bootable HDOS disk and runs the program by typing SYSMOD. The program modifies HDOS with virtually no intervention required by the user. Within a few seconds the modification is complete, and the user may type **HELP** to get a list of all the new commands (all the original commands of HDOS are also supported). A few of the abbreviated or special SYSMOD commands are:

M1, M2, M3—Mount disk in drive one, two, and three, respectively
DO, D1, D2—dismount disks
CO, C1, C2—display a disk catalog
RO, R1, R2—disk reset (dismount current disk and then mount new disk)
LAB newname—change disk label to newname
TY, REN, CO—type, Rename, and Copy files
PR filename—send a disk file to the printer
PC—send a disk catalog to the printer
DOCUM filenameCommand1;Command2;etc.—build a submit file

A DOCOM or "submit" file is a series of individual commands that the operating system will automatically execute in sequence. This allows a single command and to direct the computer to execute a series of tasks. For example: the user could build a submit file containing **M1,M2,C1,C2**, and assign it the filename "MOUNT"; then, when the user types the command **MOUNT**, the computer will sequentially mount disks on drives one and two and list a catalog for each of these disks.

SYSMOD is also available in a more powerful version



```
>set ud: help

Set Option Status  :  UD.DVD - Universal Device Driver  :  Version 1.0

Form          Feature            LP0:  LP1:  LP2:  LP3:  LP4:  LP5:  LP6:  LP7:

PORT n        Port Address.......[illegible]
BAUD n        Baud Rate..........[illegible]
WIDTH n       Characters/Line....132...132...132...132...132...132...132...132
FORM n        Lines/Form.........[illegible]
PAGE n        Lines/Page.........[illegible]
MARGIN n      Set Left Margin....[illegible]
MLC N or Y    Map Lower Case.....No....No....No....No....No....No....No....No
MUC N or Y    Multiple Copies....No....No....No....No....No....No....No....No
PAG N or Y    Pagination.........No....No....No....No....No....No....No....No
PFF N or Y    Pseudo Form-Feed...No....No....No....No....No....No....No....No
FFF N or Y    Final Form-Feed....No....No....No....No....No....No....No....No
XON N or Y    XON/XOFF...........No....No....No....No....No....No....No....No
DSR N or Y    Data Set Ready.....No....No....No....No....No....No....No....No
IVD N or Y    Inverse DSR........No....No....No....No....No....No....No....No
CTS N or Y    Clear To Send......No....No....No....No....No....No....No....No
IVC N or Y    Inverse CTS........No....No....No....No....No....No....No....No
ASC N or n... ASCII String.......No....No....No....No....No....No....No....No
>
```

Under HDOS, a device driver program is responsible for communicating with the system printer. Jim Teixeira's Soft Shop sells a very versatile driver called UD.DVD (Universal Device Driver). This display shows the programmable options of this unique driver.

called SUPERSYSMOD. This package offers alphabetized disk directories, a powerful file management system, and more.

The SoftShop also offers a unique HDOS printer driver call UD.DVD. A printer driver is a short program that tells the CPU *how* to send information to a printer. Ordinarily the driver is configured with the SET program—supplied with HDOS and explained in the HDOS manual—to adjust variable features in the driver. If the user changes printers or decides to change any of the printing parameters, the driver can be reconfigured with SET to suit the new requirements.

But the SoftShop universal device driver (UD.DVD) is, in effect, eight printer drivers in one. Each of the driver modules may be configured for different printers, or special printing situations, and each is addressed separately as LP0:, LP1:, LP2:,...LP7: (instead of the standard HDOS printer driver address—LP:). An additional feature of SYSMOD is the ability to send disk files to any of the eight drivers by simply typing **PRx filename** (where x is the driver address). UD.DVD (and a version specifically for the Z/H-25 printer—H25.DVD) is available for under $30.00.

The Software Toolworks also offers utility programs that enhance computer usage (see chapter 6). One of particular interest produces an organized listing of all files on the user disks. As each disk is inserted in the drive, the disk catalog is recorded in RAM, and when all disks have been cataloged, a sorted (alphabetical or by disk number) listing of all programs and data files can be produced on the printer. This utility can be a tremendous aid to users with extensive disk files.

Handy, inexpensive utility programs are also available from the Heath Users' Group (see appendix). Among the numerous offerings are programs to recover erased files, renumber disks (different from renaming disks), test memory, etc. In addition, enhancement programs developed by independent programmers are often introduced and described in BUSS (a newsletter for Zenith/Heath owners; see appendix), and Zenith/Heath's commitment to support HDOS on the Z/H-100 will ensure a continued supply of useful HDOS utilities.

CP/M ENHANCEMENTS

Because CP/M is the most popular operating system for microcomputers, thousands of CP/M programs are available, and a large portion of this software is in the public domain (free to be copied and used by anyone). It all started several years ago. Most of the people who first used CP/M were hobbists, and they were not interested in selling software for a profit; they just passed copies on to their friends. Now, most of these programs are considered part of the public domain (copyright protection has been forfeited), and one of the largest collections of public domain software is offered by the CP/M Users Group (CPMUG) in New York. A recent CPMUG catalog listed over 50 volumes (8-inch disks), each packed with programs for producing sorted directories, patching and examining disk files, fast disk-to-disk copies, memory testing, and hundreds more. The cost is only $8.00 per volume (little more than the cost of a blank disk). The programs are not extensively described in the catalog, but any CP/M user should find the catalog of interest.

Another source for CP/M (and, to a lesser degree, HDOS) programs is the growing number of telephone bulletin boards springing up throughout the U.S. and overseas. Bulletin boards are usually operated by hobbists who have programmed and equipped their systems for telephone communications with other computers/terminals. When the user is "logged on" the bulletin board, he may read messages left by other users, leave his own messages or comments, and with some bulletin boards the user may "download" programs (some of which are true gems of programming). The bulletin-board concept has become so popular that the National Aeronautics and Space Administration has set up a system that allows space enthusiasts to get the latest news on the Space Shuttle and other NASA missions.

SOFTWARE SPOOLERS

A Zenith/Heath computer, combined with a printer, is a powerful word-processing or accounting system, however, when the computer is sending data to the printer, it is not available for further writing/editing or data processing. With short files this is not a problem, but with large text or data files, the system may be tied up for several minutes while the printer is plodding through the output. To circumvent this limitation, a type of program called a *printing spooler* can be used to "time share" the computer between printing and another task.

Most printers have a small data-storage RAM called a *buffer*. Data is fed from the computer to the printer at a very high rate until the buffer is full; then the computer stops sending data until it receives a signal from the printer that the buffer is nearly empty. When the computer receives this signal, it sends another burst of data to the buffer, and waits for the next signal. During the periods when the computer is *not* sending data to the printer, the CPU is idle, and a spooler takes advantage of this idle time to use the CPU for other tasks. The spooler program switches the CPU between the printing transmissions and the secondary task/program. As a result, the computer and the user do not have to sit idle while a long file is transmitted to the printer.

Spoolers are complex programs that use sophisticated programming techniques. Since the CPU is running two programs at the same time, the overall processing speed is slower, and occasionally there may be conflicts with some secondary tasking, causing unpredictable results. The user is advised to thoroughly test any spooler program before committing important data to a spooler-controlled printing operation.

The Software Toolworks sells an HDOS spooler called SPOOL-N-GO. Zenith and Heath sell a CP/M spooler called DESPOOL. Either program can be purchased for under $50.00.

There is another approach to speeding the printing process; instead of a software spooler, *a hardware spooler* —what is, in effect, a very large buffer—may be used. This accessory will be discussed in the following section, along with other hardware enhancements.

HARDWARE CUSTOMIZING AND ENHANCEMENTS

For some tasks the user may wish to add specialty hardware, or customize the system for added convenience. There is a wide array of accessories currently available for the Z/H-89.

ANTI-GLARE SCREENS

The CRT screens on early Z/H-89's had a smooth surface, which reflected ghost images from ambient room lights. A few companies (see appendix) make anti-glare filters that can be installed over the front surface of the video display. These filters have a matte surface, however, and some of them may make the display appear slightly fuzzy. If the user works in an area with strong overhead or side lighting, an anti-glare screen may be helpful. Zenith and Heath market an excellent filter—part number HCA-4—that is available in the Heath catalog and in Zenith/Heath stores. Most filters cost about $20.00.

Late-model Z/H-89s are equipped with CRT screens that have an anti-glare front surface. The screen retains a very sharp image and virtually eliminates reflected glare. It is a $30.00 option in the Heath H-89 kit, but it is generally a worthwhile investment. All Zenith Z-89 and Z-90 systems have, as standard equipment, a green-phosphor display with the anti-glare treatment.

CUSTOM KEYS

If you use a software package that requires many control-key sequences in its command structure, custom keytops can greatly reduce frustration and familiarization troubles. Replacing the keys on a Z/H-89 is really quite simple. Grip the edges of the key securely with needlenose pliers, and gently pulled upward, separating the keytop from the keyswitch assembly.

At least one specialty company, The Graven Image, produces a wide variety of custom and specialty keys. The proprietor, James Scott, claims that the most requested set of specialty keys is for the WordStar word processor. Special sets for avid game-players are also popular. To deter theft, Mr. Scott can custom engrave a replacement spacebar with the owner's name and/or a unique i.d. number (driver's license or social security number, etc.).

For dedicated applications, such as the extensive use of a complex word processor (like WordStar), custom keys can make day-to-day operation much easier for the user. These custom keys are from the Graven Image (see appendix).

Warning or reminder plaques add a practical custom touch to any system, and engraving a license number (or other identifying number) on the keyboard is an excellent deterrent to theft.

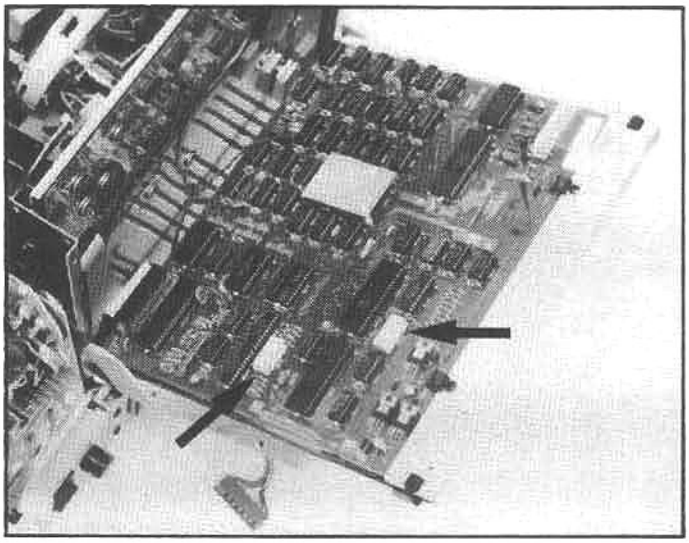
The Graven Image offers a variety of adhesive-backed plastic legend plates to complement the front of the Z/H-89 or the Zenith/Heath accessory cabinets. These plates can be made in any color, with any lettering the customer desires. There are several standard plates favored by users, including: drive labels, "Drive SY0:," "SY1:," etc.; warning labels, "Warning! Remove disk at power-up/down" (this replaces the Zenith/Heath label below the drive opening in the computer cabinet); and plates that display company or individual names.

INTERNAL CONFIGURATION SWITCHES

The Z/H-89 contains two large circuit boards that hold a majority of the electronics: the terminal logic board (TLB) controls the keyboard and screen, and the CPU logic board (CPUB) provides the interconnects for the Z80 microprocessor and support logic. The CPUB is located directly behind the neck of the CRT and the TLB is mounted just behind the CPUB (placing it at the far rear of the cabinet). Each of these boards contain at least one DIP switch (dual-inline-package switch), that controls the system default settings—establishing the "operational condition" of computer after initial powerup.

Switches 402 and 401 on the TLB control several terminal functions, some of which are beyond the scope of this book, and the reader is advised to make changes only after reading the Zenith/Heath operating manual. As a summary, switch 402:

Selects an underscore or block cursor
Provides or inhibits a click when a key is pressed
Discards or wraps the right end of lines on the display
Provides or inhibits a line-feed with each carriage-return
Provides or inhibits a carriage-return with each line-feed
Selects Heath or ANSI escape mode
Selects normal or shifted keypad mode
Selects 50 or 60-Hz refresh



Two switches on the terminal logic board (TLB) set the configurable characteristics of the terminal section. These characterisitics can be (and often are) reset by software commands, but a user need not change the default parameters, except in specialized applications (particularly systems running dedicated custom software).

Switch 401:

Selects terminal baud rate (110 to 19200)
Selects terminal parity (none, odd, normal)
Selects full or half duplex operation

Almost all of the terminal configurations can also be reset by software commands. Some software packages—particularly word processors—expect the terminal to have specific characteristics upon startup, and the modification of these parameters should be done with care. When the terminal switch settings are modified, all programs should be tested to determine if they have been adversely affected.

The CPUB has a configuration DIP switch (switch 501) that is used, among other things, to tell the system what kind of disk drives are installed (hard-sector, soft-sector, 8-inch, etc.). The exact function of the switch is determined by the monitor ROM installed at location U518. There are two common monitors, MTR-89 and MTR-90, and the latter is required if the system uses the Z/H-37 soft-sector 5.25-inch disks. The system uses the CPUB switch settings, as defined by the monitor, to properly read and write from/to disks, and to establish whether the secondary device is a 5.25-inch or 8-inch disk, etc. Switch 501 also performs other functions:

Boot from primary or secondary disk
Perform or ignore memory test after SHIFT-RESET
Set console to 9,600 (standard) or 19,200 baud
Set normal or automatic boot on power-up

These functions are explained (though not in great detail) in the system manual. Any switch setting changes on the TLB and the CPUB should be done with the power off, both for safety and for the protection of the components on these complex boards.

HARDWARE PRINTER SPOOLERS

Earlier we examined software spoolers, and explained that this type of program "shares" the CPU between a printing task and a secondary applications program. CPU-sharing is common in mainframe computers, where a very high-speed CPU can handle numerous tasks virtually simultaneously. In comparison, the CPU in a microcomputer is modest, and when a microprocessor must perform two jobs at once, the system may balk (especially during disk access). But there is another solution; a *hardware spooler* is a large memory buffer that can receive data very quickly, store it, and pass it on to the printer at a rate that the printer can accept (usually much slower). The computer transmits text files to the spooler memory, just as it would send the file to a printer, but much faster. Within seconds the transmission is complete and the computer is free for the next task; the text file, stored in the spooler, is then sent to the printer at whatever speed it can handle. This method completely eliminates CPU time-sharing problems, and ensures rapid, accurate printing, while quickly freeing the computer for additional work.

An excellent hardware spooler is available from Compulink Corporation (see appendix). It is called the Sooper Spooler, and is equipped with 16 Kbytes of memory (expandable to 62 Kbytes) and an onboard Z80 CPU that controls the operation of the spooler and provides some very advanced printing features. The Sooper Spooler has a standard parallel port, which is compatible with the new Zenith/Heath parallel communications board, and a serial option is available that is switch-selectable for baud rates and handshaking protocols (see below, printer interfacing). The serial option is required with Z/H-89 and -90 systems unless a parallel communications board has been installed.

A user who produces a lot of printed material can save considerable time by using a hardware-type print spooler. When a large file is being printed, the computer can only transmit a portion of the file to a small (usually) buffer memory in the printer. When the printer empties the buffer, another small portion of data is sent from the computer. During this process the computer must sit idle for long periods, waiting for the printer to empty the buffer. However, a hardware spooler is like a very large buffer memory. The computer transmits a large portion of data (as much as the spooler memory will hold) at a very high rate, and the spooler sends it on to the printer at a slower rate. As soon as the computer finishes sending data to the spooler (this will only take a few seconds if the spooler memory is large enough to hold the entire data file), it is free for another task.

The Compulink Super Spooler (see appendix) is one of the most capable and sophisticated hardware spoolers available. It is controlled by its own Z80 microprocessor, and can provide advanced features like automatic pagination and space compression. A digital readout indicates how much of the internal memory is filled with data.

The Compulink Sooper Spooler is really a limited microcomputer system, capable of pre-processing the text material before it is sent to the printer. Some of the features supported by this device are:

Space compression: The spooler will convert a series of spaces in a text file (like left margin spaces, etc.) to a single byte, thereby reducing the amount of space required to store a file in spooler memory. These special byte strings are expanded to their original series of spaces upon transmission to the printer.

Pagination: A text file can be broken down into discrete pages of almost any size; page length, lines per page, and line length may be selected by instructions embedded in the text file.

Formatting: Automatic indenting of carryover lines and a programmable left margin can be specified by instructions in the text.

Headers and page numbers: A header (title) may be added to each page during printout. A feature allows automatic page numbering.

Each of these special features may be enabled or disabled by the user.

The Compulink Sooper Spooler is housed in a metal cabinet with a built-in power supply. The front panel has a numeric readout that displays how much of the buffer is full, and two push-buttons to activate the space-compression and pagination features. A reset button is also provided to clear the spooler memory and stop printing or for "scratching" a file sent in error. A built-in self-test can also be activated from the front panel. It will run a diagnostic test on internal memory and, if an error is detected, print a list of the suspected defective ICs.

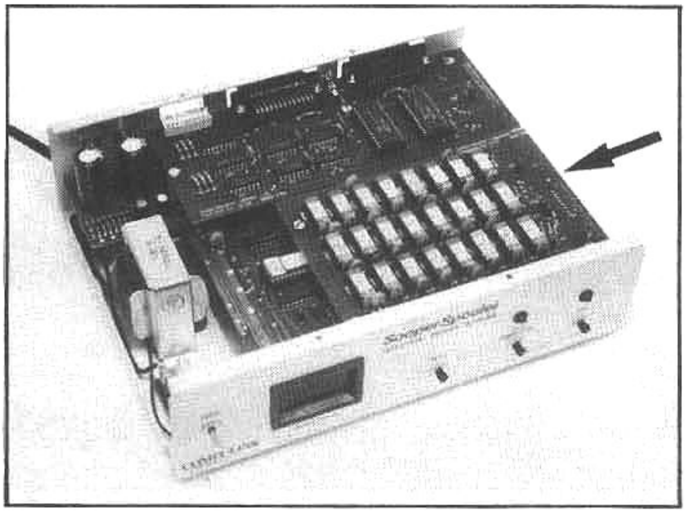
If a microcomputer is primarily used to create short documents, letters or memos, the Sooper Spooler may save the user only a few minutes during each printout, but if normal operation involves long manuscripts or contracts, the time saved by a spooler can be considerable. For example, a 55-Kbyte text file (roughly equivalent to 25 typewritten pages) can be sent to the Sooper Spooler through a 9600-baud serial I/O port in 45 seconds (a parallel connection will reduce this to only 16 seconds), and the computer will then be free for additional work, but depending on the speed of printer used, the spooling operation will continue for approximately 10 to 30 minutes before the entire file is printed.

The Sooper Spooler serial interface option is designed to use a female receptacle for connection to the computer and a male plug for connection to the printer. Special cables may be required to connect the spooler because most American-made printers (e.g., the Z/H-25) use a female-to-female cable as a connector between the printer and the spooler, while most Japanese-made printers (Epson MX-80, F-10, etc.) require a female-to-male cable. In all cases, a male-to-male cable is required to connect the spooler to the computer. Cables are available from most computer stores. If the serial interface is used, the spooler expects to see normal (not reverse) polarity RTS handshaking, and a modified LPH-24 device driver is recommended (see below, printer interfacing.)

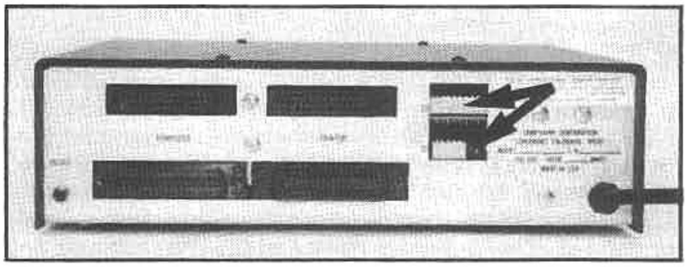
COMPOSITE VIDEO GENERATOR

Magnolia Microsystems offers a Composite Video Generator (CVG) board for the Z/H-89 and Z/H-19. This is a long name for an auxiliary circuit board that directs a copy of the current terminal display to a remote monitor. The greatest demand for this capability is in the educational area. By using this board to send the display signal to multiple monitors or a suitable large-screen display, many students can easily view the screen at once. Signals from the CVG can also be recorded on a video tape recorder, for later replay as an aid in diagnostic testing or to provide video documentation for ongoing operations.

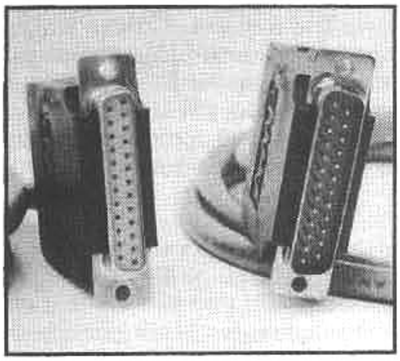
The CVG is designed to be used with a high-quality video monitor. Using this board with televisions (through an RF modulator) yields unsatisfactory results. A video monitor, like the Zenith ZVM-121, is required to display all 25 lines of 80 characters. Most inexpensive monitors and conventional televisions are not designed to "underscan" the display, rather they overscan or crop the image, preventing the full computer display from appearing on the remote screen. A 525-line monochrome (black & white) monitor, with a minimum bandwidth of 10 Mhz, incorporating underscan design, is recommended for the



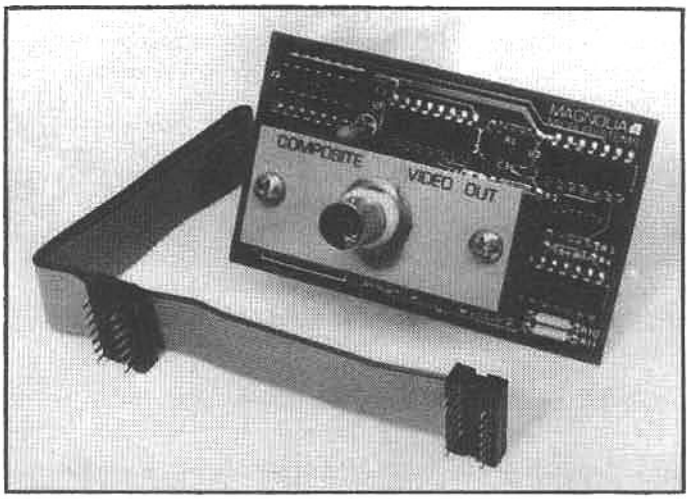
The Sooper Spooler is, in effect, a limited microcomputer dedicated to a single purpose—storing and transmitting text files to a printer. The basic unit can be upgraded with serial communications and additional memory (arrow).



The Sooper Spooler supports parallel communication (lower plug pair), and with an optional serial board, it will support RS-232 serial communications (the upper plug/socket pair). The arrow indicates the two DIP switches used to configure the system for various printers, etc.



Identifying the proper "sex" of serial cable connectors is important, and fortunately, rather easy. The plug on left is referred to as a female-socket connector and on the right is a male-pin connector. Obviously, proper connection is only possible when a female-socket is mated to a male-pin.



The composite-video generator board is useful in educational environments. It attaches to the rear of the Z/H-89; a ribbon cable connects it to the TLB. The BNC connector (arrow) outputs a video signal that will duplicate on a remote monitor the computer display.

highest reproduction quality.

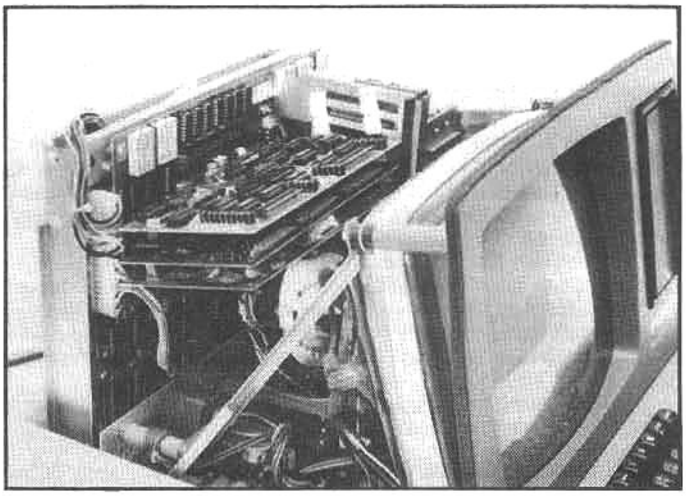
The installation of the CVG is relatively simple. No drilling is required if the computer has an unused IEEE-488 connector provision on the back panel (all late-model Z/H-89s are equipped with this provision). If the computer does not have a hole in the back panel for an IEEE-488 connector, a .50-inch hole must be drilled to mount the CVG. Normally the user removes the cover plate over the unused IEEE-488 access hole, and the CVG connector is installed through the access. The electrical connection to the CVG is completed by a 10.5-inch ribbon cable that is plugged into the terminal logic board (TLB). The TLB and one IC must be removed, and the plug provided on one end of the ribbon cable is installed in the empty IC socket. Once the TLB is reinstalled, and the plugs reconnected, the installation is complete.

The instructions make it clear that the user is responsible for connecting the CVG to the monitor, and the installation will be easier if the user has some knowledge of standard video connections. The user must also acquire BNC connectors and 75-ohm video-coaxial cable before the installation can be completed.

EXPANSION INTERFACE

Over the last few years independent manufacturers have produced a wide variety of expansion, upgrade, and special-purpose boards for the Z/H-89. There are sources for memory expansion boards, disk-controllers, speech-synthesizers, serial/parallel interfaces, digital-to-analog (D/A) converters, music and sound generators, graphic display systems, and more. The makers of these accessories have devised suitable means to adapt (interface) them to the Z/H-89 circuitry, however the Z/H-89 was not designed to readily accept accessory circuit boards. For most applications this is not a serious constraint, but for avid computerphiles this limitation is a source of frustration.

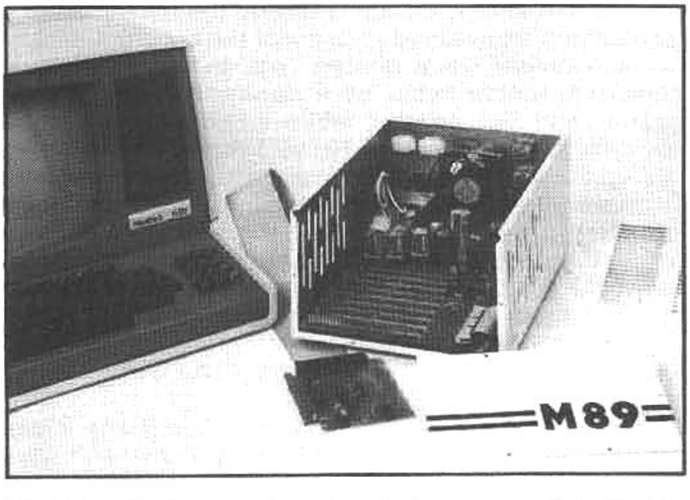
An expansion interface from Kres Engineering (see appendix) greatly extends the expansion potential of the Z/H-89. This four-layer circuit board plugs into the pins located on both sides of the CPU board, and provides expansion room for up to *seven* accessory circuit boards, all within the standard cabinet enclosure, and without any modifications to the CPU board. For added convenience the interface includes the 16-Kbyte RAM expansion (the memory chips are mounted in sockets on the expansion board), bringing the memory to a full 64K, and leaving all seven expansion slots free to accept accessories.



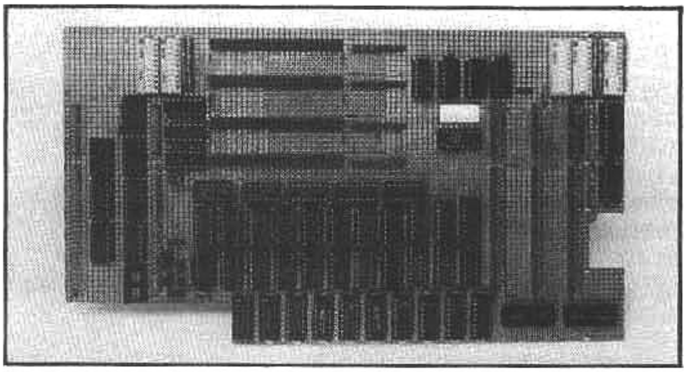
Kres Engineering (see appendix) manufactures an expansion interface board that piggybacks on the CPU board, allowing the addition of seven (instead of three) accessory circuit boards. It can utilize capabilities of the Z80 processor that are not used in the standard design.

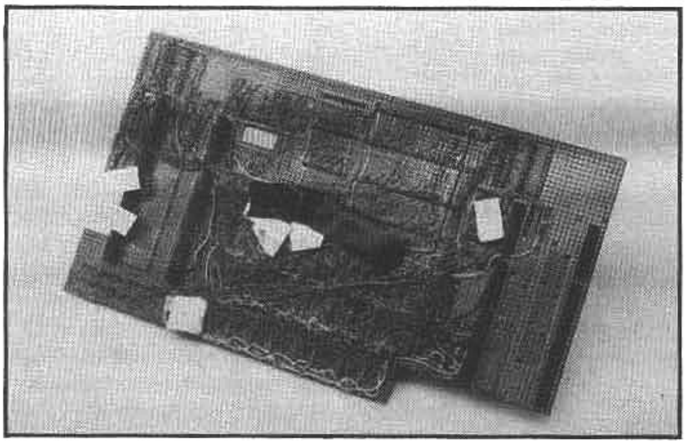
Another constraint on the Z/H-89 system is the limited number of addressable I/O ports (the CPU's connection to the outside world) supported by the standard circuitry. (This is the primary reason why only two types of disk drives may be used on the system at any one time.) The Kres interface not only supports all standard board protocols, but also opens the remaining port address lines, allowing full access to all I/O channels of the Z80 chip. Once this extended addressing is supported by auxiliary boards designed to work with the Kres interface, the limitation on interfacing accessory boards will be eliminated.

The support logic on the Kres interface is powerful enough to allow installation of *two I/O boards at the same port address.* This permits, for example, two serial inter-

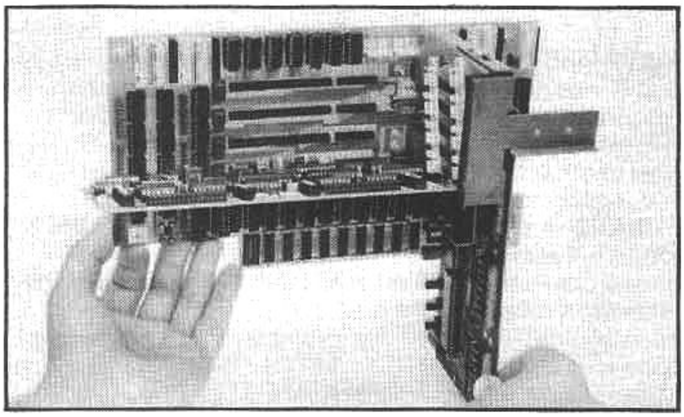


The Microflash expansion chassis (see appendix) extends the standard Z/H-89 buss lines with a four-foot ribbon cable. The chassis contains room for nine peripheral boards and a power supply. It can use relays for AC power control, and it has a wider-than-standard buss to "allow extra controlling signals."





After the complex electronics for the Kres interface were laid out, the first working assembly was built on a "prototyping" board. This is the original Kres design board; the final design has been extensively enhanced.



The production Kres interface is a four-layer circuit board. It will accept any of the several special-purpose boards that are currently marketed for the Z/H-89 system. In addition, Kres is now designing advanced expansion boards that will use many of the special features of their interface.

face boards to be used, and under software control, the expansion interface can "turn on" one board and "turn off" the other. With this provision two printers (modems, etc.) can be connected to the computer simultaneously, and the user can direct serial output to either one, at the same port address.

All of the special features of the Kres expansion interface can be selected by DIP switches located along the top of the board. A special card cage supports three circuit boards on the right side of the CPU board, and four are mounted horizontally in the top of the cabinet with their front edges tilted up to clear the CRT tube. The four horizontal boards have connection provisions for either the right- or left-side electrical signals, and specially designed Kres boards can access *both* the right and left connections *plus* the additional port address lines, *all at once* (prototyping boards are available). This flexible design promotes almost unlimited future expansion possibilities.

Kres also has several accessory boards under development, including:

A color graphics card using the TMS9918A
A music synthesis system
A system clock
D/A and A/D conversion boards

PRINTER INTERFACING

Printouts of text, data or program listings are important in most business or practical applications. This "hardcopy" is is often useful for general reference or for permanent archival storage of important records. And, connecting a printer to a Z/H-89 sounds like a relatively simple process; just plug it in and start printing! Unfortunately, in actual practice this is not as easy as it sounds.

Many experienced microcomputer users remember, with winces of pain, their first futile attempt to get a printer working. The unhappy events probably went like this: the neophyte plugged in the connecting cable (after a frustrating search for the appropriate one), turned on the printer and computer, commanded the computer to print something, and watched as nothing happened. Time passed (sometimes months) and finally—with some luck and numerous pleading telephone calls for help—the printer eventually worked, but the user's hair had turned decidedly gray. In the following sections we will discuss and offer solutions to the most frequently encountered printer interfacing problems.

There are two common ways for a computer to transmit data: serial and parallel. Serial communication is the most common method of linking a microcomputer and printer, particularly those of word processing quality, and until recently, serial interface boards were the only type available for the Z/H-89 system. We will primarily discuss serial interfacing techniques for the Z/H-89, and except for some software and hardware considerations, most of this information is applicable to the Z/H-100 system.

THE RS-232 CONNECTION

Communication between a computer and serial printer is made through an *RS-232C serial interface* connection and a cable that uses plugs and sockets called *DB-25 connectors*. These connectors can have as many as 25 pin connections and the cables can contain up to 25 wires. Most cables that are supplied with printers or purchased from computer dealers have fewer than the 25 connections; typically 9 or 13 connections are all that are required. For a Zenith/Heath computer to be properly connected to any printer, the following pins *must* have connections (the pin numbers are usually indicated on the DB-25 connector):

Pin #1—A frame ground connection
Pin #2—Data flowing from printer to computer
Pin #3—Data flowing from computer to printer
Pin #4—RTS (Request To Send) handshaking line
Pin #7—Signal ground
Pin #20—DTR (Data Terminal Ready) handshaking line

The following lines may be used by the printer and *may prevent proper operation if not present* (but are not re-

UNIVERSAL DEVICE DRIVER

With the HDOS operating system a short program handles file transfer from a storage area (e.g., floppy disk) to a system peripheral accessory (e.g., a printer). This special program is called a *device driver* and is considered part of the operating system. Because there is no "universal" interfacing and handshaking standard, Heath has established this convention to allow Zenith/Heath computers to be used with nearly any existing peripheral device. However, a suitable device driver must be available for a specific device, and *this device driver must reside on the disk as a separate system file* (where it can be accessed by the operating system) before the computer will communicate properly with the peripheral. If the user wishes to use a different peripheral device with the computer, a corresponding device driver must be "installed" in the operating system to match the new device.

This concept increases the flexibility of the operating system, but it has a built-in "snag." A Heath driver designed for a specific printer will generally *operate only that specific printer* (more or less) because the driver often sends special codes to *initialize* the printer (e.g., set line spacing, type style, etc.), and these codes vary from printer to printer. This inevitably leads to a situation where a user may buy a printer for which there is no suitable device driver. This is not uncommon. In fact, all of the printers in this chapter require a special or modified driver—except the Epson MX-80, for which MX80.DVD is available.

When Zenith/Heath released HDOS 2.0, they decided to supply one of their popular device drivers, LPH24.DVD, in source form. This means that it is provided in an easily readable and modifiable form (not in binary machine instructions). In this section we will guide the reader, step-by-step, through the modifications required to make a "universal device driver" (nearly!). This driver—called UNIVERSL.DVD—will run any serial printer with normal-polarity handshaking on the RTS line (pin #4). It is NOT designed to initialize the special features of any printer.

In the following steps we will assume that the user has only one disk drive, but if your system uses multiple drives, they can be utilized to simplify the procedure. We will provide periodic hints for multiple-drive users. They will be preceded by an "mds:" to indicate that these instructions apply to multiple-drive systems.

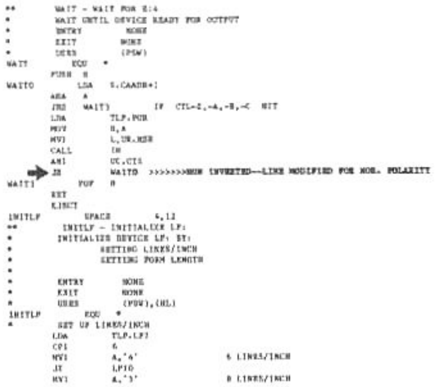
The first step is to "build" a bootable working disk under HDOS 2.0 with at least 225 sectors of free space (mds: not necessary to have a bootable disk, just init a data disk). Refer to chapter 5 and your HDOS manual for INIT and SYSGEN procedures.

HDOS 2.0 is supplied on three disks: HDOS OPERATING SYSTEM, DEVICE DRIVERS, and SOFTWARE TOOLS. You will need all three of these disks. Using ONECOPY (mds: use COPY or PIP), put the following files on the working disk—they may be on any one of the three HDOS 2.0 disks:

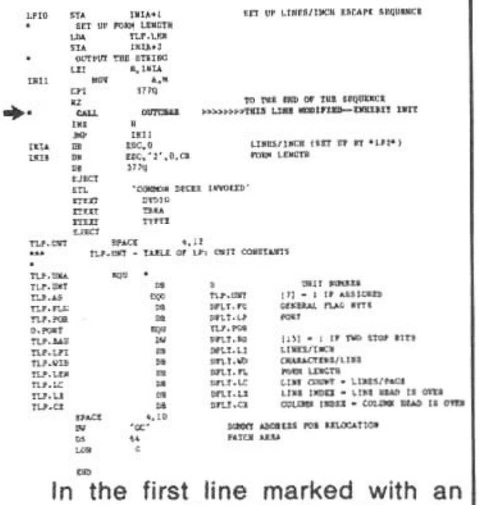
ASM.ABS
LPH24.ASM
U8250.ACM
U8251.ACM
HOSDEF.ACM
ASCII.ACM
DDDEF.ACM
MTR.ACM
HOSEQU.ACM
DIRDEF.ACM
ESINT.ACM
ESVAL.ACM
ECDEF.ACM
PICDEF.ACM
DEVDEF.ACM
DVDDEF.ACM
SETCAL.ACM
TYPTX.ACM
TBRA.ACM
DVDIO.ACM

After all the files are transferred to the working disk, you will need to transfer one additional program to the disk; a line or screen editor. A good choice is Video Scribe (see chapter 6), or you may use the line editor that comes with HDOS—EDIT.ABS (mds: the editor need not reside on the same disk). In the following section, we will only detail the modifications, the specific editing operations required to accomplish these modifications must be determined by the user (see the instructions for the editor).

The following listing is the last portion of the LPH24.ASM file. Using your editor locate the lines marked with an arrow.



```
**      WAIT - WAIT FOR E:4
*       WAIT UNTIL DEVICE READY FOR OUTPUT
*       ENTRY       NONE
*       EXIT        NONE
*       USES        (PSW)
WAIT        EQU   *
        PUSH  H
WAITO       LDA   S.CAADR+1
        ANA   A
        JNZ   WAIT1         IF  CTL-Z,-A,-B,-C  HIT
        LDA         TLP.POR
        MOV         H,A
        MVI         L,UR.MSR
        CALL        IN
        ANI         UC.CTS
        JZ          WAITO  >>>>>>>>NOW INVERTED--LINE MODIFIED FOR NOR. POLARITY
WAIT1       POP   H
        RET
        EJECT
INITLP      SPACE       4,12
**       INITLP - INITIALIZE LP:
*       INITIALIZE DEVICE LP: ST:
*               SETTING LINES/INCH
*               SETTING FORM LENGTH
*
*       ENTRY       NONE
*       EXIT        NONE
*       USES        (PSW),(HL)
INITLP      EQU   *
*       SET UP LINES/INCH
        LDA         TLP.LPI
        CPI         6
        MVI         A,'6'               6 LINES/INCH
        JZ          LPIO
        MVI         A,'3'               8 LINES/INCH
```



```
LPIO    STA         INIA+1                      SET UP LINES/INCH ESCAPE SEQUENCE
*        SET UP FORM LENGTH
        LDA         TLP.LEN
        STA         INIA+3
*        OUTPUT THE STRING
        LXI           H,INIA
INI1        MOV           A,M
        CPI          377Q
        RZ                                      TO THE END OF THE SEQUENCE
*        CALL         OUTCHAR    >>>>>>>>THIS LINE MODIFIED--INHIBIT INIT
        INX          H
        JMP          INI1
INIA    DB          ESC,0                       LINES/INCH (SET UP BY *LPI*)
INIB    DB          ESC,'2',0,CR                FORM LENGTH
        DB          377Q
        EJECT
        STL          'COMMON DECKS INVOKED'
        XTEXT          DVDIO
        XTEXT          TBRA
        XTEXT          TYPTX
        EJECT
TLP.CNT         SPACE       4,12
***        TLP.CNT - TABLE OF LP: UNIT CONSTANTS
*
TLP.UNA         EQU   *
TLP.UNT                 DB          0           UNIT NUMBER
TLP.AS                  EQU         TLP.UNT     [7] = 1 IF ASSIGNED
TLP.FLG                 DB          DFLT.FG     GENERAL FLAG BYTE
TLP.POR                 DB          DFLT.LP     PORT
D.PORT                  EQU         TLP.POR
TLP.BAU                 DW          DFLT.BD     [15] = 1 IF TWO STOP BITS
TLP.LPI                 DB          DFLT.LI     LINES/INCH
TLP.WID                 DB          DFLT.WD     CHARACTERS/LINE
TLP.LEN                 DB          DFLT.FL     FORM LENGTH
TLP.LC                  DB          DFLT.LC     LINE COUNT - LINES/PAGE
TLP.LI                  DB          DFLT.LI     LINE INDEX - LINE HEAD IS OVER
TLP.CI                  DB          DFLT.CI     COLUMN INDEX - COLUMN HEAD IS OVER
        SPACE       4,10
        DW          'OC'                DUMMY ADDRESS FOR RELOCATION
        DS          64                  PATCH AREA
        LON         C

        END
```

In the first line marked with an arrow, change the "JNZ" to "JZ" and in the second marked line add the asterisk at the beginning of the line. These two modifications will change the driver from RTS reverse to RTS normal polarity, and will prevent any special codes from going to the printer, allowing "universal" application.

After these two changes are made, you are ready to assemble the modified file into machine-readable code. The Heath assembler (ASM.ABS) will accomplish this. Type the following commands and after a couple of minutes, you will have UNIVERSL.DVD on your disk (mds: make sure that you use the appropriate drive specifier before the file names, i.e., SY1:, SY2:, etc):

ASM—then at the asterisk prompt type:

UNIVERSL.DVD=LPH24.ASM

If all goes well the assembler will display:

```
01752 STATEMENTS ASSEMBLED
XXXXX BYTES FREE
NO ERRORS DETECTED
```

If you do not get this response from the assembler, make sure that you have all the needed files (listed above) on the working disk and that modifications to the LPH24.DVD were performed as described—for additional help, refer to the assembler operating manual.

When assembly is complete, the file UNIVERSL.DVD will be on the working disk. To put the driver into use *it must reside on the disk in SY0: when the system is booted up, and it must have the name LP.DVD* (UNIVERSL.DVD must be renamed to LP.DVD) before HDOS will recognize the new driver.

UNIVERSL.DVD is programmable for baud rate using the SET program.

quired for most applications):

Pin #5—CTS (Clear To Send) reverse direction handshaking
Pin #6—DSR (Data Set Ready) reverse direction handshaking
Pin #8—DCD (Data Carrier Detect—RLSD) reverse handshaking

HANDSHAKING LINES

In the first group, pins #2 and #3 carry the data transmissions. The remaining pins have special purposes, the most important ones for this discussion are the *handshaking lines*. A handshaking line is a wire connection from the printer that tells the computer whether or not the printer is ready to receive data. It is used principally to stop data transmission from the computer when the printer buffer (temporary memory) is full, so that the printer can keep pace with data transmitted by the computer. When the buffer empties, the printer sends a signal down the handshaking line to tell the computer to transmit more data.

It may seem that, since the RS-232C standard connection exists, there should be no need to worry about details, but this is not the way it is. The manufacturers of computers and printers have not cooperated to establish a universal handshaking protocol, and several variations exist. Nonetheless, once the user understands the principles of handshaking, it is usually possible to interface any serial printer to the Z/H-89.

HANDSHAKING WITH HDOS

Depending on the specific situation, Zenith/Heath computers *may support handshaking on either RS-232C pin#4 (called the RTS line) or pin#20 (called the DTR line), and the signal to transmit data may be normal (high) or reverse (low).* When Heath released its first printer—the H-14 dot-matrix printer—the engineers chose to have HDOS "look" only at line #4 (RTS) to determine when the printer was ready to receive data. They also decided to use reverse (low) as a transmit signal and normal (high) as a stop signal.

Reverse-signal RTS handshaking is not common, and early non-Heath printers would not interface with Heath systems without some modification. Currently, most printers can be used with Zenith/Heath computers because both CP/M and HDOS (to a limited degree) will now support most handshaking conventions. Each operating system, however, has its own idiosyncrasies. The following discussion pertains to HDOS; the CP/M user may find it of interest, or the reader may wish to refer to the subsequent section on CP/M handshaking.

During the printing process the operating system monitors the handshaking line, sending data when the printer can accept it, and stopping transmission when the printer buffer is full. The portion of HDOS that handles this job is the device driver. This is a short utility program HDOS uses for communication with the outside world. There are separate drivers for each "device" that HDOS can communicate with: SY.DVD (system device driver) for communication with the system disks; DK.DVD for the secondary disk drives; AT.DVD for an alternate terminal; and LP.DVD for the line printer. It is the printer device driver—LP.DVD—that monitors the printer port and the handshaking line—pin #4 or #20—to determine if the printer is ready to accept more data.

Zenith/Heath printer drivers usually monitor pin#4 and transmit data when a *reverse* signal is detected. Most non-Z/H printers, however, are designed to handshake on *pin#20* and sends a *normal* polarity signal when more data is needed. In addition, many Zenith/Heath printer drivers will send escape-codes to the printer before the text file is transmitted. When these codes are received by the specific printer for which the device driver is designed, they are interpreted by the printer to set the length of each page, initialize the buffer, and establish various other programmable printing conventions. If these escape-codes are sent to another type of printer, the codes may cause it to malfunction or print spurious characters (portions of the improperly interpreted escape codes) at the top of the text.

RS-232C INTERFACE LINES

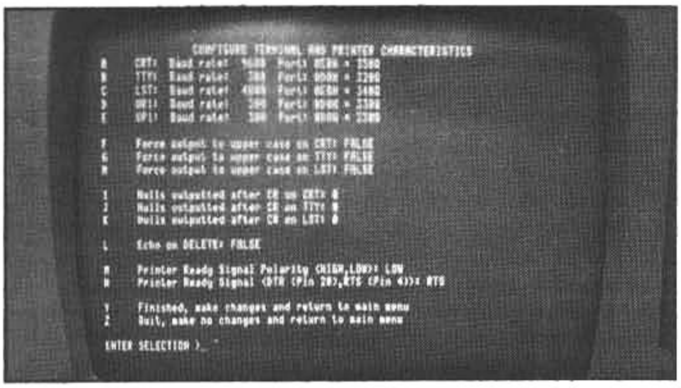
PIN	LINE NAME	LINE FUNCTION
1	FG	FRAME GROUND
2	TD	TRANSMITTED DATA
3	RD	RECEIVED DATA
4	RTS	REQUEST TO SEND
5	CTS	CLEAR TO SEND
6	DSR	DATA SET READY
7	SG	SIGNAL GROUND
8	DCD	DATA CARRIER DETECT
9		POSITIVE DC TEST VOLTAGE
10		NEGATIVE DC TEST VOLTAGE
11	QM	EQUALIZER MODE
12	DCD	SEC. DATA CARRIER DETECT
13	CTS	SEC. CLEAR TO SEND
14	TD	SEC. TRANSMITTED DATA
	NS	NEW SYNC
15	TC	TRANSMITTER CLOCK
16	RD	SEC. RECEIVED DATA
	DCT	DIVIDED CLOCK, TRANSMITTER
17	RC	RECEIVER CLOCK
18	DCR	DIVIDED CLOCK, RECEIVER
19	RTS	SEC. REQUEST TO SEND
20	DTR	DATA TERMINAL READY
21	SQ	SIGNAL QUALITY DETECT
22	RI	RING INDICATOR
23		DATA RATE SELECTOR
24	(TC)	EXT. TRANSMITTER CLOCK
25		BUSY

These are the common names for each of the handshaking lines in the standard RS-232C interface. However, equipment manufacturers may sometimes use these lines in "non-standard" ways.

We have provided a solution to this problem in the accompanying box: "How to Make a Universal Device Driver." If the reader follows these directions, the modified HDOS driver will work with most printers, although it will *not* send any escape-codes to initialize any special features that may be incorporated in "intelligent" models. It will drive *all* of the printers described in the following sections, and we will refer to it occasionally. If there is a better HDOS driver available for a specific printer, we will bring it to the reader's attention in the discussion, but this universal driver is a good "fallback" measure; it will work with most printers.

HANDSHAKING WITH CP/M

Handshaking with CP/M is somewhat more flexible than HDOS because the operating system *can be programmed* to monitor either line #4 or #20 and respond



```
              CONFIGURE TERMINAL AND PRINTER CHARACTERISTICS
A     CRT:  Baud rate:  9600  Port: 0E8H = 350Q
B     TTY:  Baud rate:   300  Port: 0D0H = 320Q
C     LST:  Baud rate:  4800  Port: 0E0H = 340Q
D     UR1:  Baud rate:   300  Port: 0D8H = 330Q
E     UP1:  Baud rate:   300  Port: 0D8H = 330Q

F     Force output to upper case on CRT: FALSE
G     Force output to upper case on TTY: FALSE
H     Force output to upper case on LST: FALSE

I     Nulls outputted after CR on CRT: 0
J     Nulls outputted after CR on TTY: 0
K     Nulls outputted after CR on LST: 0

L     Echo on DELETE: FALSE

M     Printer Ready Signal Polarity (HIGH,LOW): LOW
N     Printer Ready Signal (DTR (Pin 20),RTS (Pin 4)): RTS

Y     Finished, make changes and return to main menu
Z     Quit, make no changes and return to main menu

ENTER SELECTION >_
```

The CONFIGURE program will modify CP/M for various hardware environments. This portion of the program will select the appropriate printer characteristics, such as baud rate, handshaking line, polarity, etc.

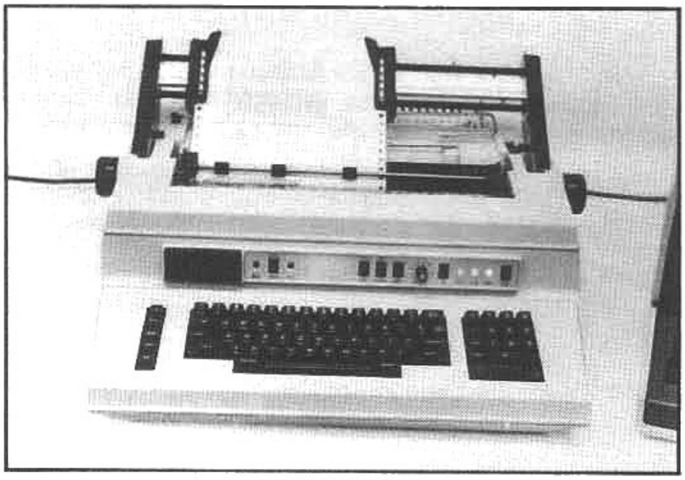
to either normal or reverse handshaking signals from the printer. Is it difficult? Nothing could be easier! The user runs the CONFIGUR program and makes selections from a menu to establish the desired handshaking protocol: RTS, DTR, normal, or reverse. (An accompanying photograph shows the menu from which the user chooses the options.) The program then reconfigures both the RAM and disk versions of CP/M (the disk is modified so that CP/M will recognize the new handshaking parameters whenever it is booted). This versatile feature is supported only on CP/M version 2.2.03 or later; users of earlier versions can contact local Zenith or Heath stores to obtain the latest update.

SERIAL INTERFACE BOARD INTERRUPT LEVEL

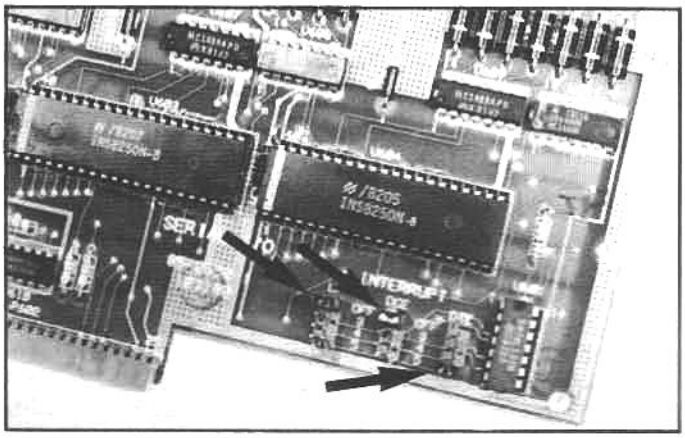
The serial interface board on the Z/H-89 system has a provision for changing the interrupt level—commands which tell the Z80 CPU to halt current processing and assume a new task. A jumper is mounted on the bottom-right edge of the card (when viewing the card so that the printing reads normally). The jumper for the LP port (port 340Q) should be set to "off," as recommended in the operating manual. However, some communications software requires the interrupt level for both the LP and the DTE ports to be set to level three or higher (REACH is an example, see the section on installing and using a modem in this chapter and the section on communications software in chapter 6). If this type of software is being used, the user should set the interrupt jumper to the appropriate "non-off" state (users report that this does not affect the normal operation of the computer; however, if there are frequent "strange happenings" during printer usage with this configuration, one remedy may be to set the interrupt jumper for the LP port to the "off" position.)

LETTER-QUALITY PRINTERS

There are two types of printers generally used with microcomputers: the "fully-formed" character printer or the dot-matrix printer. A fully-formed character printer operates very much like a typewriter, each letter is printed by striking an ink ribbon with a formed die that contains a reverse image of the letter. Instead of individual dies on the end of swinging arms (as is common in a typewriter), the typical computer printer uses a die wheel—often called a "daisywheel"—that is rotated in either direction, in precise increments, by a stepper motor. At the correct time, a hammer strikes the uppermost die on the spinning wheel, forcing it toward the ribbon to create a fully-formed character on the paper.



The Qume daisywheel printer is available in several models, rated at print speeds from 35 to 55 characters per second. This model incorporates a keyboard and is referred to as a KSR (keyboard send/receive model). It was designed to be a conventional teletype printer, but it can also be used as a computer printer.

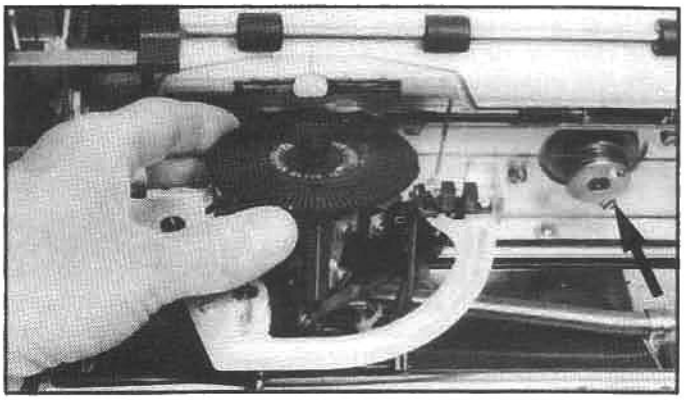


Normally, the interrupt jumpers on the serial interface should be set to "off." However, some program packages (particularly communications software) may require the interrup level on the LP and/or DTE ports to be set to "3" or higher (above: the LP and DCE are "off," the DTE is "5").

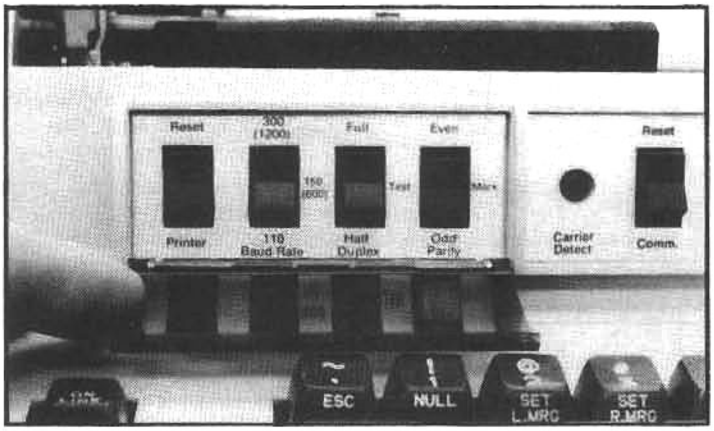
Text generated by fully-formed character printers looks very much like that created on a normal typewriter, as a result, these printers are often classified as "letter-quality" printers. They are favored for business letters and correspondence that must have a "formal" look, but because of the time interval required to rotate the print wheel to the proper position, fully-formed character printers cannot operate at speeds much higher than about 60-70 characters per second.

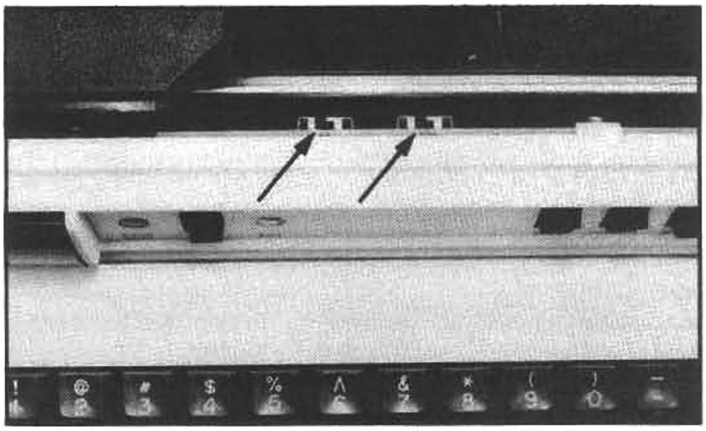
THE QUME SPRINT 5/55

Daisywheel printers have been around for several years, and until recently, there were only two major brand names: Diablo and Qume. The Qume company was started when one of the principal engineers left Diablo (the early giant in computer-driven printers) to establish a separate company and produce a "second generation" daisywheel printer. This printer is the Qume Sprint 5/55,



The daisywheel-type printer gets its designation from the characteristic shape of the multi-spoked printwheel. The wheel is rotated by a stepper motor to position the character dies. Generally, another stepper motor and some sort of drive mechanism (in this case a pulley-and-cable arrangement—arrow) controls lateral positioning of the printwheel assembly.





Several operating-mode switches are located on the left side of the front panel. These should be set for 1200 baud, full duplex, and mark parity. The two switches under the front cover (arrows) should be set to the left (1200 baud, and non-modem communications).

and when is was first introduced, it offered many advanced features that were not incorporated in other daisywheel printers. Many competitive designs are available today, but Qume has produced another advanced design, the Sprint 9, that uses compact electronics and a Gilmer belt to drive the print head, instead of the more complex (and costly) cable-and-pulley arrangement. Qume claims the new Sprint 9 is as capable as earlier printers, but it is substantially cheaper.

The Qume printer is capable of bi-directional printing (when used with appropriate software)—the print head alternately prints left-to-right and right-to-left (eliminating the time wasted by carriage returns). It uses sophisticated electronics to "look ahead" and determine optimum head maneuvering. For example: if the next line of text begins 10 spaces from the margin, the print head will not move to the margin and space in 10 columns; instead, as it prints one line, it will calculate where the next line begins and move the head directly to the proper starting position.

Some sophisticated word processors (e.g. WordStar) take advantage of the many features provided by this printer, including: bi-directional printing, bold-face characters, double-striking for emphasis, proportional spacing to produce an appearance similar to "typeset" copy, and more.

The Qume Sprint 5/55 uses handshaking with normal polarity, and the engineers at Qume have tied the connections for pins 4 and 20 together inside the printer, allowing either DTR or RTS lines to be used by the computer. To gain the proper software configuration with CP/M, use the CONFIGUR program and:

Set polarity to HIGH
Set handshaking to either DTR or RTS
Set LST: to 1200 baud, port 340Q

And with HDOS:

Use the "universal driver" described earlier (the LPH44.DVD supplied by Zenith/Heath for the Diablo printer will not work properly with the Qume)
Set the baud rate to 1200

For both HDOS and CP/M set the printer for (see the Qume instructions):

A baud rate of 1200
Mark parity
Full duplex
Non-modem communications (for the keyboard model)
Auto LF switch to OFF

The Sprint 5/55 daisywheel printer will print 55 characters per second, with very high quality, and is ideal for professional word processing. The Qume can also produce accounting and database printouts, etc., if speed is not an important criteria. Like any daisywheel printer, it isn't fast enough for anything more than modest accounting requirements.

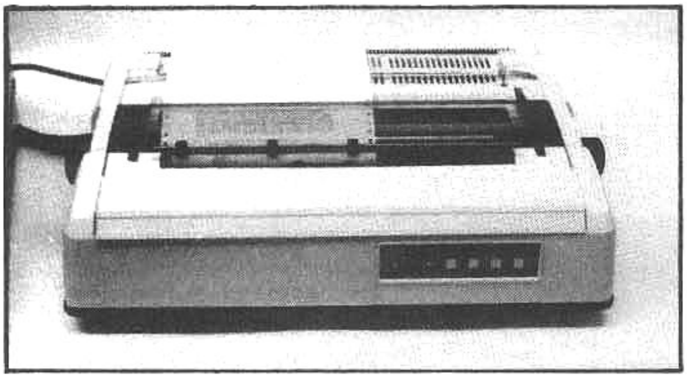
The Qume is a complex machine, it has nearly as many electronic components as the Z/H-89, and it is considered a very reliable printer. It is designed for extended operation, but a user who relies on a printer for day-to-day work should locate a local source of maintenance and repair. One national service network—The Sorbus Company—will perform onsite repairs on most major brands of daisywheel printers—including Qume and Diablo models. They can usually provide same-day service. Additional information about local Sorbus repair facilities can be obtained within the continental U.S. by calling 714-991-0243.

THE C. ITOH F-10 STARWRITER

The Qume is aimed at the business user, and the hobbyist usually cannot justify the relatively high cost of such a printer. But there are some daisywheel printers that are more modestly priced: the C. Itoh F-10 Starwriter is an excellent example.

The Japanese-made F-10 is available in two versions: the Starwriter, rated at 40 characters per second (cps); and the Printmaster, rated at 55cps. The 40cps Starwriter is the printer we will discuss in this section, however, the interfacing and overall characteristics are nearly identical for both models—except the Printmaster prints faster and costs somewhat more.

This printer uses an onboard Intel 8085 microprocessor to reduce the number of individual circuits, and to increase reliability. All of the electronics are contained on three circuit boards that can be readily removed for servicing. Despite its modest cost, the construction of the F-10 is, in a word, rugged. The chassis baseplate is die-cast aluminum, providing a lightweight and solid foundation for internal components. The stepper motors that control the print head and platen movements are heavy-duty designs and should provide virtually endless service. Operation is suspended (without losing data) when the last sheet of fan-fold paper is detected or the ribbon cartridge (carbon,



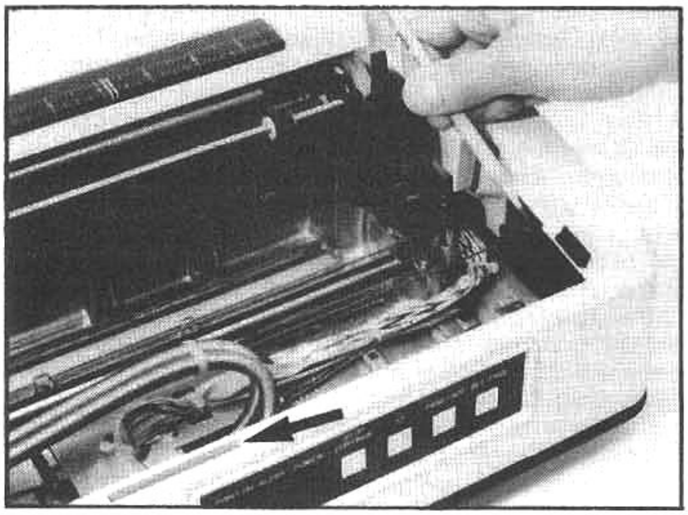
The C. Itoh F-10 Starwriter is currently one of the most popular daisywheel printer. Even though it is a low-priced letter-quality printer, it has a rugged design that utilizes state-of-the-art electronics to produce outstanding performance and tank-like reliability.

reel-type) needs to be changed. A self-test is included to ensure that the major mechanical and electronic components are operating properly.

These printers operate in a normal line mode or in a bi-directional mode. But unlike the Qume, the F-10 does not need a special bi-directional driver to take advantage of this feature. The F-10 has a look-ahead logic circuit (very much like the Qume), but the microprocessor and internal software perform the calculations necessary for efficient bi-directional printing. The F-10 can also accept a wide variety of escape-codes to set printing parameters through software control.

The F-10 is supplied with a standard pressure platen for single-sheet feeding, and it, like the Qume, has an unusually powerful and fast vertical paper feed. This feature permits the F-10 to complete a long printing job as fast as many printers that have higher ratings (cps). The feed mechanism is strong enough to easily pull fan-fold (continuous form) paper through the printer. However, it is difficult to keep the paper tracking straight through the platen feed, and if the user regularly sends more than 10-15 sheets of fan-fold paper through the printer, the tractor-feed accessory is recommended.

Serial communication with the F-10 requires normal polarity handshaking on either pins #4 (RTS) or #20 (DTR). The proper configuration for CP/M (using the CONFIGUR program) is:



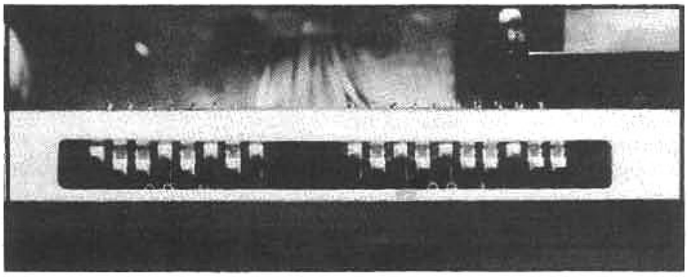
With the front cover of the F-10 removed, the cover-interlock is visible (hold this switch down to test the printer when the cover is removed). Access to the configuration switches is gained through the slot (arrow).

```
Set polarity to HIGH
Set handshaking to either DTR or RTS
Set LST: to 1200 baud, port 340Q
```

And with HDOS:

Use the "universal driver" described earlier (the LPH44.DVD supplied by Zenith/Heath for the Diablo printer will not work properly with the F-10). Use the HDOS SET utility to set the baud rate to 1200

The F-10 has two DIP (Dual-Inline-Package) switches just inside the top cover. These switches (SW-40 and SW-41) control the interface and print characteristics. The following switch settings will allow the F-10 to work with both CP/M and HDOS. Refer to the F-10 operating manual for other variations.



The F-10 configuration switches set the baud rate, character length, type style, etc. Though the configuration options are detailed in the operator's manual, the settings detailed in the text will help the beginner get the printer into conventional operation with a minimum of fuss.

```
Set switch 40:
  #1—Closed: baud rate to 1200
  #2—Open: baud rate to 1200
  #3—Open: character length to 8 bits
  #4—Closed: no parity
  #5—Open: no parity
  #6—Closed: 1 stop bit
  #7—Open: X-ON/X-OFF protocol
  #8—Unused: sets higher baud rate with Printmaster model
Set switch 41, #1 though #10 as follows:
  #1—Closed: Line communication
  #2—Open: CR at end of line
  #3—Closed: Auto CR space on
  #4—Open: Elite type (closed for Pica type)
  #5—Closed: proportional space off
  #6—Open: form length, see F-10 manual
  #7—Open: form length, see F-10 manual
  #8—Closed: form length, see F-10 manual
  #9—Open: form length, see F-10 manual
  #10—Open: hammer strike normal (closed for heavy)
```

These switch setting should get the F-10 up and running. A close look at the operator's manual will provide additional information on each of these settings, allowing the user to modify the printer to suit specific requirements.

All models of the F-10 printer and high-quality C. Itoh dot-matrix printers are available from Leading Edge Products, the exclusive distributor for C. Itoh products in the U.S. (see appendix).

DOT-MATRIX PRINTERS

There are many applications where the quality, cost, and relatively slow speed of a fully-formed character printer are inappropriate. Printouts of accounting reports like accounts-receivable listings and general-ledger reports are often used only for internal reference and do not require "typewriter-like" appearance. Likewise, the most important consideration when printing invoices and monthly statements is (generally) how fast and inexpensively they can be produced.

Compared to letter-quality printers, dot-matrix printers are usually faster and cheaper. They create printed characters from an array of dots produced by slender wires that are driven toward the ribbon and paper by electromagnets. High-speed logic circuits control which wires are driven forward to create each printed number, letter, or symbol.

The speed of the electronics, the quality and response of the printhead, and the capability of the stepper motors determine the maximum printing speed, but even relatively slow dot-matrix printers are faster than most daisywheel printers. Dot-matrix printers range in speed from 50cps to 600cps, and specialized industrial models can print at even higher rates. (Though not an impact printer, the Siemens Company manufacturers a dot-matrix laser printer capable of over 50,000cps.)

THE Z/H-25 PRINTER

Heath's first printer—the H-14—was designed for non-continuous hobbyist use, and it was not suited to the needs of a small business. But Zenith and Heath now offer a second-generation dot-matrix printer—the Z/H-25—that can handle commercial or small-business applications.

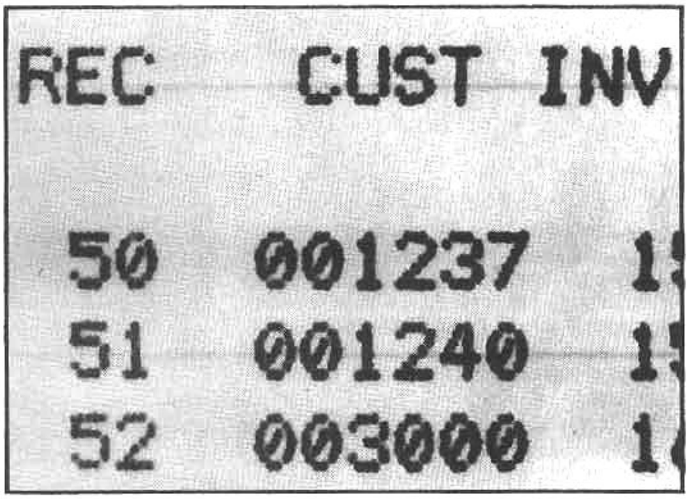
The Z/H-25 is hefty in both size and construction, and suitable for continuous-duty operation. It is reasonably priced—considering its capabilities—and it is an outstanding buy in kit form (see chapter 10). An onboard Z80 microprocessor running a ROM-based program controls the many sophisticated printing functions available from the Z/H-25. The CPU board also has two empty ROM sockets, and since the printer is controlled by firmware, future upgrades (e.g., special fonts, foreign character sets, unique graphics, etc.) can be incorporated simply by modifying or adding to the internal programming.

The Z/H-25 produces characters from a 9x9 matrix at a maximum speed of 150cps. Bi-directional printhead positioning is optimized with sophisticated logic. Some of the important additional features are: rapid paper feeding; the capacity to handle six-part forms; paper-out and jammed-paper sensors to stop printing without losing data; programmable vertical and horizontal tab stops; and 10, 12, 13.2, or 16.5 pitch selection (characters per inch).

There is probably no easier printer to interface with *any* computer. The Z/H-25 will support the 20ma current loop (the original teletype standard) and a wide variety of RS-232C handshaking protocols. The handshaking parameters are set with switches located at the rear of the printer. These switches select the baud rate, form length, character pitch, and a variable skip-over-perforations option. All of these features, and more, are also selectable by escape-codes that may be included in a printout file.

A CP/M user should select the following handshaking with the CONFIGUR program:

```
LST: device at 4800 baud, port 340Q
RTS handshaking
HIGH polarity (normal)
```



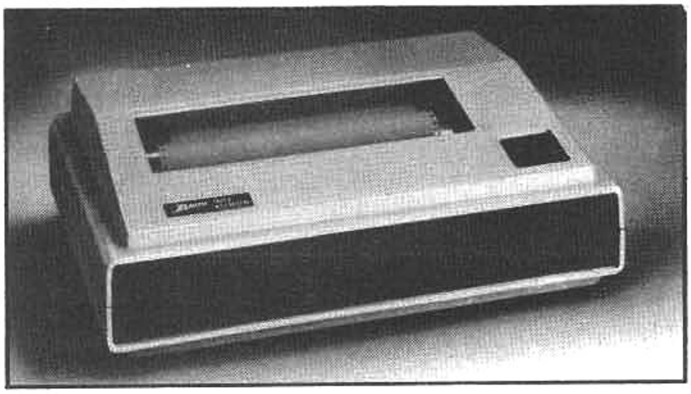
```
REC    CUST    INV
50   ØØ1237    1
51   ØØ124Ø    1
52   ØØ3ØØØ    1
```

This is an enlargement of dot-matrix characters produced by the Z/H-25 printer. Each character is formed by seven wire hammers driven by solenoids. As the printhead moves across the paper, selected wires are driven into the ribbon; transferring ink to the paper.

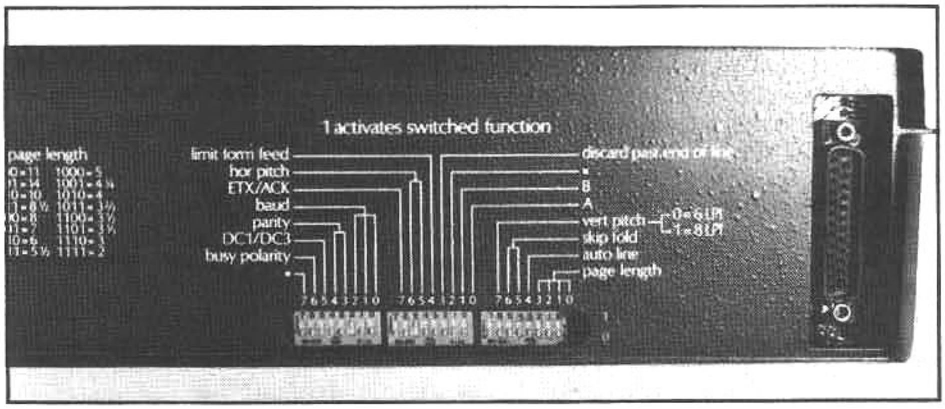
Zenith/Heath supplies HDOS users with a special driver called LPH25.DVD. This device driver supports many of the special functions of the printer, however, if unavailable, the "universal driver" (described above) can be used.

Both CP/M and HDOS require the switches at the rear of the printer to be properly set. One configuration that is known to work is:

```
From right to left (looking at the rear panel)
 Right-hand switch
   #0—OFF: Page length to 11 inches
   #1—OFF: Page lenght to 11 inches
   #2—OFF: Page length to 11 inches
   #3—OFF: Page length to 11 inches
   #4—OFF: Automatic line feed
   #5—OFF: Skip fold disabled
   #6—OFF: Skip fold disabled
   #7—OFF: Vertical pitch at 6 lines per inch
 Middle switch
   #0—OFF: Not used
   #1—OFF: Not used
   #2—OFF: Not used
   #3—OFF: Discard past end of line disabled
   #4—OFF: Limit FF
   #5—ON: Horizontal pitch to 12cpi
   #6—OFF: Horizontal pitch to 12cpi
   #7—OFF: ETK/ACK handshaking
```



The Z/H-25 printer is a rugged, reliable, and handsome high-speed printer designed for the business environment. It can withstand continuous-duty operation, and it has a quad-tractor drive that produces accurate and extremely fast form feeds.



The rear panel of the Z/H-25 printer provides access to 24 switches that set baud rate, form length, handshaking parameters, etc. It can be readily configured for virtually any standard handshaking protocol, making it a versatile and easy printer to interface with any computer.

Left-hand switch

#0—OFF: 4800 baud

#1—ON: 4800 baud

#2—ON: 4800 baud

#3—OFF: No parity

#4—OFF: No parity

#5—OFF: DC1/DC3 handshaking disabled

#6—OFF: Busy line (high)

#7—OFF: Not used

A SPECIAL Z/H-25 DRIVER FOR HDOS

Z/H-25 users operating under HDOS may be interested in an excellent specialty device driver available from Jim Teixeria's SoftShop. The H25.DVD is a multi-part driver that allows the user to set up to eight unique configurations for the Z/H-25 at one time (with one driver). For example: by calling LP1:, you can produce double-strike printing for letters; LP2: may provide pagination and multiple copies; LP3: double-space printing; etc. Each of the eight drivers can be individually programmed to suit your requirements. The programmable features are:

PORT: between 300 and 377Q

BAUD RATE: 110 to 19200

FORM: lines per form, 0 to 255

PAGE: lines per page, 0 to 255

CPI: characters per inch, 10, 12, 13.2, 16.5

LPI: lines per inch, 6 or 8

RMARG: right margin, 20 to 220

LMARG: left margin, 0 to 220

PAG: printout is paginated with titles, if desired

MUC: produce multiple copies, if desired

OVP: overprint for enhanced appearance

TAB: allows transmission of TAB characters

DBL: double line spacing

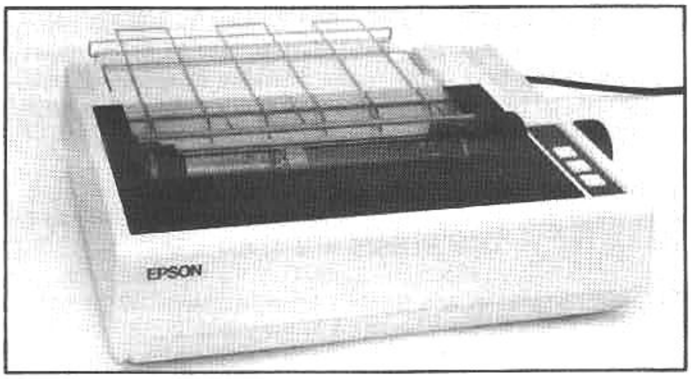
FFF: send a final form feed

ASC: send an ASCII ESCAPE string

The H25.DVD uses DC1/DC3 handshaking, and the DC1/DC3 switch at the rear of the printer must be set to "1" for this driver to function properly. The driver is supplied with over 50 Kbytes of thoroughly documented source code, making custom modifications easy (for a qualified programmer). Anyone with a Z/H-25 printer running under HDOS should consider this unique driver. It is priced at $29.95. Contact Jim Teixeria at the Soft Shop (see appendix) for more information.

THE EPSON MX-80 PRINTER

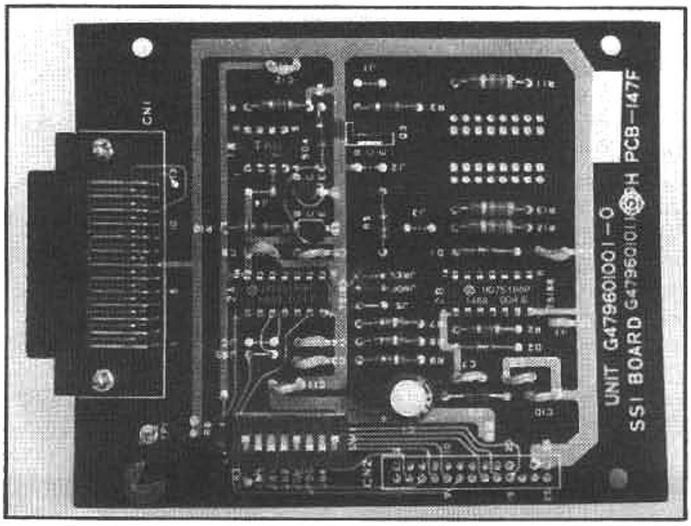
The MX-80 is perhaps the most popular dot-matrix printer in the U.S. It was the first printer to fill the need for a high-quality, small-systems printer that cost less than $700. When the Japanese-made MX-80 was first introduced, comparable printers were selling for well over $1000 (though the low-cost printer market is currently more competitive), and it became an instant hit. In addition to its attractive price, it is small, relatively fast, and has a tractor-feed that can be adjusted up to 9.5 inches wide to accept fan-fold paper or continuous-form mailing labels. Following their initial success, Epson produced two additional models: the MX-80FT, which has a standard platen feed (to accept single sheets) but can be fitted with an optional tractor-feed accessory; and the MX-100FT model, which is functionally identical to the comparable MX-80FT model but has a 13-inch wide carriage.



The Epson MX-80 is unquestionably one of the most popular printers sold in the U.S. (and probably the world). This popularity is easy to explain. The Epson is inexpensive, but it has gained a reputation for rugged reliability. It is a compact and attractive design that produces high-quality print, and it has several advanced features not generally found on comparable printers.

These Epson printers have become a standard by which other low-cost dot-matrix printers are judged. They operate at 80cps, with bi-directional head movement optimized by microprocessor logic. The print quality is exceptional, and different print fonts—including condensed and extended styles—can be selected by sending the appropriate initializing codes. Two of the styles—double-strike and emphasized—produce results suitable for most correspondence or business purposes.

Epson printers are supplied with a standard parallel interface, however, Epson offers a serial interface board—Epson part 8141—as an option, and when this is installed in the Epson, communication can be established with the Zenith/Heath serial port. The serial interface has several jumper-selectable handshaking options. Normal or reverse polarity is supported; normal is the factory setting. If you do not alter any of the factory jumper positions, the serial board will function as described below. The serial board also has an 8-segment DIP switch to select operational parameters. Set the switches as follows:



This is the Epson serial interface board (PN 8141). The jumper selections can be left "as is" (see text). The DIP switch selects baud rate, parity, etc. (for operation as described in the text, set them as shown here). The top portion of the enclosure must be removed to install the serial board.

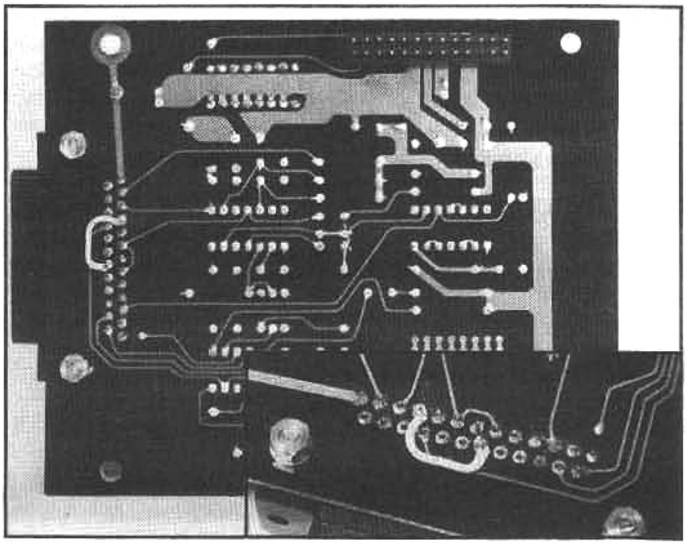
#1—ON: baud rate to 4800
#2—OFF: 7-bit word length
#3—ON: baud rate to 4800
#4—OFF: baud rate to 4800
#5—either ON or OFF: not used
#6—OFF: no parity
#7—OFF: baud rate to 4800
#8—either ON or OFF: not used

Note: There are two other DIP switches on the main circuit board (below the serial board position). These switches need not be altered from their factory setting.

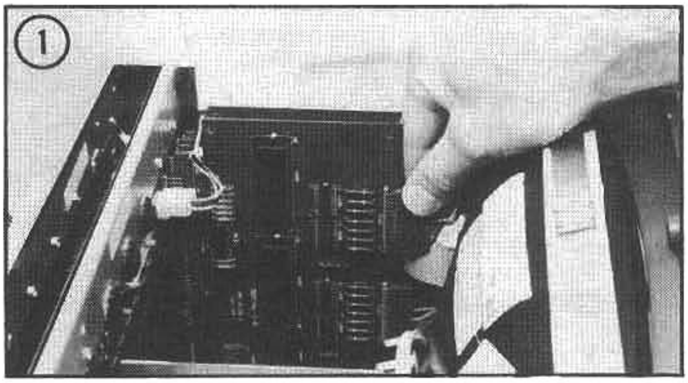
For operation with CP/M use CONFIGUR to set the following:

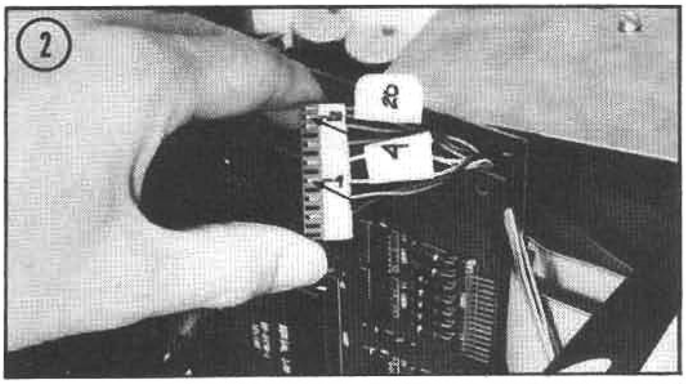
LST: to 4800 baud, port 340Q
Handshaking polarity to HIGH
Handshaking line to DTR (pin #20)

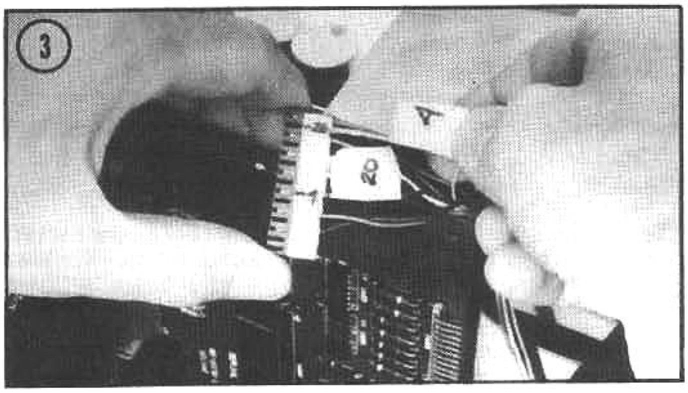
For operation with HDOS (2.0), use the Zenith/Heath LPMX80.DVD driver. It supports several special features of the Epson printer, and is designed to work with the standard MX-80 switch settings. If this driver is not available, use the "universal driver" described above, however this driver expects RTS (pin #4) handshaking (the Epson uses pin #20). To solve this problem, jump the handshaking lines on the Epson serial board (see illustration). The handshaking line swap (pin #4 and #20) can also be accomplished at the serial-board plug (see accompanying photos). There is also a third way to accomplish the same effect: a special computer-to-printer cable that has the handshaking lines swapped so that the #20 pin on the



For proper operation with HDOS 1.6 (or when the special LPMX80.DVD is not used with HDOS 2.0) the printer handshaking line—pin 20—must be connected to the RTS handshaking line—pin 4—on the Z/H-89. This is accomplished by installing a jumper as shown above. This will not affect normal operation with other drivers or with CP/M.







Instead of installing a jumper on the Epson serial board, the handshaking lines may be swapped at the serial board plug in the Z/H-89. The upper plug on the serial-interface board (photo 1) is connected to port 340Q (the normal printer port). Remove the plug and label the wires as shown (photo 2). By gently pushing in the metal retaining clip (arrow), the wires can be removed from the plug. Switch the two wires marked 4 and 20 (photo 3). Reinstall the plug on the serial board (the brown wire should be positioned up). This swaps the DTR (pin 20) and RTS (pin 4) handshaking lines, which will provide suitable operation with all HDOS device drivers that monitor pin 4 for handshaking signals.

printer end is connected to the #4 pin on the computer end. For both CP/M and HDOS operation, an interface cable with male plugs on each end is required to connect the Epson printer to the Z/H-89 computer (the Z/H-100 requires the same cable, but the software requirements are different.

AN EXTENDED BUFFER FOR THE EPSON

A hobbyist may not need the features of an advanced print spooler (like the Compulink Sooper Spooler), but a simple, self-contained modestly-priced spooler with about 8-16 Kbytes of memory is perfect. This is exactly what Practical Peripherals Inc. (see appendix) offers for the Epson.

The Practical Peripherals MBS-8K is an 8-Kbyte "buffer board" that contains a serial interface and enough storage for about 8000 characters (roughly 4-5 typewritten pages). The buffer board mounts inside the MX-80, in place of the Epson 8141 serial board, and when configured properly, it will interface similarly to the Epson board.

First, set the MBS-8K jumpers to the following specifications:

Switch S1:
 B—JUMPED: Normal polarity handshaking
 B(minus)—OPEN: Reverse polarity
 X—OPEN: X-on/X-off software handshaking
 W—JUMPED: Hardware handshaking
Switch S2:
 WL—OPEN: Word length to 8-bit
 EP—OPEN: Not used
 SP—JUMPED: One stop bit
 PI—OPEN: Parity disabled
Switch S3:
 Pins #1 thru #6—OPEN: baud rate to 19200
 Pin #7—JUMPED: baud rate to 19200

For CP/M operation use CONFIGUR to select the following:

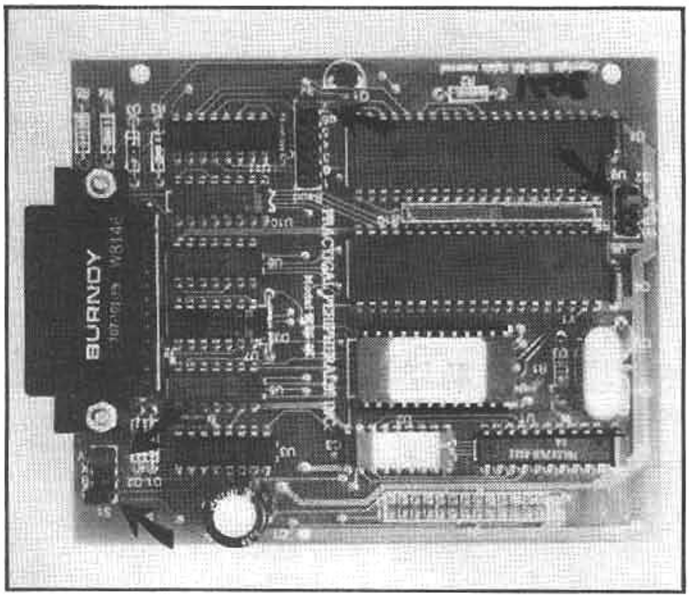
LST: to 19200 baud, port 340Q
Handshaking polarity to HIGH
Handshaking to DTR (pin #20)

For HDOS operation, use the "universal driver" and use the HDOS SET utility to set the baud rate to 19200. Since the buffer board is designed to handshake on the same line as the Epson board (DTR, line #20), a small modification must be made to the MBS-8K board before it will work with the universal driver. There is a solder pad on the bottom side of the board (non-component side) that must be connected. Directly under the DB-25 plug there is a small, round pad with a slit through the center (marked "RTS"). Scrape the varnish off this pad, and join the two halves of the pad together with solder (see accompanying photo). This will tie pin #4 to #20.

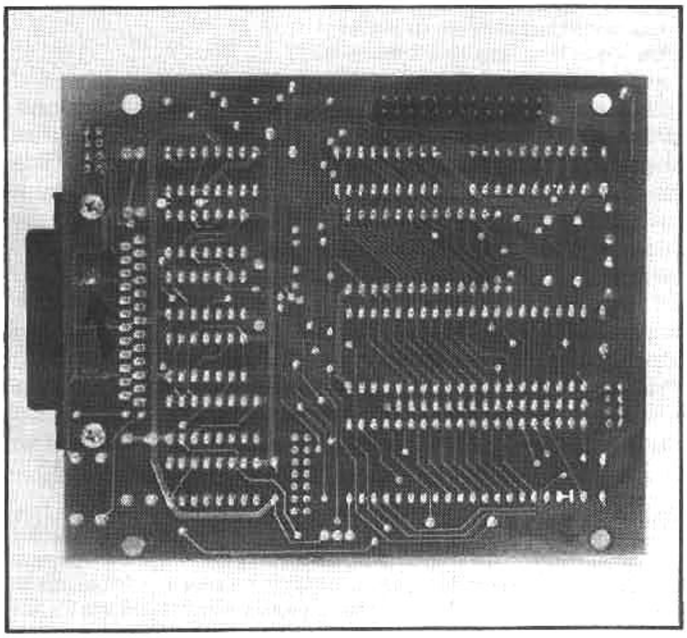
An additional note for HDOS users: do not use the LPMX80.DVD device driver with the MBS-8K board. This driver expects to see the exact hardware environment of the Epson, and this is not found on the Practical Peripherals board—as a result data will not be transmitted. However, Practical Peripherals claims to be developing a new version of this buffer board which will provide additional memory capacity and completely "Epson-like" interfacing. (Apparently the MBS-8K was designed to be used with Apple computers, and the interface features are not identical to the original Epson serial board).

USING THE EPSON WITH WORDSTAR

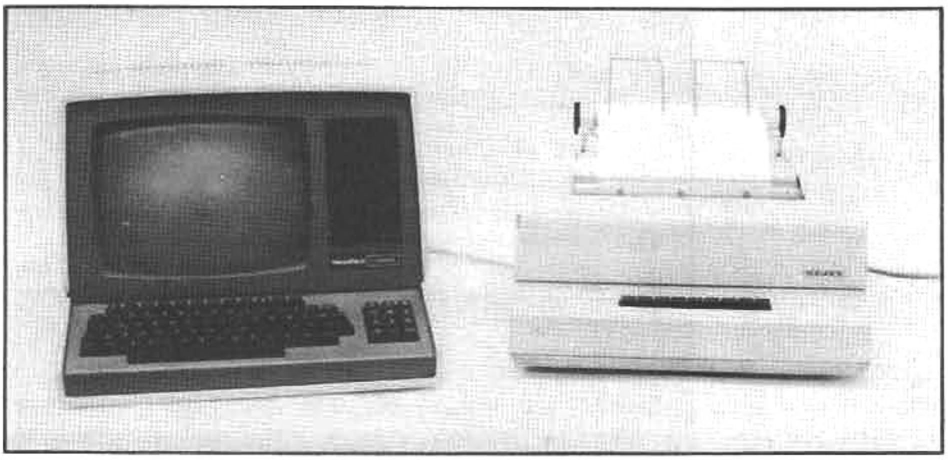
The popular WordStar word processing program (see chapter 6) can control many special features of the Epson MX-80, through an enhancement package called EP/Star. This program, developed by Patrick McNally, modifies WordStar, and allows commands within the text file to call up the special features in the MX-80. In addition, if the Graftrax option is installed (an Epson upgrade that allows the MX-80 to print graphics characters), EP/Star will gen-



The Practical Peripherals MBS-8K accessory board for the Epson MX-80 is a serial interface and extended buffer that mounts in place of the Epson serial board. It operates like a mini-spooler, holding incoming data in the 8-Kbyte memory, and freeing the computer for additional tasking. The jumper configurations shown are properly set for operation with either HDOS or CP/M at 19,200 baud.



For the MBS-8K to operate properly with HDOS 2.0 (using the universal device driver), the RTS line—pin 4—must be strapped to the normal DTR handshaking line—pin 20. This is easily accomplished by soldering the small pad (arrow) marked "RTS" on the bottom of the board.



The Siemens ink-jet printer is an innovative bit of engineering that produces high-quality dot-matrix printing at the remarkable speed of 270 characters per second, and with almost no noise. The printer "sprays" ink at the paper (under microprocessor control) to produce all of the standard ASCII characters plus several unique graphics characters.

erate an italic type font in either standard, bold, or wide versions.

EP/Star costs $19.95. For further information on EP/Star and other unique CP/M program contact Patrick McNally (see appendix).

THE SIEMENS 2712 INK-JET PRINTER

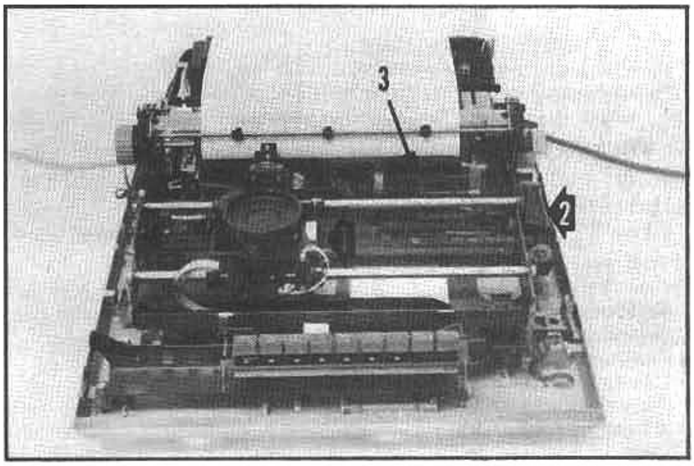
Daisywheel and dot-matrix printers are noisy! In a crowded business environment the staccato bark of an impact printer can become quite annoying, and in the home environment it can aggravate an otherwise placid family relationship. There is, however, an alternative: the Siemens 2712 ink-jet printer. The operation of this unique printer is, to say the least, impressive. It produces virtually no noise, other than a low-level hum from the print-head stepper motor, and the print speed is rated at a very quick 270cps.

The 2712 generates characters within a 9x12 dot matrix that is formed by 12 tiny ink-spray jets mounted on a moving printhead. A droplet of ink forms each dot in the character matrix, and under the control of the built-in microprocessor, the droplets are "fired" at the paper by the ink jets. There is a tubular piezoelectric transducer in each of the 12 passages feeding ink to the jet nozzles, and when one of the transducers is stimulated with an electrical charge, it produces a shockwave that dispels an ink droplet (about .004-inch in diameter) from the nozzle. Each nozzle can shoot up to 2500 ink droplets per second. The piezoelectric crystal is the only "moving" part in the system, and the print-head life is claimed to be more than 10 billion (not million) characters.

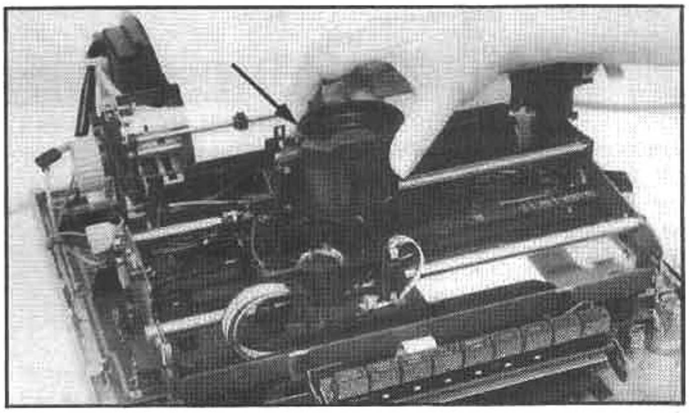
The ink jets do not clog with ink or dirt because the ink never dries and dirt cannot get into the ink supply. The ink is a special formula that includes a hygroscopic agent similar to automotive antifreeze. The ink does not dry immediately, but it is smearproof because it is absorbed by the paper. It is supplied in a sealed plastic bladder, which is replaced as a unit. Each bladder contains enough ink for about ten-million characters, and when the ink supply is low, an indicator flashes on the front panel before the print quality degrades.

Although the type is not as dense or dark in appearance as that of an impact printer with a fresh ribbon, it is easy to read and just as permanent. The best print quality is obtained by using Siemens' own brand of paper, however, the 2712 will print on conventional paper with only a slight loss of quality due to the dispersion of the ink by the larger "grain" of the paper.

Interfacing the 2712 is fairly straightforward. The printer contains a parallel port, an RS-232C serial connection (discussed here), and a 20ma current-loop (teletype) interface. This printer was designed for the "world market," and rather than force a compromise by choosing one serial handshaking convention, Siemens put the handshaking line on pin #25, a pin that is rarely used. The accompanying photos show the two wires that must be swapped in the plug on the Siemens printer logic-control board. This modification moves the handshaking line to pin #4 (RTS), which is compatible with Zenith/Heath computers. The modification is easy to do, since the 2712 provides easy access to the logic-control board. The logic board also contains several DIP jumpers. They should be configured as follows:



The ink-jet assembly (arrow 1) of the Siemens printer is moved with a stepper motor (arrow 2). The platen is also moved by a stepper motor (arrow 3). Both are controlled by microprocessor-based logic mounted below the mechanical print mechanism.



The ink-jet nozzles receive ink from a reservoir (arrow) that is sealed to protect the ink free from contamination. Ink is supplied to the print head through a small needle that punctures a rubber bladder within the reservoir housing.

Strapping plug W28:
#1—Jumped: form length 66
#2—Jumped: CR only with CR & LF
#3—Jumped: No CR with FF
#4—Jumped: No CR with VT
#5—Not used
#6—Jumped: Elite type
#7—Open: Elite type
#8—No control character printing
Strapping plug W29:
#1—Jumped: No parity
#2—Jumped: No parity
#3—Jumped: 9600 baud
#4—Jumped: 9600 baud
#5—Jumped: 9600 baud
#6—Jumped: select handshaking convention
#7—Jumped: select handshaking convention
#8—Open: Xon/Xoff disabled

For CP/M operation use CONFIGUR and select:

LST: to 9600 baud, port 340Q
Handshaking polarity to HIGH
Handshaking line to RTS (pin #4)

For HDOS operation, use the "universal driver" discussed earlier, and use the HDOS SET utility to set the baud rate to 9600.

The Siemens ink-jet printer has many additional features, including:

Built-in self test modes
Very fast form feeds
Optimized logic-seeking head movement
Downloadable character sets
Proportional spacing
A full graphics mode
Vertical and horizontal tabs
10, 12, and 16.5 characters per inch (pitch)
Built-in foreign language character sets

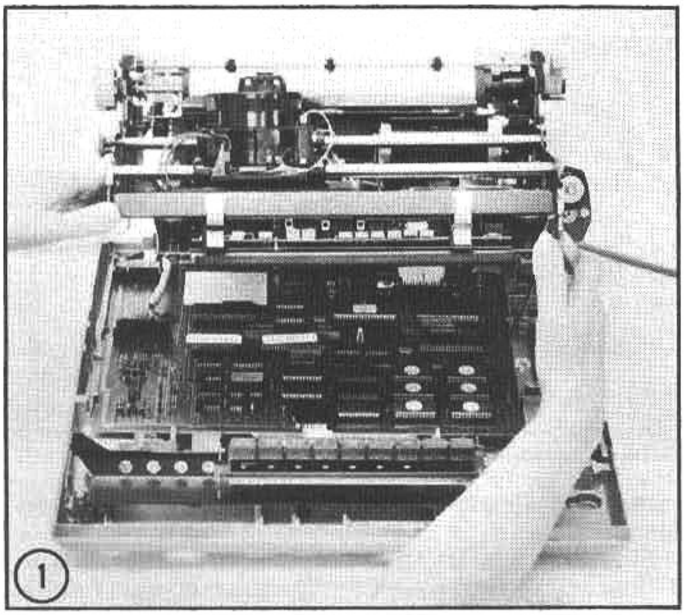
The 2712 can only be used with single-weight paper; since nothing strikes the paper with a measurable impact, no carbon copies can be made during printing. It also has a somewhat high price. Depending on the configuration—8.5- to 15.75-inch widths, with or without tractor feeds—the price ranges from about $3200 to $3650. For further information contact the Siemens Corporation (see appendix).

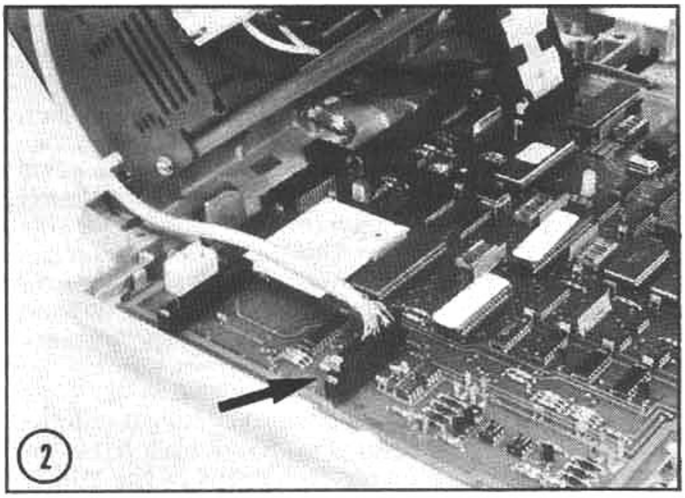
SWITCHING RS-232C SERIAL PRINTERS

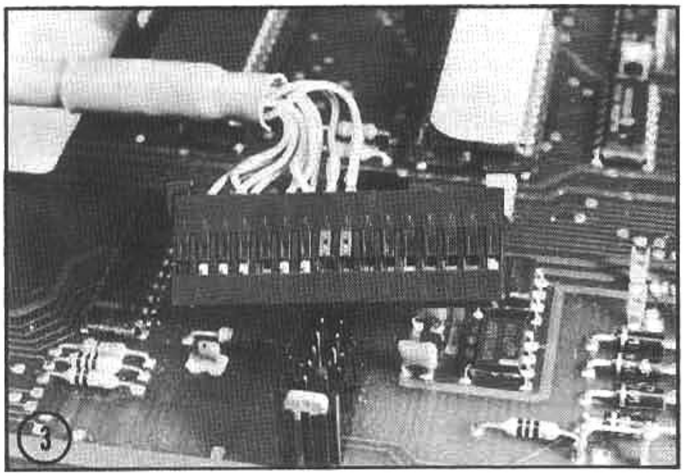
Many computer installations require both the speed of a dot-matrix printer *and* the print quality of a daisywheel printer. Specialized applications may even require two printers of the same or similar type for data output. For example: a large mail-order warehouse may need two high-speed printers, one set up for continuous invoice forms and the other for mailing labels, bills of lading, etc. For these situations, a device that will switch computer output between two serial devices is useful.

Giltronix, of Palo Alto, California, manufacturers an excellent line of low-priced, high-quality RS-232C switches. One model (part GRS 232-S8ABC) allows switching between three printers, computers, terminals, etc. Each model can be ordered to switch eight, 12, or 24 (of the 25) lines. The eight-line switch (identified by the "8" in the part number) works well with Zenith and Heath equipment. The model illustrated here has six indicators to display the status of all the handshaking and data-transmission lines, a helpful feature during installation and diagnostics.

Giltronix also offers a programmable model (part ASU)

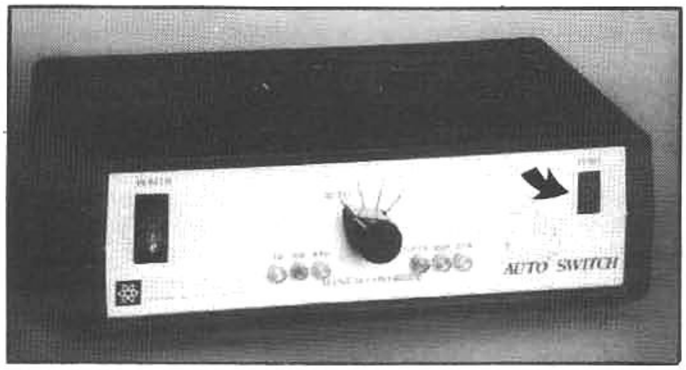




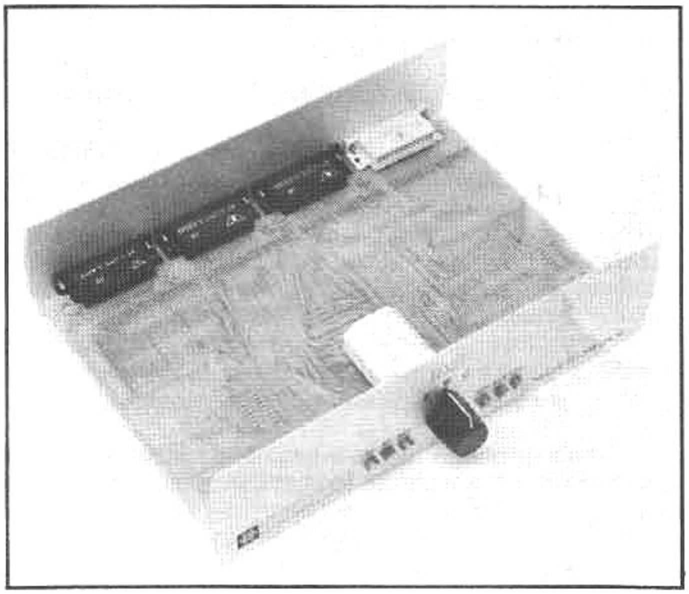


The Siemens printer must be modified slightly before it will handshake properly with the Z/H-89. The twist-lock screws must be released and the print mechanism can be raised to gain access to the electronics. Locate the serial port plug on the left side of the main electronics board (photo 2, arrow). Bend the plastic retaining clips outward on the two wires shown—the 6th and 7th wires from the right edge (black tab), on the rear-facing length of the plug, and swap these leads. This moves the handshaking line from pin 25 to pin 4. Set the two strapping plugs (W28 and W29) as indicated, and then follow the Siemens manual and the guidelines in the text to complete the configuration.

The Giltronix switch box (on top of the Z/H-89) can switch RS-232 serial data transmission between any of three devices. This system can send data to either the Z/H-25 on the left or the Siemens ink-jet printer on the right. The switch will operate with CP/M or HDOS.



Giltronix (see appendix) also has a programmable 3-way RS-232 switch. Upon receipt of an escape-sequence from the computer, the switch will connect it with the appropriate device, and indicate that selection on a digital readout (arrow).



Instead of troublesome individual or ribbon-wire interconnects, the Giltronix design utilizes high-quality printed circuits and gold-plated contacts in the switch assembly. This ensures a reliable, noise-free data path through the switch. The rear panel has female-socket plugs for the computer input (on left) and peripheral output (three on the right).

that will switch the computer output from one channel to the next when the appropriate escape-sequence is transmitted by the computer. This unit allows full automatic control of up to seven printers, and includes a switch that will override automatic operation and allow manual selection of any single device. The front panel includes a digital display that shows which device is active.

In the standard configuration, all the interface plugs at the rear panel are female DB-25 sockets; therefore a male-to-male cable is required to connect the computer printer port to the "data-in" port of the Giltronix switch. The "data-out" ports on the selector switch are identical—female-pin—to those on the rear of Zenith/Heath computers (LP port 340Q on the Z/H-89, not the modem port). Therefore, male-pin plug connectors are required on all the cable ends that connect to the selector switch. If the printer is fitted with a male-pin connector (e.g., the Z/H-25), a female-to-male cable will be needed for printer connection to the Giltronix switch. However, if the printer is fitted with a female-pin connector (e.g., the F-10 and the Epson), a male-to-male cable will be needed. And, if the printer is equipped with a built-in cable (e.g., the Qume and the Siemens), it is probably a male-pin plug which will plug directly into the Giltronix switch.

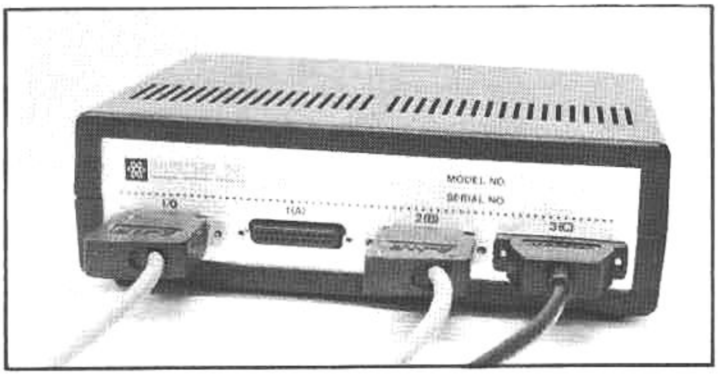
All of the RS-232C switches manufactured by Giltronix have printed circuit boards, not point-to-point wiring. This reduces production costs and increases reliability. The main switching elements are sealed, and use gold-plated contacts to eliminate electronic "noise," a common cause of data loss. All switches include a 5-year warranty. For more information contact Giltronix, Inc. (see appendix).

INSTALLING AND USING A MODEM

In chapter 3 we discussed basic computer communications and in chapter 6 we described some of the software that turns a Zenith/Heath computer into a powerful communications terminal. In this section we will show how to configure the serial interface board for modem communications and install acoustic and direct-connect modems.

All Z/H-89s and 90s are currently supplied with a serial interface that contains three ports through which the CPU can communicate with the outside world. Two of the ports are designed for printers (the top two) and the third is designed to function as a modem connection.

The serial board is normally installed at the center position of the three buss connectors on the right side of the CPU board (see illustration), at plugs P505 and P511. The serial board has three sets of interrupt jumpers located in the lower right corner (when the board is positioned so the printing on the component side reads correctly). Modem communication software *usually* requires that these

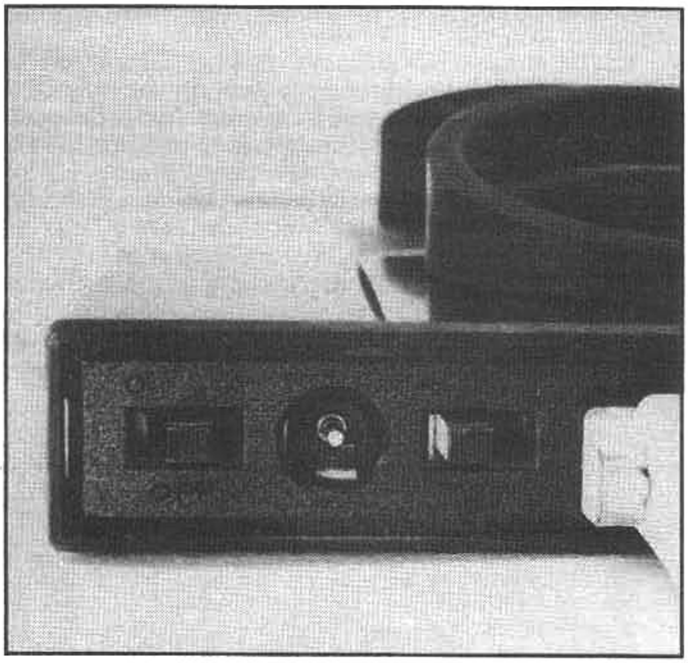


The very popular CAT acoustic modem is one of the many models made by Novation (see appendix). A modem converts digital signals (data) into "tones" that are transmitted over telephone lines just like voice communication. When these tones are received by another modem, they are converted back into digital signals and sent to the terminal or computer.

jumpers be set to level three or higher on the LP and DTE ports (refer to the software documentation for details). However, if there is trouble while sending data to a printer (when communication software is not being used), setting the LP port interrupt level to "off" may solve the problem (it will have to be reset to the proper level when using the communications software).

THE CAT ACOUSTIC-COUPLED MODEM

The *acoustic-coupled* modem is virtually universal. It has rubber muffs to isolate the telephone handset from ambient noise while the modem "talks" and "listens" to the transmitter and receiver in the handset. There are many manufacturers of acoustic modems, but one of the most popular models is the CAT, made by Novation, Inc. (see appendix).



The slide switches on the CAT modem select half- or full duplex operation and answer or originate modes. One of the switches also activates a self-test mode that can check for proper operation of the modem (helpful in tracing communications problems).

The installation and operation of the CAT (and similar modems) is quite simple. The modem is powered by a modular transformer that plugs into an ordinary wall outlet. This design prevents any hazardous voltages levels at the modem or at any connections to the phone line. The modem draws only three watts of power and the user need not remove the wall unit when the modem is out of service.

Data connection to the computer or terminal is made with an RS-232C cable. The modem has a female-pin connector and the modem port in the rear of the Z/H-89 (designated as port 377Q) has a male-pin connector. Therefore, a male-to-female cable is required, and it may have the nine standard wires (pins #1, 2, 3, 4, 5, 6, 7, 8, and 20), although only six are used by the modem. The CAT has two switches that control the originate/answer mode and set the modem for full- or half-duplex operation.

THE ANSWER-ORIGINATE SWITCH

The *originate mode* is used when the user calls a remote computer. When a remote system "answers" an incoming call, it sends a continuous tone over the phone, alerting the user that the system is ready to complete the communication link. When the user's modem is set in the originate mode, it "listens" for this tone from the remote system before sending a response. Conversely, when the user's modem is switched to the *answer mode*, it generates a continuous tone for a remote to acknowledge.

For a successful communications link to be established between two modems, one must be in the originate mode and one must be in the answer mode. The important thing to remember is: if you hear a tone from the remote system, set the modem to originate mode, and if the remote system does not produce a continuous tone, set the modem to answer.

THE FULL- OR HALF-DUPLEX SWITCH

When a key is depressed on the terminal, the character is transmitted through the modem to the remote system. In the *full-duplex mode*, the remote system echoes the character back to the sending terminal. The terminal displays the character on the screen only after it receives this echoed character. The full-duplex mode is the most common form of telephone data communications. If the connection is poor and the remote receives "garbled" data, erroneous characters will be echoed back to the user's screen, alerting him of transmission difficulties.

In the *half-duplex mode*, the modem transmits characters to the remote system and simultaneously displays the character on the terminal screen. This mode should be used when the remote system can not echo received characters back to the sender. If the modem is set to half-duplex when the remote is in the full-duplex mode, characters will be displayed twice on the terminal screen (lliikkee tthhiiss).

THE SELF-TEST MODE

The CAT has a self-test mode that checks the internal electronics and the acoustic transmitter and receiver in the modem; to a limited extent, the telephone headset is also tested. In the test mode, data sent to the modem is converted to acoustic signals and transmitted to the handset. The modem "listens" for the transmitted signals, and since a normal telephone handset echoes sound from the transmitter to the receiver, the data echoed by the handset is converted back to the equivalent ASCII code by the modem, and sent back to the terminal. If the characters that are typed on the terminal are correctly displayed on the screen, the modem is functioning properly.

The telephone must have a "quiet line" for the test to work properly; there should be no dial-tone or other background noise. Dialing only one or two digits will normally provide a quiet line for 15-20 seconds before the telephone switching equipment takes over the line and transmits a recorded message to indicate a dialing error. If more time is needed for the test, it is also possible to call a friend, and have him or her cover the transmitter of the handset to keep the line quiet during the self test.

USING THE MODEM

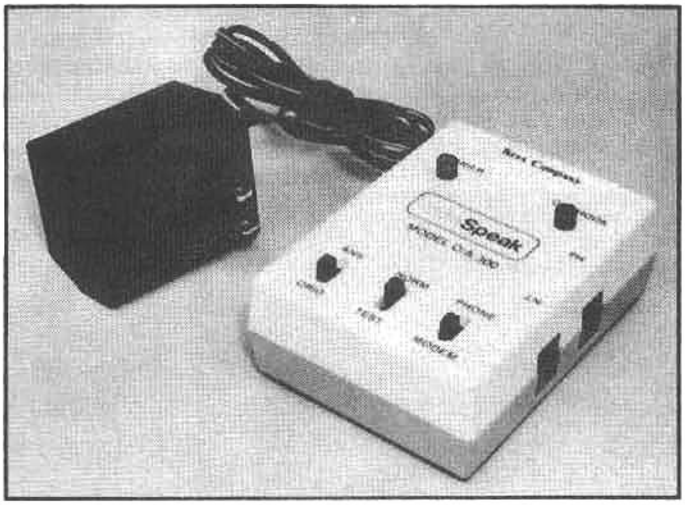
Once the modem is properly connected to the computer, and the modem switches are in the proper positions—usually originate mode and full duplex—the user is ready to establish communication with a remote computer. (If a Z/H-89 is being used, software will be required to convert the computer into an intelligent terminal—see chapter 6; with a Z/H-19 terminal, software is not needed.) The user dials the telephone number of the remote computer. When it responds to the call, it will send the "answer" tone. As soon as the user hears this tone, the handset should be placed in the modem (as indicated by the modem instructions), and the "ready light" on the modem will go on when the communication link is established. A few seconds later, the remote computer will send a sign-on message to the user's terminal, and some prompt will generally be provided by the remote to indicate what commands must be used to gain access to the computer/database. If a sign-on message is not received, the user should press the Return key a few times; the remote will then usually respond with the sign-on.

(Novation provides a modem-accessible computer that can be contacted at 213/881-6880. It primarily provides promotional information about CAT products, but it is a typical remote computer, and may be of interest to beginners who want to experiment with modem communications. After the sign-on message is received, type **CAT** and **RETURN** to get a list of available information.)

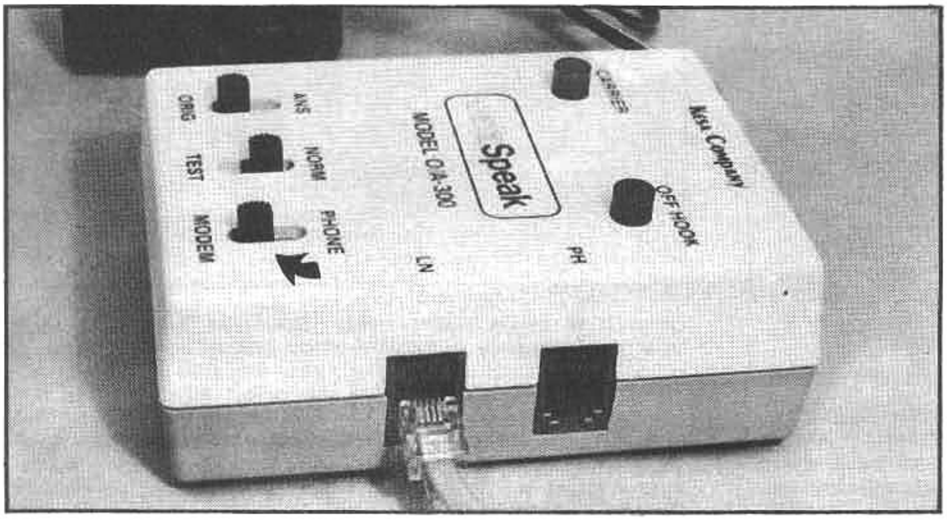
Occasionally there will be communication difficulties. This usually results from a bad connection somewhere along the telephone network. The first course of action is to hang-up and redial the remote, to try to gain a clean line. If trouble persists, try dialing another remote computer (a listing of public-access "bulletin boards" is provided on the Novation computer). If clear communications is still impossible, the modem self-test should be run.

In the event that the modem passes the test, but will not communicate properly, an additional test can verify if the trouble is in the computer/interface or the modem. Remove the RS-232C connector from the modem and use a short jumper or piece of wire to connect pins #2 and#3 together. Type a message on keyboard; the message should be properly displayed on the screen. If the problems persist during this test, the malfunction exists somewhere in the computer/interface or the software, and not in the modem. Verify that the communications software is functioning properly, and inspect the interrupt jumper on the serial interface board to ensure that it is correctly positioned. If these tests do not turn up any faults with the computer/interface, the modem may, in fact, be the source of the trouble, and will require inspection by an authorized repair service.

The CAT modem is available at many computer stores and through many mail-order supply houses. For additional information on this and other modem products contact Novation, Inc. (see appendix).



A direct-connect modem operates in the same general manner as an acoustic modem, however, data is not converted into audible tones and transmitted into the telephone handset. The data signals are introduced directly into the telephone line, reducing distortion and noise, and improving data reliability. This compact direct-connect modem is manufactured by the Kesa Company (see appendix).



The Kesa modem is connected to the telephone line with standard modular jacks. The phone is plugged into the PH socket, and a short cable connects the LN socket to the wall. A switch (arrow) activates either the voice or data communication mode.

THE KESA DIRECT-CONNECT MODEM

A *direct-connect* modem is wired directly into the telephone line, and because there is no audio link, distortion is reduced, ambient noise is no longer a factor, and data-transfer reliability is generally improved. Direct-connect modems offer a few features not available with acoustic modems and they are generally more expensive. The Kesa company, located in Sunnyvale, California, offers a direct-connect modem that is relatively inexpensive and small in size. It is easily connected to any standard (rotary or touch-tone) telephone that uses modular jacks.

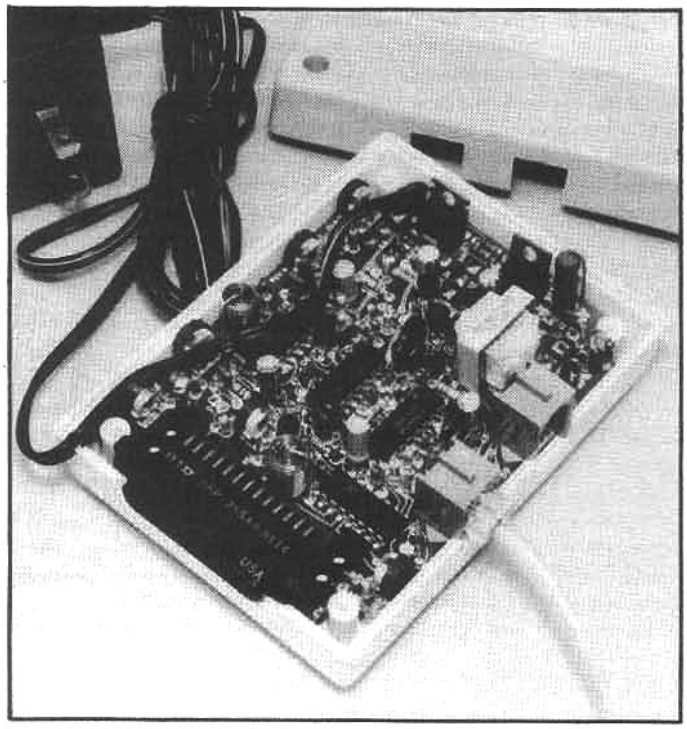
The Kesa O/A-300 DataSpeak modem uses a "phase-locked loop" design that supposedly increases its immunity to noise and improves data transmission reliability. Power is supplied to the modem by a low-voltage modular transformer that plugs into a wall outlet (similar to the CAT), and both originate and answer modes are provided, as well as a self-test feature. Three switches are provided to control the DataSpeak: an answer/originate mode switch; a normal/test switch, which activates the self-test feature; and a phone/modem switch that disables telephone communication and enables modem communication. There are two indicator lamps on the modem: the one marked "Carrier" indicates that communication is established with the remote system, and the one marked "Off-Hook" indicates that the phone is being used by the modem and that incoming callers will hear a busy signal.

The DataSpeak modem is easily connected to any phone with modular jacks. The telephone line is unplugged from the telephone and plugged into the modem at the "LN" (line) jack. An auxiliary connecting cable (supplied with the modem) connects the modem jack marked "PH" (phone) to the modular jack in the telephone. This connection installs the modem in series with the telephone.

Using the Kesa modem is easy. After the computer or terminal is properly connected to the modem—a male-to-female RS-232C cable is required—and the communications software is running (if required), the user usually switches the DataSpeak to the originate mode, sets the function switch to "Norm," and dials the remote computer. When the tone from the remote system is received, the DataSpeak is switched from "Phone" to "Modem" and the telephone handset is returned to the cradle. The "Off-hook" light will come on, and within a few seconds, the "Carrier" light will also come on, indicating that a communications link has been established. When the user is finished communicating with the remote and ready to break the communication link, the DataSpeak is switched from "Modem" to "Phone." This breaks the link and hangs up the telephone line, and the "Carrier" and "Off-hook" lights will go off.

An interesting capability that a direct-connect modem provides is the ability to "answer" incoming calls. The Kesa modem does not support this capability, but some direct-connect modems, like the Hayes Smartmodem, can be used for unattended E-mail operation (see chapter 3), or a hobbyist can use a direct-connect modem to establish a local computer "bulletin board." Of course, suitable software is needed to support these operations.

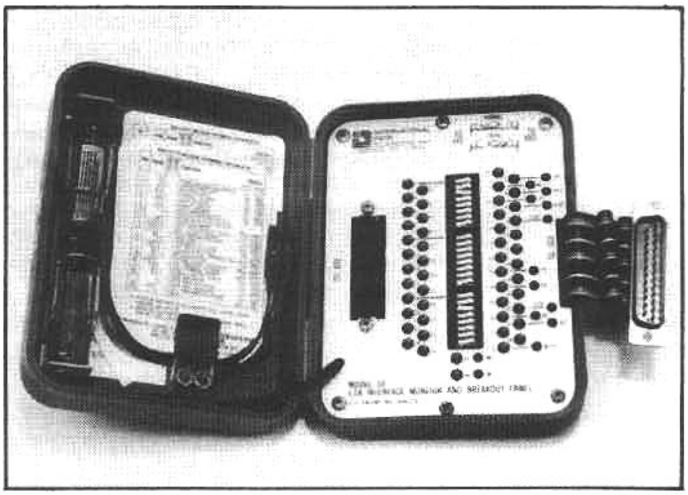
The Kesa DataSpeak is an ideal modem for home or business use, but it can not be used on commercial multi-line telephones or any type of telephone that does not have a modular jack at the telephone end of the incoming line. It occupies very little desk space; it is extremely easy to use; it is reliable and virtually immune to both ambient and electrical noise. For more information contact the Kesa Company (see appendix).



The interior of the Kesa modem reveals the simple and compact design of a direct-connect modem. The components convert digital signals to audio and back again, while retaining compatibility with the normal telephone voltages, signal strengths, etc.

SOLVING INTERFACING PROBLEMS

Although the problems aren't readily apparent to the neophyte, establishing communication between a computer and an accessory component can be tricky, at best. The difficulties are minimized if the accessory also carries



Sooner or later nearly every computer user confronts interfacing or data communication problems with peripheral accessories. Without some reliable means to accurately determine the state (on or off condition) of the handshaking lines, the user is completely in the dark. The only sure-fire solution is a line monitor like this "blue box" from International Data Sciences (see appendix) It can prevent hours of frustration when coordinating communication between unfamiliar hardware.

the Zenith/Heath brand name, because the system accessories are designed to work together, and adequate information is generally provided to help the user complete the hookup. However, the situation gets much more complicated if the user is trying to interface non-Zenith/Heath components with the system.

To successfully complete a communication interface, the user must determine: which I/O line(s) the computer uses for handshaking; which I/O line(s) the accessory uses for handshaking; what polarity (normal or reverse) the computer observes on the handshaking line(s); what polarity the accessory observes on the handshaking line(s); and what sort of software is required (CP/M "configuration" or HDOS device driver).

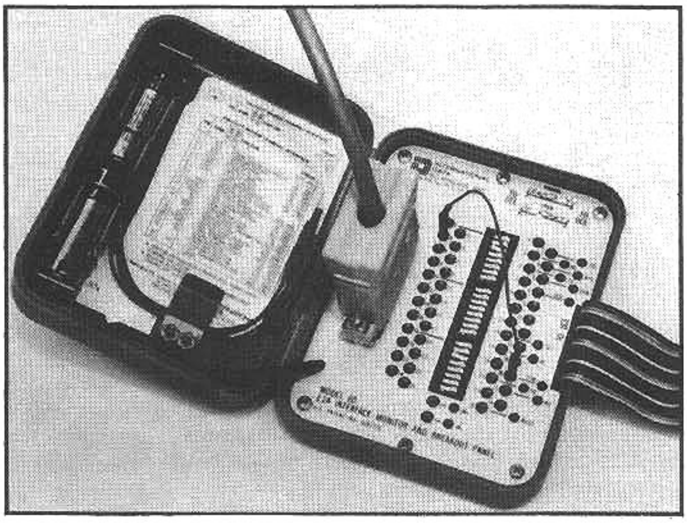
Finding *accurate* answers to these questions can be frustrating, especially for the beginner. Unfortunately there are no universal handshaking standards, and as much as the manufacturers or distributors would like to help, they can not provide much assistance, because it is simply not practical for each of them to test their equipment with *every* other product on the market. Often the only certain solution is to get first-hand advice from someone who has already solved the problems (if you can find such a person), or a determined user can learn to solve interfacing problems with the aid of some useful test equipment.

To experienced computer technicians the "blue box" is one of the requisite tools-of-the-trade. Manufactured by International Data Sciences, Inc., the Model 60 Monitor and Breakout Panel (fondly known as the "blue box" because it is enclosed in a blue plastic carrying case) is a portable, self-contained tester that allows monitoring and cross-connecting (if desired) of all 25 lines in the standard RS-232C interface connector.

The blue box is installed in the RS-232C communications line between the computer and the device to be interfaced: it has a built-in DB-25 female-socket—called the DTE connector—to allow the cable from the peripheral to be plugged into the box, and a short ribbon cable with a DB-25 male-pin plug—called the DCE connector—to connect the box to the computer serial port (the LP port on the Z/H-89). The box contains switches (see illustration) that permit the user to switch or or off each of the 24 signal lines (pin #1 is a universal ground and is not switched). In addition, a slip-pin connector is provided on both sides of each switch. The pins are labeled with the RS-232C line number to which they correspond and, by using short jumpers (supplied with the box), the user can cross-connect signals from one line to another.

The box also provides a series of twelve (plus two) LEDs that monitor the conventional RS-232C handshaking and data-transmission lines. These LEDs are useful for reading the "state" of each line during communication. Each LED lights up when the particular line it monitors "goes high" (indicating it is carrying a small positive voltage). However, it is important for the user to realize that the LEDs in the box always monitor *the conventional source of the signal*; and if the DTE and DCE sides of the line are separated ("opened") by one of the switches, the LED does not directly indicate which side of the line—the computer-side or the peripheral-side—is high or low (one must refer to the line chart supplied with the box).

The lines in an RS-232 connection do not carry bi-directional signals. Each line is controlled—driven high or pulled low—only by the peripheral or the computer, *not both*. For example, under normal conditions, line #2 is the "transmitted-data" line and is called the TD line. This line only carries a signal *from the peripheral* to the computer and the LED on the TD line only monitors the *peripheral side* of the line. When it lights up, it indicates that the DTE side of line—and the corresponding jumper pin—has been *driven high by the peripheral*. Conversely, line #3 is a "received-data" line and is called the RD line. It only carries a signal from the computer to the peripheral. When the monitor on this line lights up, it indicates that the computer has driven the DCE side of the line—and the jumper pin on the DCE side—high. Therefore, in order to completely understand the information provided by the LEDs, *it is essential to know whether the line is controlled by the computer or the peripheral*. The conventions of RS-232 communications are explained in the documentation supplied with the blue box.



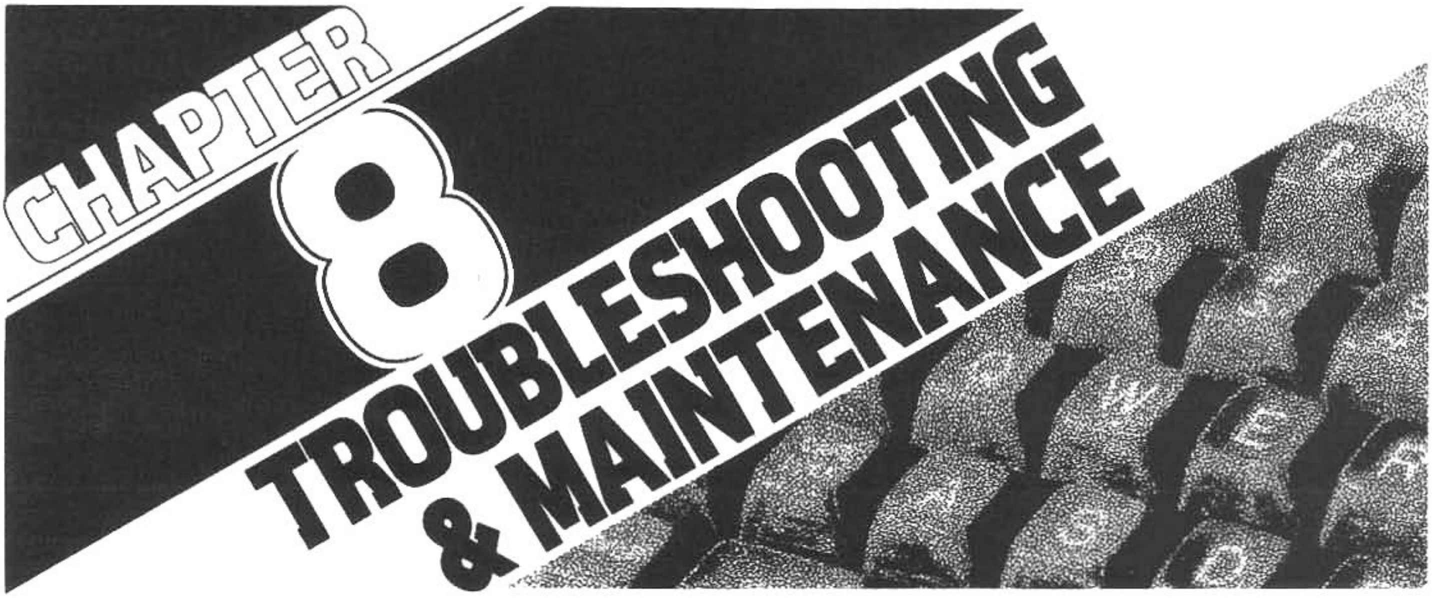
In this example the IDS line monitor is used to test a cross-connection between handshaking lines 20 and 4 (a modification that is often required when interfacing Zenith/Heath computers to serial printers). This technique will quickly identify what alterations are required to establish proper communication with peripheral accessories.

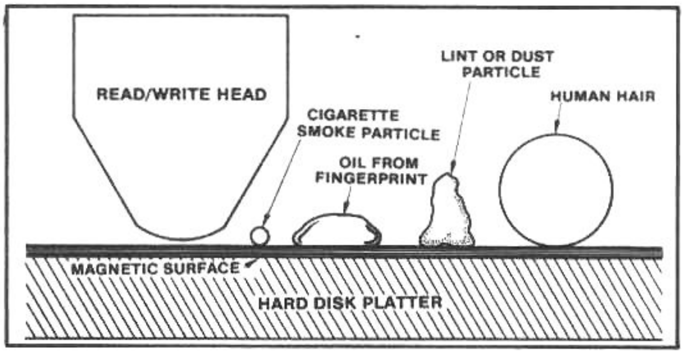
Once the concept and conventions of handshaking (and the operation of the blue box) are understood, most interfacing problems can be readily resolved. For instance, suppose the box is connected between a printer that is transmitting handshaking signals on line #20 and a computer that is "looking" for handshaking on line #4. First, the switches on lines #4 and #20 are turned off (creating open circuits) and a jumper is installed from the pin on the DTE (printer) side of line #20 to the pin on the DCE (computer) side of line #4. This jumped connection directs the handshaking signals coming from the printer, on line #20, to the computer, on line #4; and the computer and the printer should be able to communicate correctly. (The switches are turned off to prevent short circuits to the printer or the computer that might occur when pins #4 and #20 are jumped together—the open switches isolate pins #4 and #20 on both the computer and printer side; and the signal from the printer is directed *only* to the computer).

Even though the blue box makes interfacing easier, it is not likely that every microcomputer user will want to buy a box just to connect one printer to a computer. However, if the reader has problems of this sort, the easiest solution may be to find a local computer technician who has a box and is willing to help. The box *will* answer your questions.

The Model 60 Monitor and Breakout Panel is only one of many useful testing and switching instruments manufactured by International Data Sciences (see appendix); additional information is available from IDS upon request.

CHAPTER 8 TROUBLESHOOTING & MAINTENANCE





Microcomputer systems contain sensitive electronic and mechanical components. These devices are normally very reliable, but one of their worst enemies is dirt. Many electromechanical problems can be traced to accumulations of dust, lint, hair, fingerprints, or even airborne smoke particulates. A hard disk drive is especially sensitive to these troubles because there is an extremely small gap between the read-write head and the disk surface.

There are many reasons why electronic and mechanical things fail, and fail they will! As rugged and reliable as a Zenith or Heath computer is, it may one day require repair. This chapter will help a user understand what can go wrong, give some simple trouble shooting advice, and provide some suggestions for implementing practical preventive maintenance.

THE SOURCES OF TROUBLE

Three enemies of computer equipment are around us every day: dust, water and static electricity. Of these, dust is the most common, and the component most susceptible to the ravages of dust is the disk drive. A disk drive is an electromechanical system that must function with precision, but dust accumulations can interfere with the operation, increasing disk read/write errors and reducing media life. High-density drives are manufactured to even closer tolerances, and are more susceptible to the effects of dirt. To a lesser degree, circuit boards can be affected by airborne contaminants—if a circuit board is covered with a layer of dust, random errors or lost data may result.

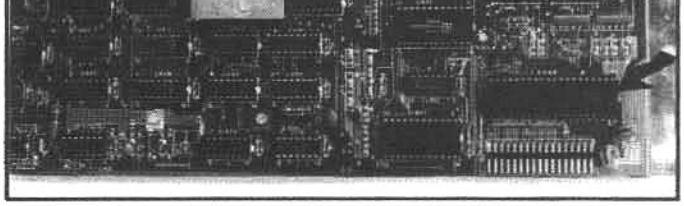
Another source of dust—believe it or not—is smoke and ashes from cigarettes. Leaving an ashtray next to the computer is like putting a compost pile in the kitchen. Ashes are chunks of dirt, and smoke particles, though smaller, are large enough to seriously damage a hard disk drive. This is why a hard disk is sealed and cooling air admitted to the drive cabinet is thoroughly filtered.

Zenith and Heath computers do not use a filter for incoming air. The manufacturer places the responsibility of providing a "computer room environment" on the user. Surgical cleanliness is not necessary, however anything that can be done to provide clean, cool air to the system will increase reliability.

TOO WET

Water and other liquids, such as coffee and soft drinks, are found next to computers more often than ashtrays. Sugar or milk in coffee, or the syrup from a cola drink, can be "the kiss of death" to any electromechanical connection. The most common repair procedure for a keyboard inundated with such a liquid is simple: it is replaced at the owner's expense.

Other than from a leaky roof, water rarely enters the computer (except through the keyboard), so elementary



Some early Z/H-89 systems were plagued with keyboard encoder IC failures. Static electric charges, transferred from the user's fingers to the keyboard, would occasionally zap the silicon circuitry in the chip, but this was eliminated by adding a wire to ground the static charge before it reached the chip.

caution is all that is necessary to avoid trouble. Water can also spill onto disks or into printers, and the effects are worsened if the water has anything else mixed into it. Be alert. Nobody's saying: "Don't drink coffee while you work." Just be very careful or, better yet, keep the cup on a separate table. Make sure that any other user is made aware of the effects of coffee on computer electronics—first-hand experience is not always the best experience.

ZAP

Static electricity is a killer. It is the hardest to control, because it is generated so quickly and so quietly. A liquid spill is usually accompanied by a loud human exclamation, and prompt action can often restore what the liquid threatens to destroy. But when static electricity strikes, its effects are devastating and there is often no recovery—other than replacing parts.

One of the most common problems with Z/H-89s and Z/H-19s manufactured before 1981 was a failure of the keyboard encoder chip on the terminal logic board. Many chips were replaced, and the failed ones were sent back to Zenith for examination. Quality control was apparently not to blame. What had happened was that accidental discharges of static electricity, through the users' fingers had "zapped" the tender chip. A simple grounding circuit was installed in new machines to absorb the shocks, and the problem abated.

Some of the effects of static electricity are: loss of data, a confusion of displayed characters, temporary or permanent loss of components, and a painful sensation both at the tip of the finger and in the mind of the user upon contemplation of the damage.

There are several ways to reduce static electricity, though none of them is foolproof. There will always be some risk, but if a user is careful, it will be a negligible risk.

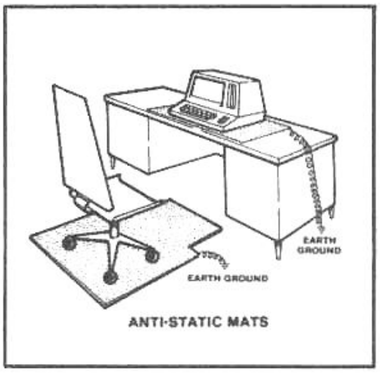
Humidify the room, if you can do so without causing dampness (see the problems of water, above). People in humid or tropical climates almost never encounter static electricity because water vapor is a natural conductor, and it will "damp out" static electricity. In climates where the air is naturally dry, static electricity is a fact of life.

During winter, heating systems tend to make interior air dry, and people tend to wear wool clothing—a well-known static generator. So the problem may be at its worst during those times. If a clothes dryer is nearby, the static will probably be impossible to counteract. A user who can not eliminate static electricity from the computer environment should seriously consider moving the machine to a more suitable location. The inconvenience may be offset by not having to put up with the frustration (and expense) caused by static discharges.

ANTI-STATIC MEASURES

There are "anti-static mats" for installation between computers and desks or on which the user's chair can roll. These usually have wires which are grounded to cold water pipes or other good earth grounds. Mats are priced according to size, and should cure most static problems.

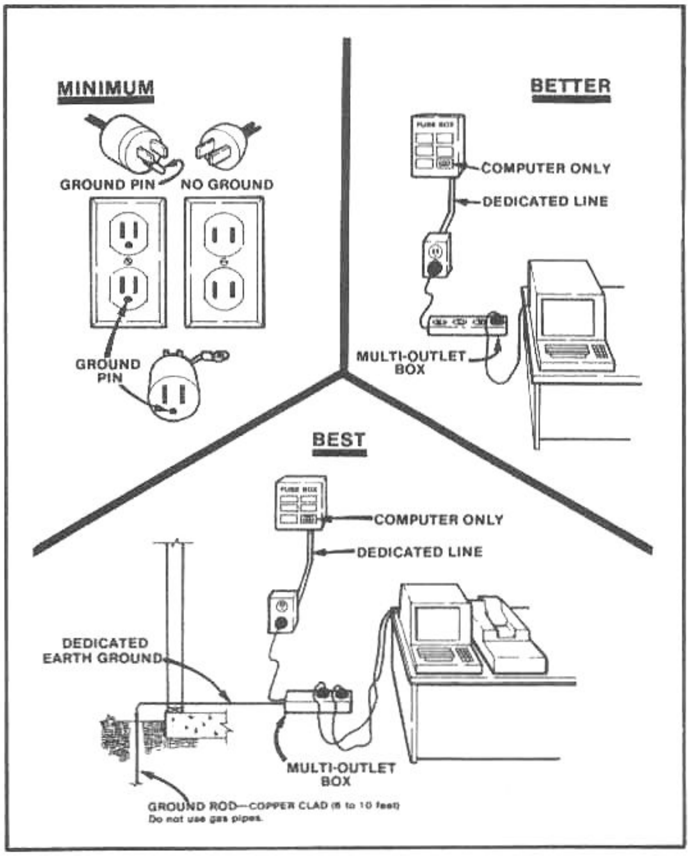
Successfully keeping an entire *room* free of static electricity, however, is a mammoth task; it is better assigned to a professional electrician or computer hardware consultant who has experience with this problem. Zenith technicians suggest one "home remedy" for static. If you have carpeted floors, gently spray them with a plant sprayer, using plain water to dilute some laundry fabric softener. Fabric softeners are usually sold to "reduce static cling," and they will have the same effect on the computer room carpet.



Special mats that suppress static electricity can reduce static electricity problems. A small mat can be installed under the computer, or a larger one can be used under a desk or chair. These mats are connected to an earth ground, draining static charges away from the computer and operator.

GROUND WIRES

Your Zenith computer or terminal comes with a power cord that has a plug with three prongs. The flat ones carry the current, and the tubular one is a ground. If there were no ground wire, static electricity could build up, and in the event of an electrical short, a serious safety hazard could



If you are installing a new system, it pays to provide your computer with a "clean" source of power. The power line should be a functional three-wire system (the third wire is a ground). If possible, it is wise to provide "dedicated" power to the computer. This means that the line is routed directly from the fusebox to the computer system, with no other appliances, lights, etc. on the same circuit. *The best arrangement has both a dedicated line and a separate earth ground.* These measures will isolate a small system from most external disturbances, and provide the best data protection (other than employing an expensive line conditioner and "non-interruptible" power supply).

result. The best connection for your computer is a three-hole, grounded socket, which is common in practically every office building and in most modern homes.

If you do not have a three-hole, grounded wiring system, you have a two-wire system in which one of the wires doubles as a ground. This is not good for sensitive electronic devices, so you have two choices: ignore it or adapt it. If you ignore it, (i.e., plug three-pronged plugs into two-pronged adapters and plug them into your wall sockets), you may risk electrical shock and/or damage to your computer. But you can adapt your wiring in three ways:

The best way is to use a "dedicated line" to the computer with a special earth ground. A dedicated line is a single circuit run just to the computer, with no other appliances connected on that circuit. The ground for the system should be a "dedicated" ground: a ground wire connected to a copper (or thickly copper plated) ground rod, sunk in moist soil at least six feet. This will provide the most reliable power for your computer system (called a "quiet line"). It is standard practice in commercial buildings. (All such installation work should be done by an experienced electrician.)

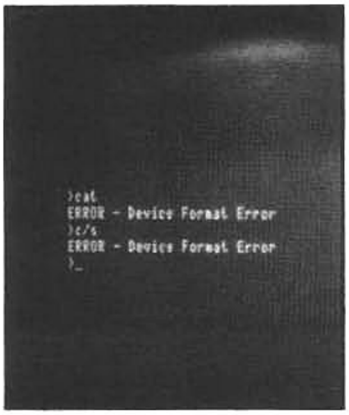
If you cannot use a dedicated line, the next-best method is to run a separate ground wire from the socket feeding your computer to a water pipe (a ground rod is better, as most modern plumbing now uses plastic pipe), and you will have a suitable installation for most applications.

The cheapest, and the most risky, way is to use a three-to-two-prong adaptor. If you must use this method, at least *ensure that there is nothing else on that circuit*, including lamps and especially any motor-driven appliances (use a multiple-outlet box for all of the computer equipment). What you will have, then, is a system with a common ground. While the possibility of faulty grounding and electrical shock still exists, the common connection will at least keep all the computer components "together." This may help the system withstand electrical "noise," although in some installations, a printer can introduce sufficient interference to affect the computer. If at all possible, install an earth ground to the multiple-outlet box. The result will be a much "quieter," and safer installation.

POWER OUTAGE OR LOSS

If a computer loses power, it will stop. A severe storm or natural disaster, of course, may interrupt electric utility service for minutes, hours or even days, so it is unwise to begin a major computing project when violent weather is forecast. Any computer that is running when the power goes off will suffer unless some provision has been made for immediately switching to an alternate power source. But backing up the source of electricity is generally impractical for a home or small business. In large companies, and in communication industries, where voice and data traffic must not fail, rooms full of batteries are kept at full charge all the time, to take over if the power goes out. This backup capability is expensive, and requires not only batteries, but also a sensitive electronic device that recognizes the loss of power and switches in the batteries within a very small fraction of a second. "Uninterruptible" power supplies are available for small (personal) computer systems, but the cost is still rather high. When compared to the cost of lost accounting records, however, they can be much less expensive.

If you are working at your computer when the power fails, you will lose whatever file you were working on at that moment, since it will disappear from the main memory (RAM). This happens because RAM needs to be "refreshed" every few thousandths of a second. It is done automatically, without the user being aware of it, but if the power stops, so does the refreshing, and the orderly pattern of data stored in RAM disappears. The same thing happens when the user switches off the power intentionally.



```
>cat
ERROR - Device Format Error
>c/s
ERROR - Device Format Error
>_
```

This upleasant message indicates that HDOS cannot read a portion of the floppy disk. The result is almost always a loss of some disk data, or if the surface defect is in a critical area, the entire disk may be unreadable. To avoid this, *keep the area around your computer clean, use a dedicated power line, do not operate your system during severe weather, and ALWAYS MAINTAIN BACKUP DISKS!!!*

However, files that are stored on disk are not usually lost when power fails, unless the computer was in the process of writing to the disk; in this case, the disk can be left in an unpredictable condition and all data may be unreadable. If power is lost, remove the disks from the disk drives immediately. The files which have been stored on the disk may not have been affected by the power outage, but they can be damaged when the power returns because a sudden surge of power can fool a drive into "writing" spurious data on a disk. *Then turn the computer and all the peripherals off.*

This advice is familiar to most people who own refrigerators and other heavy appliances, but for different reasons: motor-driven appliances demand so much current when they are turned on (i.e., when normal power resumes), that they may overload the fuse or circuit breaker. A computer's internal circuits can also be damaged by a surge in electrical power. So first let the power come on and stabilize, *then* turn the computer back on.

POWER FLUCTUATIONS

A fluctuating supply of electricity is another problem that plagues computers, and is much more common than most people think. Electrical fluctuations can have a varying effect, from no effect at all to approximately the same effect as turning the power off and then on again. If you do not wish to have your computer randomly RESET, you will take care to prepare for power fluctuations.

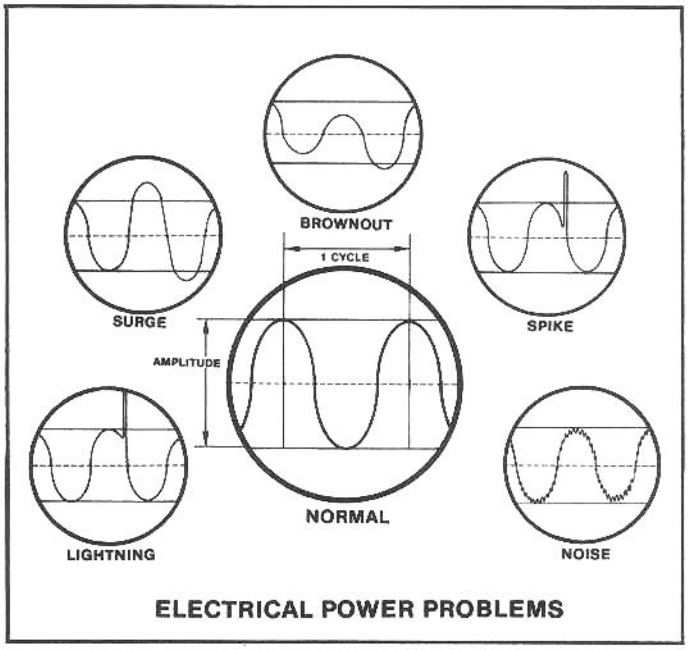
In the U.S., household current comes through wires in pulses, 60 times per second. (In Europe and many other regions, it cycles 50 times per second, and equipment sold there is designed to work at that rate.) If the power should skip a few cycles, the all-important internal timing for "refresh" will be upset, and the computer will then do what it is programmed to do when power first comes on: ask you to boot the operating system.

A momentary loss of power is unpredictable. It may come from the utility's grid system, as when a generator goes down, and another is switched on. More common, however, it is generated inside your home or business. Allowing a washing machine, refrigerator, or other motor-driven appliance to share an electrical circuit with a computer is asking for trouble. Did you ever notice how the lights flicker when somebody plugs in a vacuum cleaner?

THE SPIKE

Another power problem comes not from too little power but from too much. Just as electricity is supposed to keep a steady frequency (60 Hz), it is also supposed to maintain a constant "amplitude." That means that each pulse of electricity should be exactly as strong as the last and the next.

If you look at an oscilloscope display (an electronic "graph") of 60-cycle electricity, the frequency is shown by how many peaks or valleys fit into a given time period: each cycle can be measured from peak to peak, with a valley between. The amplitude is shown by how high above the center line (or how deep below it) the wave goes. An interruption of power is represented by a straight line (no amplitude). If power hesitates for a moment, the trace will show a flat segment between the peaks and valleys. When the power comes back on, however, the amplitude may take a few cycles to reach its original height. Similarly, if a surge of power comes through the line, the amplitude may jump to an unusually high point before settling back to normal—this is called a "spike."



ELECTRICAL POWER PROBLEMS

These illustrations represent power line conditions (as traced on an oscilloscope). A normal flow of AC power produces 60 uniform cycles (Hertz—Hz) per second. (50 cycle is standard in Europe.) Even if the cycle rate does not change, problems can arise if the amplitude (height) of the waveform varies excessively. Undesirable power line "noise" can change the smooth curve to a rough one, and upset sensitive computer circuits. A surge is a sudden increase in amplitude, and a brownout is a decrease in amplitude. Either of these conditions can overload sensitive components in the computer power-supply circuits, or create internal voltage fluctuations that may destroy data stored in the RAM. A spike is a brief increase in amplitude, and can also affect the power supply or wreak havoc in the integrated circuits. Lightning is an extreme form of a spike—very short in duration but very powerful in amplitude—that can permanently damage major components.

A spike is just about the most frustrating electrical event that can happen to a computer user, short of a power outage. The problem is not that data can be lost (as from an outage) but that it can be subtly changed. When a spike whips through a computer system, it may overload a component and cause unpredictable results. If the component is a memory chip, its loss may not be immediately apparent, and the next time a file is accessed, data may be erroneous or missing.

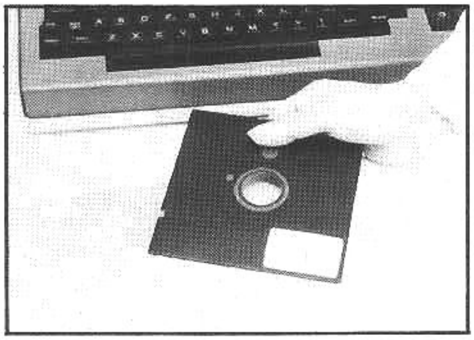
Spikes can be smoothed out by placing a special device between the power source (i.e., the wall socket) and the computer. Some multi-socket outlet boxes have a fuse that will break if too much power comes through, but that is only useful for a very severe condition; spikes are much more subtle. Some outlet strips, (various models are sold in Heath electronic centers), have a surge protector. Aside from line-conditioning multiple outlet boxes, there are "black boxes" that plug into the wall socket, and the cord from the computer plugs into the "black box." Regardless of their design, these "watchdogs" detect a spike and shunt the excess power to ground while keeping the normal power flowing.

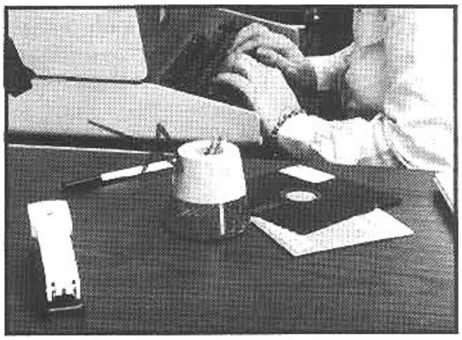
DISK DILEMMAS

So much has been written about disk and disk drive maintenance that people are tempted to skip a section like this. Don't. Disk reliability is essential, and we will present some information here that every computer user should know.

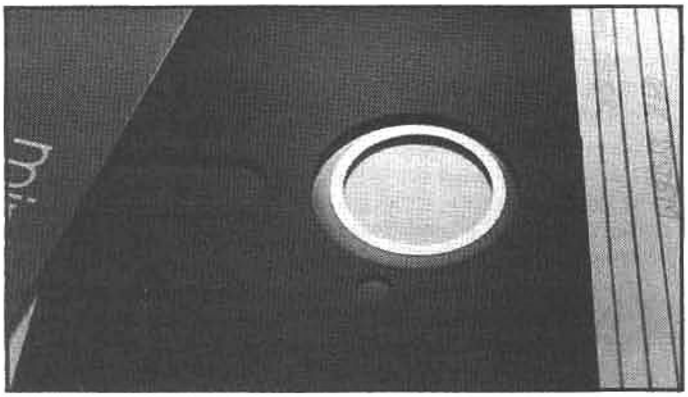
There are several things that can render your disks useless. Of these, the most common troublemakers are dirt, heat, and magnets. Dust and fingerprints can be kept off your disks with one simple caution: don't grab them through the access slot. Always replace them in their protective envelope when not in use. Keep cigarettes, food, cosmetics, solvents and lubricants far, far away from your disks. Do not touch the magnetic media, the coated disk surface; handle only the protective cover, and then only gently. It may be known as a "floppy" disk, but don't try to find out just how floppy it can get.

The most common form of abuse is writing on a disk label with a ballpoint pen. Write with a soft felt-tip pen to prevent damage to the thin mylar media inside the jacket. If a new label is needed, put it over the old one, and don't try to remove the old label unless it peels off very easily—you could damage the disk inside the cover in the process. Since the disk drive isn't "concerned" with the cover (only with the disk inside), a few extra layers of labels won't af-





The average floppy disk can store more than 2,000,000 bits of data. If you would like all of these bits to be there the next time you need it, *keep the disk in a protective envelope when it is not being used*, do not bend the disk or touch the magnetic surface, and never use the disk near magnets or when the ambient temperature is unusually hot or cold.



Disk support rings help "center" the floppy disk in the disk drive. If the disk seats improperly when it is grasped by the drive spindle, the recording tracks will not be registered on the true center, and data read or write problems can occur. The user can install accessory rings on unsupported disks, but this is not as accurate as rings that are installed by the disk manufacturer (the rings should be concentric with the true center of any existing data tracks). Quality disks with support rings are considered a necessity when using high-density drives (such as the Z/H-37).

fect normal operation.

An often-overlooked source of trouble is heat. Don't lay disks on the vents in the computer cabinet (there's dust there, too). Watch out for window ledges, heaters, and sun-dappled spots on your desk or file cabinet. And don't store floppy disks in your car on a hot day, and never leave them where direct sunlight can overheat them.

One source of trouble that is only obvious after it is explained is exposure to magnets. *Don't use any magnetic objects near your computer work area.* Magnets are used in paper clip dispensers, small tools, certain types of bulletin boards, etc. A magnetic object will erase data the instant it touches the disk jacket (or comes very close). Don't get caught by this potential problem—such surprises you don't need!

Be very careful when inserting or removing a disk from a disk drive. Make sure the power is on before you insert a disk, and never turn off the power with a disk inside the drive. Handle the disks gently, but seat them in the drive with a firm pressure before closing the door. If you feel any resistance when closing the door, stop: you may be pinching the disk (it happens when the disk is not inserted all the way).

The very best preventative measure is to get in the habit of making backup copies of all your disks. Besides ensuring that the working copies of software are reliable, backup copies of important files can be time- (if not life-) savers when the "horror" message appears on your screen: "BIOS Error on B: Bad Sector." This problem is usually irreversible, in CP/M, and means that for some reason—usually a magnetic/physical discrepency in the disk—your file can not be read, and is therefore no longer available to you.

Zenith technicians recommend against trying to make "flippies" out of single-sided disks by cutting an additional write-protection notch and "flipping" the disk over to record on its reverse side. This is a hobbyist trick that does work, but disks which have been manufactured for single-sided recording are generally not quality-controlled on the reverse side, i.e., the quality of recording on the reverse side is not guaranteed by the manufacturer.

Whatever your disk drive's specifications, it is wise to buy only the appropriate and corresponding disks. For example, if you have a double-side, double-density 5.25-inch drive (the Z/H-37), be sure to purchase disks which are certified for operation at 96 tpi (tracks per inch). They are more expensive than those for single-track density recording (48 tpi) but they are essential for reliable operation at high track densities. In addition, disks that have a reinforced ring around the center drive hole are usually more accurately "centered" by the drive when installed, and are worth the money (this design is essential when using the Z/H-37 drives).

UNDERSTANDING THE TECHNOLOGY

This section will cover more technical material than the rest of the book does; none of it is essential (though it is helpful) for operating the Z/H-89 or Z/H-100 systems successfully. Those users who have built their computers from kits may already have learned much of the following information. However, this technical information is presented in a non-tutorial manner, addressed at anyone interested in understanding more of what goes on behind the CRT screen.

There are diagnostic tests that professional Zenith and Heath service people use to verify the proper operation of the internal electronics in Z/H computers. Some of these can also be run by users, partly for their own education and enjoyment, but also as a way of learning more about the workings of their computers, or ensuring that their systems are functioning properly. Before running those tests, however, a user should have a clear idea of what is being tested, and how the test results may be interpreted. To do this requires a basic understanding of computer jargon, which will be summarized here, as it relates to the Zenith/Heath computers.

The concepts that must be understood are: bits and bytes, binary and split-octal number systems, dynamic RAM, the memory map, and the monitor. Where disk drives are involved, the concepts of disk I/O, tracks and sectors are also needed.

Diagnostic programs are very important troubleshooting tools. Some diagnostics are extremely complex, requiring extensive knowledge of electronics and test equipment, others produce results that are easy for the ordinary user to analyze. With a basic understanding of binary and split-octal numbers, a Z/H-89 user can competently test the CPU, RAM, and disk drives.

BITS AND BYTES

There are two electrical states: on and off. If something is "on," it is like writing a number "1;" if it is off, it is like a "0" (zero). Each distinguishable "on" or "off" state (1 or 0) is a bit of information. If a computer can absorb and act on eight bits simutaneously (as the Zenith/Heath computers do), it is said to have an "eight-bit byte." The word "byte" is something of a pun on "bite," as when a person can eat eight beans in a single mouthful (in the same genre, four bits are called a "nibble.").

BINARY NUMBERS

The number systems that modern civilizations use are based on the idea that the position of a number determines its value. The common base-10 number system uses this notation. The number 22, for example, tells the reader than there are two "ones" and two "tens:" (2 X 10)+(2 X 1)=22. The "tens place," "hundreds place," "thousands place," etc. are familiar words from our childhood, and their meanings are well-understood. Each "place" is a power of 10:

10^0=1
10^1=10
10^2=100 ("10 squared")
10^3=1,000 ("10 cubed")
.
.
.
10^9=1,000,000,000

Moving digit by digit to the left, each "place" increases in value by one power of ten, and in each position—or place—a number can take on any one of 10 values (0,1,2,3,4,5,6,7,8 or 9) The "nature" of the base-10 number system runs in tens: 10 possible values for each place, and the next higher place being ten times greater in value. For example: the number "5,794" is understood as (5 X 1,000)+(7 X 100)+(9 X 10)+(4 X 1).

In the binary number system (base 2), there are only two possible values for each place: 0 or 1. Therefore, each place represents a power of 2 (2 multiplied by itself, compared with 10 multiplied by itself in the base-10 system).

2^0=1
2^1=2
2^2=4 ("2 squared")
2^3=8 ("2 cubed")
.
.
.
2^{16}=65,536

The binary number "1" equals 1, but the binary number 11 equals 3 (one "two" plus one "one"); the binary number 111 equals seven: (1 X 4)+(1 X 2)+(1 X 1)=7. Now let's breakdown a more complex binary number step-by-step. The binary "11011" can be expressed as: (1 X 2^4)+(1 X 2^3)+(0 X 2^2)+(1 X 2^1)+(1 X 2^0) or, in simplified terms: (1 X 16)+(1 X 8)+(0 X 4)+(1 X 2)+(1 X 1) is equal to (16)+(8)+(0)+(2)+(1)=27.

In computers that use eight-bit bytes, binary numbers are eight "places" wide—even if some of those places are filled with zeros. Thus, the number 10101010_2 (to the base 2) is numerically equal to 170_{10} (to the base 10). But it is not the same to a computer. The number 170 is meaningless to the logic of a computer because it can only "think" in terms of electrical states being either "on" or "off," which are represented as "1" and "0," respectively. So the number 10101010, in binary notation, is very meaningful to a computer. Reading right-to-left, it recognizes that the first position is off, the next is on, the next is off, the next is on, etc. Each position is "understood" from an "on" or "off" electrical state, with the extreme right-hand position known as the "least-significant bit" (LSB), and the extreme left-hand position known as the "most-significant bit" (MSB). The significance of "significance" is that the MSB has the greatest place value in the byte, while the LSB has the least place value. Additional meaning to LSB and MSB will be developed later, as we explore the "memory map" of the computer.

DECIMAL	BINARY	DECIMAL	BINARY
0	00000000	32	00100000
1	00000001	33	00100001
2	00000010	•	
3	00000011	•	
4	00000100	•	
5	00000101	63	00111111
6	00000110	64	01000000
7	00000111	65	01000001
8	00001000	•	
9	00001001	•	
10	00001010	127	01111111
11	00001011	128	10000000
12	00001100	129	10000001
13	00001101	•	
14	00001110	•	
15	00001111	•	
16	00010000	•	
17	00010001	•	
•		•	
•		•	
•		254	11111110
31	00011111	255	11111111

This is the relationship between decimal and binary numbers. Note that the decimal zero (0) is represented by the binary number *00000000*, and the highest decimal number that can be represented by a one byte (8-bit) binary number is 255, which is the binary *11111111*. A two-byte pair (a 16-bit binary number called a "word") can represent decimal values from zero (0) to 65,535.

SPLIT-OCTAL NUMBERS

A split-octal number is a shorthand commonly used with Zenith and Heath computers. It uses decimal numerals to express binary values. The split-octal notation represents a 16-bit number in the form of two 8-bit bytes of information. For example:

3-7-7 3-7-7

is the split-octal representation of 65,535. But before we get too far ahead of ourselves, we we must look more closely at binary notation.

The largest binary number that eight bits can represent is 11111111 (128+64+32+16+8+4+2+1) 255. By adding eight more bits to the left, making a 16 bit "word," the binary number 11111111 11111111 equals the same 255 (for the right eight bits plus (32,768+16,384+8,192+4,096+2,048+1,024+512+256=65,280) for the left eight bits. Therefore 65,280 (for the left eight) plus 255 (for the right eight), equals 65,535—the base-10 number that represents the largest 16-bit binary number.

Split-octal notation breaks down each "half" of a 16-bit binary number into three base-10 numbers. The upper eight bits are displayed as the first three decimal numbers and the lower three decimal numbers represent the lower eight bits (see the above example). In each of these eight-bit groups (the lower and the upper eight-bit bytes), three decimal numbers represent specific portions of the binary number as follows:

One digit for the 1st, 2nd, and 3rd places
One digit for the 4th, 5th, and 6th places
One digit for the 7th and 8th places

If we divide the 16-bit binary number up so that each "group" is represented by one decimal number, it will take on the following form:

11-111-111 11-111-111

Notice that each "half" has two groups of three bits and one group of two bits. In split-octal notation, each group is treated as an individual binary number, and listed by its decimal equivalent. Thus, the binary number 00-100-000 10-101-011 is broken down into decimal groups (from left to right for the upper eight bits): 00=0, 100=4, 000=0; (and now for the lower eight bits), 10=2, 101=5, and 011=3. Thus, equivalent of the binary:

00-100-000 10-101-011

is the split-octal number:

0-4-0 2-5-3

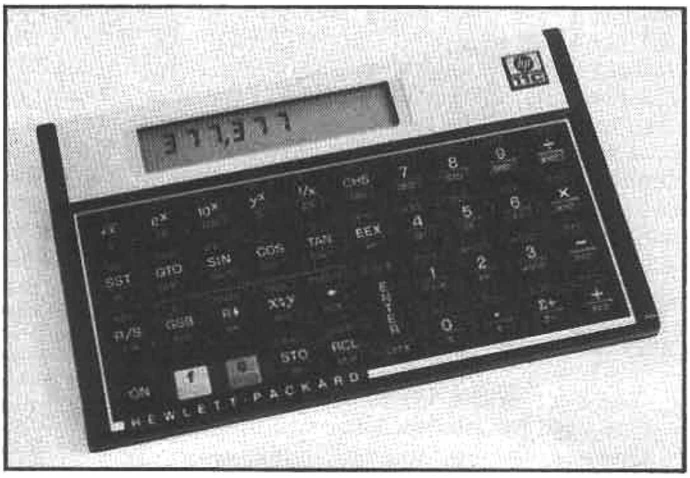
The highest binary number (11111111 11111111) thus becomes 377 377 in split-octal form (11-111-111 11-111-111=3-7-7 3-7-7).

DYNAMIC RAM

The random-access memory chips in a Z/H-89 or Z/H-100 are packed with thousands of tiny circuits. Each can be "on" or "off," therefore each circuit is capable of representing a "1" or a "0." But the computer must electronically "refresh" the memory used in Z/H computers. That is, it must reestablish the current state (a "1" or a "0") by "recharging" each memory cell every few clock cycles (a clock cycle is one two-millionths of a second); if it did not do so, the memory cell would lose track of the bit it was storing. Because each cell requires constant refresh, the memory is known as "dynamic" RAM. (There are other RAM technologies that do not require refresh; one of them is called a "static" RAM, but it cannot be manufactured as dense—with as many bits per chip—or as inexpensively as dynamic RAM.) The refreshing requirement of dynamic RAM explaines (in part) why, when power is switched off, the data in RAM is lost.

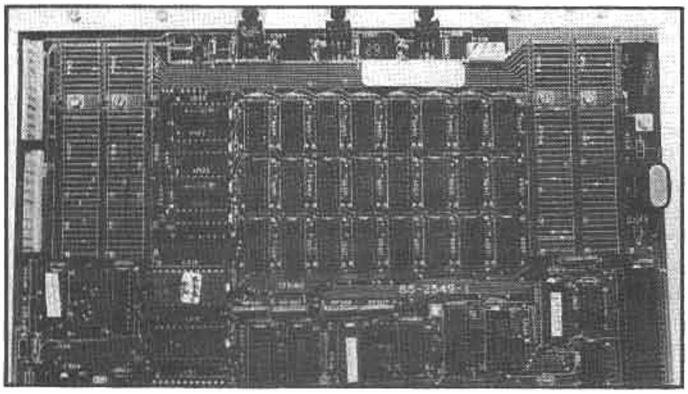
MEMORY MAP

The Z/H-89 uses 24 dynamic RAM chips (on the main CPU board) which are each capable of holding 16,384 bits of information. 1K=1024 and 16K=16,384. Therefore,



STEP	KEY ENTRY	KEY CODE	COMMENTS
01	42,21,12	LBL B	Initilize
02	42,34	CLR REG	
03	48	.	Separate high & low order bytes
04	0	0	
05	0	0	
06	1	1	
07	20	X	
08	36	↑	
09	43,44	INT	Isolate integer
10	44,2	STO 2	Store high order byte
11	33	↓	
12	42,44	FRAC	Isolate fraction
13	1	1	
14	0	0	
15	0	0	
16	0	0	
17	20	X	
18	44,1	STO 1	Store low order byte
19	3	3	Decimal place position
20	44,25	STO I	Store decimal place position
21	1	1	Digit value
22	44,3	STO 3	Store digit value
23	32,4	GSB 4	Analyzing subroutine
24	8	8	Digit value
25	44,3	STO 3	
26	32,4	GSB 4	
27	6	6	Digit value
28	4	4	
29	44,3	STO 3	
30	32,4	GSB 4	
31	45,2	RCL 2	Analyze high order byte
32	44,1	STO 1	
33	3	3	
34	44,25	STO I	Store decimal place position
35	2	2	Digit value
36	5	5	
37	6	6	
38	44,3	STO 3	Store digit value
39	32,4	GSB 4	
40	2	2	Digit value
41	0	0	
42	4	4	
43	8	8	
44	44,3	STO 3	
45	32,4	GSB 4	
46	1	1	Digit value
47	6	6	
48	3	3	
49	8	8	
50	4	4	
51	44,3	STO 3	
52	32,4	GSB 4	
53	45,4	RCL 4	Recall the result
54	42,7,0	FIX 0	Display the result
55	43,32	RTN	END
56	42,21,4	LBL 4	Analyzing subroutine
57	45,1	RCL 1	Recall byte for analysis
58	48	.	Isolate individual digit
59	0	0	
60	0	0	
61	1	1	
62	20	X	
63	45,25	RCL I	Recall digit place position
64	13	10^x	
65	20	X	
66	43,44	INT	
67	48	.	
68	1	1	
69	20	X	
70	42,44	FRAC	
71	1	1	
72	0	0	
73	20	X	End of digit isolation
74	45,3	RCL 3	Recall digit value
75	20	X	
76	44,40,4	STO + 4	Add to total
77	42,5	DSE	Decrement place counter
78	43,32	RTN	
79	43,32	RTN	

A programmable handheld calculator is helpful to an advanced user who regularly works with computer number systems (binary, split-octal, etc.). With the program listed here, the popular Hewlett-Packard HP-11C can convert any split-octal number (up to six digits) into its decimal equivalent. The user simply enters the split-octal number (the program does not check for an illegal entry) and selects the label-B function.



These are the three 16-Kbyte banks of RAM on the Z/H-89 CPU board. Because the bottom 8 Kbytes of the memory map contains special system RAM and the monitor (in ROM), the rows of chips do not correspond to the first, second, and third 16-Kbyte banks of system RAM; instead, they are the first, second, and third 16-Kbyte banks of *user-available RAM*.

these chips are called "16K RAM chips." They are arranged in three "banks" (rows) of eight chips each, so each bank is responsible for 16,384 8-bit bytes, or 16K bytes. The three rows of 24 (total) chips combine to provide 48 Kbytes (49,152) of RAM memory. An additional 16 Kbytes can be added—for a maximum of 64 Kbytes—with a plug-in circuit board containing eight more 16K chips. With this addition, the eight additional 16K RAM chips are, conceptually, a fourth "bank," bringing the total to 65,536 bytes. However, since the first byte in RAM is the binary number 00000000 00000000, equal to the decimal zero, the highest decimal number that can be counted with 16 bits (when zero is the starting point) is given by the formula (16K X 4)-1=65,535 (an odd number). This is the decimal equivalent of the highest available memory location, or "address," with four banks of 16K RAM.

In common computer terminology, 64K is equal to the decimal number 65,535. This fact is often the source of confusion. For example, some computer vendors advertise that their machines have "65K of memory," and tout this as being somehow better than those having "only 64K." In reality, however, their system is no different; the difference is in the nomenclature. A microprocessor (like the Z80 used in the Z/H-89) that uses a 16-bit address buss, can not address more than 65,535 bytes (remember that a 16-bit binary number can "count" no higher than the decimal 65,535 when the starting point is zero). The Z/H-89 and the H-8 computers are fully capable of using all 65,535 bytes. The "65K" trick confuses "k" (meaning 1000 in metric notation) with "K" which is a computer term for 1024. Strange as it seems at first, 65,535 equals 65.535k, which equals 64K.

The organization of those 49,151 or 65,535 addresses is controlled by a memory-map decoder chip. But before exploring the organization of RAM and the function of the memory-map decoder, one additional concept is required: the system monitor.

THE MONITOR

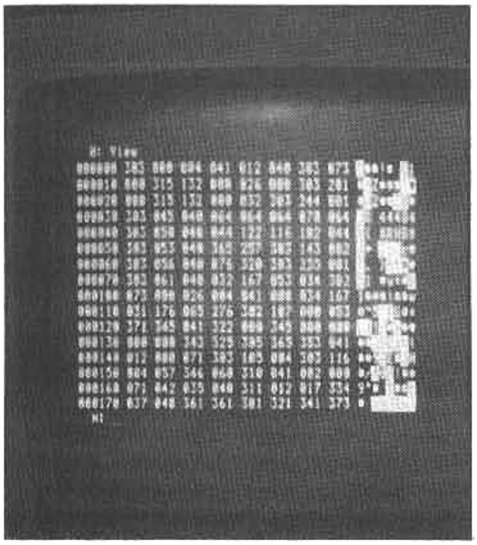
When the user switches on the Z/H-89, or presses SHIFT-RESET, the screen clears and the letter "H," followed by a colon, appears in the upper-left corner of the CRT. Indicating that the system is ready to begin work, the "H:" (called the "H-prompt") is produced by a program stored in the monitor ROM; this program controls the basic operation of the system (usually until the disk operating system is booted), and is called the monitor.

The monitor is what the Z/H-89 uses to start everything from scratch. It is a program, similar in some ways to other computer programs, but instead of being stored on a disk or a cassette tape, it is permanently stored in the computer in a ROM chip. The ROM with the Z/H-89 monitor program in it is called MTR-89; the one in the Z-90 is MTR-90 (which is also supplied with the Z/H-37 high-density disk upgrade kit). When power is switched on (or the SHIFT-RESET is pressed), the monitor "tells" the terminal logic board to clear the screen and displays the "H:." Then the monitor waits for the user's instructions.

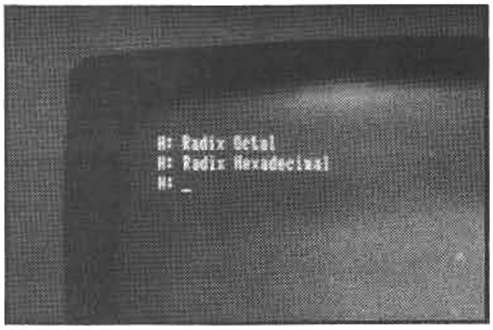
MONITOR COMMANDS

"Booting" an operating system from a disk is a familiar job, even for novice users, so many people are not aware of other operations that can be performed at the appearance of the "H:." There are four commands that the monitor MTR-89 can execute for the user, they are:

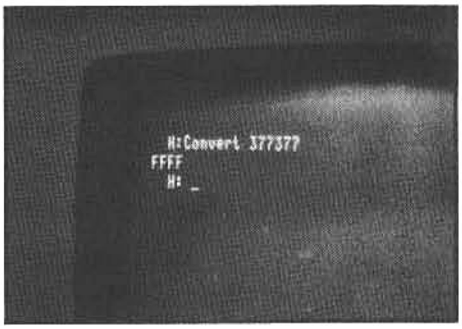
B (boot) is used to "bootstrap" the disk operating system (DOS) into computer RAM and then transfer operation of the computer to the DOS. When the user presses "B," the monitor program completes the word with "oot" and waits for the user to press RETURN. This technique of word-completion is a way of being "user-friendly," but it also serves as an unambiguous indicator of the ac-



```
H: View
000000 303 000 004 041 012 040 303 073
000010 000 315 132 000 026 000 303 201
000020 000 315 132 000 032 303 244 001
000030 303 045 040 004 064 064 070 064
000040 303 050 040 044 122 116 102 044
000050 303 053 040 365 207 303 143 002
000060 303 056 040 076 320 303 333 001
000070 303 061 040 032 167 053 034 302
000100 073 000 026 004 041 000 034 167
000110 031 176 065 276 302 107 000 053
000120 371 345 041 322 000 345 000 000
000130 000 000 343 325 305 365 333 041
000140 012 000 071 303 105 004 303 116
000150 004 037 346 060 310 041 002 000
000160 071 042 035 040 311 032 017 334
000170 037 040 361 361 301 321 341 373
H:
```



```
H: Radix Octal
H: Radix Hexadecimal
H: _
```



```
H:Convert 377377
FFFF
H: _
```

These photos illustrate some of the new commands that have been incorporated in MTR-90, the monitor for the Z/H-37 soft-sector controller. The left photo shows the result of the "VIEW" command; it will display the contents of memory in either octal or hex. The "RADIX" command is shown in the middle; it will select between the octal or hex mode. On the right is the "CONVERT" command, a built in number converter; split-octal is converted into either octal or hex. In addition, MRT-90 will accept "IN" and "OUT" commands (communication with ports) and will enter the memory test routine by pressing "T" for "TEST MEMORY."

tion to be performed. When the return key is struck, the monitor directs the CPU to begin reading the DOS from disk and transfer it to a specific location in RAM. When the initial transfer of DOS is complete, the monitor directs the CPU to "jump" to the starting location of the DOS (in RAM).

S (substitute) is used to examine and modify the contents of any location within the range of the CPU (all 64Kbytes). When the user types "S," the monitor completes the word with "ubstitute" and waits for the user to enter the desired memory location, which must be written in split-octal form. The user enters an address and then presses RETURN. The monitor will then display the contents of this location. The contents may be changed by entering the new data or, if the SPACE bar is pressed, the contents of the next location (in numerical sequence) will be displayed. If the "-" (minus) key is depressed, the monitor will display the previous location; pressing the RETURN key will send the monitor back to the "H:" prompt again. If any of the memory locations that are addressed with the substitute command are in ROM (such as the monitor locations) they can only be read from and not written to.

P (program counter) tells the monitor to remember a particular location in memory; usually this location is the beginning of a program to which the user wishes to transfer operation. When "P" is entered, the monitor completes the word with "rogram counter" and waits for the user to enter the memory location. The user can change the program counter value at this point by entering a new number, or can leave it unaltered by pressing the RETURN key.

G (go) is used to start running a program. When "G" is typed, the monitor completes the word with an "o" and waits for the user to type the address (in split-octal notation) where the program is to start. If no address is entered and the user presses RETURN after the "Go," the monitor will transfer the CPU to the memory location stored in the program counter (which has been set with the "P" command).

There are two additional monitor commands which may be found in Heath's MTR-88—a monitor program for computers with cassette tape storage devices. These commands are: "L" (load), which loads programs from a tape into RAM, and "D" (dump), which sends the contents of RAM back to the tape. These commands are not found in MTR-89. If you are not sure which monitor program your computer has, enter "D" after the "H:" prompt comes up; if the word "DUMP" appears, you have MTR-88. The MTR-88 will allow the use of cassette tape (a marginal value), but will not allow more memory than 48 Kbytes or the use of CP/M. Your machine can be upgraded simply by removing the MTR-88 EPROM chip and replacing it with the MTR-89 chip, available from authorized Zenith and Heath dealers. The MTR-89 will allow 64 Kbytes of RAM and the use of CP/M, but it will not allow the use of the high-density Z/H-37 5.25-inch drives. If the Z/H-37 upgrade is installed the system must use the MTR-90 (which accepts all the MTR-89 commands plus four more—see accompanying illustrations).

DIAGNOSTIC TESTS

A series of tests is used by Heath/Zenith technicians to determine if all the systems are functioning properly and working together harmoniously. If you are careful, you can run these tests yourself, although you may not be able to correct problems without professional assistance. These tests will merely confirm that everything is running correctly. They do not necessarily diagnose specific causes of failures, since this requires test equipment.

THE ASCII DISPLAY

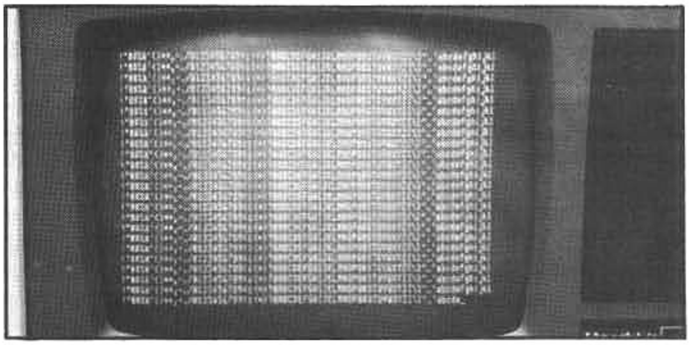
This test is actually a short program, written in the "machine" language that the computer understands. It uses monitor commands to load the program into RAM, and then transfers CPU operation to that portion of RAM containing the test program. The ASCII test program produces a continuous display of characters on the CRT screen. It is a very simple test to determine if the terminal logic board (TLB) is displaying every ASCII character correctly, but the Zenith/Heath service manual says the test is "primarily intended to reassure the customer that their (sic) computer is working properly."

The display test is performed by the following steps:

1) **SHIFT-RESET** the computer and, when the "H:" prompt appears, type **S**

2) Enter the memory location **040100** and press **RETURN**

3) When the computer echoes "040100" plus a a three-digit number representing the current byte at that address, type **076** and then hit the **SPACE** bar.



The ASCII-display CPU test (described in the text) produces this result. The test reassures the user that the CPU and terminal section are functioning properly.

4) When the computer displays the next memory location and its contents, enter the rest of the numbers in the same way:

011 SPACE
315 SPACE
302 SPACE
003 SPACE
074 SPACE
376 SPACE
155 SPACE
312 SPACE
100 SPACE
040 SPACE
303 SPACE
102 SPACE

040 SPACE RETURN (i.e., hit **RETURN** after the **SPACE** bar.)

5) Now go back, using the "S" command to check your work. Enter the starting address (040100) and then *one by one* check the contents of each address; if the contents are correct, use the SPACE bar to go on to the next. If they are incorrect, change them by typing the correct number and then pressing SPACE to go on.

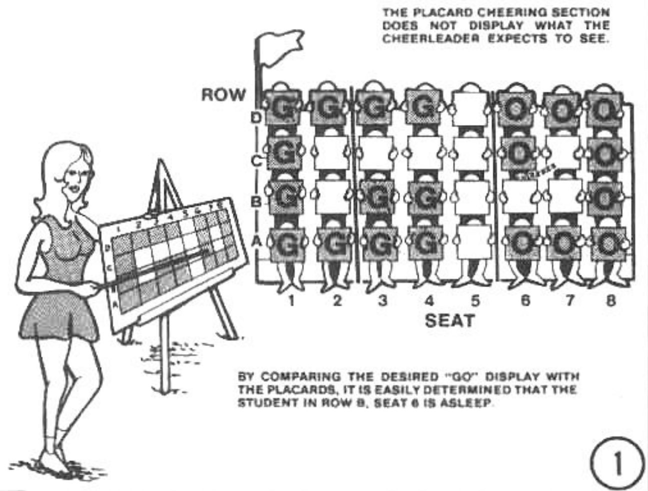
6) Now **SHIFT-RESET** and, when the "H:" prompt appears, type **G** and the number **040100**. Press **RETURN**. This tells the monitor to begin running a program that it will find, starting at memory location 040100. That program was written by you in the form of the numbers you entered.

7) If all goes well, the computer will display all the ASCII characters on the screen, scrolling them over and over again until you stop them. You may do this by pressing **SHIFT-RESET.**

THE "WALKING-BIT" RAM TEST

The built-in memory test for the Z/H-89 and -90 (initiated by typing **Go 7375** at the H prompt) can detect faulty *memory cells*, and with proper interpretation of the test results, the user can determine exactly which integrated circuit (IC) memory chip is defective.

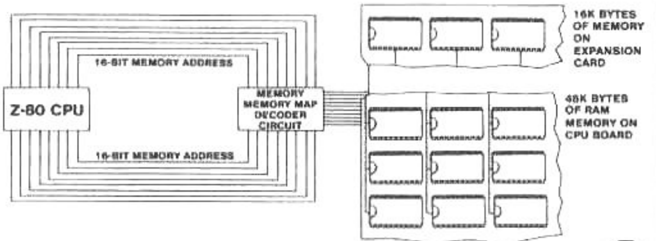
"RAMTEST" FOR A SLEEPING STUDENT


THE PLACARD CHEERING SECTION DOES NOT DISPLAY WHAT THE CHEERLEADER EXPECTS TO SEE.
ROW
D
C
B
A
G
O
1
2
3
4
5
6
7
8
SEAT
BY COMPARING THE DESIRED "GO" DISPLAY WITH THE PLACARDS, IT IS EASILY DETERMINED THAT THE STUDENT IN ROW B, SEAT 6 IS ASLEEP.
1

The arrangement of memory chips can be likened to a series of students in a college flash-card cheering session. The students are arranged in rows and columns (like the memory chips on the CPU board), and the students are given flash-cards that are lettered on one side and white the other. Each student is given instructions, telling him whether to hold the lettered or white side up when instructed. Suppose, however, that the designated pattern spells the word "GO," but when the cheerleader calls for this pattern, one student fails to display the proper card and the pattern is faulty. A seat-by-seat comparison is made with the cheerleader's diagram to find the row and column location of the sleeping student, and when located (row B, seat 6), he is "gently awakened." This is roughly the same pattern of events used to find a bad memory chip.

The memory test writes a specific code (either a 1 or a 0) to each memory cell in each row (bank) of memory chips, and sequentially reads the memory cells and compares the retrieved code with the written code; any discrepancy indicates a faulty memory element.

THE MEMORY-MAP DECODER

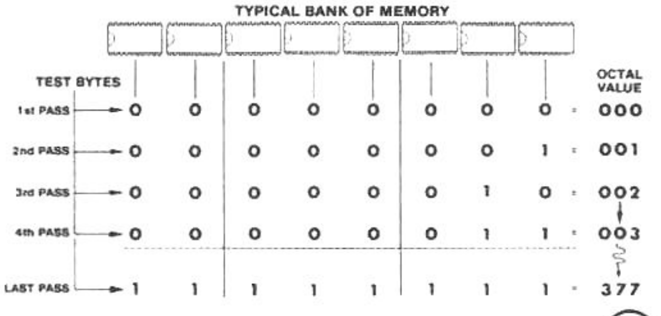


THE MEMORY-MAP DECODER CIRCUIT INTERPRETS THE 16-BIT ADDRESS FROM THE CPU AND DIRECTS DATA FLOW TO AND FROM THE APPROPRIATE MEMORY CHIP.

When the memory test program reads or writes data to individual rows of memory chips, a circuit called the *memory-map decoder* ensures that the flow of data is directed to the appropriate bank, chip, and memory cell.

The 16 memory-address lines from the Z80 CPU (capable of addressing up to 64 Kbytes) carry an individual address from the CPU to the memory-map decoder, which provides access to the specific byte of data stored at that memory address.

48 Kbytes of memory are located on the CPU board in a three-row deep by eight-column wide "block" of ICs, and 16 Kbytes of additional memory is available on a plug-in expansion card. The memory expansion contains, in effect, one row of eight ICs (although the chip arrangement on some accessory expansion cards is not this straightforward). The accompanying illustrations show the 48 Kbytes of memory on the CPU board as the lower block of 24 chips, and the 16K bytes on the expansion board as a separate row of eight chips above the CPU board.

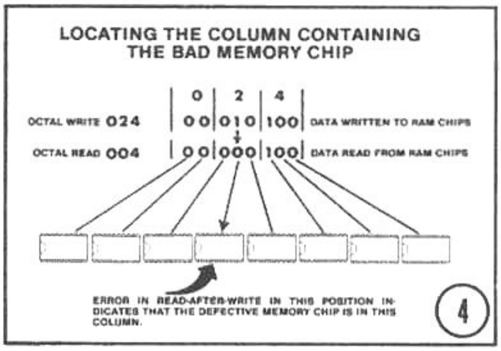


INCREMENTING BINARY TEST BYTES

The memory test routine writes a bit (a 1 or a 0) to each memory cell by sending an 8-bit byte—*with each bit assigned to one of the eight chips in the row (one bit per column)*—starting with *00000000* and incrementing by one until *11111111* is reached. This constantly changing pattern of bits ensures a thorough test of every memory cell in every chip, since after "filling" of memory, a test comparison (readout) is made to determine that each and every cell has properly retained either a high (1) or low (0) bit.

Referring to illustration #3, for simplicity let's consider just one row (bank) of RAM chips. (However, all banks are tested as a group, just like the one in this example.) On the first test pass, the computer writes a 00000000 test byte to all of the 16K memory locations in the row (bank), then it reads back the data byte-by-byte, and the results are compared with the original test byte (i.e., 00000000). If the results match, the test byte is incremented to 00000001 and rewritten back. On the second pass, each byte location is read and checked against 00000001, and if it matches, the test byte is again incremented, and a 00000010 is written. If no errors are detected, this process continues until, at the last pass, the incremented test byte reaches 11111111. The memory test will then repeat, starting again with 00000000.

Note in the drawing that the rows of chips are divided up into three groups. The right-most group is three chips wide, the middle group is three wide, and the left-most group is two chips wide. The right group represents the right-hand octal number value (that is, in the octal number XYZ, "Z" is the right-hand octal number, with its value represented by the right-hand group of three chips). The middle group represents the next octal digit ("Y"), and the left group equals the left-hand octal digit ("X"). For example, on the last pass of the memory test (the test byte is 11 111 111) the right-hand and middle groups each total "7" (1 plus 2 plus 4) and the left group totals "3" (1 plus 2) for an octal value of "377." The least-significant bit of the binary test byte is stored in the right-most chip location (right column) and the most-significant bit is stored in the left-most chip (left column). Armed with this information and the memory test routine (plus a knowledge of the split-octal number system), it is possible to determine which chip contains a defective memory cell.


LOCATING THE COLUMN CONTAINING THE BAD MEMORY CHIP

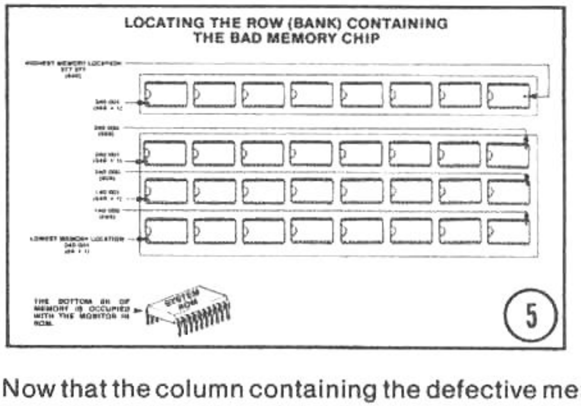
0
2
4
OCTAL WRITE 024
00 010 100
DATA WRITTEN TO RAM CHIPS
OCTAL READ 004
00 000 100
DATA READ FROM RAM CHIPS
ERROR IN READ-AFTER-WRITE IN THIS POSITION INDICATES THAT THE DEFECTIVE MEMORY CHIP IS IN THIS COLUMN.
4

Let's assume that during a memory test the binary test byte was successfully incremented to *00 010 100*. This is equivalent in value to the octal number 024, indicating that the octal value 024 was written to all memory locations. Let's further assume that the read-after-write portion of the memory test read an 004 instead of an 024 at memory address 140 000 (split-octal notation). The CRT screen would display this error message:

```
Dynamic RAM test
    LWA = 377377
    Pass = 024
    ERROR @ 140000 = 004
```

The second line in the display indicates that the highest address of memory is 377 377 (when 64 Kbytes of memory are installed). The third line gives the octal value of the binary byte written to all memory locations, and the last line indicates that an error occurred at the address 140 000. The byte read at that location was 004 (should have been 024).

Referring to drawing #4, notice that when the error byte (004 octal) is converted to binary, the 1 in the fifth column (from the right) was read back as a 0 (it was written as a binary 1 in the octal byte 024). This indicates that a chip with a bad memory cell is located in the fifth column.


LOCATING THE ROW (BANK) CONTAINING THE BAD MEMORY CHIP

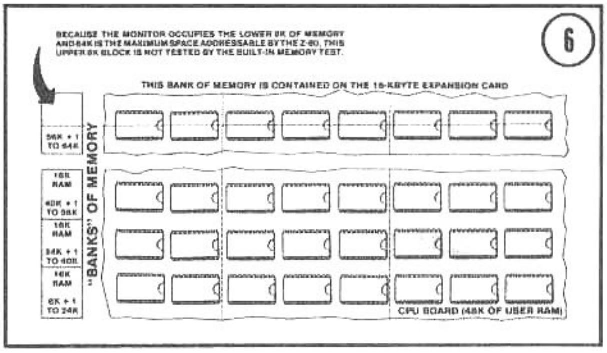
HIGHEST MEMORY LOCATION
LOWEST MEMORY LOCATION
THE BOTTOM 8K OF MEMORY IS OCCUPIED WITH THE MONITOR IN ROM.
SYSTEM ROM
5

Now that the column containing the defective memory chip has been located, the row (bank) must be located. This can be determined by the relationship between memory addresses and the physical location of those addresses on the CPU board (and the expansion board).

Drawing #5 illustrates the memory map of the Z/H-89 and -90 computers. The bottom 8K of memory (zero to 8K) is occupied by the system monitor located in ROM (the memory test we are running is part of this monitor). The bottom row of chips on the CPU board contains the next 16 Kbytes in address space; from 8K+1 to 24K (or 040 001 to 140 000 in split-octal). The next row occupies the next 16K block and so on. The error we are looking for occurred at the split-octal address of 140 000. This is the very last byte addressed by the bottom row of memory chips. The defective chip is therefore in the bottom row, the fifth chip from the right.

If the memory test locates a defective memory chip in the last 8K bank of memory on the expansion board, it may be difficult to pinpoint which chip is faulty because of the physical arrangement of chips on that board. The simplest solution may be to swap all the memory chips on the expansion board with those in bank 2 or 3 and then rerun the memory test. When the faulty chip is located in those banks, it is easily isolated.

The highest address on the CPU board is 56 Kbytes or 340 000 split-octal. When the 8K address space of the monitor is subtracted from this high address, the result is 48 Kbytes; the maximum *user addressable memory* on the CPU board (the 8K monitor is not considered *user* space).



6
BECAUSE THE MONITOR OCCUPIES THE LOWER 8K OF MEMORY AND 64K IS THE MAXIMUM SPACE ADDRESSABLE BY THE Z-80, THIS UPPER 8K BLOCK IS NOT TESTED BY THE BUILT-IN MEMORY TEST.
THIS BANK OF MEMORY IS CONTAINED ON THE 16-KBYTE EXPANSION CARD
56K + 1 TO 64K
16K RAM
40K + 1 TO 56K
16K RAM
24K + 1 TO 40K
16K RAM
8K + 1 TO 24K
"BANKS" OF MEMORY
CPU BOARD (48K OF USER RAM)

There is another interesting point to consider. Since the Z80 can only address 64 Kbytes and the highest memory address on the CPU board is already 56K, there are only 8 Kbytes more that can be added to the memory map. And, of the 16 Kbytes that reside on the expansion board, only 8K are accessible to the Z80 (remember the bottom 8K of memory is occupied by the system monitor). This means that the upper 8K of the 16K expansion board cannot be tested by the built-in memory test, but since it is not addressed by the Z80, does it matter whether it is tested or not? When running HDOS, no! But when the computer is booted up with CP/M, things are different.

Under CP/M the system monitor cannot be a part of the memory map, because CP/M is designed to run at origin zero. That is, it expects to have continuous memory starting at location 000 000. So Zenith engineers have designed a special memory-map ROM that switches out the monitor upon CP/M boot-up and uses the *upper 8K block* of the memory expansion board as the replacement for the previously ROM-addressed *lower 8K block*. However, since this upper 8K block on the expansion board is not tested by the built-in memory test, the CP/M user should consider running a more sophisticated memory test program (one that relocates itself as the memory test is underway) to ensure trouble-free operation.

THE "WALKING-BIT" TEST

It happens, though not very frequently, that one of the RAM chips can fail. This is a very subtle failure, because it affects only a tiny part of the whole, available memory. Nonetheless, it must be isolated and replaced. The text in the accompanying box takes a close look at a built-in RAM test that is included with all Z/H-89 computers.

This RAM test is one of the very first that Zenith and Heath technicians run when a user brings in a misbehaving computer. The user may, of course, run the test, but interpreting the results is not always straightforward. According to Mark Groeger, technical training manager at Zenith's training center in Chicago, "This test will detect some 80 to 90 percent of memory failures. But it will not necessarily reveal a 'slow' or intermittent chip. A more vigorous test, like the ones discussed next, can detect up to 98 percent of failures. Even so, no single test can find every defect."

The most difficult faults to locate, of course, are intermittent or pattern-fault problems. These may show up only once in a while, or only after the chips grow hot from long periods of use. Some faults occur only when a particular bit-pattern sequence is read into the chip, but not otherwise. These are frustrating even to a professional service technician.

"Generally," Groeger says, "we run this test before checking any other part of a Zenith computer. However, even a perfect set of memory chips may fail this test if the memory-map decoder chip is faulty. But if the problem continues, despite a memory-map decoder chip swap, then the problem is usually in the chips themselves."

THE MOST COMMON HARDWARE PROBLEMS

Here is a list of the most common hardware problems of the Z/H-19 terminal, and the Z/H-89 and 90 computers, as they have been reported to Zenith's technical training center. In running diagnostic tests, Groeger and his staff have learned two important lessons: "Never use the customer's software for troubleshooting" and "If a machine passes the tests on our diagnostic disk, then—as far as Zenith is concerned—the hardware is OK and the customer has a software problem." If the user is running Zenith Data Systems' software, it will have come with a telephone number for help. Most professional-quality software comes with similar "hotlines."

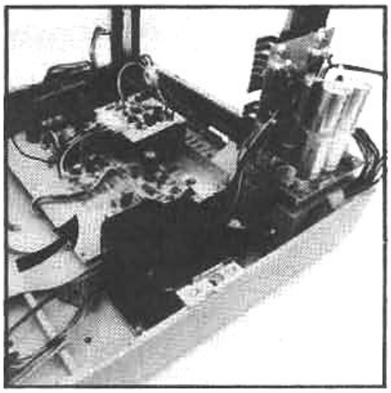
Some 80 percent of all hardware malfunctions can be attributed to failures located on the video board, the terminal logic board, and the serial I/O board.

THE VIDEO BOARD

Within the video board, the horizontal output transistor, the 53-volt regulator transistor and the vertical-output transistors account for most of the failures. The individual components can be replaced, but the fastest remedy is to replace the whole board. This is the usual procedure at most Zenith Service Centers and Heath Electronic Centers.

THE TERMINAL LOGIC BOARD

As was mentioned in the previous chapter, static electricity is notorious for causing chips to fail. The keyboard encoder chip, mounted on the terminal logic board, was—until 1981—very susceptible to static discharge, usually from a user's finger on the keyboard. A simple additional ground connection has since been added to new machines, and retrofitted to older ones that experienced this problem.



The video board controls the "television-like" functions of the terminal CRT. It regulates focus and brightness; places the display in the center of the screen; synchronizes the display (a lack of "sync" causes the display to roll); etc. Trouble with any CRT function can usually be traced to this circuit board.

Another terminal logic board failure has been noticed at the tie between the asynchronous communications element (8250 ACE) and the converter chips: it is subject to abuse by improper installation of interfaces. The user should be forewarned when connecting the system to "alien," external equipment. Testing the line with an interface "breakout" box is helpful (see chapter 7).

The CRT controller chip is also on the terminal logic board, and has a failure rate higher than some of the other components on the board.

THE SERIAL I/O BOARD

The most failure-prone devices on the serial I/O are the ACE converter chips. These circuits directly "see" the outside world and can be damaged by the incorrect signal levels. Proper installation of interface devices eliminates this problem.

OTHER FAILURES

Most of the remaining 20 percent of failures can be attributed to the CPU board, on which no single component is more likely than any other to fail. Neither the power supply, which includes a printed-circuit board with diode rectifiers, and regulators mounted on a heat sink, nor the disk-controller board, nor the disk drive itself are prone to failure. The original-equipment disk drive is not manufactured by Zenith: it is made by an outside supplier under contract. If it is properly installed, failure of this unit—a Siemens FDD 100-5B—is virtually unheard-of.

DISK I/O

All floppy disks record and play back digital signals by selectively magnetizing portions of the specially-treated metal-oxide surface of a spinning mylar plastic disk. The head (which is like the record/playback head of a tape recorder) records data in concentric circles, called tracks, on the disk surface (rather than in a spiral, as on a phonograph record). The computer disk-controller board directs the disk-drive head to move across the disk surface, "stepping" from track to track to record or read data.

The outside tracks are "longer" and would appear to be able to hold more data than the shorter, inside tracks. However, since it takes the the same amount of time to rotate through one revolution (track) whether that track is located on the inside or outside, and because data is written to the disk surface in equally timed electrical pulses, the amount of data storage is the same for the inside and

5.25-inch HARD SECTOR DISKETTES

HEATH/ZENITH CONTROLLER	RECORDING DENSITY	TRACK DENSITY	READ/WRITE DISK SIDES	TRACKS /INCH	SECTORS /INCH	BYTES PER SECTOR	KBYTES FORMATTED STORAGE
Z-17	SINGLE	SINGLE	ONE	48	10	256	90
Z-17	SINGLE	SINGLE	BOTH	48	10	256	180
Z-17	SINGLE	DOUBLE	ONE	96	10	256	180
Z-17*	SINGLE	DOUBLE	BOTH	96	10	256	350
5.25-inch SOFT SECTOR DISKETTES							
Z-37	**SINGLE**	**SINGLE**	**ONE**	**48**	**10**	**256**	**90**
Z-37	SINGLE	SINGLE	BOTH	48	10	256	184
Z-37	SINGLE	DOUBLE	ONE	96	10	56	190
Z-37	SINGLE	DOUBLE	BOTH	96	10	256	388
Z-37	**DOUBLE**	**SINGLE**	**ONE**	**48**	**16**	**256**	**148**
Z-37	**DOUBLE**	**SINGLE**	**BOTH**	**48**	**16**	**256**	**304**
Z-37	DOUBLE	DOUBLE	ONE	96	16	256	308
Z-37	**DOUBLE**	**DOUBLE**	**BOTH**	**96**	**16**	**256**	**624**
Z-37	EXTENDED	SINGLE	ONE	48	5	1024	182
Z-37	EXTENDED	SINGLE	BOTH	48	5	1024	382
Z-37	EXTENDED	DOUBLE	ONE	96	5	1024	386
Z-37	EXTENDED	DOUBLE	BOTH	96	5	1024	782
MAGNOLIA CONTROLLER MMS	DOUBLE	SINGLE	ONE	48	9	512	162
MMS	DOUBLE	SINGLE	BOTH	48	9	512	342
MMS	DOUBLE	DOUBLE	ONE	96	9	512	340
MMS	DOUBLE	DOUBLE	BOTH	96	9	512	700
8-inch SOFT SECTOR DISKETTES							
Z-47	**SINGLE**	**SINGLE**	**ONE**	**48**	**26**	**128**	**241**
Z-47	SINGLE	SINGLE	BOTH	48	26	128	490
Z-47	**DOUBLE**	**SINGLE**	**ONE**	**48**	**26**	**256**	**482**
Z-47	**DOUBLE**	**SINGLE**	**BOTH**	**48**	**26**	**256**	**980**
Z-47	EXTENDED	SINGLE	ONE	48	8	1024	596
Z-47	EXTENDED	SINGLE	BOTH	48	8	1024	1208
IBM STANDARD	**SINGLE**	**SINGLE**	**ONE**	**48**	**26**	**128**	**241**
MAGNOLIA CONTROLLER MMS	DOUBLE	SINGLE	ONE	48	16	512	594
MMS	DOUBLE	SINGLE	BOTH	48	16	512	1210
MAGNOLIA CONTROLLER M-47	EXTENDED	SINGLE	ONE	48	8	1024	594
M-47	EXTENDED	SINGLE	BOTH	48	8	1024	1210

* Available from Magnolia Microsystems

Floppy disk formats vary according to data density and track density, and may be either single or double-sided. Some of these formats are unique to Z/H disk systems, and other than the single-side, single-density 8-inch format (the IBM 3740 format), none of them can be considered "standard." This chart illustrates the formats that a user is likely to encounter, with the more common ones listed in bold print.

outside tracks. However, the data on the inside tracks is more densely packed, with the magnetized spots on the surface placed closer together.

The disk-controller electronics divide the disk into segments, called sectors. These pie-shaped portions of each track are defined by the disk controllers in two common ways: a fixed, hard-sector indexing; or by a soft-sector indexing.

HARD-SECTOR DISKS

A hard-sector disk can be thought of as a pie whose wedge-shaped sectors are permanently assigned. Their sizes are uniform and unchanging.

On the Z/H-89 system the standard 5.25-inch disks use ten hard sectors. To identify the position of each sector on the disk surface, there are ten (sector) holes in the disk, spaced equally in a ring that surrounds the central drive hole. As the disk spins in the drive, a light-emitting diode (LED) sends a beam of light through the holes, while a light-sensitive receptor keeps track of the flashes that result. This hole-position information is used to define and identify the sectors.

Between two of the sector holes, an extra hole has been made: this is called the index hole, and it tells the disk controller that the "next" hole to appear is the one corresponding to "sector one." The disk controller needs that information to create a unique "address" on the disk for the data located at each sector.

On the standard Zenith and Heath disk drive, there are 40 tracks (i.e., concentric rings) around the drive hole. The recording head is positioned to read or write to each of these tracks through an opening (the head access slot) in the protective cover. The 40 track disk is often called "48 tpi" for 48 tracks per inch, but since the effective surface (that portion of the disk that the drive head can read and write) is less than one inch in length, only 40 tracks are available for data). With ten sectors, there are 400 track-sectors available for data. Each track-sector holds 256 bytes, so the whole disk, when full, will hold 102 Kbytes. For "housekeeping" purposes, however, the disk-controller needs approximately 12 Kbytes of the available space for the recording of "index" information, so the actual storage space available to the user is 90 Kbytes.

This approach to creating sectors is called hard sectoring because the size of the sectors is fixed by the sector holes in the mylar disk, and can not be changed.

SOFT-SECTOR DISKS

On 8-inch disk drives, and on the Z/H-37 5.25-inch drives as well, a different method of sectoring is used. Rather than locating each sector by reading a series of sector-locating holes, another method, called soft sectoring, is under the control of sophisticated software that resides in a ROM on the disk-controller board. A soft-sectored disk has only one hole—the index hole. Every sector is created as it is needed by a sensing circuit that calculates the speed of the disk from the timed appearance of light through the index hole. While it is true that soft sectoring makes more efficient use of disk space, it is achieved by a greater reliance on housekeeping—therefore, there is actually slightly less storage available on a soft-sectored disk than there is on a hard-sectored disk of the same size and density. After formatting a user can store up to about 80 Kbytes on a single-side, single-density disk.

There is no "standard" for 5.25-inch disks as there is for 8-inch disks. In the 8-inch size, single-density, soft-sector recording was standardized by IBM in their 3740 format. In the 5.25-inch market the most common recording method is considered to be single-density, soft-sector recording. One advantage of using soft-sector recording is the ease with which disks can be transferred from one system to another (more on this later).

DENSITY

There are ways to increase the storage capacity of a floppy disk. Besides recording on both sides of the disk (using a double-sided disk drive, not flipping the disk over), the most common method increases the density of the data recorded on each track, and less frequently a greater number of tracks are used. The first method is usually called double-density recording, and the second method is called double track density.

As we have mentioned, about 80 Kbytes can be stored on a single-sided, 5.25-inch soft-sector disk, with normal recording and track density. By increasing the recording density (to double density) 161 Kbytes are available, and by further increasing the storage area by doubling the track density (to 96 tpi) the available space is 322 Kbytes. By using the Z/H-37 disk controller card, the standard

Z/H-89 can be upgraded to accept disks with 161 Kbytes of storage on them. By adding the external cabinet with Z/H-17-4 (Tandon) drives, each external drive (operating with double-density recording, double track density) can store 322 Kbytes. (These drives will not operate properly inside the Z/H-89 cabinet.) The Z/H-17-4 drives will also record on both sides of the disks at the same time without removing and "flipping" the disk over. Using the drives in this mode—double-side, double-density recording, and double track density—each disk will store 640 Kbytes.

STANDARDIZATION OF DISK FORMATS

Unfortunately for the users of 5.25-inch disks, there is no industry-wide standard for track- or bit-density, and hence no way to guarantee that a disk recorded on one computer system can be read by another. However, the most common recording format is single-side, single-density (SS/SD), 48 tpi, using soft-sector indexing. And since the most common operating system is CP/M, a disk recorded under CP/M in a SS/SD, 48 tpi format will most likely be compatible with a similar system.

For 8-inch diskettes, however, the IBM 3740 single-side, single-density standard exists. Any computer (including Zenith and Heath systems) which has a disk controller supporting SS/SD 8-inch disks (and using the same operating system), will be able to read to and write from those disks. This is why most software for microcomputers is distributed on 8-inch SS/SD disks. Zenith/Heath owners are advised to check with a vendor before purchasing software on a 5.25-inch disks; many vendors offer Zenith/Heath format disks, but not all of them do.

TESTING THE ROTATIONAL SPEED OF THE DISK

A 5.25-inch disk should revolve at 300rpm in a normal disk drive. To verify that the speed is correct, Zenith has built into every Z-89 and Z-90 a test program that resides in the monitor PROM. To initiate this test routine:

1) Remove all disks and SHIFT-RESET.

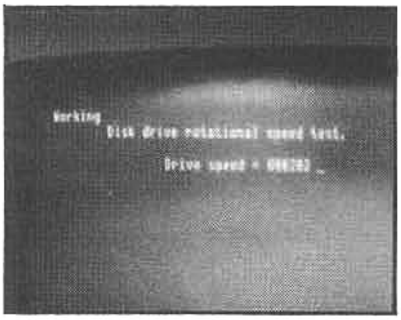
2) Insert a *blank* disk into the on board drive. If the disk has data stored on it, that data may be altered or lost as a result of this test.

3) Type **G (go)**, and the numbers **7372**, and hit RETURN.

4) The computer will now run the rotational speed test. It will display on the screen a number between 166 and 212. (This is not the RPM; it is an octal number representing the RPM.)

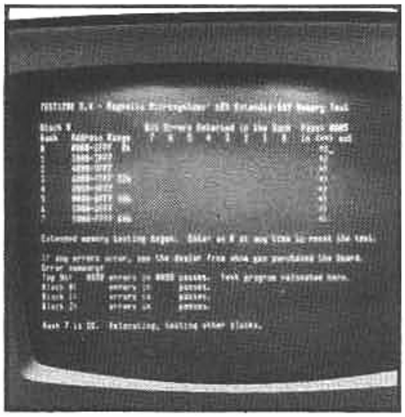
5) The optimal reading for this test should be "200," with an acceptable range of between 176 and 202.

6) Adjustments of the rotational speed are made by "fine tuning" the adjusting screw (its position is indicated in Zenith/Heath literature). If you are unsure about making this adjustment, refer the procedure to an authorized service technician.



```
Working
        Disk drive rotational speed test.

                Drive speed = 000202 _
```

System test software is an important aid for ardent users and system technicians. This screen illustrates an extensive memory test program supplied with Magnolia Microsystems' 128-Kbyte disk-emulator. The program exercises the MMS board and verifies system operation.



```
[illegible]
Extended memory testing begun.  Enter an R at any time to reset the test.

If any errors occur, see the dealer from whom you purchased the board.
Error summary:
[illegible]

Bank 7 is OK.  Relocating, testing other blocks.
```

The monitor program contains a test to check the rotational speed of the onboard disk drive. The drive is within normal tolerance if the displayed number falls between 176 to 202. Outboard drives must be checked with other system test software (e.g., TEST17.ABS).

ADDITIONAL DIAGNOSTIC SOFTWARE

In addition to the memory test and disk-rotational speed tests that are built-in the Z/H-89 system, a user can perform several diagnostic tests by running special software. System test programs are available from several sources; some can be purchased from software writers (see appendix), others are available at no charge from Heath bulletin boards (over the telephone), and some are used by Zenith/Heath technicians (these may be difficult to locate, but personnel at your local Zenith or Heath repair centers may be able to help).

These software tests come in various forms (depending on the source, author, etc.). We present the following list as a reference to some of the diagnostic software that is currently available. These programs can help the knowledgeable user locate some malfunctions, and determine if the problem resides in software (user programs) or hardware (CPUB, TLB, disk drives or controller, etc.).

Test programs are usually written in machine language, and often run under HDOS. They are presented here by program names (which may vary, depending on the source) and will be followed by the "ABS" (if HDOS) or "COM" (if CP/M) extension.

CRTAD is used to adjust the CRT display. The video board is displayed in a graphic representation, with its controls identified. A "ruler" is displayed, for setting height, width, centering, focus, and brightness; and a "grid" is used to adjust for geometric distortions.

ESCAP runs several escape-sequences as a way of ensuring that the terminal logic board is operating properly. The operator answers "Y or N questions" that are displayed. Computer "errors" are listed at the bottom of the screen.

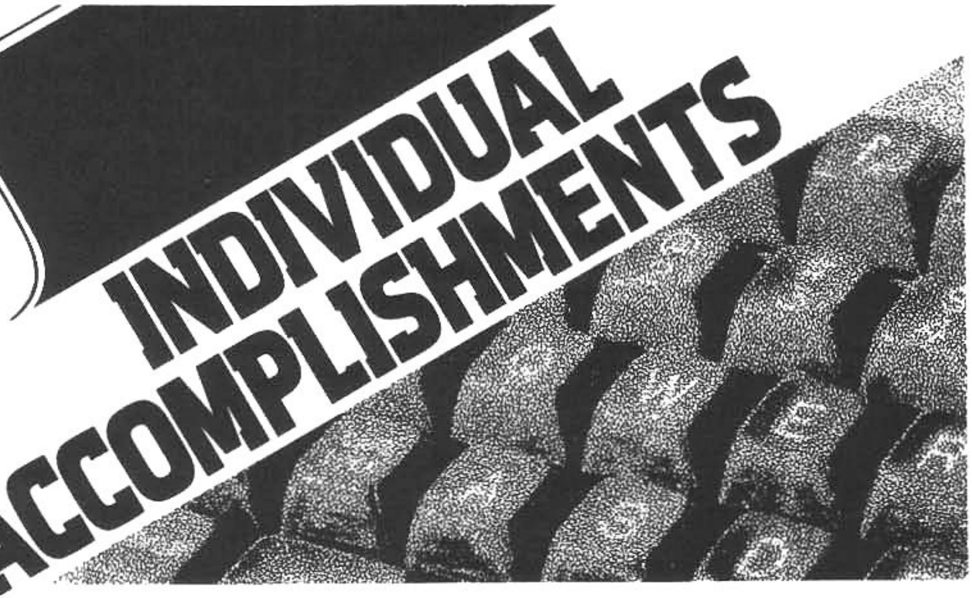
KEYBR checks the operation of the keyboard. Failure of this series of tests indicates a failure of either the keyboard itself or the terminal logic board.

Z89CP contains nine separate tests of the CPU and serial I/O boards. All nine run automatically.

RAMTST checks the RAM chips associated with main memory and floppy disk. Like the user's "walking bit" internal test, a series of data patterns is written to each possible memory location and compared with what is read back. Several versions of this test are available.

TEST17 is a series of tests that check the operation of the disk drive (both the internal and external units). This test program (TEST17) is designed to test hard-sector units, while other versions (TEST37 and TEST47) will test soft-sector disks and drives. These programs are supplied with HDOS and are very helpful to the user in diagnosing disk drive problems.

CHAPTER 9 INDIVIDUAL ACCOMPLISHMENTS



One reason for the success of Zenith/Heath computers has been the ingenuity and resourcefulness of their customers. In this chapter you will meet some of them: people from many walks of life—not just computer "hackers"—who have developed their own uses for these versatile machines.

REAL ESTATE APPRAISALS

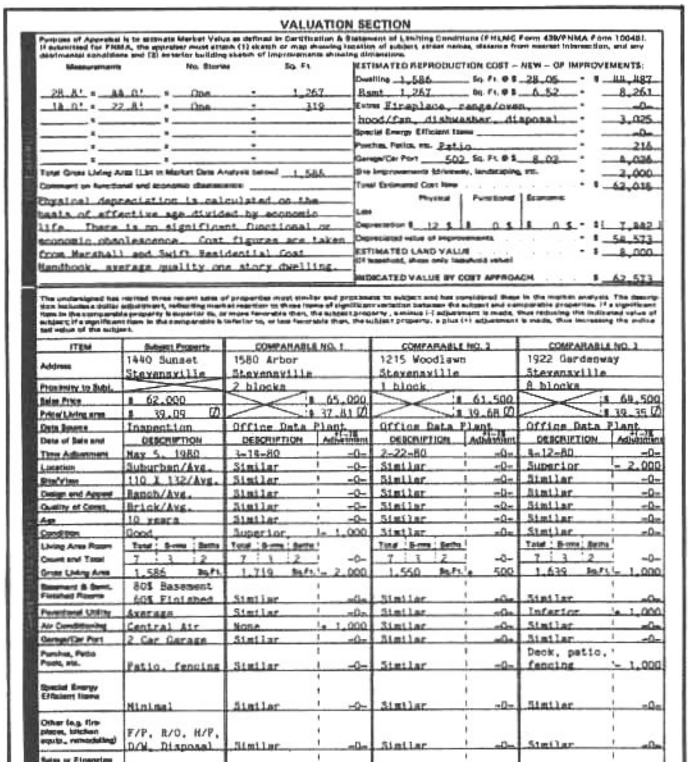
"The computer was an effective tool for writing narrative reports, so we developed our own software for processing forms," says Richard Heyn, of St. Joseph, Michigan, who works not far from the Heath factory. An appraiser and consultant who is not a broker himself, Heyn faced a dilemma a few years ago.

"Most appraisals of single-family dwellings in this country are made on the Federal National Mortgage Assistance (FNMA, or 'Fannie Mae') form 1004. Either we had to enlarge our staff or devise an alternate means of increasing productivity without sacrificing quality. At that time, we extensively researched microcomputer systems, with primary emphasis on word processing and on 'Fannie Mae' forms capability," he says. "We found that nearly all systems offered word processing, but we were unable to find anything in an affordable system with software for the FNMA 1004 form."

"About 50 percent of the narrative forms are boilerplate, and some of the word processing systems were offered as having forms capability, but we learned that this basically amounted to tab setting, with no math functions, automatic defaults to common words, storage of commonly used phrases, etc. We finally decided to buy the highest quality hardware in our price range and have our own software developed."

Heyn decided on a Zenith Z-89, with the Z-37 floppy disk system and a Diablo 630 letter-quality printer. His software is written in Microsoft Basic, and runs under HDOS; it is compatible with Electronic Typing, the word processing package, and was so useful in his appraisal business that he became a consultant to others, and a dealer for the hardware and software as a turn-key package, which he sells for under $7000.

The FNMA forms program includes the ability to retrieve commonly-used phrases and comments, automatic defaults to typical responses, math functions for



VALUATION SECTION

Purpose of Appraisal is to estimate Market Value as defined in Certification & Statement of Limiting Conditions (FHLMC Form 439/FNMA Form 1004B). If submitted for FNMA, the appraiser must attach (1) sketch or map showing location of subject, street names, distance from nearest intersection, and any detrimental conditions and (2) exterior building sketch of improvements showing dimensions.

Measurements		No. Stories		Sq. Ft.
28.8' x	44.0' x	One	=	1,267
14.0' x	22.8' x	One	=	319
x	x		=	
x	x		=	
x	x		=	
x	x		=	

Total Gross Living Area (List in Market Data Analysis below) 1,586

Comment on functional and economic obsolescence: Physical depreciation is calculated on the basis of effective age divided by economic life. There is no significant functional or economic obsolescence. Cost figures are taken from Marshall and Swift Residential Cost Handbook, average quality one story dwelling.

ESTIMATED REPRODUCTION COST – NEW – OF IMPROVEMENTS:

Dwelling 1,586 Sq. Ft. @ $ 28.06	=	$ 44,487
Bsmt. 1,267 Sq. Ft. @ $ 6.52	=	8,261
Extras Fireplace, range/oven,	=	-0-
hood/fan, dishwasher, disposal	=	3,025
Special Energy Efficient Items	=	-0-
Porches, Patios, etc. Patio	=	216
Garage/Car Port 502 Sq. Ft. @ $ 8.02	=	4,026
Site Improvements (driveway, landscaping, etc.)	=	2,000
Total Estimated Cost New	=	$ 62,016
Less Physical | Functional | Economic Depreciation $ 12 $ | $ 0 $ | $ 0 $	=	$(7,442)
Depreciated value of improvements	=	$ 54,573
ESTIMATED LAND VALUE (If leasehold, show only leasehold value)	=	$ 8,000
INDICATED VALUE BY COST APPROACH		$ 62,573

The undersigned has recited three recent sales of properties most similar and proximate to subject and has considered these in the market analysis. The description includes a dollar adjustment, reflecting market reaction to those items of significant variation between the subject and comparable properties. If a significant item in the comparable property is superior to, or more favorable than, the subject property, a minus (–) adjustment is made, thus reducing the indicated value of subject; if a significant item in the comparable is inferior to, or less favorable than, the subject property, a plus (+) adjustment is made, thus increasing the indicated value of the subject.

ITEM	Subject Property	COMPARABLE NO. 1		COMPARABLE NO. 2		COMPARABLE NO. 3	
Address	1440 Sunset Stevensville	1580 Arbor Stevensville		1215 Woodlawn Stevensville		1922 Gardenway Stevensville	
Proximity to Subj.		2 blocks		1 block		8 blocks	
Sales Price	$ 62,000		$ 65,000		$ 61,500		$ 64,500
Price/Living area	$ 39.09 ☐		$ 37.81 ☐		$ 39.68 ☐		$ 39.35 ☐
Data Source	Inspection	Office Data Plant		Office Data Plant		Office Data Plant	
Date of Sale and Time Adjustment	DESCRIPTION	DESCRIPTION	+(–)$ Adjustment	DESCRIPTION	+(–)$ Adjustment	DESCRIPTION	+(–)$ Adjustment
	May 5, 1980	3-14-80	-0-	2-22-80	-0-	4-12-80	-0-
Location	Suburban/Avg.	Similar	-0-	Similar	-0-	Superior	- 2,000
Site/View	110 X 132/Avg.	Similar	-0-	Similar	-0-	Similar	-0-
Design and Appeal	Ranch/Avg.	Similar	-0-	Similar	-0-	Similar	-0-
Quality of Const.	Brick/Avg.	Similar	-0-	Similar	-0-	Similar	-0-
Age	10 years	Similar	-0-	Similar	-0-	Similar	-0-
Condition	Good	Superior	- 1,000	Similar	-0-	Similar	-0-
Living Area Room Count and Total	Total 7 | B-rms 3 | Baths 2	Total 7 | B-rms 3 | Baths 2	-0-	Total 7 | B-rms 3 | Baths 2	-0-	Total 7 | B-rms 3 | Baths 2	-0-
Gross Living Area	1,586 Sq.Ft.	1,719 Sq.Ft.	- 2,000	1,550 Sq.Ft.	+ 500	1,639 Sq.Ft.	- 1,000
Basement & Bsmt. Finished Rooms	80% Basement 60% Finished	Similar	-0-	Similar	-0-	Similar	-0-
Functional Utility	Average	Similar	-0-	Similar	-0-	Inferior	+ 1,000
Air Conditioning	Central Air	None	+ 1,000	Similar	-0-	Similar	-0-
Garage/Car Port	2 Car Garage	Similar	-0-	Similar	-0-	Similar	-0-
Porches, Patio Pools, etc.	Patio, fencing	Similar	-0-	Similar	-0-	Deck, patio, fencing	- 1,000
Special Energy Efficient Items	Minimal	Similar	-0-	Similar	-0-	Similar	-0-
Other (e.g. fire-places, kitchen equip., remodeling)	F/P, R/O, H/F, D/W, Disposal	Similar	-0-	Similar	-0-	Similar	-0-
Sales or Financing Concessions	None	Similar	-0-	Similar	-0-	Similar	-0-
Net Adj. (Total)		☐ Plus; ☒ Minus	$ 2,000	☒ Plus; ☐ Minus	$ 500	☐ Plus; ☒ Minus	$ 3,000
Indicated Value of Subject			$ 63,000		$ 62,000		$ 61,500

Comments on Market Data All three comparables are of average quality, brick veneer construction and are located within close proximity to Subject. All are three bedroom, ranch style dwellings with appeal and utility similar to that of Subject. Overall, #2 is most similar to Subject

[illegible] and is given primary emphasis in final figure. $ 62,000

[illegible] (If applicable) Economic Market Rent $ ______ /Mo. x Gross Rent Multiplier ______ = $ N/A

This appraisal is made ☒ "as is" ☐ subject to the repairs, alterations, or conditions listed below ☐ completion per plans and specifications.

Comments and Conditions of Appraisal: Single family dwellings in this price range are seldom purchased for income purposes and rental data is all but nonexistent. Therefore, the income approach to value lacks good rationale and is not used in this report.

Final Reconciliation: Both applicable approaches to value fall in a close range and are mutually supportive. Given the high quality of available sales data, the figure indicated by the market approach is used as the final estimated value, with support from the cost approach.

Construction Warranty ☐ Yes ☒ No Name of Warranty Program ______ Warranty Coverage Expires ______

This appraisal is based upon the above requirements, the certification, contingent and limiting conditions, and Market Value definition that are stated in

☒ FHLMC Form 439 (Rev. 10/78)/FNMA Form 1004B (Rev. 10/78) filed with client July 7, 19 79 ☐ attached.

I ESTIMATE THE MARKET VALUE, AS DEFINED, OF SUBJECT PROPERTY AS OF May 5, 19 80 to be $ 62,000

Appraiser(s) ______ Review Appraiser (If applicable) ______

Richard Heyn, SRA, RM ☐ Did ☐ Did Not Physically Inspect Property

FHLMC Form 70 Rev. 7/79 [illegible] REVERSE FNMA Form 1004 Rev. 7/79

Using his Zenith computer, Richard Heyn has developed a program that can fill out the federal FNMA 1004 (Fannie Mae) form used for placing a value on real property.

either initial calculation or as a check against figures computed elsewhere, error-trapping routines, and automatic reentry of recurring data. The program is widely customized for individual customers' needs, Heyn says.

"You don't have to be a programmer to use this system," he insists. "The software has been designed with the user in mind. Menus lead you through the process."

The accompanying sample form 1004 shows several of Heyn's innovations. The page is from the "Valuation Section," an estimate of a property's market value. The paragraph-length texts are boilerplate comments, explaining how the evaluation was made; since they are standardized, each has been assigned a code number and the user can recall them by typing the code, instead of the whole text.

In the column labeled "Subject Property," the address and description of the house are items which have appeared on a previous page of the form, and so can be called back for inclusion here. The repetitive task of entering the word "similar" in the adjacent three columns is an automatic default option for the person filling out the form. The numerical values, such as the last entries in each column, were computed by the math functions of the program.

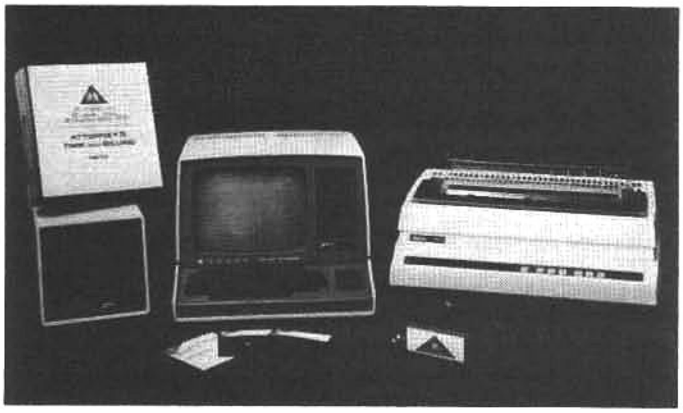
"We also offer database management and accounting software," says Heyn, "and have some interesting financial and statistical programs in progress. We've just introduced a program that will sort and link comparable sales data with the FNMA program."

HEARING TESTING

Oto-Data, Inc, of Atlanta, Georgia, has converted the Z-19 into a hearing-testing device. Instead of a disk drive next to the screen, Oto-Data has mounted a narrow paper printer; earphones emerge from a rear port.

The Audiometric Computer Model AC 2001 automatically administers a hearing test to a subject, using standard clinical procedures. The progress of the test can be monitored continuously on the CRT screen, as the computer "walks" the tester through the subject's interview and validates all entries of data. Standardized audiometric procedures are repeated for each threshold of hearing.

Oto-Data, Inc., adapts a small printer into the disk drive opening and installs a special accessory board so the Z/H-89 can be used to test for hearing impairments. Tones of varying frequencies and intensity are sent through the earphones, and as prompted by the CRT, the test subject presses certain keys to respond. The test results are recorded by the printer.



P.K. Systems offers a time-billing program for attorneys; it requires a Z/H-89, at least two disk drives and a printer. The package is designed for a small office with two or three attorneys. P.K. Systems also offers a Docket Control System that keeps an attorney or legal secretary apprised of upcoming court appearances.

To the Z-19, Oto-Data has added a printed circuit board that operates the audiometer; their software resides on six 2K PROMs. The board is mounted in the place where the Z-89 board would be fitted. The AC 2001 can use software available for record keeping and is also usable as a normal computer terminal for other business functions.

Besides being printed on paper, the results of the hearing tests are sent through the RS-232C port and can be copied onto disk or tape, or shared with a larger computer.

ATTORNEY'S OFFICE MANAGEMENT

"We got into the Zenith line through the back door, while we were a software house for Digital Equipment Corporation (DEC). The Z-19 is a great emulator for the DEC VT-52 terminal," admits Patrick Kauffold, of P.K. Systems, Bloomington, Illinois. "We could practically put in two Z-19s for the price of one VT-52; the performance is really outstanding."

Now, with a major commitment to Z-89s, his software helps manage many busy law offices. The time billing program is written in Microsoft Basic, Kauffold says, "because it lets us write fairly structured programs, and cut down on the testing time." With three single-sided, hard-sectored, 5.25-inch disk drives, a small office of two to three attorneys can be easily managed. Using soft-sectored 5.25-inch or 8-inch disk drives will vastly increase the number of time records that can be stored and the billings that can be computed.

P.K. also offers a Docket Control System, which is a calendar program for reminding (or "tickling") the attorney or the legal secretary of court dates and appearances. "It works like a 3x5 index card file," says Kauffold, "when somebody looks at each week's list. It might say, for instance, 'statute date minus 10,' meaning that ten days remain to file a brief before a statute of limitations expires. The program also includes a way to check on conflicts of interest. It searches the previous dockets that the attorney handled and makes sure he or she didn't represent the opposing party on some previous occasion.

"Eventually, of course, we want to integrate that kind of calendar program with the time billing program, and make the same technology available to other professionals, such as accountants."

The Datamedic System 500 is a localized timesharing system for doctor and dental offices. It utilizes standard Z/H-19 terminals that allow the individual offices to communicate with the main computer system.

DOCTORS' PATIENT/PRACTICE MANAGEMENT SYSTEM

The Datamedic System 5000 handles data entry and retrieval for the offices of doctors or dentists. It is a timeshared computer service which places a Z-19 at each office site, coupled with a large, hard disk drive, and a dot-matrix printer for record-keeping.

In a typical installation, the system handles recalls, insurance claim preparation, 15-month ledger lookups, day sheets and deposits; it also determines doctor productivity and improves monitoring of cash flow. The CRT display shows a summary of each patient's data in the top third, and activity areas in the center which are filled in line by line; the bottom third of the screen can display tutorial instructions for new users.

Formats are provided for adding patients, determining charges, responding to patient questions from the ledger, and reviewing insurance claims. Computer-to-computer communication permits claims to be submitted directly to the insurance carrier. The Datamedic main system polls the local systems during off-hours.

HELPING THE BLIND TO READ

Using Z-19 terminals, Triformation Systems, Inc., of Stuart, Florida, helps blind and visually impaired people gain access to databases. An audio speech device converts text into sound.

The Free Scan Speech Terminals (FSST) are interactive computer terminals that deliver their output in audible form. A blind operator can call up "pages" of information, and scan or review them horizontally as well as vertically. Varying audible signals tell the operator where he or she is on the "page."

When specific data is located, the terminal verbally reads out the information at a rate which the operator has selected, and which can be increased as practice and efficiency permit. A speaker, mounted in the space where a disk drive is normally installed, delivers the text either as full words or as spelled-out words, letter by letter.

The numerical keypad is the operator's tool for accessing data. When not used for number-keying, it can be shifted into a cursor-control unit: up, down, left, right, home, previous word, present word, and next word are its options.

Their equipment has provided jobs and educational access to blind students, programmers and clerical people. Coupled with peripheral equipment they also make, Triformation's FSST terminals can also translate text into braille.

The Z/H-19 has also been modified to assist blind and visually impaired people. The Triformation system shown here produces an audible representation of data or keyboard input.

UNDERSTANDING THE HUMAN VOICE

Interstate Electronics Corp., of Anaheim, California, makes voice-recognition products for office and factory automation, word processing, computer-aided design, building security, toys and games, order entry, appliances, and other applications that its customers want. "Tell us your needs and we will meet them," its flyers proclaim.

With a 48 Kbyte RAM Z-89, CP/M, and one built-in disk drive, Interstate's Voice Recognition Terminal, VRT 101, provides direct voice interaction with software through a 100-word vocabulary. That capability comes from a voice-recognition module (VRM): a printed circuit board which is connected to the Z-89 through one of the RS-232C serial ports.

The VRM contains a special chip that analyzes incoming sound, converts it to computer-recognizable digital data, and creates a "template" for each vocabulary item desired. New sounds are then compared with those stored in the templates and the Z-89 rejects those which do not sufficiently agree with the utterance previously stored.

Conceptually, the VRM sits between the keyboard and the computer's CPU, so the input to the CPU is ASCII characters from the VRM. Any words or numbers that form an utterance of 1.5 seconds or less can be spoken, and converted by the VRM to ASCII characters.

Voice recognition on the Z/H-89 is particularly useful for data entry, reducing training time for new users, and offering freedom of hand movement for other tasks.

Interstate Electronics Corp. gives the Z/H-89 the ability to understand human speech or other sounds. An accessory component installed in the Z/H-89 compares sound coming through the microphone with phonic representations stored in memory; if they match, the computer performs a predetermined function.

CHAPTER 10 ASSEMBLING THE HEATH H-89 COMPUTER KIT

(1) The kit is complete with all of the parts required to build a functional H-89. Small parts—screws, nuts, resistors, etc.—are collected in brown paper bags to prevent loss. The bags are opened as needed, generally one for each subassembly.

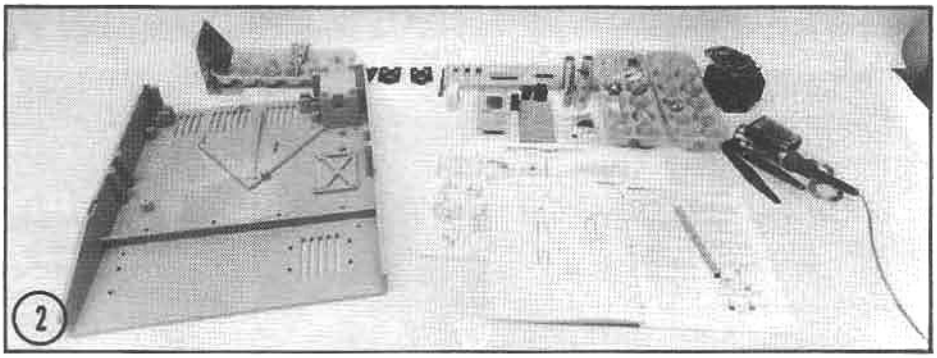
It doesn't seem possible that an average person can build a sophisticated digital computer from a kit. Afterall, the word computer has sacred overtones, and most people are intimidated by the image of an immensely expensive machine housed in a surgically clean room. In this light, a "kit computer" sounds like a toy, but nothing could be further from the truth.

The computer world has changed dramatically in the last few years, and before long, computers may be viewed as common household appliances. Millions of people are beginning to realize the value of a new breed of computer—the *microcomputer*. These compact, inexpensive, yet powerful machines are being purchased by thousands of individuals, for a thousand different reasons, and as the Computer Revolution continues, building a small computer on the kitchen table may become as common as baking a cake.

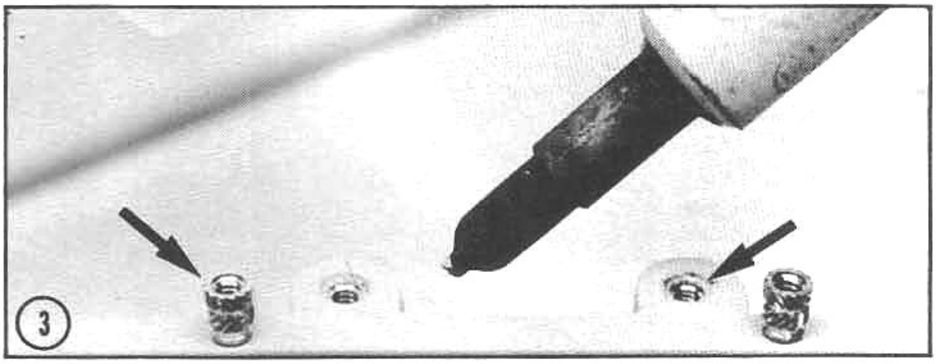
Building a computer is currently a little more complicated than baking a cake, and it takes longer, but it's not as difficult as it sounds. Heath was the first company to sell a "usable" small computer in kit form, and their long-established reputation for engineering and customer support makes such a project entirely feasible for even a beginner.

But why a "Heathkit?" Why would someone want to assemble a computer from a pile of parts. There are many reasons: an urge to learn "what makes it tick," a desire to save money, or the opportunity to have some fun! That's right, fun. Building a Heath computer can be an enjoyable and educational experience. The assembly manuals are superb, all of the parts are high quality, the procedures are not difficult, and the Heath company virtually will not let a kitbuilder fail. If unexpected problems overcome the builder, Heath can—and does—provide national and local support through their network of Heath Electronic Centers.

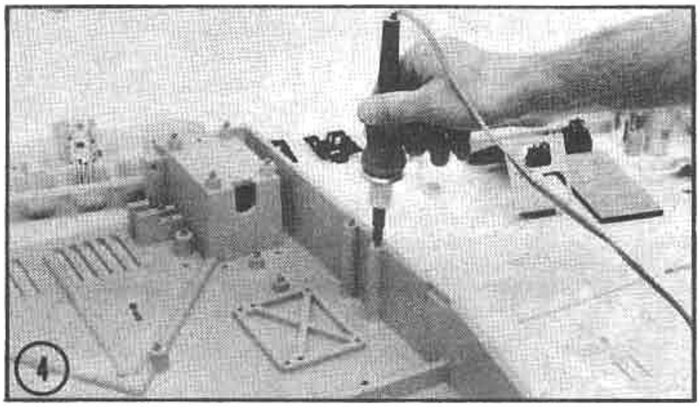
Nonetheless, it is hard for many people to imagine what it must be like to put together a computer kit. To give readers an overview of the procedures, we present here an illustrated guide to "building a computer from little brown paper bags." It isn't as difficult as it sounds. Have a look. Who knows, in a few days you may be at the kitchen table, amidst strange-looking gadgets and wisps of smoldering solder, assembling your own "personal" computer.



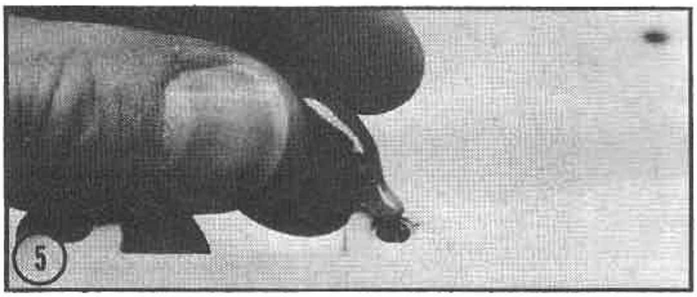
(2) It is best to work at a large table. Before you begin a series of subassembly steps, scan the instructions and become generally familiar with the major components and assembly sequence. Some subassemblies use many small parts and it is surprisingly easy to mislay tiny pieces. If you follow the instruction implicitly and work carefully, the assembly will proceed smoothly.



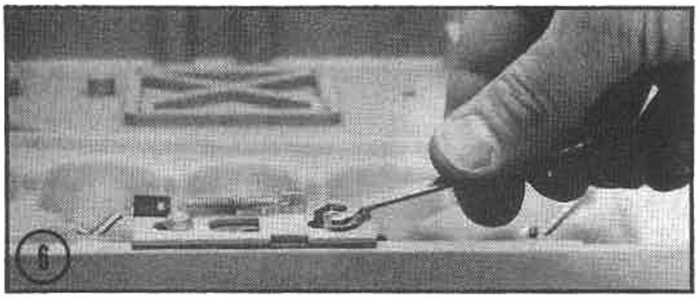
(3) The assembly begins with the base of the cabinet. Brass inserts must be installed in the base, providing anchoring points to secure the major subassemblies to the base. The inserts (left arrow) are heated with a soldering pencil tip and slipped into place (right arrow).



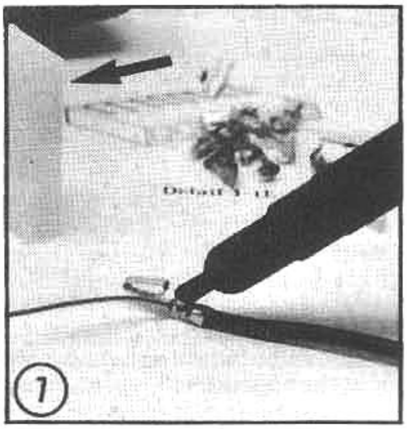
(4) Over 40 brass inserts must be installed in this manner. When these inserts are heat-welded into the plastic base, they provide solid anchors for the subassembly fasteners. Here the soldering pencil is heating an insert for the upper cabinet lockplate mount.



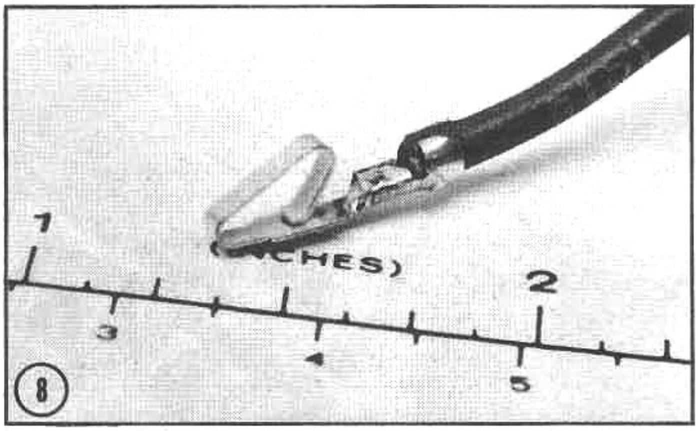
(5) The lockplate assemblies for the cabinet cover are installed next. The threads in the brass inserts are coated with Vibra-Tite sealant. This compound will lock the screws in place, and once positioned, they will not vibrate loose or require further adjustment.



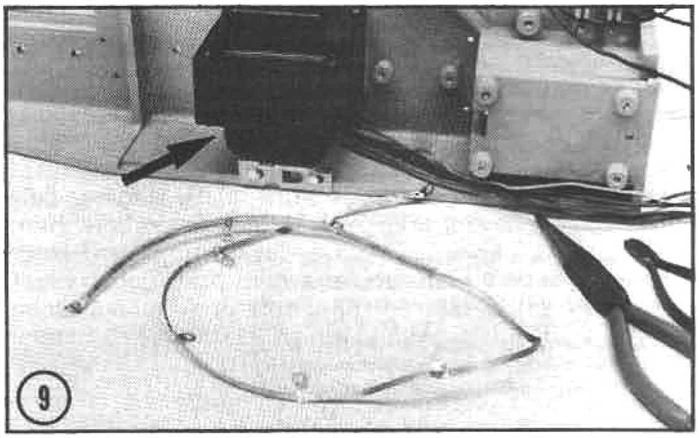
(6) Each cover lockplate assembly consists of three slide plates and two hold-down screws. The latches have return springs, and when the computer is completely assembled, the cover can be latched in place by applying a modest downward pressure on the top.



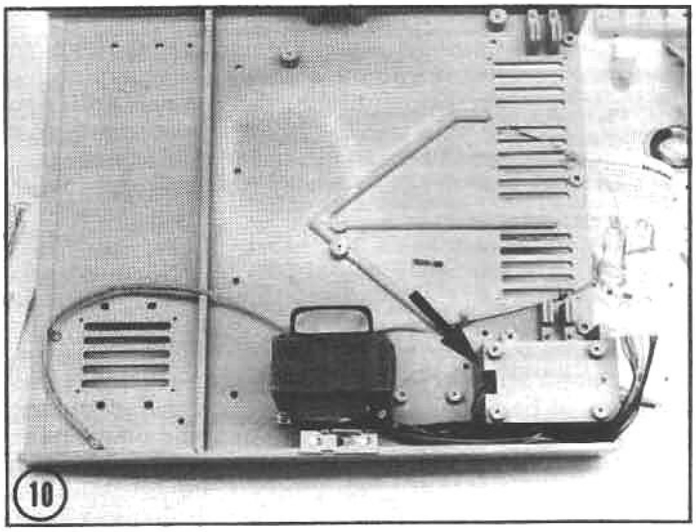
(7) It is necessary to make up several cables during the assembly. Spring terminals are crimped and soldered to the individual wires, which are then assembled in the connector shell (arrow) to form a multiterminal cable connector. This cable is used primarily for electrical distribution between the internal circuit boards.



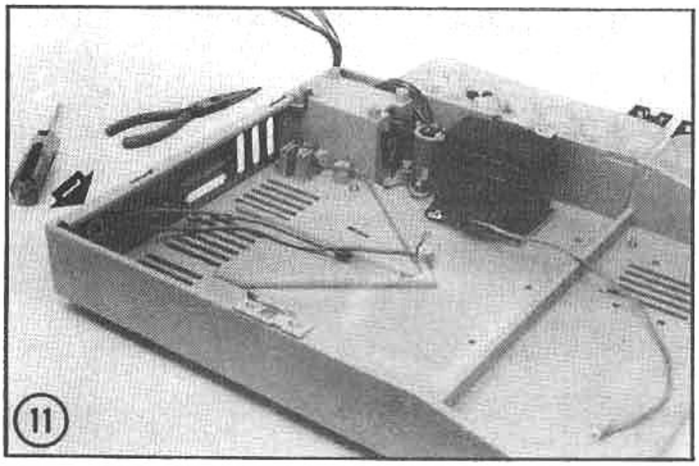
(8) Proper soldering techniques must be developed to ensure reliable electrical connections. The solder should flow smoothly over the connection and have a shiny appearance when hard (cold solder joints have a dull surface sheen and can cause electrical problems). The instructions clearly explain basic soldering techniques, and the first-time builder should practice before beginning the final assembly (extra solder is provided for this purpose).



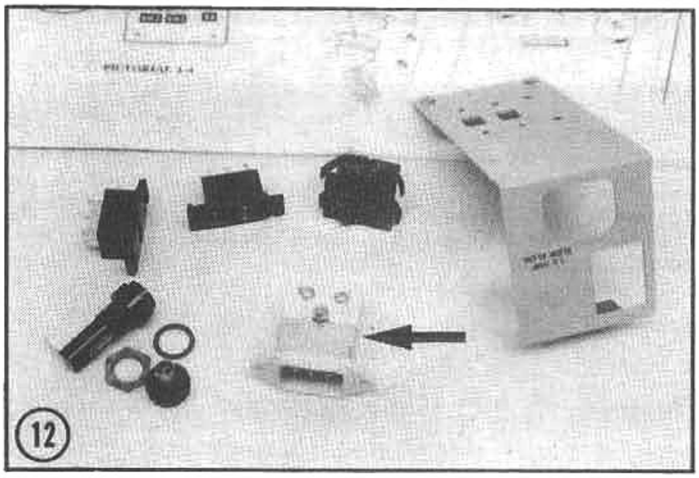
(9) The power transformer (arrow) is mounted to the baseplate and a braided ground connector is constructed. The braided lead is part of the RFI (radio frequency interference) shielding built into the new H-89 models. These measures substantially reduce radio and television interference.



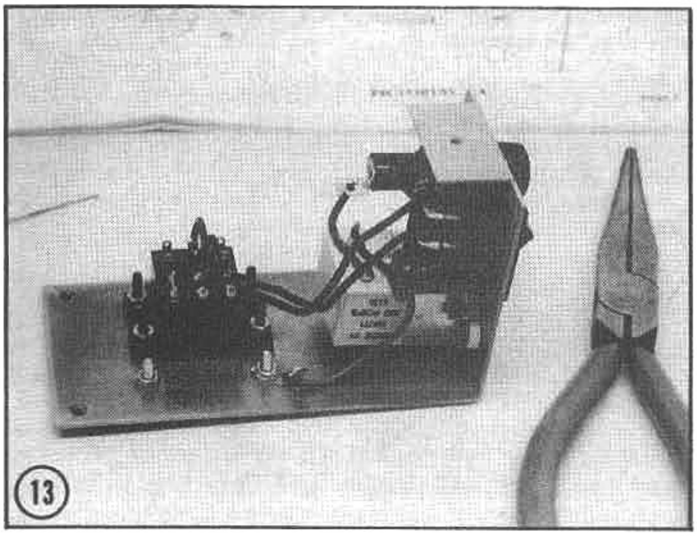
(10) The RFI ground lead is attached to the power transformer and the transformer leads are routed across the baseplate. The leads disappearing into the hole (arrow) connect to the AC (power line) chassis.



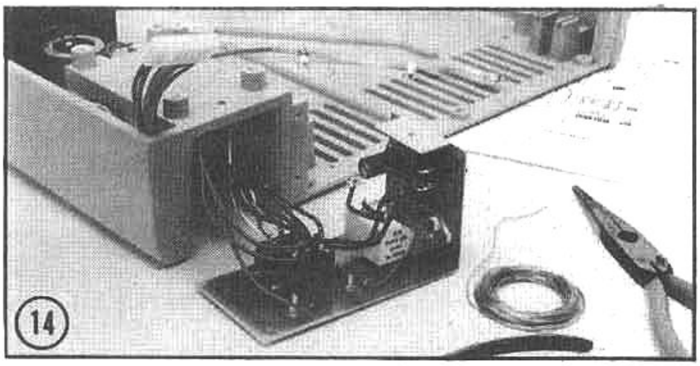
(11) The rear chassis plate and brightness control (arrow 1) and a 1500uf capacitor (arrow 2) is installed on the baseplate. The RFI ground braid is connected to both the capacitor mounting clamp and the rear chassis plate.



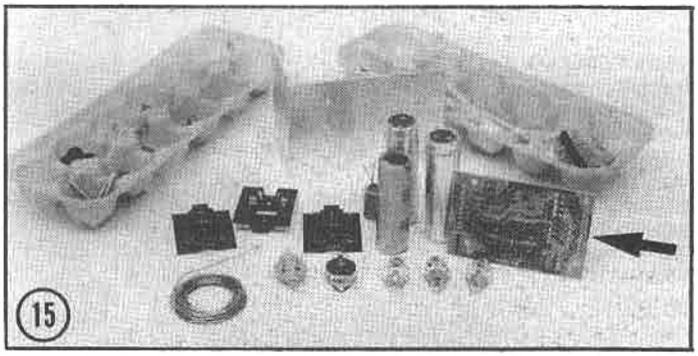
(12) The next step is to assemble the AC chassis. This small subassembly houses the AC line cord plug, on-off switch, 110-220 switches, and fuse assembly. The line cord plug contains a transient supressor (arrow) to reduce problems caused by electrical irregularities in the power line.



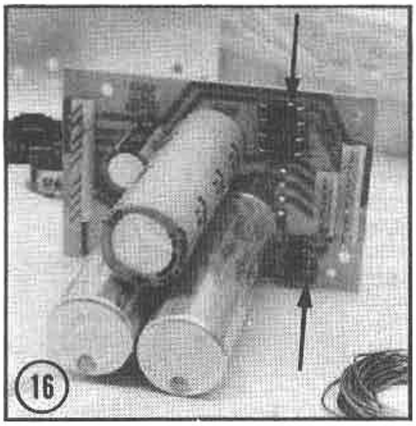
(13) Once the AC chassis components are mounted, several short interconnecting wires must be crimped and soldered in place. Where an additional wire will be later connected, the assembly manual notes "NS" for "not soldered." This means the connection should be crimped and not soldered at this time.



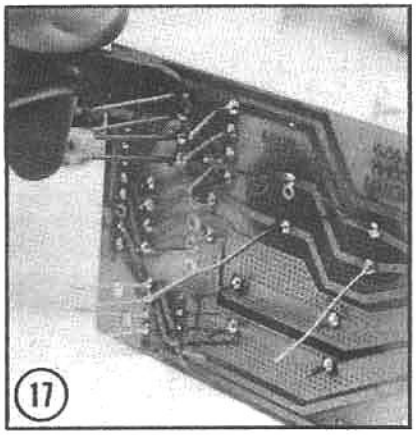
(14) The final connections to the AC chassis are the leads from the power transformer. The completed unit is then installed in the baseplate, shielding the hazardous AC line voltage from accidental contact.



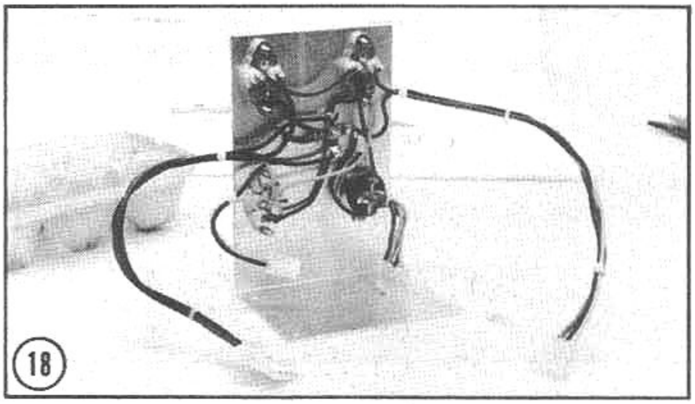
(15) The next project is the power supply. All of the components for this subassembly are located and organized (note that a plastic egg carton is being used to hold the small parts). The power supply is one of the three circuit boards (arrow) that must be assembled by the builder. It is a simple board to build.



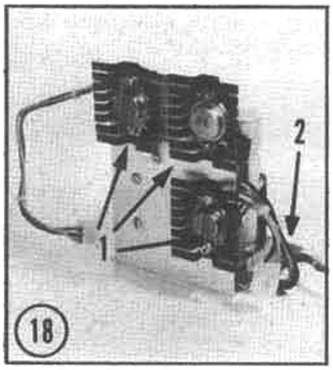
(16) The power supply components (capacitors, diodes, and plugs) are installed on the circuit board. Most of these components must be oriented in the proper direction (note the banded ends of the diodes), but this is carefully detailed in the assembly manual.



(17) After the components are installed and soldered in place, the excess leads are cut off. Using a small pair of diagonal cutters makes this job very easy (although eye protection should be worn, as the wire leads can shoot off in any direction).

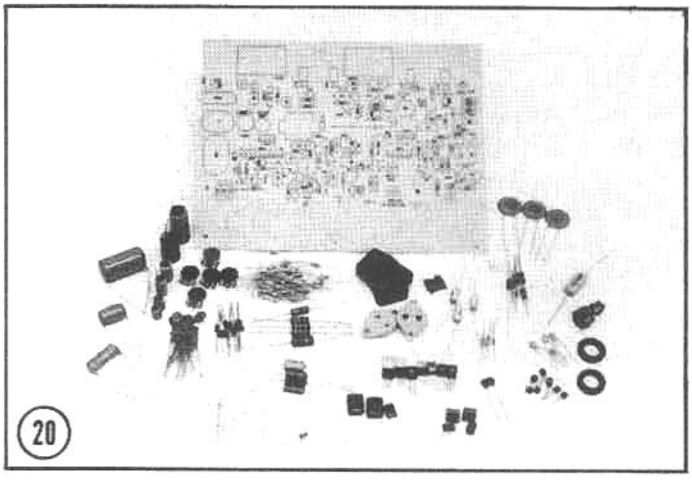


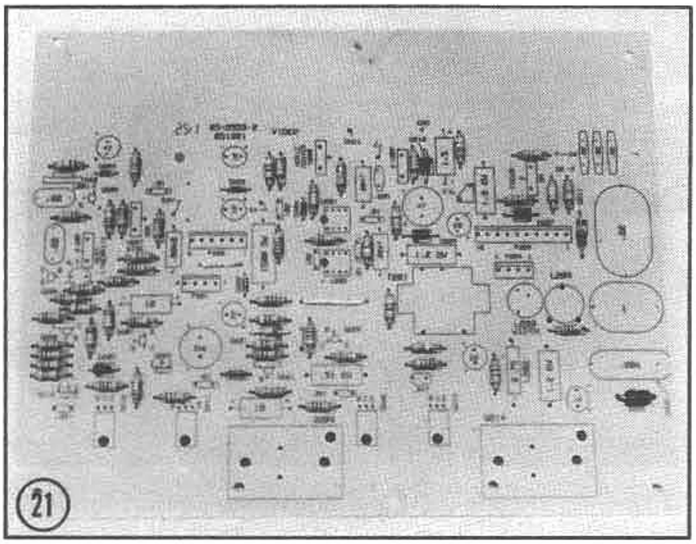
(18) The power supply heat sink and interconnecting cables are assembled next. Three integrated circuits mounted on the heat sink (arrow 1) are the equivalent of several hundred discrete components, and regulate (with high accuracy) the DC voltages that power the components inside of the H-89. The circuit board that was assembled in the previous step (arrow 2) is mounted on the heat sink.



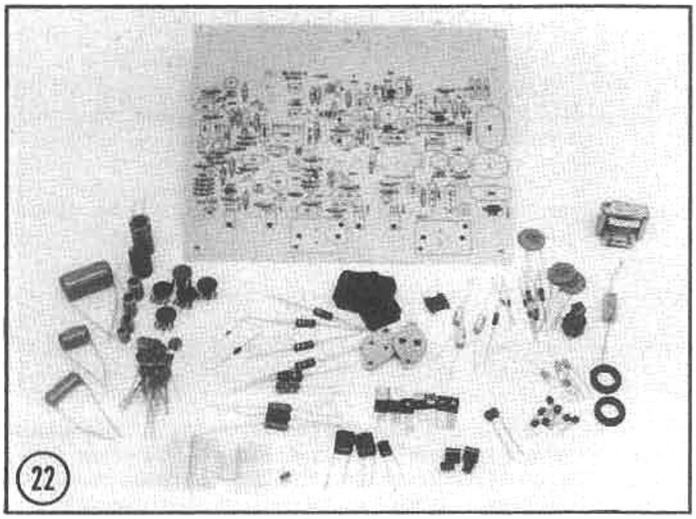
(19) The finished power supply (shown here mounted on a completed computer) is tested for proper voltage levels before the assembly proceeds. A faulty power supply could damage the integrated circuits on the complex CPU and terminal logic boards that are installed in later steps.

(20) Next, the builder must locate and identify all of the parts for the video circuit board. This is the most difficult board that the builder must actually put together. *The most important and complex boards—the CPU and terminal logic boards—are assembled and tested at the factory.*

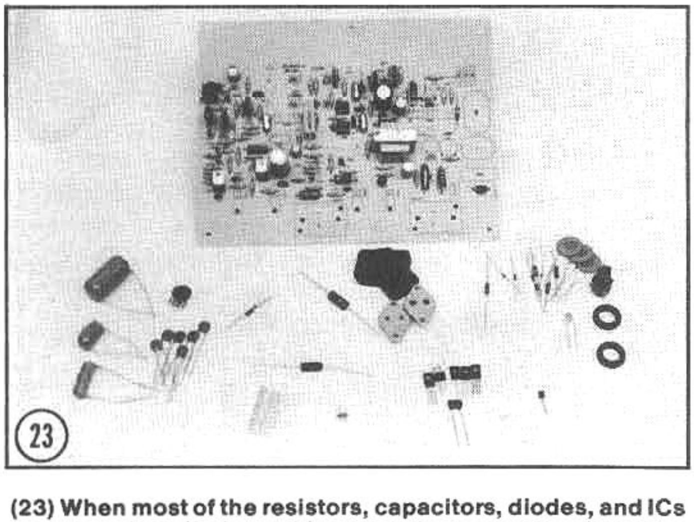




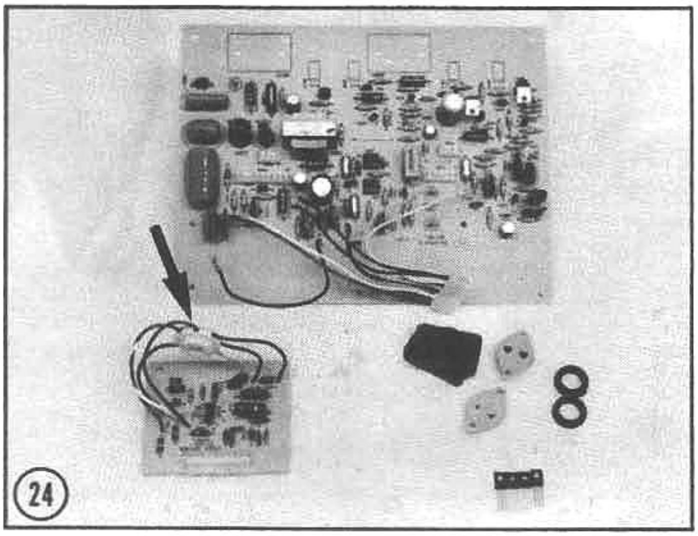
(21) Parts are installed and soldered to the video board in groups. Here the resistors and diodes have been installed. The circuit board clearly indicates the position of all parts with either the part number or value of the component.



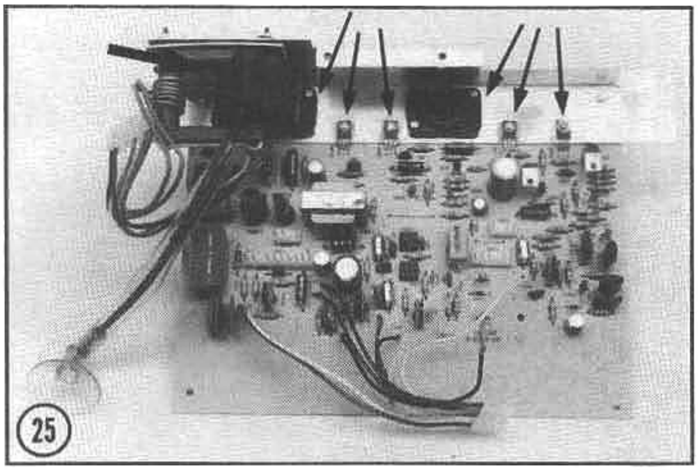
(22) After the resistors and diodes have been installed, the next group of components—the ICs, transistors, and capacitors—must be located.



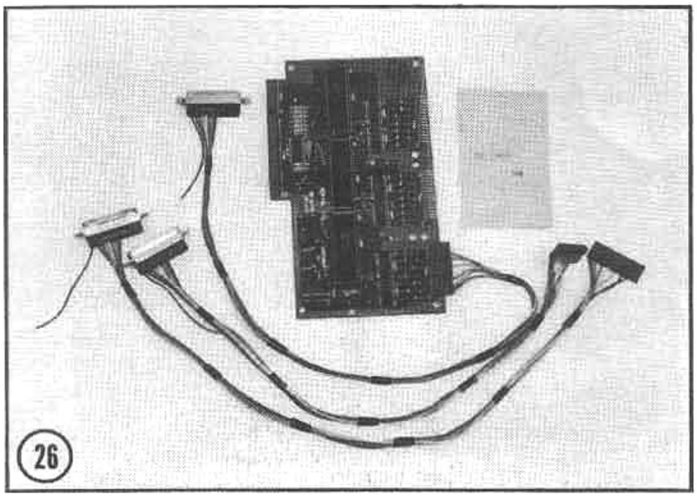
(23) When most of the resistors, capacitors, diodes, and ICs have been installed, the video board is almost complete. The assembly time spent on the video board to this point is about 1-2 hours.



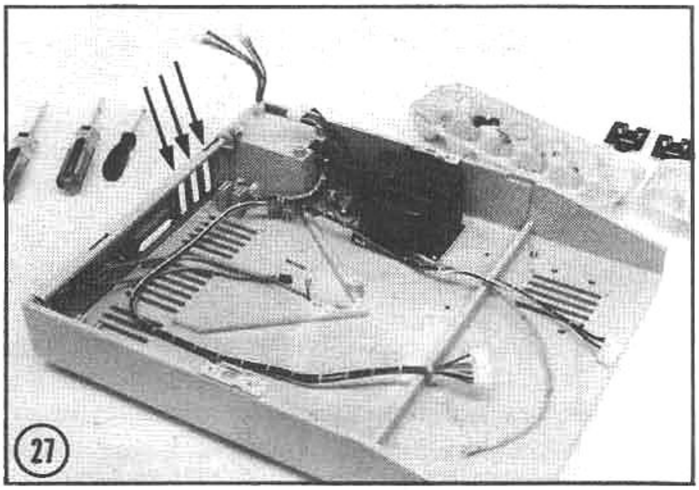
(24) All components have been installed on the TLB, except those that will mount on the aluminum heat sink (next step). The video driver circuit board (small board in the lower left) is also assembled. The plug that attaches to the rear of the CRT is soldered to this board (arrow).



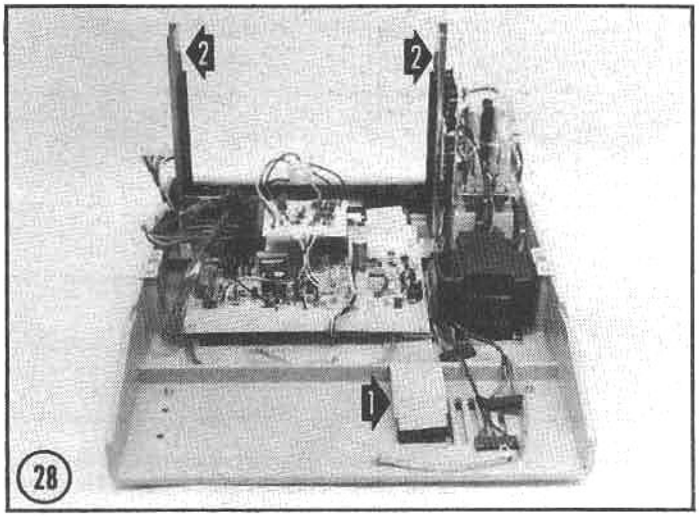
(25) Here is the completed video circuit board. The final assembly includes the driver transistors and flyback transformer (arrows) on the aluminum heat sink.



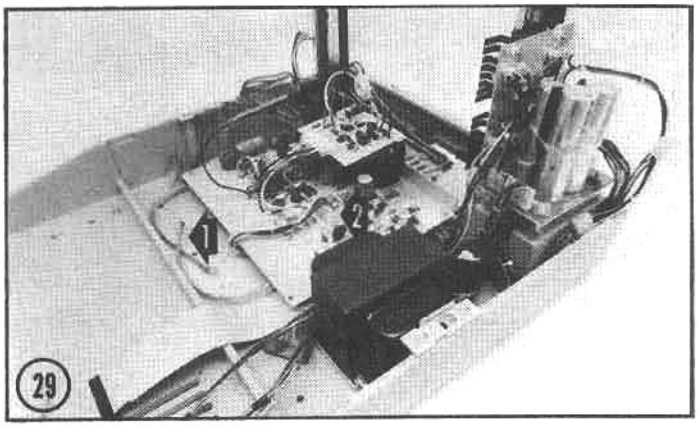
(26) This is the serial interface board, connecting cables, and hardware. This board is supplied assembled and tested. The connecting cable assemblies are removed from the package and used in the next step.



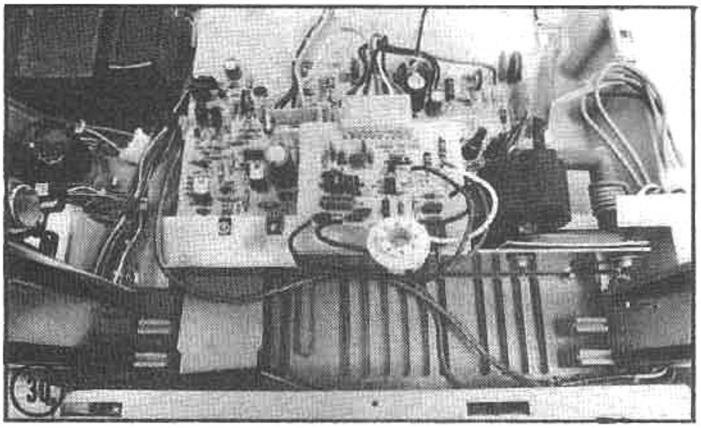
(27) The serial interface cables are installed in the three slots provided at the rear of the baseplate (arrows). During the rest of the assembly, the computer will begin to take shape as subassemblies are added to the baseplate.



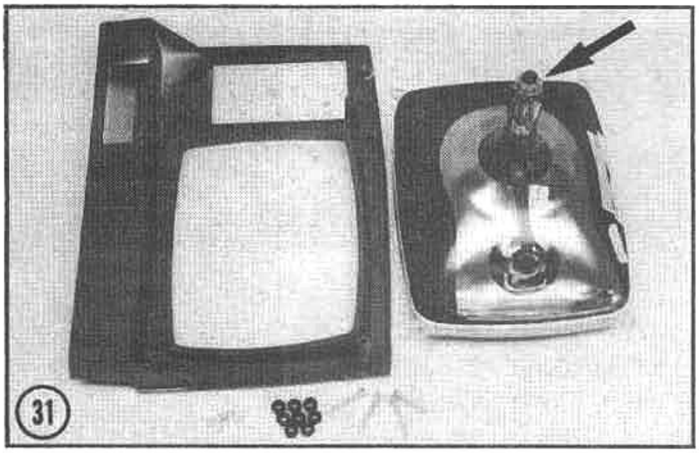
(28) Here the video circuit board and video driver boards are installed in the baseplate (the video driver mounts on the aluminum heat sink). The interconnecting cable for the keyboard (arrow 1) is routed under the circuit board. The CPU and TLB circuit board supports are installed (arrow 2) here.



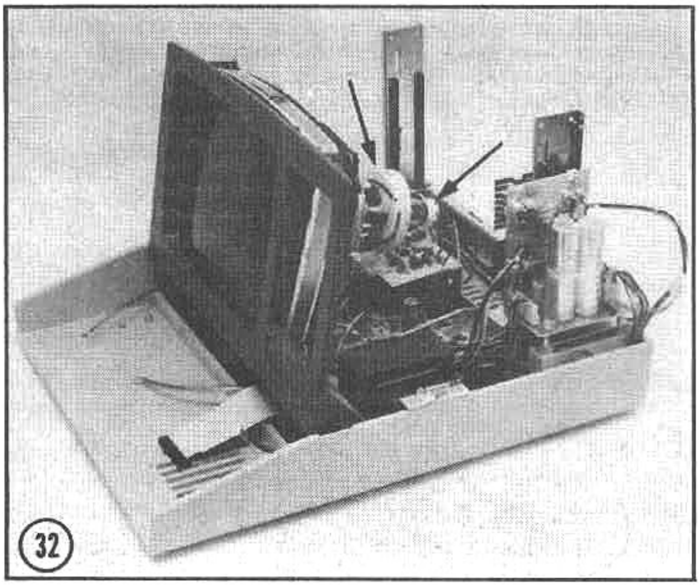
(29) When the circuit board is mounted to the baseplate, the RFI ground leads (arrow 1) are attached between the board and the mounting bosses on the baseplate, providing a solid ground connection for the video electronics. Power will be supplied to the video circuit board by a cable from the power supply (arrow 2).



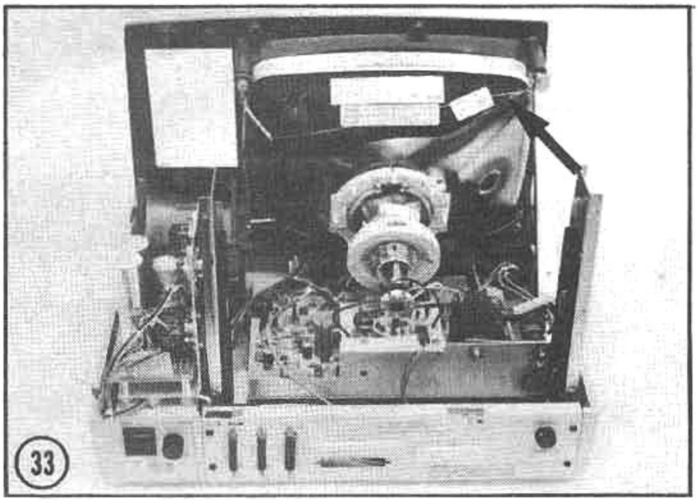
(30) Looking down the card guides, it is easy to see the cables routed to the rear of the cabinet. The RS-232C serial plug and power-supply cables are directed around the left (as illustrated) guide. The power-supply cable continues along the back and around the right guide, where it will eventually be connected to the CPU Board.



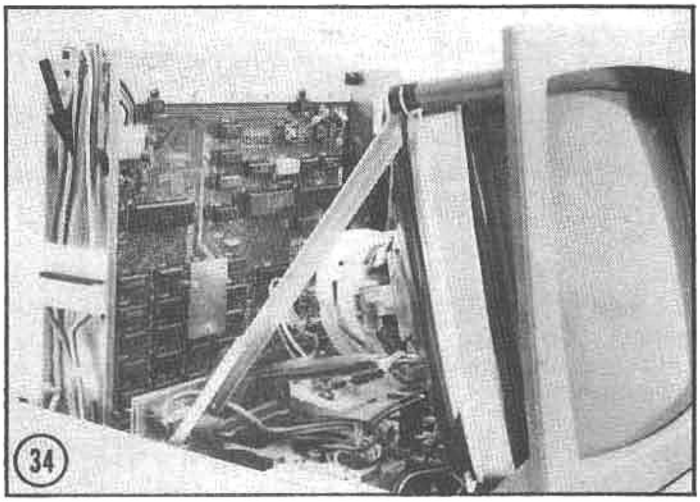
(31) Next, the CRT is attached to the front panel. This operation requires great care because the CRT neck (arrow) is fragile and can easily be broken.



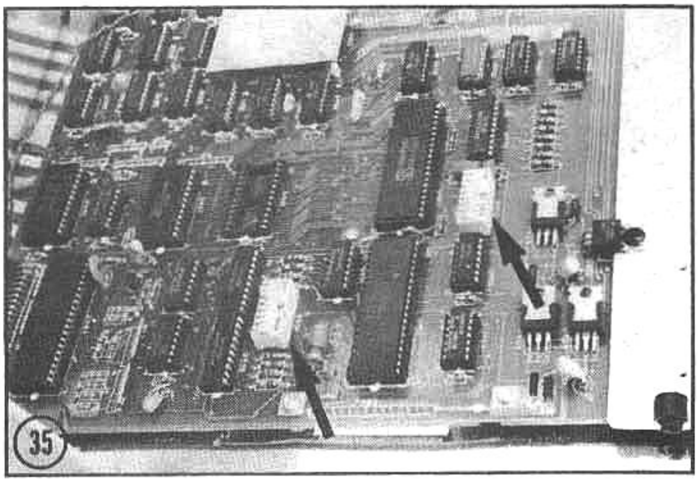
(32) Finally the computer is beginning to take shape! The front panel and CRT assembly is installed in the baseplate, routing the RFI-ground and keyboard-cable leads through slots in the lower panel assembly. The deflection yolk and CRT neck plugs are also installed (arrows).



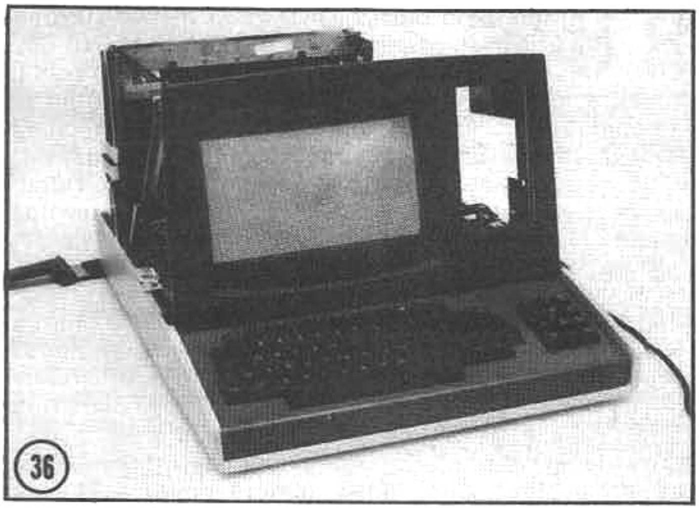
(33) The CRT installation includes a ground wire (arrow) that is held in contact with a conductive coating on the outside of the tube back. This drains away static charges that would otherwise affect the sensitive electronics located in close proximity to the CRT.



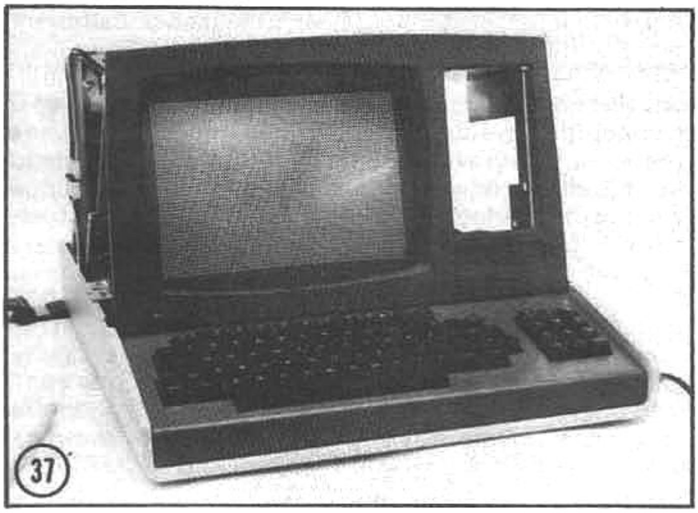
(34) The terminal logic board is installed in the rear card guides. A short jumper connector and cable (arrow) will supply power to the TLB for the initial checkout performed in the next few steps.



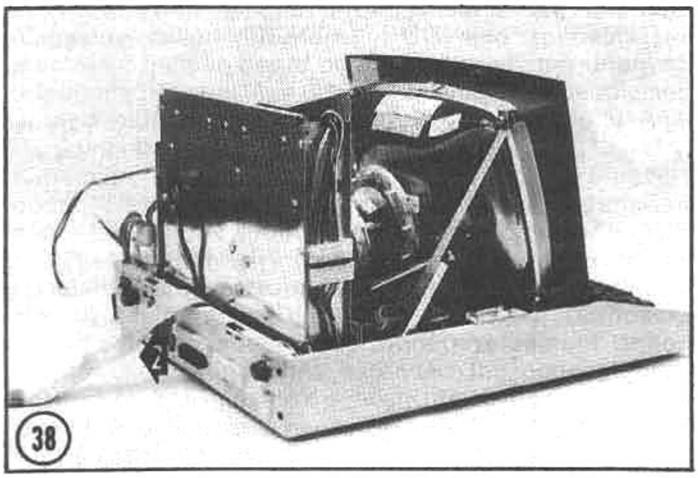
(35) The terminal logic board has two DIP switches—SW401 and SW402—that control the configuration of the terminal section These switches (arrows) should be set as indicated in the instructions.



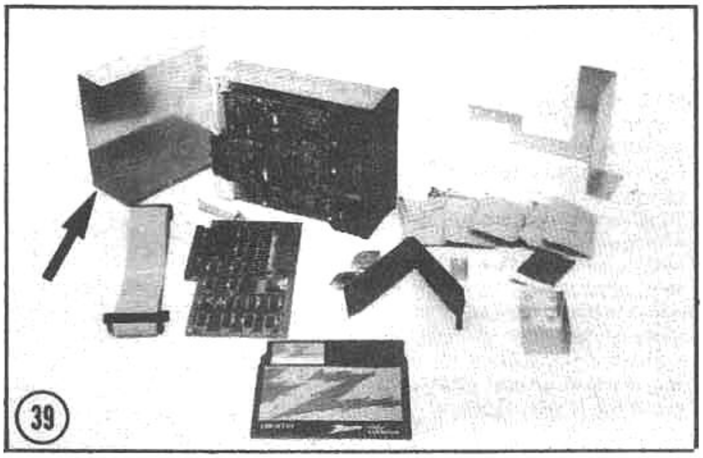
(36) The power is now applied to the computer, without the CPU board in place. This allows the user to verify power cable connections and to perform video board adjustments (to tune the CRT display). Note the screen display is too short and the background is not dark enough.



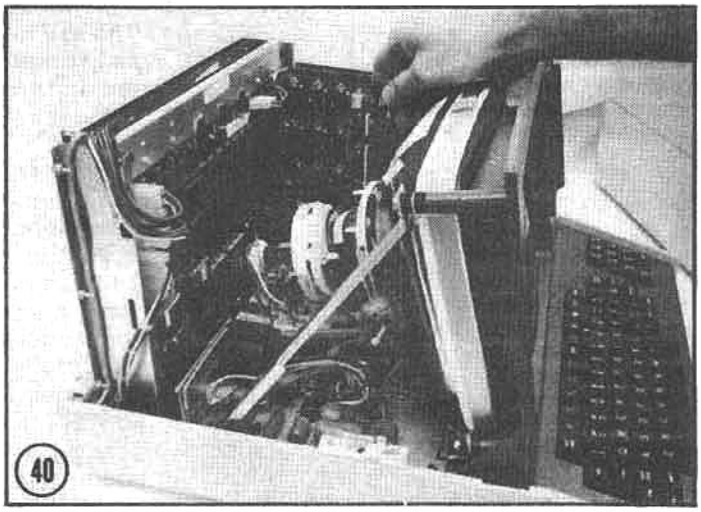
(37) After the adjustments on the video board have been completed properly the display fills the screen and the background (raster) is dark. So far...so good!



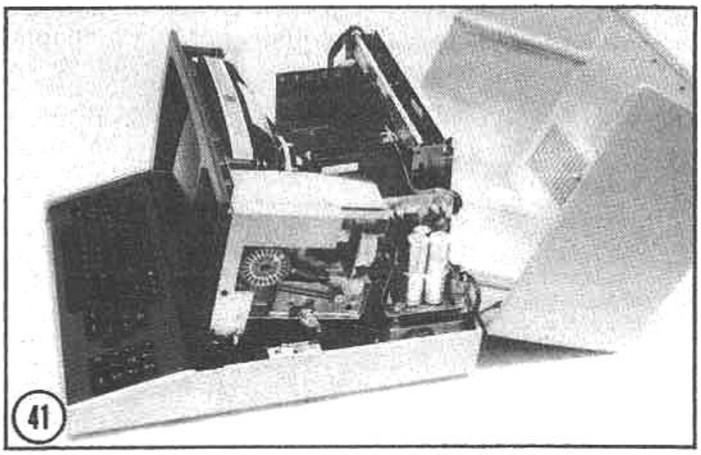
(38) Power is now turned off and the remaining assembly is performed. The interconnect cable between the TLB and the CPUB is installed at the rear of the TLB (arrow 1). The floppy disk drive accessory cable is temporarily positioned out of the way (arrow 2).



(39) The floppy disk drive is installed next. It comes in a separate box with accompanying hardware. Special shielding is included (arrow) that permits reliable drive operation next to the CRT.



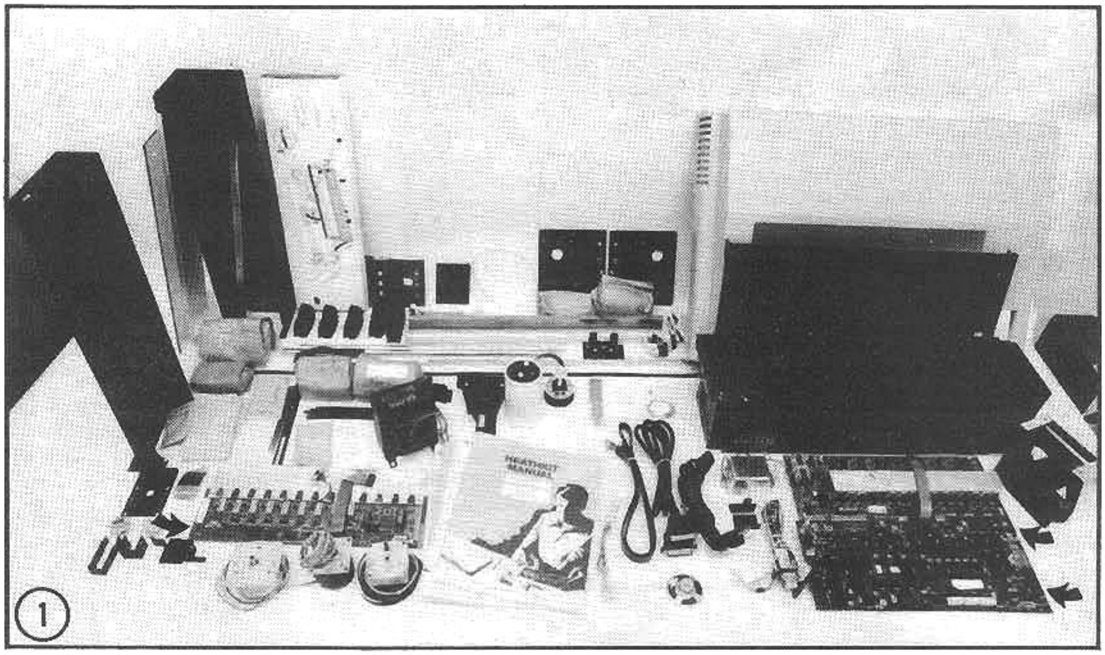
(40) After the floppy drive and shielding are installed, the CPU board is slipped in front of the TLB and the power cables are connected. The floppy disk controller and RS-232C serial cards are plugged into the accessory connections on the right side of the CPU board.



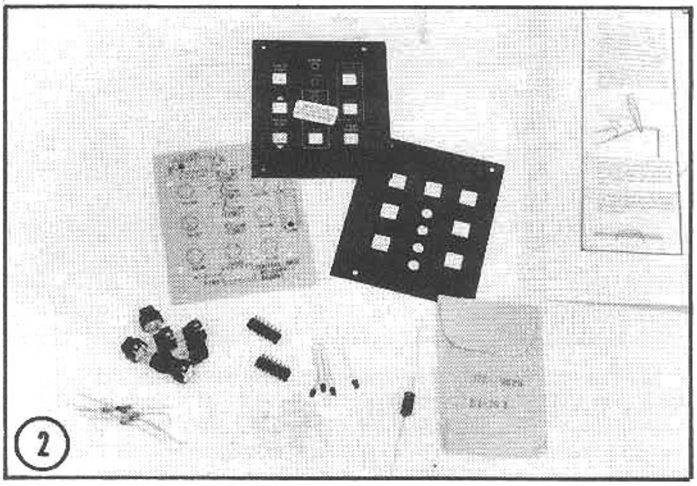
(41) Here is the finished computer. And, yes, it did work properly the first time! As a matter of fact, these words were written on the H-89 pictured here (with Heath word processing software). The total assembly time was about 25 hours, but this will vary, depending on the builder, between 20 and 30 hours. Compared to a similar factory-assembled unit, this kit H-89 was about $1000 cheaper!

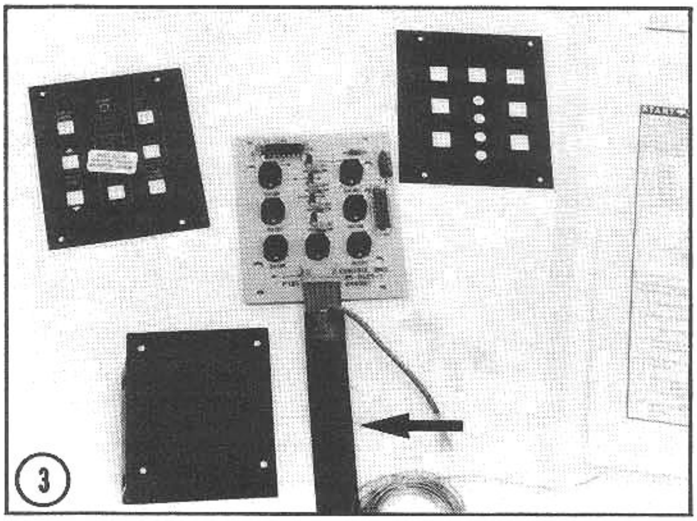
ASSEMBLING THE HEATH H-25 PRINTER

(1) Despite the intimidating bulk of the shipping cartons, the Heath H-25 printer is not a difficult kit to assemble. As in the H-89 assembly, the main circuit boards (arrows), which contain the complex electronics, are assembled and tested at the factory. Most of the assembly procedures involve mechanical components (nuts, bolts, etc.).

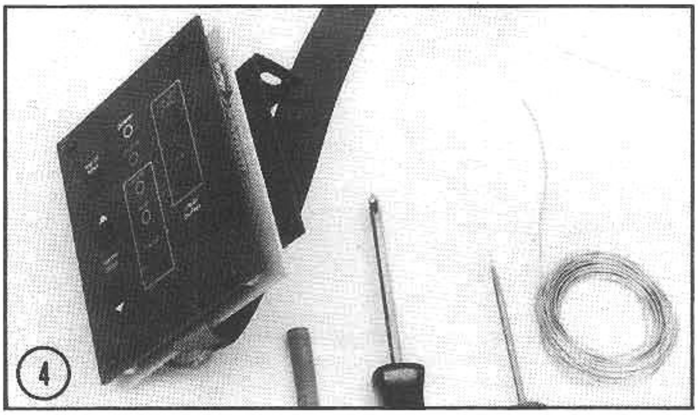


(2) The kit is accompanied with Heath's excellent documentation, and once the primary components have been identified, the assembly begins with the control panel circuit board. This board is the only electronic assembly required in the kit.

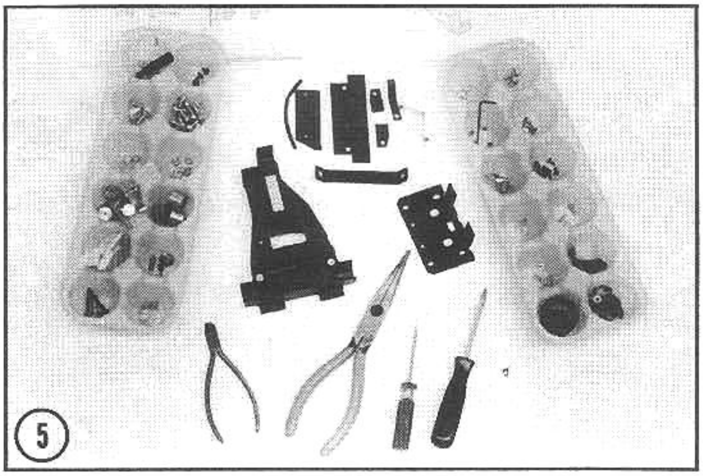




(3) The main components are mounted to the board. They comprise a series of pushbutton switches, two integrated circuits, and several LEDs (light emitting diodes). The ribbon cable (arrow) connects the panel to the main electronics.

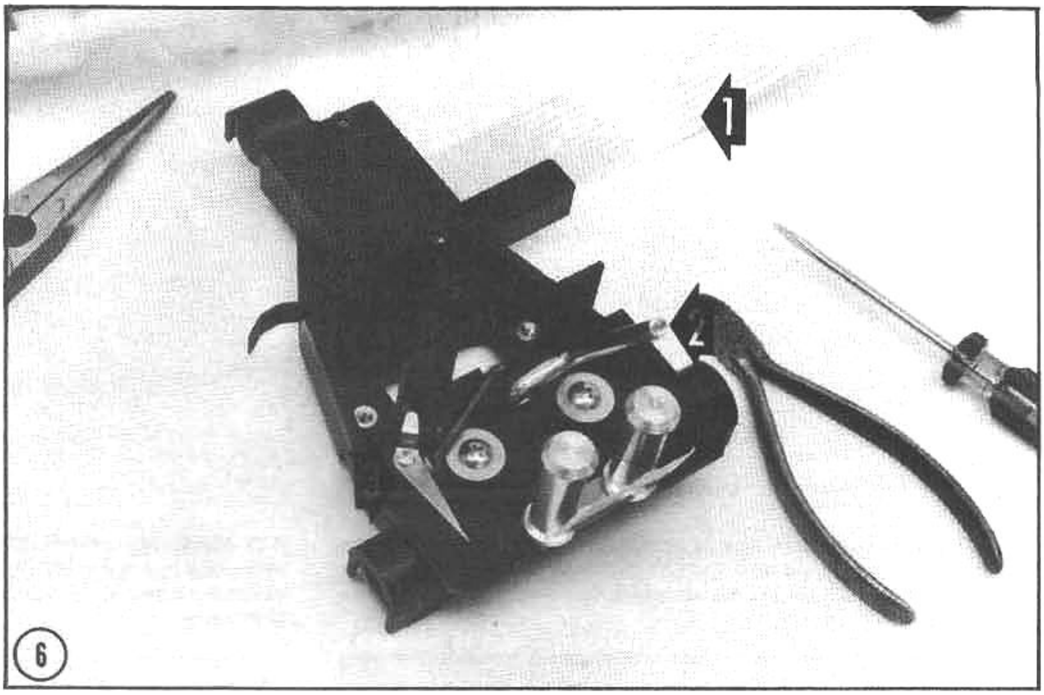


(4) The control panel sandwiches the circuit board between the top plate (with control descriptions) and the lower mounting bracket. In the finished printer, the attractive panel is accessible through an opening in the acoustic cabinet.

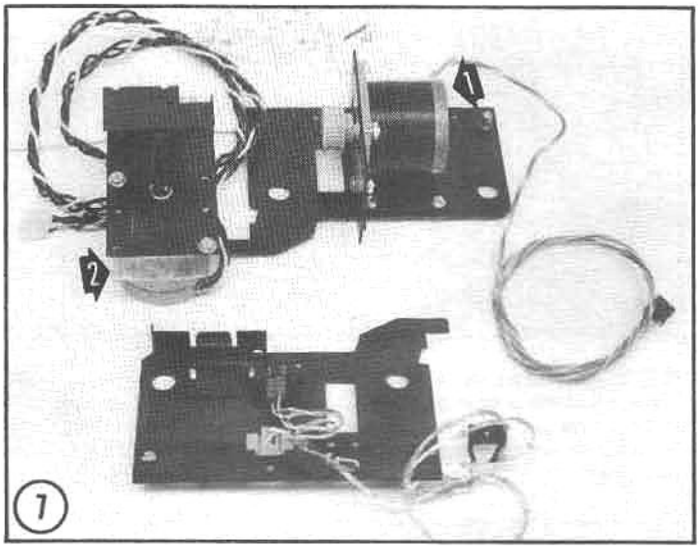


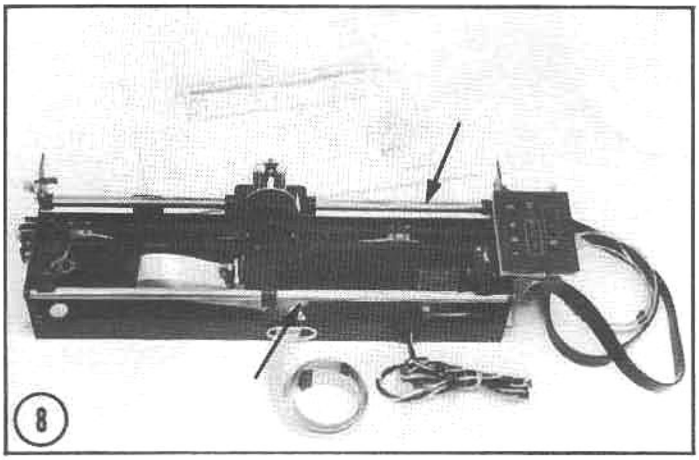
(5) The mechanical assembly begins with the printhead carriage. Organizing the parts is especially important with a kit that uses many small screws, nuts, washers, clips, pins, and specialized components.

(6) The carriage assembly is a base for the matrix-wire printhead (installed on the carriage in a later step). Printhead power is supplied through a flex-ribbon cable (arrow 1), and an adjustment for the thickness of multi-carbon paper forms is provided (arrow 2) with an indicator and scale (arrow 3).

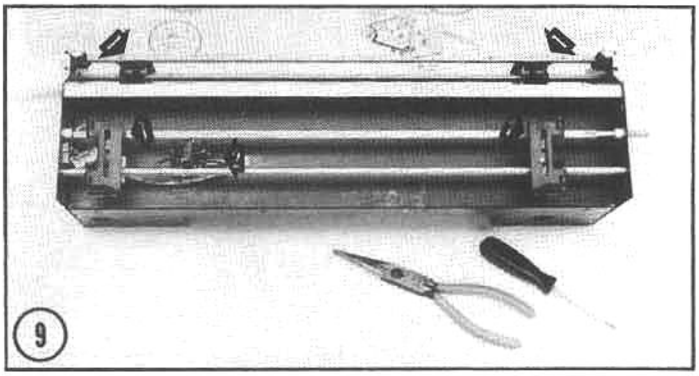


(7) The printhead moves within the carriage assembly, which consists of precise supports for the carriage, and suitable stepper motors that provide carriage motion and recirculate the continuous ribbon. Here the printhead motor (arrow 1) and ribbon motor (arrow 2) are attached to brackets that comprise the carriage assembly.

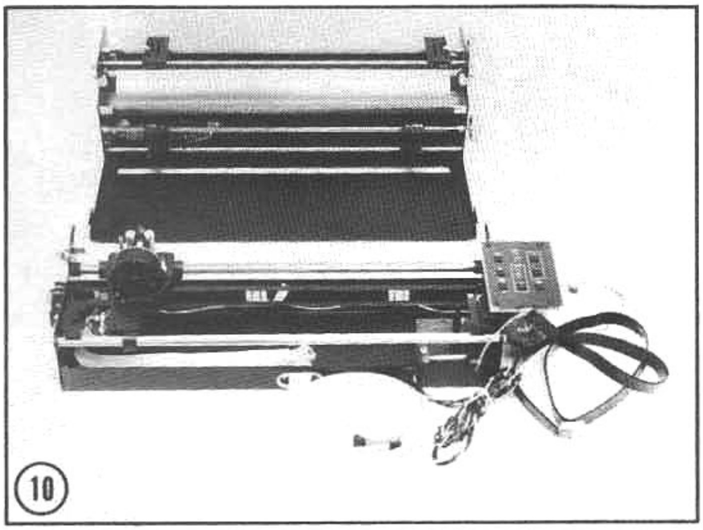




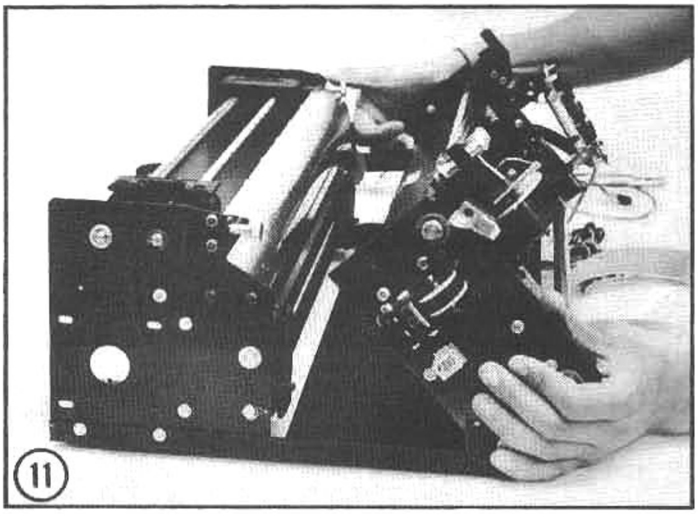
(8) Here is the completed carriage assembly. The printhead stepper motor is mounted directly under the control panel on the right, and drives the printhead with a toothed belt. Two .50-inch diameter steel shafts (arrows) guide the printhead as it moves across the paper.



(9) The next subassembly is the paper-drive unit, shown here nearly complete. This assembly uses four adjustable tractors (arrows 1) to draw the paper through the printer. If the printer runs out of paper or a paper jam occurs, the paper-out sensor (arrow 2) halts the printer.

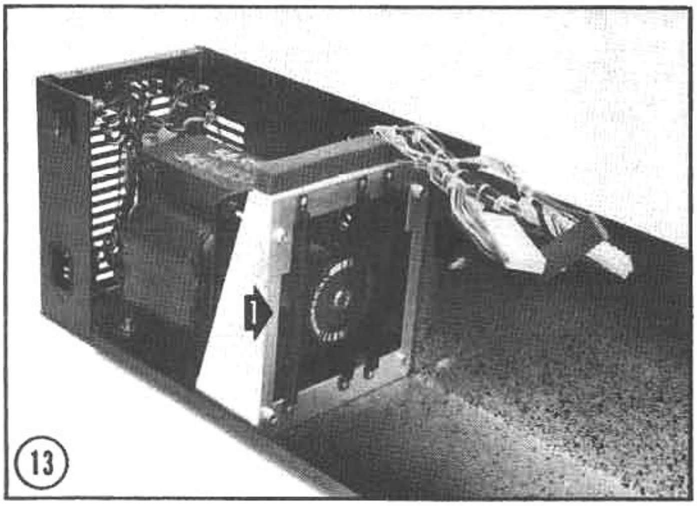


(10) The assembled paper-drive unit is mounted to a steel baseplate that will also support the carriage-drive assembly. The two subassemblies are mated together in the next step.

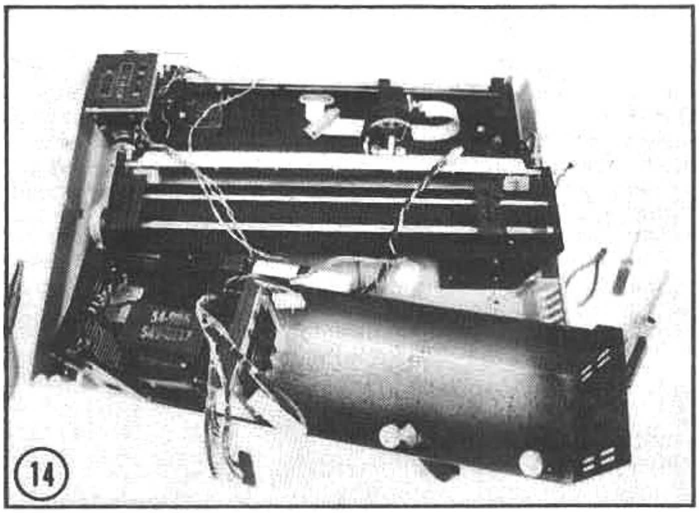


(11) The carriage-drive unit is attached to the baseplate with two swivel mounts, permitting the carriage assembly to tilt up for paper installation. When lowered, two clips hold the units together securely.

(12) The final subassembly is the power supply and electronics cage. This unit will house the three circuit boards that control the H-25. Once again, assembly begins with parts identification and organization.

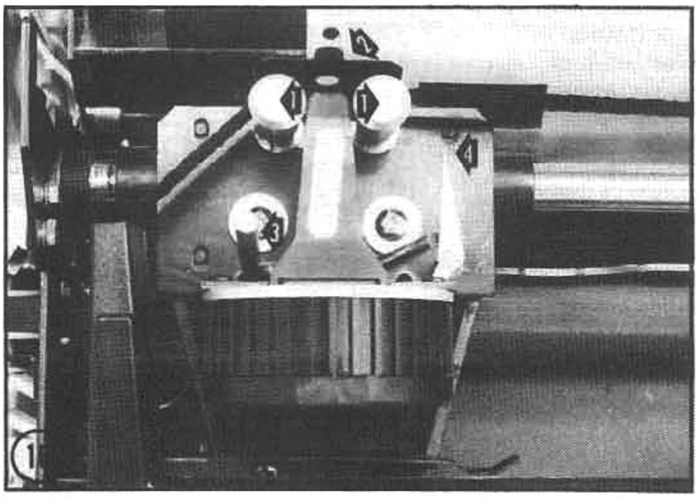


(13)The power supply is assembled in one end of the housing, including a cooling fan (arrow 1) that draws air across the circuit boards (the three circuit board guides are visible on the fan mounting bracket). The harnesses with connectors (arrow 2) are pre-assembled by Heath; they will be connected to the circuit boards in a subsequent step.

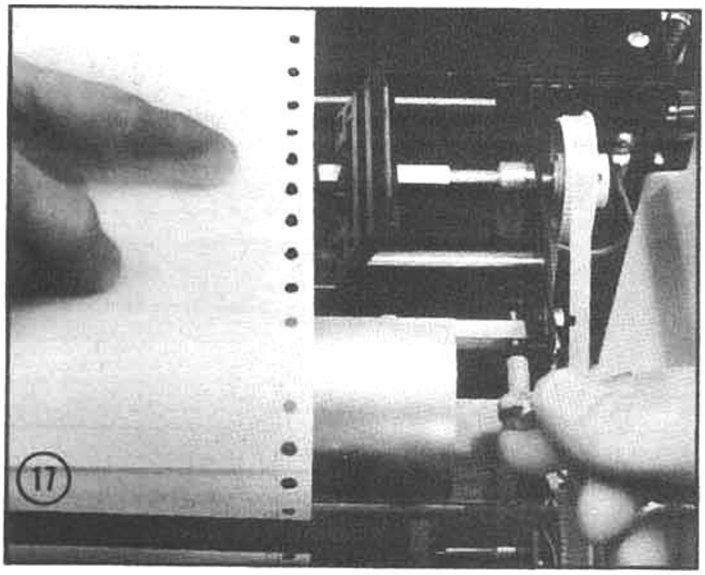


(14) Next, the power supply, carriage and paper-drive assemblies are installed in the printer baseplate. The small harnesses connect to various sensors located in the subassemblies.

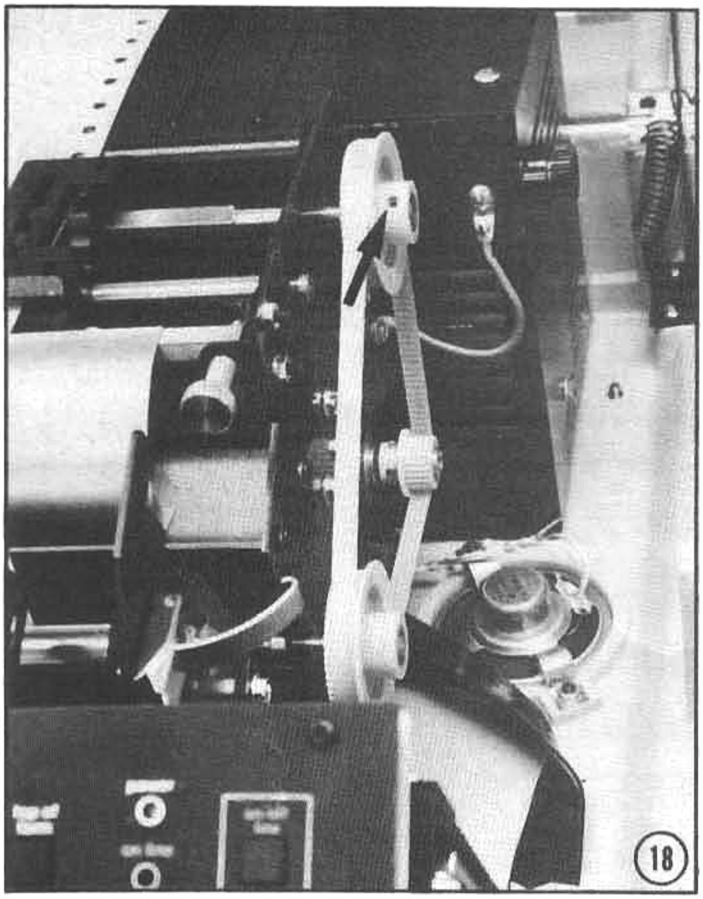
(15) When the power supply and cables are installed, the main circuit boards are slipped in place and the power supply lines are connected. At this point, a test (with an ohm meter) is performed to ensure proper wiring of the AC lines and circuit ground.



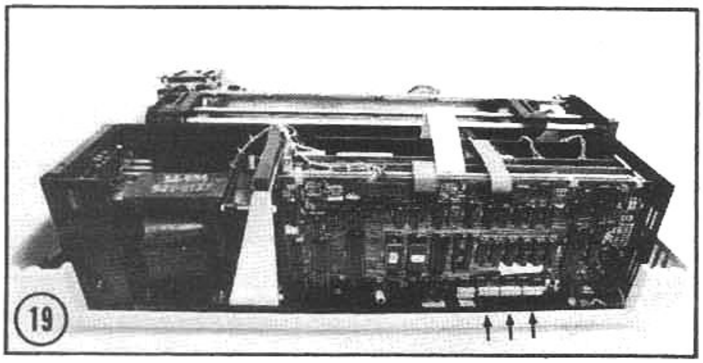
(16) Final adjustments are made, and a "power-on" test is performed. Note that the ribbon passes between the ribbon guides (arrow 1) and the ribbon shield (arrow 2). This is effectively prevents the ribbon from smudging the paper. Printhead-to-platen spacing is also adjusted with the lever (arrow 3) using the pointer as an indicator (arrow 4).



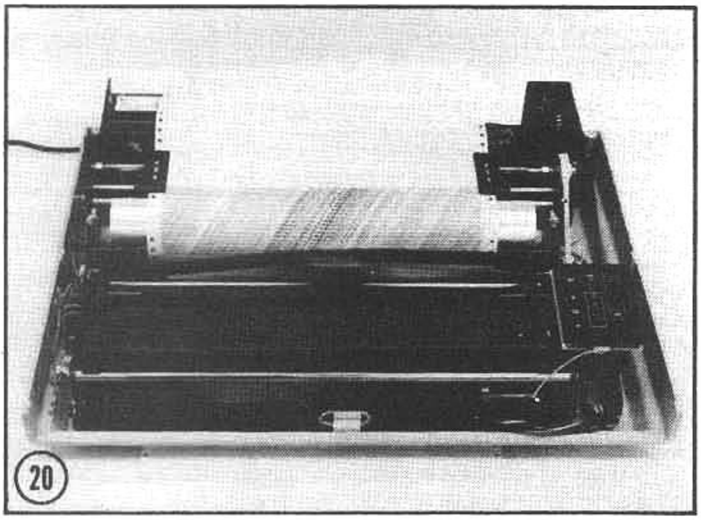
(17) The next two adjustments are interrelated, and if they are not completed properly, paper will not feed straight or reliably. The knurled knob (one is located on each end of the printer drive assembly) centers the punched holes in the paper with the pins in the upper tractor drives. Rotate the adjustment until the paper holes are centered on each end, if the range-of-adjustment is inadequate to center the holes, the next step is required.



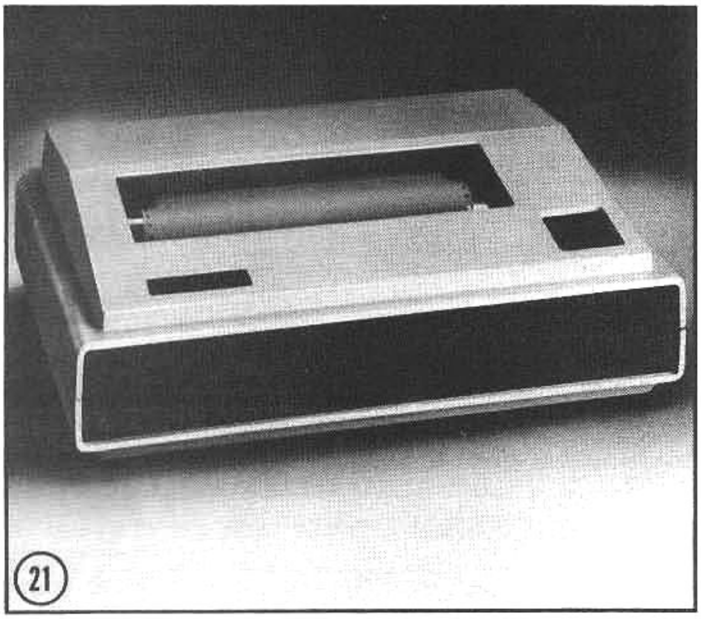
(18) Since the paper is grasped by four tractors, adjusting the space between the upper set of tractors and the lower set is important. This adjustment is made with the adjustment knobs (as indicated in the previous step). If insufficient travel is provided by the knobs, center the knob adjustments, loosen the set screw (arrow) and manually move the upper tractor shaft to center the drive pins in the paper holes. When completed, retighten the set screw.



(19) After these adjustments, the printer configuration switches (arrow) and all connecting cables should be rechecked. The printer is then ready for paper and a full power-on test.



(20) Power is applied, and the printer is instructed to perform the internal self-test routine. And, it works! The entire character set of the H-25 is printed (as part of the self test), as the builder watches (hopefully, with a self-satisfied smile).



(21) After a few minutes of final assembly, the completed H-25 printer is ready for service. This is an interesting kit that can be assembled by anyone with basic mechanical skills. Assembly time is slightly less than the H-89; varying between 20-30 hours, depending on the builder.

GLOSSARY

absolute binary source (.ABS) An ".ABS" suffix (extension) on an *HDOS* filename indicates the file is an "absolute code" program, i.e., it is written in *binary-coded machine language* that is directly *executable* by the *CPU*. The equivalent file in *CP/M* is labeled with a ".COM" extension.

ACE See *asynchronous communications element*

algorithm In programming: the sequence of logical procedures through which the computer accomplishes a task or solves a problem. (In mathematics: a formula).

analog An electrical signal that is continuously variable. An ordinary thermometer produces an *analog* measurement of temperature; the column of mercury rises and falls proportional to changes in the heat around the thermometer. Compared this with a *digital* signal that can only have a very restricted range of values.

ANSI escape-sequences A standardized scheme of *escape-sequences* defined by the American National Standards Institute; these sequences are recognized by most computer. They are not the same as *Heath escape-sequences*, but the Z/H *CPU board* can be reconfigured to recognize ANSI escape-sequences.

A-prompt Under *CP/M*, the prompt: "A>"

ASCII Standard *binary* codes used to represent the letters, numbers, symbols and non-displayable command codes used in computer communications; acronym for the American Standard Code for Information Interchange. See also *EBCDIC*.

assembly language A form of computer language that uses mnemonic names to stand for *machine language* instructions. Assembly language is a convenient "shorthand" form of machine language, and is therefore much easier to use, but it is more difficult to master than a *high-level language*.

asynchronous communications element (ACE) The *chip* inside a Zenith/Heath computer that regulates the flow of data between the *CPU* and the *terminal logic board* or *peripheral equipment*.

asynchronous transmission A technique for *serial transmission* of data along a wire or radio link, which surrounds each block of data *bits* with start and stop bits to separate the data blocks. It is not as efficient as *synchronous transmission*, but is better suited to man/machine communication, and is less expensive.

backup *(v)* The process of copying software, files, documents, etc. as protection against accidental erasure or loss. *(n)* A storage device on which *backup* copies are made, such as the second of two *disk drives*.

Basic A language for *programming* computers; considered easy for beginners to learn.

baud or baud rate A term for the rate of *data* transmission, in *bits per second (bps)*. For example, a modem that can send or receive *data* at the rate of 300 bits per second is sometimes called a *300-baud* modem.

binary Numbers in the system 0-1, in which either of the two numerals in the system can be used in any position. Numbers so written are also said to be "in the base 2." The binary number is ideally suited to computers, since the 1s and 0s are analogous to the "on" and "off" states in a *digital* computer.

binary code *Instructions* coded in the *binary* number system that a *CPU* can directly *execute*; no *compiling*, *interpreting*, or conversion is required.

bit An electrical state (in a computer's logic circuits) that is either "on" or "off." These states are also called (respectively) "high" or "low." *Bits* are represented in writing as (respectively) "1" or "0." These are the fundamental building blocks of all computer communication. Eight bits are commonly grouped together to form a *byte*.

bit density, recording density The organization of bits stored on a disk *track*.

bits per second (bps) The number of *bits* (not *bytes*) that pass through a communication channel in one second. For example, the main *CPU* and the *terminal section* of a Z/H-89 can transmit *data* back and forth at rates as fast as 9,600bps. Also called *baud* or *baud rate*.

boot, bootstrap After a computer has been switched on, the *operating system* may be loaded—*booted*—into *RAM memory* from a storage device (e.g., a *floppy disk*). Because the initial *boot* procedure loads only a few instructions—which the computer then uses to load the remainder of the operating system—this procedure is often likened to someone "lifting himself by his bootstraps," and has been given the fanciful nickname *boot* or *bootstrap*.

buffer A *chip* or circuit which can store characters temporarily and release at the appropriate rate for a *computer*, *terminal*, *printer* or *disk drive* to accept them.

bus A path over which information is transferred inside a computer; e.g., a *bus* links the *CPU* with the printed circuit boards on which the *RAM* chips are mounted.

byte In microcomputers, the most fundamental meaningful segment of *data* or computer *code*. A *byte* is made up of eight *bits*, and the pattern of bits in each *byte* distinguishes one *byte* from another. A *byte* may represent a letter, number or symbol.

cassette An audio cassette tape used for storing programs, *data* or text.

central processing unit (CPU) The *microprocessor* and direct support elements: the component that processes and executes software *instructions*, either from the *operating system* or auxiliary *programs*.

character-generator ROM The *ROM* that contains instructions for shaping characters displayed on the *CRT* screen.

chip A silicon wafer etched with tiny electronic circuits. Some are used for processing *data* (*microprocessors*); others are used to store *data* (*RAMs* and *ROMs*), and still others perform simple logical operations (gates, encoders, etc.).

code A widely used term with broad meaning; most commonly refers to *instructions* in *ABS* or *assembly language*, but can also mean *statements* in *high-level languages*, like *Basic*. See *binary code*. Less often refers to Morse code used in radio communication (dots and dashes).

cold boot, cold start The first time a *bootstrap* operation is performed following the switching on of power; or a boostrap performed after a *shift-reset* has been commanded.

column The vertical orientation of characters on the *CRT display*. There are 80 *columns* in which characters can be displayed on the screen of a Z/H-19, -89 or -90.

command A directive supplied to the computer by the user; usually typed on the *terminal keyboard*.

command-driven software Software is said to be *command-driven* if the user must type in the *commands* individually, rather than selecting them from a *menu*.

compiler A *programming* tool that converts an entire *high-level language* (source) program into a separate, directly//F2executable, machine language program.

computer or computer system An electronic computational machine which can execute *programs* for many and varied purposes (e.g., complex mathematical calculation, word processing, etc.

control-sequences Any key or sequence of keys that is pressed while the key marked CTRL (control) is held down. Control-sequences are used primarily to add software-defined commands to the keyboard; e.g., in the Magic Wand word processor, control-T (holding down the control key while simultaneously pressing the T key) moves the *cursor* to the top (beginning) of a file.

CPS (characters per second) A measurement of the speed of (usually) a printer: how many characters (letters, numbers or symbols) it can transfer to paper every second.

CPU Abbreviation for *central processing unit*.

CPU board or CPU card The printed circuit board that contains the *CPU*. It is, in effect, the "brain" of the computer because it contains logic circuits that control all primary system tasks.

CRT An abbreviation for Cathode-Ray Tube or Cathode-Ray Terminal: the television-like screen on which a terminal or computer displays information.

CRT-controller In a Heath/Zenith *terminal assembly*, the silicon *chip* which controls the appearance and placement of characters on the CRT screen. It is located on the *terminal logic board*.

CRT screen The *CRT* display.

cursor The point on the *CRT display* where *data* or *commands* are implemented. The Zenith/Heath terminals and computers use a flashing underscore or a flashing *reverse-video* block cursor.

daisy-wheel The element in some *letter-quality printers* which holds the fully-formed character dies at the tips of spokes in a wheel (like petals on a daisy).

data Numbers or *characters* that are manipulated by the *CPU*. When data is loaded into *RAM*, it can be altered, transferred to other storage devices, displayed on the *CRT* screen, etc. Data is not composed of *instructions* for the CPU, instead, data is manipulated by the CPU under *program* control; see *instructions* and *program*.

Database A body of knowledge stored according to some pre-set pattern, e.g., a mailing list with a person's name first, followed by a house number and street name, etc.

Database management system (DBMS) A computer program that accesses and/or rearranges a *database* in ways determined by the user.

decimal Numbers written in the system 0-1-2-3-4-5-6-7-8-9, in which any one of the ten numerals in the system can be used in any position. Numbers so written are also said to be "in the base 10."

default option The parameters of a computer system or its components which are pre-set, or will be assumed, unless changed by the user. For example, after booting up the first time and entering a date, this date will become the *default option*; every time the computer is warm booted the operating system will assume the same date is current, unless it is changed by the DATE operating system *command*.

digital Referring to electrical signals which have only a very limited number of possible values—usually two: "on" or "off;" "1" or "0," etc. In a way, a digital signal can be likened to a household light switch; it is either on or off and has no condition in between. Compare this with an *analog* signal, which can have any value along a continuous scale. Digital computers can perform mathematics using *binary* representations of *decimal* numbers (binary arithmetic).

destination The *disk* to which a file or a program is to be copied.

disk-controller (board or card) A printed circuit card, located inside a computer, that controls the action of (and the *data* flow to and from) one or more *disk drives*.

disk drive The mechanism that rotates the storage media (usually a *floppy disk*) and positions the *head* to record or retrieve *data* on the disk.

diskette A *floppy disk*.

disk operating system (DOS) An *operating system* that supports interaction between the *CPU* and *floppy disk drives*. At times, synonymous with *operating system*.

displayable character A letter, number or symbol which can be displayed on a *CRT* screen; as compared with a command code that can not be displayed (e.g., the code corresponding to the pressing of the scroll key, or the repeat key).

documentation Printed matter concerning computer hardware or software; including instruction or operator's manuals and technical specifications.

dot-matrix printer A printer which forms each letter from a matrix of individual dots, usually 5x7 or 5x9, rather than from the fully-formed letters. Dot-matrix printers may use ribbons (impact printing), or they may use an electrostatic or thermal process (non-impact).

double-density A method to store more *bits* on each disk *track*: approximately twice as many bits are stored on each track, compared to standard *single-density* recording.

dumb Antonym for *intelligence*. A *dumb* terminal is merely an input or output station for a computer; it usually does not contain a microprocessor.

dynamic RAM *Random-access memory* which must be re-recorded periodically if it is to remain stored in the semiconductor chip. The process of re-recording is called *refreshing*.

EBCDIC A unique set of *binary codes* used in most IBM terminals and computers to represent the letters, numbers, symbols and non-displayable command codes of computer communications; most *EBCDIC* codes are not compatible with *ASCII* codes.

emulate, emulator Am element that imitates the operation of something else; e.g., the Z/H-19 terminal sends signals to a Digital Equipment Corporation (DEC) computer that exactly mimic those sent by the DEC VT-52 terminal. Thus, the Z/H-19 is said to be a VT-52 *emulator*.

escape-sequence Any key or series of keys preceded by the key marked ESC (escape). Escape-sequences are used primarily to configure or send operational commands to the *terminal* or other *peripheral devices*. For example, when a Z/H terminal is *off-line*, an escape-B (the escape key, followed by the B key) moves the *cursor* down one *line* on the display. Note: the escape key is not held down while the subsequent keys are pressed.

EPROM An *erasable/programmable read-only memory*; see *PROM*.

execute The act of a computer following the elements (*instructions* or *statements*) of a *program* to perform a specific function or task. For instance, when a user commands a computer to *execute* a program, the computer begins to sequentially follow the elements contained in that program.

FIFO First-in, first-out: referring to the movement of *data* in and out of a *buffer*.

files A contiguate collection of *data* stored and manipulated as a whole; normally a file is identified by a unique name (e.g., WordStar.ABS, Readme.DOC, Invaders.BAS, Letter.BJ, etc.).

firmware Computer *instructions* or *program(s)* permanently or semi-permanently recorded in a silicon integrated circuit (a *ROM*) so that the user cannot change them.

floppy disk A thin, flexible vinyl disk coated to accept magnetic recording. Also called a *diskette*, the disk may be either 8 inches, 5.25 inches, or 3 inches in diameter. The 5.25-inch floppy disks may also be called *minifloppies*.

format or formatting (1) The adjustment of margins tab stops, page length, etc. to control or alter the printed appearance of text. (2) The process of preparing a *floppy disk* for recording or playing back *data*, by encoding *track* and *sector* information for the *disk-controller* to follow.

full-duplex When a *terminal* is *on-line* with a *host computer* and in the *full-duplex mode*, data or command characters entered on the terminal *keyboard* are tranamitted only to the *CPU* of the host, which then "echos" the characters back to the terminal for display on the *CRT*. In this mode the CRT only displays characters echoed from the host CPU, verifying that the transmitted characters were received by the computer.

function keys Special keys on a *terminal* keyboard that accomplish a pre-programmed task. On the Zenith/Heath terminal assembly some function are labeled with a specific task (e.g. Delete, Repeat, etc.); others are called special-function keys (F_1, F_2, etc.) and may be programmed by the *software* to do unique tasks. A third group of function keys is the numerical keypad; in the shifted mode these keys can be programmed to do special functions (e.g. insert character, delete line, etc.). All Zenith/Heath function keys work by sending an *escape-sequence* to the *computer*.

half-duplex When a *terminal* is *on-line* with a *host computer* and in the *half-duplex mode*, data or command characters entered on the terminal *keyboard* are simultaneously transmitted to the *CPU* of the host and the terminal *CRT* screen. In this mode the CRT will display the characters even if (for some reason) the host computer does not receive them.

handshaking Signalling methods used by computer devices to coordinate the transmission of *data* from one to the other. Handshaking ensures that *data* is transmitted and received without loss.

hard disk A rigid metal disk coated to accept magnetic recording (see magnetic tape). Unlike a *floppy disk*, a hard disk is sealed to exclude dust and moisture, and hard disks can usually store much more information.

hard-sector On a *floppy disk* structured for hard-sector *data* recording, the physical size of the *sectors* is predetermined and inflexible. See also *soft-sector*.

hardware The machinery of a computer: everything from the smallest silicon *chip* to the largest enclosure that houses the equipment.

hard-wiring, hard-wired Two or more components of a system are said to be *hard-wired* when they are physically connected to one another, as by wires. Also refers to the attributes of a computer that are determined by the electronic design, as opposed to *software* characteristics.

HDOS The Heath Disk Operating System, designed by Heath. Popular among hobbyists, it is similar in general operation to *CP/M*.

head The record/playback device inside a *disk drive* which creates or reads the magnetic structure on the *floppy disk*; storing or reading *data*. Similar in general function to the head of an audio tape recorder.

HEATH or ZDS escape-sequences A unique scheme of *escape-sequences* which Zenith/Heath and some other computers recognize. See also *ANSI escape-sequences*.

Heath Users' Group (HUG) An affiliation of Heath and Zenith computer users.

high-level language A *programming language* that uses statements which are similar to written English; it may even use English-like grammar rules (e.g., "LET A=2.1"). High-level languages, although easier to understand and use than *assembly language* or *machine language*, consume more computer resources because the *statements* must be "translated" into fundamental expressions (*machine language instructions*) that the computer (*CPU*) can understand.

host, host computer The computer which directs the activity of a terminal.

H-prompt Under *HDOS*, the prompt: ">"

index hole A small hole in a *floppy disk* that signals the *disk-controller board* that the next *sector* represents sector number one.

initialize or initializing See *format (2)*.

input/output (I/O) device Any mechanism for entering and/or extracting data: a *terminal*, a *disk drive*, and a *cassette tape* are all I/O devices.

instruction, CPU instruction A single (usually assembly language) operation that a *CPU* can process. A series of instructions that are combined in a logical sequence is a *program* (assembly-language program).

instruction set All of the *instructions* that a particular *CPU* recognizes as legitimate operations. The Z/H-89 computer uses a *Z-80* CPU that recognizes all of the instructions used by the earlier 8080 processor, plus many unique to the Z-80.

intelligent (device) Usually refers to the ability of a device to perform complex tasks independent of the computer (CPU). However, can refer to any device that performs logical operations, as if under the control of a computer.

interpreter A *programming* tool that converts *high-level language* instructions into *machine language* instructions, usually one by one, while a program is being *executed*.

I/O See *input/output*.

Kbytes 1,024 *bytes*; abbreviated *K*, *KB* or *Kbytes*. Generally, a large capital (upper-case) *K* refers to 1,024 *bytes*, and a small (lower-case) *k* refers to multiples of 1000 *bytes*.

keyboard The flat area on the front side of a terminal, containing plastic buttons representing letters, numbers and symbols or special functions. This may also be called an "alphanumeric" keyboard, and is similar to that on a conventional typewriter.

keypad The group of keys on the right side of the Z/H terminal or computer keyboard which contain the number and cursor-arrow keys, usually used for rapid entry of numerical *data*; may also be called a numerical keypad.

letter-quality printer A printer which mimics the type quality of an electric typewriter, achieved by striking paper through a ribbon with a metal or plastic die of the desired character. These dies may be arrayed on a rotating-and-tilting ball or a rotating thimble or flat wheel (*daisy-wheel*).

line The horizontal orientation of characters on the *CRT display*. There are 25 lines on which characters can be displayed on the screen of a Z/H-19, -89 or -90.

machine-independence A *program* that is not limited to one kind of *CPU*, or one brand of computer, is said to be *machine-independent*. For example, the *CP/M operating systems* is said to be "machine-independent" because it can be *configured* to work with several different computer systems; whereas *HDOS* is "machine-dependent" because it can only be used with Zenith/Heath computers.

machine language The fundamental computer language consisting of *binary-coded* instructions (1s and 0s) that can be understood by the *CPU*. Originally, it referred to a program as it was punched on paper tape; sometimes used incorrectly as a synonym for *assembly language*. Machine language is the most economical language, in terms of computing power (i.e., high speed, small size, etc.); but it is difficult to learn and work with, since a thorough understanding of computer fundamentals and specific *CPU instructions* are required.

magnetic tape A ribbon of vinyl plastic, coated with tiny iron particles that can be magnetically polarized; thereby storing audio or *data* signals.

mainframe computer A non-specific term that is generally used to refer to the largest and most powerful category of computers. Such a computer can process millions of *instructions* per second and store billions of *bytes* of *data*.

Mbytes 1,048,576 *bytes* (1000 Kbytes); abbreviated *M*, *MB* or *MByte*. See *Kbytes*.

memory A place in the computer where information is stored on silicon integrated circuits (*chips*). The size of computer memory is measured in the number of *bytes* it can retain. Sometimes synonymous with *RAM*.

memory map, memory-map ROM When an operating system is *booted*, the *CPU* organizes the *memory* into defined portions according to a structure called the *memory map*. The specific instructions for the memory map are permanently encoded in the *memory-map ROM*.

menu A list of possible actions (usually displayed on the computer screen) from which the user can select.

menu-driven software Software is said to be menu-driven if the selection of tasks is directed by the *user* based on *menu* prompts.

merge Combining information from two or more sources; e.g., when a word processor merges address lists with form letters.

microcomputer A small computer, sometimes no bigger than a typewriter. The Zenith/Heath *Z/H-89* and -90 are microcomputers. Inside them, a single chip—the size of a thumbnail—contains the *microprocessor*.

microprocessor A silicon integrated circuit that contains the *central processing unit* of a computer.

minicomputer Smaller than a *mainframe computer*, but larger than a *microcomputer*, a minicomputer is widely used for scientific research, and in medium-sized businesses. They usually cost about $50,000 to $100,000.

minifloppy A *floppy disk* 5.25-inches or smaller in diameter.

mode Often refers to the way a *terminal* transmits data and commands; see *off-line*, *on-line*, *half-duplex*, and *full-duplex*.

modem The device that links computers and terminals by telephone linè. The word is derived from its ability to *modulate* and *demodulate* digital signals.

monitor A *software* program, residing in a *PROM* that tells the computer how to perform certain tasks, such as the *bootstrap* operation.

octal Numbers in the system 0-1-2-3-4-5-6-7, in which any of the numerals can be used in any position. Numbers so written are also said to be "in the base 8." Compare with *split-octal*.

off-line A terminal is *off-line* (with respect to a computer) when it can not send or receive information (the communication link is broken). See also *on-line*.

on-line A terminal is *on-line* (with respect to a computer) when it can send or receive information. See also *off-line*.

operating system A *program* (or series of programs) that allow the user to communicate with the computer *CPU*. At times, synonymous with *disk operating system*.

package or packages (software) A set of computer *instructions* designed to complete a task, or a group of related *programs* that accomplish a specific purpose. For example, an accounting *package* may include several programs to handle individual bookkeeping chores for a small business.

parallel port A *port* that is designed to support *parallel transmission* of data, i.e., one *byte* (eight *bits*) at a time.

parallel transmission A technique for sending data along comparatively "wide" communication links, one *byte* (eight *bits*) at a time.

p-Code See *pseudo-code*.

peripheral, peripheral device Any equipment which is "outside" the computer or terminal; e.g., a *disk drive*, *printer*, *modem*, etc.

port, communication port An access for data flow to or from a computer or terminal. For example, inside a Z/H-89 computer, the *disk-controller board* is mounted on the *CPU board* with connectors that access specific *ports* to the CPU, permitting the CPU to send and receive data to/from the *floppy disk drives*.

CP/M A popular *disk operating system* for *microcomputers*; an abbreviation for Control Program for Microprocessors; Zenith offers CP/M as an alternative to its own operating system, *HDOS*.

printer A mechanism for transferring information from a computer to paper. See *letter-quality printer* and *dot-matrix printer*.

printout Generally, any reproduction of text on paper by a computer *printer*.

program A series of *instructions* that a computer (*CPU*) can follow (execute) to accomplish a task.

program generator A program that "writes" other programs, guided by menu selections from the user. Generated programs are usually not "bug free," but can save a programmer considerable time in producing the main body of the program.

programming The act of selecting and arranging the elements of a *programming language* into a specific arrangement that the computer (*CPU*) can successfully follow (*execute*) to accomplish a task.

programming language A complete collection of elementary *instructions* or *statements* and a set of "rules" (for using these elements) that a computer (*CPU*) can understand. Like any human language—written or spoken—a *programming language* has syntax, grammar and other structure requirements. If the rules of a programming language are violated, the resultant program will not function properly. Languages such as Basic, Cobol, Fortran, Pascal, and others are known as *high-level languages* because they utilize statement that approximate human language, and are easier to comprehend than fundamental *assembly language* or the *machine language* instructions that the CPU *executes*.

PROM A *read-only memory* which can be altered or re-programmed with special equipment; hence a *programmable read-only memory (PROM)*. An *erasable/programmable read-only memory (EPROM)* can be "wiped clean," usually by exposure to a strong source of ultra-violet light, and re-programmed with special equipment.

prompt A unique symbol, displayed on the *CRT* screen, which marks the place where the user is expected to type the next character or command. Under *HDOS*, the prompt looks like this: ">" and under *CP/M* the symbol "A>" is the prompt.

pseudo-code, p-code An "in-between" *language* that does not really relate to a specific *CPU* but to a universal, imaginary one. This permits *programs* to be transported from computer to computer without having to be rewritten, even if the computers use different *microprocessors*. A *hardware-dependent compiler* converts the p-code to a *machine-language* program executable by the specific CPU.

RAM See *random-access memory*.

random-access memory (RAM) Data storage space, usually within silicon chips, that stores and retrieves information electronically. The address of each bit or byte is maintained so that the *data* can be found irrespective of the location of any other bit or byte—at random.

read Information retrieved electronically is said to be *read* from its storage device. To record it is to *write* it.

read-only memory (ROM) A silicon *chip* which contains a permanently or semi-permanently recorded set of *instructions* (a *program*). These *instructions* can not be changed or altered by attempting to *write* new *instructions* or *data* to the *ROM*. ROM-based instructions are intended to be *read* only, and are used for implementing rigidly controlled tasks, such as the shape of characters on the *CRT display*, or the specific signals encoded by each key on the *keyboard*.

records This term is used in many ways, however, a common use is the reference to a block of related *data* stored by a *database management programs* (e.g., one employee's records); *data* within each *record* is usually located in *fields* (e.g., name, address, phone number, etc.).

refresh The process by which *data* stored in a *dynamic RAM* is re-recorded periodically to preserve it.

relational database A *database* in which items are accessed by their relationship to one another; e.g., if names of employees can be found according to criteria such as sex, seniority, and salary, then it should be possible, with a relational database program, to find the name of any male employee, who has been at his job at least three years, and makes more than $25,000 per year.

reverse-video When a *CRT* displays black characters against a white background, it is said to be displaying them in *reverse video*; the normal video display is white characters on a black background. Both display modes are supported on Z/H-19 terminals, Z/H-89 and Z/H-100 computers.

ROM See *read-only memory*.

routine A previously-written *software* task that may be *executed* at the user's discretion without having to be re-written. In cooking, for example, the use of a ready-made, frozen pie shells is a "routine" compared with the mixing, rolling, etc., needed to make a pie shell from scratch.

RS-232C A standardized electronic connection (located on the rear chassis plate of Zenith/Heath equipment) which is used for connecting a *printer*, a *modem* or other *peripheral device*. The RS-232C connection is often used as a *serial communication port*.

scratch pad RAM Storage space in a *random-access memory* chip, which is used as a temporary holding area.

scroll, scrolled, or scrolling The movement of text off the top or bottom of the *CRT display*. Scrolling up is moving toward the end of the text (file) and scrolling down is moving toward the beginning of the text.

sector hole(s) A series of small holes in a *floppy disk* that mark the starting points of each sector in *hard-sector formatting*.

sector(s) In the structuring of *data* on a *floppy disk*, the wedge-shaped sections which designate portions of each *track*. As a pie is cut into pieces, a disk is divided into *sectors*.

serial port A *port* that is designed to support *serial transmission* of data, i.e., one *bit* at a time.

serial transmission A technique for sending data along comparatively "thin" communication links, one *bit* at a time.

shift-reset, reset In a microcomputer, causes the *CPU* to halt its operation and jump to the memory location of the system *monitor*, and begin monitor execution. In the Z/H-89, the right-hand shift and reset keys are pressed simultaneously to initiate a system reset.

single-density The standard density of *bits* recorded on each disk *track*.

smart See *intelligent*.

soft-Sector On a *floppy disk* structured for soft-sector *data* recording, the physical size of the *sectors* is flexible, and may be set by the *disk-controller board* and the software controlling the system. See also *hard-sector*.

software Software is an all-encompassing term for computer *programs*. Software is often classified according to the task it performs. A *utility program* accomplishes elementary—but important—system-oriented tasks for the user (e.g., initializing disks, transferring files, etc.). An *application program* accomplishes specific user-oriented tasks (e.g., word processing, accounting, data collection, etc.). Software which is permanently or semi-permanently recorded in a special silicon chip (a *ROM*) may be called *firmware*.

sort Arranging things in meaningful patterns, e.g., names alphabetically, invoices numerically, or letters chronologically.

source (1) The *disk* from which a file or a program is to be copied. (2) The *assembly language* or *high-level language* program listing, before conversion into binary code.

speech recognition A computer program that compares the acoustics (sounds) of a spoken word (received from a *microphone*) with phonic representations stored in memory; if it finds a match, it displays the word or executes instruction assigned to the word.

split-octal Numbers in the system 0-1-2-3-4-5-6-7, in which there are restrictions to number and place positions, particularly in the split-octal notation used by Zenith/Heath to identify locations in *RAM*. Compared with *octal* numbers, split-octal numbers are longer, using two octal numbers separated by a space to represent larger numbers.

statement A single program step in a *high-level language* (i.e., Basic, Fortran, Cobol, etc.). A statement is *compiled* (or *interpreted*) into several *instructions* that are executed by the *CPU*.

static RAM *random-access memory* which does not have to be re-recorded periodically, and remains unchanged in a semiconductor chip unless it is deliberately altered or erased by the user (or the power is removed from the system).

synchronous transmission A technique for *serial transmission* of data along a wire or radio link, which alternates each block of data *bits* with a special block of bits called a "synchronization character." This technique is more efficient than *asynchronous transmission* and better suited to computer/computer communication, but it is also more expensive to implement.

telecommunication The sending and receiving of information over a distance by wires (ground-based phone links) or radio waves (satellite links).

terminal logic board (TLB) In a Z/H-19, the printed circuit board that controls the operation of the *terminal* and its interaction with the user and the computer. The same terminal logic board is used in the Z/H-89 and -90 to control the *terminal* section of the computer system.

terminal, terminal section or terminal assembly Generally, a typewriter-like keyboard and television-like display screen (*CRT*) that enable a user to communicate with a computer (*CPU*).

time-sharing or time-shared Many *terminals* or *microcomputers* (with special software) sharing the *CPU* of a host computer. The powerful CPUs of some large computers can simultaneously communicate with—and answer the demands of—many users.

track density A measurement of the number of *tracks* recorded on a disk. "Double track" density will approximately double the number of tracks recorded on a single disk. See also *tracks-per-inch*.

track(s) In the structuring of *data* recorded on a *floppy disk*, the concentric rings on which *data* is recorded.

track-sector One sector of one track on a disk: all the track-sectors on any one disk have identical storage capacities.

tracks-per-inch (tpi) A measurement of the number of *tracks* which are recorded on a *floppy disk*. See also *format*.

turn-key A computer system which can be used, essentially, without elaborate setup procedures, or "right out of the box." The term is derived from the idea that to use an automobile, a user has only to "turn the key" and drive it away.

user-friendly software Software is said to be user-friendly if it is designed for easy operation by a user who is not familiar with the program, often through the display of menus and error messages, etc.

video RAM The *random-access memory* inside Zenith/Heath *terminals* or *computers* which stores the character-generation *information* of *data* that is to be displayed on the *CRT screen*.

warm boot, warm start A *bootstrap* operation used to return to the *operating system* after an *applications program* has been *run*; also to log on a new disk that is loaded into a *disk drive*. In the *CP/M operating system*, a *warm boot* is initiated by a CTRL-C (holding down the control key and pressing the C key).

wildcard When files are being manipulated at the operating system level (*HDOS* or *CP/M*), certain "wildcard" characters can be used in filenames in place of other, undefined characters. For example, the character "?" in the filename "BOB????" tells the computer to find any file whose first three letters are "BOB," such as "BOBLETR," "BOBNOTE" or even "BOB," irrespective of the other characters used in the filename. This is handy for simultaneously manipulating several related files.

write Information recorded electronically is said to be *written* to a storage device; e.g., on a *floppy disk*. To retrieve that text is to *read* it.

Z80 The trade name of the microprocessor chip used in Zenith/Heath Z/H-89 computers and Z/H-19 terminals. It was developed by Zilog, Inc.

INDEX

WE'RE INTERESTED IN YOUR OPINION

We would appreciate your opinion of this publication. Please return this portion of the page (or a copy) to the publisher. Mail to:

S-A DESIGN BOOKS
515 West Lambert, Bldg "E"
Brea, CA 92621

How did we cover the subjects?
□ Covered well □ Adequate □ Poorly done

On a scale of 1 to 10 (10 being best), I would rate this book:
□ 1 to 3 □ 4 to 6 □ 7 to 9 □ 10

Other comments (how could we have improved this book, what subjects would you like to read about, etc.).

____

____

____

____

____

____

HOW TO USE ZENITH/HEATH COMPUTERS

JOIN THE S-A DESIGN "BOOK USERS GROUP"

As a member of the Users Group, you will receive periodic updates on new books and upcoming supplements, plus special membership discounts. And it's all free! Just mail in the attached post card or use this coupon (or a copy). Mail to:

S-A DESIGN BOOKS
515 West Lambert, Bldg. "E"
Brea, CA 92621

Name ______________________________

Street ______________________________

City ____________ State ________ Zip ________

NOTES

LIST OF SUPPLIERS

SYSTEM MANUFACTURERS

HEATH COMPANY
BENTON HARBOR, MI 49022
616-982-3285
BUILDS ZENITH & HEATH COMPUTERS

ZENITH DATA SYSTEMS
ST. JOSEPH, MI 49085
312-391-8192
BUILDS ZENITH & HEATH COMPUTERS

CONSULTATION

LOGICAL SOLUTIONS
904 FIRST PLACE
HERNDON, VA 22070
703-471-1408
HDOS AND CP/M BUSINESS APPLICATIONS

SOFTWARE HELPLINE
16811 EL CAMINO REAL
SUITE 131
HOUSTON, TX 77058
SOFTWARE HELP

HARDWARE

ACT
11301 SUNSET HILLS RD.
RESTON, VA 22090
703-471-6288
DISK UPGRADES

AMERICAN RESEARCH & ENGINEERING
1500 EXECUTIVE DRIVE
ELGIN, IL 60120
LOW-PROFILE DIP SWITCH REPLACEMENT

C. D. R. SYSTEMS, INC.
7667 VICKERS ST.
SUITE C
SAN DIEGO, CA 92111
714-275-1272
DISK SYSTEMS

CAMEO ELECTRONICS, INC.
1626 CLEMENTINE ST.
ANAHEIM, CA 92802
714-535-1682
HARD-DISK SYSTEM

COMPULINK CORP.
1840 INDUSTRIAL CIRCLE
LONGMONT, CO 80501 303-651-2014
HARDWARE PRINTER SPOOLER

D-G ELECTRONIC DEVELOPMENTS CO.
P.O. BOX 1124
DENISON, TX 75020
214-465-7805
VARIOUS PLUG-IN BOARDS

DATA COMPASS
2730 REGA PARK DR.
ANAHEIM, CA 92806
714-630-7450
DISK DRIVE SYSTEMS

EPSON AMERICA, INC.
23844 HAWTHORNE BLVD.
TORRANCE, CA 90505
213-378-2220
EPSON MX PRINTERS

GILTRONIX
970 SAN ANTONIO AVENUE
PALO ALTO, CA 94303
415-493-1300
RS-232 SWITCHES

HAYES MICROCOMPUTER PRODUCTS, INC.
5835 PEACHTREE CORNERS EAST
NORCROSS, GA 30092
404-449-8791
ADVANCED MODEMS, CHRONOGRAPHS

INTERNATIONAL DATA SCIENCES, INC.
7 WELLINGTON ROAD
LINCOLN, RI 02865
401-333-6200
TEST EQUIPMENT—RS-232 "BLUE BOX"

KRES ENGINEERING
P.O. BOX 17328
IRVINE, CA 92713
714-559-8579
EXPANSION INTERFACE, PLUG-IN BOARDS

KESA COMPANY
774 SAN MIGUEL AVENUE
SUNNYVALE, CA 94086
408-746-2738
COMPACT DIRECT-CONNECT MODEM

NOVATION, INC.
18664 OXNARD STREET
TARZANA, CA 91356
213-996-5060
MODEMS

THE LEADING EDGE PRODUCTS, INC.
225 TURNPIKE STREET
CANTON, MA 02021
800-343-6833
DISTRIBUTOR OF C. ITOH PRINTERS

MI-5
822 EAST COUNTY RD. 30
FT. COLLINS, CO 80525
303-669-4116
VOICE SYNTHESIZER

MAGNOLIA MICROSYSTEMS
2264 15TH AVE., WEST
SEATTLE, WA 98199
206-285-7266
DISK DRIVE SYSTEMS

MAXTEK
2908 OREGON COURT, G-3
TORRANCE,CA 90503
800-421-1423
GRAPHICS

MICROFLASH CO.
4916 B CAROL
SKOKIE, IL 60077
312-677-4928
BOARDS & EXPANSION INTERFACE

OPTIMAL TECHNOLOGY, INC.
BLUE WOOD 127
EARLSVILLE, VA 22936
804-973-5482
EPROM PROGRAMMER

PERCOM CORPORATION
11220 PAGEMILL RD.
DALLAS, TX 75243
DISK DRIVES

PF RESEARCH
866 HUMMINGBIRD DR.
SAN JOSE, CA 95125
ANTI-GLARE SCREEN

PRACTICAL PERIPHERALS, INC.
31245 LA BAYA DRIVE
WESTLAKE VILLAGE, CA 91362
213-991-8200
HARDWARE SPOOLER FOR EPSON PRINTER

QUME CORPORATION
2323 INDUSTRIAL PARKWAY WEST
HAYWARD, CA 94545
415-783-6100
DAISY-WHEEL PRINTERS

R.P. ENTERPRISES
P.O. BOX 122
LIVERMORE, CA 94550
415-443-3703
AUTO-REPEAT FEATURE FOR KEYPAD

SCITRONICS, INC.
523 SOUTH CLEWELL ST.
P.O. BOX 5344
BETHLEHEM, PA 18015
215-868-7220
REMOTE CONTROLLER

SIEMENS CORPORATION
240 EAST PALAIS ROAD
ANAHIEM, CA 92805
714-991-9700
PRINTERS, DISK DRIVES

THE GRAVEN IMAGE
205 STEVEN ST.
MASON, MI 48854
517-676-4648
ENGRAVED KEYS

TRU-DATA
19 SANDS POINT DR
TOMS RIVER, NJ 08753
201-349-3842
PARALLEL BOARD, EPROM BURNER

LANGUAGES

TINY C ASSOCIATES
P.O. BOX 269
HOLMDEL, NJ 07733
201-671-2296
TINY-C

UCSD P-SYSTEM USER'S SOCIETY
P.O. BOX 1148
LA JOLLA, CA 92038
UCSD P-SYSTEM USERS GROUP

PERIODICALS

BUSS
THE INDEPENDENT NEWSLETTER
325 PENNSYLVANIA AVENUE, S.E
WASHINGTON, DC 20003
202-544-0900
NEWSLETTER EVERY 3 WEEKS

BYTE MAGAZINE
70 MAIN STREET
PETERBOROUGH, NH 03458
603-924-9281
LARGEST MAGAZINE IN THE INDUSTRY

DESKTOP COMPUTING
80 PINE STREET
PETERBOROUGH, NH 03458
603-924-9471
MONTHLY MAGAZINE FOR THE BEGINNER

H-SCOOP
2918 S. 7TH. STREET
SHEBOYGAN, WI 53801
414-452-4172
MONTHLY NEWSLETTER FOR H8 AND H89

LIFELINES
1651 THIRD AVENUE
NEW YORK, NY 10028
MAGAZINE FOR CP/M USERS

MICRO MEDIA MAGAZINE
P.O. BOX 402286
GARLAND, TX 75040
BIMONTHLY MAGAZINE ON DISKETTE

REMARK
HEATH USER'S GROUP MAGAZINE
HILLTOP ROAD
ST. JOSEPH, MI 49085
616-982-3463
HEATH USER'S GROUP MAGAZINE

QUICKDATA COMPUTER SERVICES
2918 SOUTH 7TH ST.
SHEBOYGAN, WI 53081
414-452-4172
MONTHLY NEWSLETTER ON H-8 & H-89

SOFTWARE

ACME BUSINESS COMPUTERS
1727 EAST SPRAGUE
SPOKANE, WA 99202
509-535-1111
BUSINESS APPLICATIONS

CALIFORNIA DATA BASE
4554 CRISTY WAY
CASTRO VALLEY, CA 94546
415-886-0847
GAMES

CHERRY ENGINEERING
BOX 2b16
LAUREL, MD 20810
301-596-4236
CP/M-GENERAL PURPOSE

CLARK SYSTEMS CORPORATION
P.O. BOX 490156
ATLANTA, GA 30349
404-964-1262
BUSINESS PROGRAMS

COM BUSINESS COMPUTER SYSTEM, INC.
551 EAST GENESEE ST.
P.O. BOX 159
FAYETTEVILLE, NY 13066
315-637-3125
CP/M & HDOS PROGRAMS

COMPUTER CONSULTING SERVICES
184 WASHINGTON ST.
NEW BEDFORD, MA 02740
617-994-4619
WORD PROCESSING

COMPUVIEW PRODUCTS INC.
618 LOUISE
ANN ARBOR, MI 48103
313-996-1299
CP/M WORD PROCESSING

COMSOFT
665 MAYBELL AVE.
PALO ALTO, CA 94306
415-493-2184
HAM RADIO PROGRAMS

CPMUG
1651 THIRD AVENUE
NEW YORK, NY 10028
LARGEST CP/M USERS GROUP

DAVID A. SHAW
469 NORTH HOWARD
ELMHURST, IL 60126
312-823-4271
HDOS ASSEMBLY PROGRAMMING

DFD SYSTEMS
4805 NORTH 107TH ST.
OMAHA, NE 68134
402-493-1390
HAM RADIO PROGRAM

EASYSOFT
122 YANKEE DR.
ST. CHARLES, MO 63301
314-946-1968
HDOS SPOOLER

ESSEX COMPUTER SCIENCE
1827 ST. ANTHONY AVE.
ST. PAUL, MN 55104
LANGUAGE—HDOS FIG-FORTH

EVRYWARE
P.O. BOX 60802
SUNNYVALE, CA 94088
GAMES

FAR FIELD SOFTWARE
1031 RE VERAND
SLIDELL, LA 70458
504-641-3741
HDOS SUPPORT FOR SUMMAGRAPHICS
DIGITIZER

FBE RESEARCH COMPANY, INC.
P.O. BOX 68234
SEATTLE, WA 98168
206-246-9815
SUPPORT FOR MX-80 & CENTRONICS 737

HEATH USERS' GROUP
HILLTOP RD.
SAINT JOSEPH, MI 49085
616-982-3463
MAIN USERS SOFTWARE LIBRARY

HOYLE & HOYLE SOFTWARE
716 SOUTH ELAM AVE.
GREENSBORO, NC 27403
919-378-1050
GAMES

J.J. THOMPSON
281 WARREN AVE.
KENMORE, KY 14217
716-873-0380
COPY PROGRAMS, CP/M TO HDOS, ETC.

JAY H. GOLD, M.D.
P.O. BOX 2024
DES MOINES, IA 50310
515-279-9821
PERSONAL FINANCE

JERRY A. PHELPS
9013 INNES TRACE RD.
LOUISVILLE, KY 40222
502-425-4765
GAMES

JESSE E. UNDERWOOD
4424 THRUSH
EL PASO, TX 79922
MODEM PROGRAMS FOR CP/M

JIM TEIXERIA/SOFTSHOP
62 CHURCHILL ST.
SUDBURY, MA 01776
617-443-9693
HDOS, SYSMOD 2, DRIVERS, UTILITIES

JOHN O'RYAN
949 SPANISH OAKS BLVD.
PALM HARBOR, FL 33563
DATABASE SOFTWARE

LIFEBOAT ASSOCIATES
1651 THIRD AVE.
NEW YORK, NY 10028
212-860-0300
VARIOUS CP/M PROGRAMS

LINDLEY SYSTEMS
21 HANCOCK ST.
BEDFORD, MA 01730
617-275-6821
CUSTOM DEVICE DRIVERS

M.H. KELLICUT, PH.D.
679 SPINDRIFT WAY
HALF MOON BAY, CA 94019
415-726-5607
WORD PROCESSING

MAKO DATA PRODUCTS
1141 #B N. RED GUM
ANAHEIM, CA 92806
714-632-8583
HDOS 2.0 MODS

MICRO ARCHITECT, INC.
96 DOTHAN ST.
ARLINGTON, MA 02174
617-643-4713
MBASIC BUSINESS APPLICATIONS

MICROPRO INTERNATIONAL CORP.
33 SAN PABLO AVENUE
SAN RAFAEL, CA 94903
415-499-9420
QUALITY PROGRAMS (WORDSTAR, ETC.)

MICROTECH EXPORTS
467 HAMILTON
PALO ALTO, CA 94301
415-324-9114
IBM SOFTWARE CONVERSION

MICROTRAN
76 FLINTWELL WAY
SAN JOSE, CA 95138
408-226-4122
HDOS FORTRAN SUPPORT

MLM ASSOCIATES
5621 MAPLE HEIGHTS CT.
PITTSBURGH, PA 15232
412-683-4742
MBASIC MORSE CODE PROGRAMS

MRZ DATA SYSTEMS INC.
P.O. BOX 2571
WARMINSTER, PA 18974
CP/M SINGLE DISK UTILITIES

NEWLINE SOFTWARE
P.O. BOX 402
LITTLETON, MA 01460
617-486-8535
WORD PROCESSING & BUSINESS

OASIS SYSTEMS
2765 REYNARD WAY
SAN DIEGO, CA 92103
714-291-9489
THE WORD PLUS—SPELLING CHECKER

PALM BAY SOFTWARE
1190 S.E. CYPRESS LN.
BAY, FL 32905
305-724-9092
HDOS UTILITIES

PATRICK M. MCNALLY
P.O. BOX 576
HALEIWA, HAWAII 96712
WORDSTAR MODS & SIMULATION PGMS.

POWERLINE SYSTEMS
P.O. BOX 97
LINCROFT, NJ 07738
201-842-5731
PERSONAL SOFTWARE

REACS
4005 EAST 32ND ST.
DES MOINES, IA 50317
515-266-2382
REAL ESTATE SOFTWARE

REDDI-SOFTWARE
9141 EAST GRANDVIEW ST.
MESA, AZ 85207
602-986-7536
HDOS SOFTWARE

REICHERT DIGITAL SYSTEMS
29 BLAZIER RD.
WARREN, NJ 07060
201-469-3854
DATA PLOTTING, TECH APPLICATIONS

S & M SYSTEMS INC.
P.O. BOX 1225
HAVERHILL, MA 01830
617-373-1500
CP/M BUSINESS PROGRAMS

SCOTT WITT
79 OLD HAVERSTRAW RD
CONGERS, NY 10920
914-268-6183
HDOS EDITOR

SD ENTERPRISES
9228-E REGENTS RD.
LA JOLLA, CA 92037
714-455-0892
HDOS WORD PROCESSOR

SIEBERT ICKLER
3709 BENTWOOD
CORPUS CHRISTI, TX 78415
512-855-6748
VARIOUS HDOS PROGRAMS

SOFTSTUFF
HEATH CO.
BENTON HARBOR, MI 49022
HEATH SOFTWARE

SOFTWARE DIMENSIONS
1061 BOOT RD.
DOWNINGTOWN, PA 19335
HDOS GAMES

STEVEN ROBBINS
4610 SPOTTED OAK WOODS
SAN ANTONIO, TX 78249
512-492-3384
SPECIAL HDOS EDITOR—EDIT 19

STRESS ANALYSIS ASSOCIATES
4529 ANGELES CREST HWY
SUITE 104
LA CANADA, CA 91011
213-684-2478
HDOS ENGINEERING STRESS ANALYSIS

SUNFLOWER SOFTWARE INC.
6304 WEST 76TH ST
PRAIRIE VILLAGE, KS 66204
913-648-1333
HDOS & CP/M PROGRAMS

T & E ASSOCIATES INC.
P.O. BOX 362
MILLERSVILLE, MD 21108
HDOS UTILITIES

TARCO DEVELOPMENT CORPORATION
1445 KOLL CIR. SUITE 111
SAN JOSE, CA 95112
408-947-1101
MPM, GRAPHICS

TBT DIGITAL
741 OAKWOOD AVE.
HURST, TX 76053
HDOS UTILITIES

THE KEYBOARD STUDIO
125 ASPEN
BIRMINGHAM, MI 48009
313-645-5365
DATA BASE, FINANCIAL, ETC.

THE SOFTWARE SUBSCRIPTION
P.O. BOX 5379
RICHMOND, CA 94805
415-237-2115
HDOS GAMES & EDITORS

THE SOFTWARE TOOLWORKS
14478 GLORIETTA DR.
SHERMAN OAKS, CA 91423
213-986-4885
MANY PROGRAMS FOR HDOS AND CP/M

TIDEWATER SOFTWARE INC.
P.O. BOX 4465
VIRGINIA BEACH, VA 23454
SOFTWARE UTILITIES

ULTIMETH CORP.
24025 FERNLAKE DR.
HARBOR CITY, CA 90710
213-539-4276
HDOS SUPPORT FOR MAGNOLIA 128K
RAM BOARD

WALTER A. RISON
P.O. BOX 134
PISGAH, MD 20640
HDOS SCREEN PRINTER

WILLIAM BEGGS
12400 VIEW CT. N.E.
ALBUQUERQUE, NM 87112
SPECIAL CIRCUIT ANALYSIS PROGRAM

WILLIAM W WALKER JR.
1127 LOWELL ROAD
CONCORD, MA 01742
617-369-8061
HDOS MBASIC MODELING & SIMULATION

WISCONSIN INTELLIGENT SYSTEM ENG.
BOX 344
422 THIRD ST.
BARABOO, WI 53913
608-356-9432
OSBORNE BUSINESS PROGRAMS

WORDS AND NUMBERS
614 EAST UNION STREET, #204
SEATTLE, WA 98122
206-324-7285
OSBORNE BUSINESS PROGRAMS

XTRASOFT INC.
P.O. BOX 91063
LOUISVILLE, KY 40291
502-499-1533
HDOS EDUCATIONAL & SMALL BUSINESS

SPECIAL

BEELINE SERVICES
211 EAST ALLEGAN ST.
OTSWEGO, MI 49078
616-694-9141
TYPESETTING FROM ASCII FILES

COMPUCOVER
P.O. BOX 324
MARY ESTHER, FL 32569
904-243-5793
DUST COVERS

COVER CRAFT
P.O. BOX 555
AMHERST, NH 03031
603-889-6811
DUST COVERS

DAVE POWERS
BOX 1154
TROY, MI 48099
313-879-1115
INDEX FOR REMARK ON DISK

GREGORY J. MARSH
5279 MILES CT.
WOODBRIDGE, VA 22193
703-590-3360
HDOS HARDWARE & SOFTWARE

SIGMASOFT AND SYSTEMS
9668 ARBORHILL DR.
DALLAS, TX 75243
214-341-6755
SOUND EFFECTS BOARD

SPEED O PRINTS
300 RT. 59
TALLMAN, NY 10982
914-357-2223
FORMS SUPPLIER

SUN-FLEX COMPANY INC.
20 PIMENTEL CT.
NAVATO, CA 94947
415-883-1221
NON-GLARE SCREEN

THE PC&J GRAPHICS CO. INC.
P.O. BOX 108
DEER PARK, NY 11729
516-667-8076
SUPPLIES, RIBBONS ETC.

$19.95
U.S. CURRENCY

S-A DESIGN

ISBN 0-931472-18-0