THREE ‘NEW GENERATION’ Z-80 CPUs COMPARED

See Pages 22–24

Also in this Issue

The CP/M Connection by Chris Terry.......................................................... 18
Dot Graphics on the IMSAI-VIO by Gary Sabot........................................... 30
Selecting Functions Via Tables by Fred Gohlke........................................... 42
New Life for an Old Text Editor by Bill Machrone....................................... 44

and more

Complete Table of Contents on Page 3

*CP/M is a registered trademark of Digital Research.*
First compare quality. Then compare cost.

Morrow Designs' 10 megabyte hard disk system: $3,695.

MORE MEMORY. LESS MONEY.
Compare Morrow Designs' DISCUS™ M26™ hard disk systems to any system available for S-100 or Cromemco machines. First, compare features. Then, compare cost per megabyte. The M26 works out to under $200 a megabyte. And the M10 is about half the cost of competing systems.

COMPLETE SUBSYSTEMS.
Both the M10 (8"), and the M26 (14"), are delivered complete with disk controller, cables, fan, power supply, cabinet and CP/M® operating system. It's your choice: 10 Mb 8" at $3,695 or 26 Mb 14" at $4,995. That's single unit. Quantity prices are available.

BUILD TO FOUR DRIVES.

S-100, CROMEMCO AND NORTH STAR:
The M26 and M10 are sealed-media hard disk drives. Both S-100 controllers incorporate intelligence to supervise all data transfers through four I/O ports (command, 2 status and data). Transfers between drives and controllers are transparent to the CPU. The controller can also generate interrupts at the completion of each command ... materially increasing system throughput. Sectors are individually write-protectable for multi-use environments. North Star or Cromemco? Call Micro Mike's, Amarillo, TX (806) 372-3633 for the software package that allows the M26 and M10 to run on North Star DOS. MICAH of

Sausalito, CA, (415) 332-4443, offers a CP/M expanded to full Cromemco COOS compatibility.

AND NOW, MULT-I/O™
Multi-I/O is an I/O controller that allows multi-terminal and multi-purpose use of S-100 and Cromemco computers. Three serial and two parallel output ports. Real time clock. Fully programmable interrupt controller. Designed with daisy-wheel printers in mind. Price: $299 (kit), $349 assembled and tested.

MAKE HARD COMPARISONS.

5221 Central Avenue
Richmond, CA 94804

*CP/M is a trademark of Digital Research Corp.
*Cromemco is a trademark of Cromemco, Inc.
*North Star is a trademark of North Star Computers, Inc.
MEMORY PRODUCTS FOR THE 80'S

16K Byte — Fully Static

8/16 RAM

Designed with the Future in Mind...

- Able to use the full 24-bit address bus of the IEEE S-100 Standard for a 16-megabyte address range.
- Fully static design eliminates system timing problems. Promotes reliable operation with a wider range of CPU cards and DMA devices.
- Easy to integrate into your system. Addressable on 4K boundaries. PHANTOM, extended addressing, 16-bit operation may be switched off if desired.
- Can act as either an 8-bit or 16-bit wide memory. Dynamic bus switching per the IEEE Standard.
- Fast 200 nanosecond memory chips help you keep up with the ever-rising clock speeds of newer CPUs.
- The 8/16 is the only memory board made which is designed to run at full speed with our 16-bit 8Mhz. 8086 CPU card. And, it has plenty of speed to spare.

From Seattle Computer, the Static Memory Experts

Why static memory? First, compatibility. Most S-100 products do not meet the IEEE Standard. Because of their critical timing requirements, dynamic memory boards must often be configured for a particular CPU—a situation that could make future upgrading difficult. Static memories are inherently more versatile.

Second, speed. The access times of static and dynamic memories are not comparable numbers. Static memories get a substantial head start while dynamic memories wait for a "clock." The IEEE S-100 Standard guarantees this head start is at least 70 nanoseconds. The result: most of today's dynamic memory boards will not run at full speed with tomorrow's (or even today's) faster CPUs. The 8/16 will.

The 8/16 memory card is fully assembled, tested, guaranteed one full year. Suggested retail price: 1-4, $395; 5-9, $345. Manual only — $3. Overseas purchasers add $15 per board for air shipment.

Seattle Computer Products, Inc.
1114 Industry Drive, Seattle, WA. 98188
(206) 575-1830
16 BIT 8086 MICRO SYSTEM
Tec-86

IEEE S-100 Bus — Available Now
Complete system up and operating
8086 System with Dual Floppy Disks
Available Now for only $3990
CP/M-86™ only $250

CP/M-86
TEC-86
BREAK-THRU

Complete 8086 System for CP/M-86™
CP/M-86 Available Now - $250 • IEEE 696 S-100 Compatible • 5 MHz 8086 CPU (Optional 4 or 8 MHz) • Vectored Interrupts • 24 Parallel I/O lines (three 8 bit ports) • Two RS-232 Ports • Baud rates from 50 to 19200 baud • Independent baud rate generator for each serial port • 32 Kilobytes of static RAM (Expandable) • ROM boot for CP/M-86™ • Dual 8-inch Shugart floppy disks with controller • Attractive, all metal desk top enclosure

Data Acquisition Systems and Video Digitization Systems Available
23414 Greenlawn • Cleveland, OH 44122

TECMAR, INC.
(216) 382-7599
In This Issue

The CP/M Connection ........................................... 18
Chris Terry

Three 'New Generation' Z-80 CPUs Compared ............ 22
Bill Macchrone

The SSM VB3 Video Board ..................................... 26
Jon Bondy

Choosing Between CRT Output and Printer Output ....... 29
Bob Kowitt

Dot Graphics on the IMSAI-VIO ............................ 30
Gary Sabot

An Ultra Simple Index for Sequential Files ............... 35
Fred Gohlke

Patching a CP/M Diskette on a North Star System ....... 36
Tom Wiens

Selecting Functions Via Tables ............................... 42
Fred Gohlke

New Life for an Old Text Editor ........................... 44
Bill Macchrone

An 88-S4K Design Update .................................... 48
W. Howard Adams

Two Approaches to CP/M ..................................... 51
Steven H. Leibson

Departments

Editors Page ......................................................... 6
Letters to the Editor ............................................ 8
News and Views .................................................. 12
The CP/M Bus ...................................................... 14
Software Directory ............................................ 53
New Products ..................................................... 58
Advertiser Index ................................................ 64

*CP/M is a registered trademark of Digital Research.

MICROSYSYEMS (ISSN#0199-7955) is published bi-monthly by Microsystems, P.O. Box 789-M, Morristown, NJ 07960. Controlled circulation pending at Allentown, PA.

POSTMASTER: Send address changes to: MICROSYSTEMS, P.O. Box 789-M, Morristown, NJ 07960.

Copyright © 1980 by MICROSYSTEMS--A Subsidiary of Creative Computing.
IEEE S-100 Standard—Where Art Thou?

The IEEE proposed standard for the 696-bus, better known as the S-100 bus, was developed in the spring of 1979. The official version of the proposed standard was published in the July 1979 issue of COMPUTER magazine (the IEEE's official computer magazine) and was also reprinted in the January 1980 issue of MICROSYSTEMS magazine. The authors of the standard were Kells Elmquist, Ithaca Intersystems Inc., Howard Fullmer, Parasitic Engineering Inc., David B. Gustavson, Stanford Linear Accelerator Center and George Morrow of Morrow Designs. George Morrow served as the committee chairman. Another important participant in the committee work was Robert Stewart, also the chairman of the IEEE Computer Standards Committees.

Nearly two years have passed since the standard was proposed. It has still not been adopted. The question is—what has happened? What is happening currently and when can we expect the standard to be adopted? After all, most of the other IEEE standards have moved through the adoption process in considerably less time.

It has been difficult for me to determine exactly what has happened with the standard. The primary problem is that I am on the east coast while all the committee work is being done on the west coast, and I am simply not able to attend the meetings. I have over the past year and a half made numerous phone calls to all the gentlemen listed above and have even managed to have personal meetings with two of them. I have come to the following conclusions:

The S-100 bus standard represents a substantially different situation from that of other bus standards such as the 488 and Multibus standards. In those cases the buses were first developed by one company, Hewlett-Packard in the case of 488 and Intel with Multibus, and they had clearly defined the bus before submitting them to the IEEE for adoption as standards. This was not the case with the S-100 bus.

The S-100 bus as developed by MITS in early 1975 was only very loosely defined. In fact there were about 20 undefined lines in the MITS definitions. MITS never saw fit to publish any timing definitions for the bus. Further, during 1975 through 1979 several dozen manufacturers entered the S-100 hardware market and implemented many variations on pin assignments and developed their own timing specifications.

Thus the committee was faced with the task of trying to develop a standard which would cause the least problems for existing products and yet provide features for systems that were now significantly different from the original MITS Altair system (the first computer system to use the S-100 bus). The participants in the committee all had different opinions on how these problems should be dealt with. The problem was compounded by a significant turnover on the committee and the failure of all committee members to attend all meetings.

Because of differences that developed, George Morrow left the committee in the fall of 1979, and George Fullmer took over as committee chairman.

The committee had solicited comments and criticism on the proposed standard during the fall of 1979. Close to a hundred responses were received. Some pointed out errors; most made suggestions which were easily integrated into the standard. However, there were a few questions raised on which there were differing opinions among the committee members. These led to numerous discussions during 1980. George Fullmer has informed me that these issues have been resolved and that hopefully the standard will be formally submitted for adoption by the IEEE in the late spring. George promised that I would receive a copy
The InterSystems price/performance/reliability story now has three versions.

While everyone's been busy trying to convince you that large buses housed in strong metal boxes will guarantee versatility and ward off obsolescence, we've been busy with something better. Solving the real problem with the first line of computer products built from the ground up to conform to the new IEEE 5-700 Bus Standard. Offering you extra versatility in 8-bit applications today. And a full 16 bits tomorrow.

We call our new line Series II. And even if you don’t need the full 24-bit address for up to 16 megabytes (!) of memory right now, they’re something to think about. Because of all the performance, flexibility and economy they offer. Whether you’re looking at one of our three mainframes, at a new mainframe, expanding your present one or upgrading your system with an eye to the future. (Series II boards are compatible with most existing S-100 systems and all IEEE S-100 Standard cards as other manufacturers get around to building them.)

Consider some of the features: Reliable operation to 4MHz and beyond. Full compatibility with 8- and 16-bit CPUs, peripherals and other devices. Eight levels of prioritized interrupts. Up to 16 individually-addressable DMA devices, with IEEE Standard overlapped operation. User-selectable functions addressed by DIP-switch or jumpers, eliminating soldering. And that’s just for openers.

The best part is that all this heady stuff is available now! In our advanced processor—a full IEEE Bus Master featuring Memory Map addressing to a full megabyte. Our fast, flexible 16K Static RAM and 64K Dynamic RAM boards. An incredibly versatile and economical 2-serial, 4-parallel Multiple I/O board. Our 6-serial I/O board. Our Double-Density High-Speed Disk Controller. And what is undoubtedly the most flexible front panel in the business. Everything you need for a complete IEEE S-100 system. Available separately, or all together in your choice of DPS-1 mainframe styles.

Whatever your needs, why dump your money into obsolete products labelled “IEEE timing compatible” or other words people use to make up for a lack of product. See the future now, at your Intersystems dealer or call/write for our new catalog. We’ll tell you all about Series II and the new IEEE S-100 Bus we helped pioneer. Because it doesn’t make sense to buy yesterday’s products when tomorrow’s are already here.

Ithaca Intersystems Inc.,
1650 Hanshaw Road/P.O. Box 91,
Ithaca, NY 14850
607-257-0190/TWX: 510 255 4346

InterSystems™
Ithaca Intersystems Inc.

Micros for bigger ideas.
ELECTRONIC CONTROL TECHNOLOGY's R210 is an S-100 Bus I/O Board with 3 Serial I/O Ports (UART's), 1 Parallel I/O Port, 4 Status Ports, 2K of ROM with Monitor Program and 2K of Static RAM. The R210 provides a convenient means of interfacing several I/O devices, such as CRT terminals, line printers, modems or other devices, to an S-100 Bus Microcomputer or dedicated controller. It also provides for convenient Microcomputer system control from a terminal keypad with the 8080 Apple ROM monitor containing 26 Executive Commands and I/O routines. It can be used in dedicated control applications to produce a system with as few as two boards, since the R210 contains ROM, RAM and I/O.

The standard configuration has the Monitor ROM located at F000 Hex with the RAM at F800 Hex and the I/O occupies the first block of 8 ports. Jumper areas provide flexibility to change these locations, within reason, as well as allow the use of ROM's other than the 2708 (e.g. 2716 or similar 24 pin devices). Baud rates are individually selectable from 75 to 9600. Voltage levels of the Serial I/O Ports are RS-232.

**8080 APPLE MONITOR COMMANDS**

A - Assign I/O  
B - Branch to user routine A-Z  
C - Undefined  
D - Display memory on console in Hex  
E - End of file tag for Hex dumps  
F - Fill memory with a constant  
G - GOTO an address with breakpoints  
H - Hex math sum & difference  
I - User defined  
J - Non-destructive memory test  
K - User defined  
L - Load a binary format file  
M - Move memory block to another address  
N - Nulls leader/trailer  
P - Put ASCII into memory  
Q - Query I/O ports: QI (N)-read I/O; GD(N, V)-send I/O  
R - Read a Hex file with checksum  
S - Substitute/examine memory in Hex  
T - Types the contents of memory in ASCII equivalent  
U - Unload memory in Binary format  
V - Verify memory block against another memory block  
W - Write a checksummed Hex file  
X - Examine/modify CPU registers  
Y - ‘Yes there’ search for ‘N’ Bytes in memory  
Z - ‘Z END’ address of last R/W memory location

Specializing in Quality Microcomputer Hardware  
Building Blocks for Microcomputer Systems, Control and Test Equipment  
Card Cages, Power Supplies, Mainframes, CPU’s, Memory, I/O

**ELECTRONIC CONTROL TECHNOLOGY**

763 Ramsey Ave., Hillside, N.J. 07205

(201) 888-8080
Dear Editor:

I must protest the article published in the September/October issue of your new magazine supposedly evaluating the Intel 8088.

To repeat a worn-out cliche, the article is comparing apples to oranges. How can anyone compare a 16-bit machine with an 8-bit data bus to a machine that is only 8 bits wide and only a few crude 16 bit operations. It compares a machine that is a downgrade from a full blown 16-bit micro to an enhancement of a regular 8 bit micro. The Z-80 would have been a better choice to compare to the 8080.

The article states that the 8088 is “more than 4 times faster.” Well I hope so! If it’s running at 5 MHz, I hope it executes code faster than a machine running at 1 MHz. Does this mean if it ran at 1 MHz it would actually run slower?

Of special interest was the selection of the benchmarks program. Most were designed with a 16-bit functions of the 8086 in mind; especially the graphics, 16 bit multiply and word shift routines. This isn’t exactly fair to expect an 8-bit machine to execute 16-bit operations as easily as a 16 better. After all, isn’t that the advantage of the 16-bit machines? And wasn’t it curious to note the closeness of the two machines when it came to routines that both machines could run equally well, especially when you consider the extra speed the 8086 is running?

The question of the programmer comes up at this point. Since we don’t get to see the actual routines we can only assume an Intel programmer is most familiar with the Motorola device. Sure!

Up to this point I realize that this article was given to S-100. The article may or may not reflect the facts but one is for sure. Your magazine is responsible for its publishing. How irresponsible to print an article without even an author’s name. Smacks of newspapers and television newsmen’s “informed sources.” Just another way to report a rumor. I work with Z-80 and 8086 systems and own two 6809 systems and I believe the Motorola device is the best 8-bit micro on the market. It has taken the best of its predecessors and combined them into a powerful device. I am becoming familiar with the 8088 and feel it does have many advantages. I think you could exercise a little responsibility and check facts before you publish.

John Groves
Indiana University/Purdue University/Indianapolis

Response from the author:

The comparisons between the 8088 and 6809 were extracted from a published Intel benchmark report.

I agree with Mr. Grove’s points regarding the choice of benchmarks and the competence of the Intel programmer. These issues are both subject to evaluation by each reader.

The 6809 is an excellent microprocessor. It is, in fact, for many (particularly real-time, interrupt driven) applications, the 6809 is faster than the 8088. Mr. Groves is in error when he attempts to draw conclusions based on each machine’s clock rate. First, the comparisons made were between the 5MHz 8088 an the 2MHz 6809. Second, the 8088 implements a four clock bus cycle, while the 6809 uses a one clock bus cycle. This makes direct comparisons based on clock rates meaningless.

T.W. Cantrell
San Jose, CA

Dear Editor:

With reference to “PASCAL Speed Comparisons” by Fred Greeb (Sept/Oct 1980), I want to report the microengine times. This is now appropriate for the S-100 BUS. Program A and B ran at 0.7 seconds, program C ran at 0.8 seconds, program D required 7.1 seconds. Of course this is 16 bits. I work alternately with the IMSAI and have not verified Fred’s results for the 8-bit operation.

Robert C. Luckey, MD
Richland, WA

Mr. William Gates
C/o MICROSOFT
10800 NE Eighth
Suite 619
Bellevue, WA 98004

Dear Mr. Gates,

I am writing this letter to you because a very serious question has arisen in my mind about the pricing policies of software from various vendors. I am also certain that the same questions have arisen in the minds of many other S-100 bus and CP/M users. It is for this reason that I am also sending a copy of this letter to Mr. Sol Libes of MICROSYSTEMS magazine and will also inform him of your reply.

In the November 1980 issue of BYTE magazine (page 123) you have an advertisement for mu-MATH and mu-SIMP running under CP/M with an advertised price of $250 for the package.

In this same issue (page 321) your Consumer Products Division offers ostensibly the same product for the TRS-80 at $74.95. While there is also an editorial by Gregg Williams starting at page 324, in which he alludes to certain supposed differences, I would appreciate your comments and explanation for the vast difference in price.

Furthermore, if there is a difference and the cheaper package is available for the TRS-80, should not this option also be available to those of us who use a Z-80 on the S-100 bus under CP/M?

To be frank, there has been a flurry of articles of late about the concern of various software houses about having their products ‘ripped-off’ by unscrupulous
**NEW! TPM for TRS-80 Model II**

**NEW! System/6 Package**

Computer Design Labs

The Osborne package. Requires C Basic II. 5" disks $124.95 (manual not included)

8" disks $99.95 (manual not included)

- New 8080 disk operating system
- This is not CP/M
- But unlike CP/M, this operating system was written specifically for the 8080 and takes full advantage of its extra powerful instruction set. In other words, it's not warmed over 8080 code! Available for TRS-80 (Model I or II), Tarbell, Xitan DDCC, SD Sales "VERSA-FLOPPY", North Star (SD&D), and Digital (Micro) Systems.

**SYSTEM/MONITOR BOARD (SMB II)**

A complete I/O board for 50-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2716 RAM, 2708/2716 EPROM, and 8K ROM, jump on reset circuitry. Bare board $49.95/$20.

**PAYROLL**

<table>
<thead>
<tr>
<th>Source Code Only</th>
<th>Machine Language</th>
<th>Manual $20.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Osborne package. Requires C Basic II</td>
<td>5&quot; disks $124.95 (manual not included)</td>
<td>8&quot; disks $199.95 (manual not included)</td>
</tr>
</tbody>
</table>

**ACCOUNTS PAYABLE/RECEIVABLE**

<table>
<thead>
<tr>
<th>Source Code Only</th>
<th>Machine Language</th>
<th>Manual $20.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Osborne, Requires C Basic II</td>
<td>5&quot; disks $124.96 (manual not included)</td>
<td>8&quot; disks $199.95 (manual not included)</td>
</tr>
</tbody>
</table>

**GENERAL LEDGER**

<table>
<thead>
<tr>
<th>Source Code Only</th>
<th>Machine Language</th>
<th>Manual $20.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Osborne, Requires C Basic II</td>
<td>5&quot; disks $99.95 (manual not included)</td>
<td>8&quot; disks $99.95 (manual not included)</td>
</tr>
</tbody>
</table>

**LIFER**

How many times have you written the same subroutine in each new program? Top notch professional programmers have libraries of these subroutines and use a LIFER to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL LIFER, start writing programs in a fraction of the time it took before. LIFER is compatible with Macro I, II, as well as TDL/Xitan assemblers version 2.0 or later. $99.95/$20.

**ZED**

A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95/$15.

**NEW! ZED**

8080 version of Zapple

$34.95/$15.

**DEBRIEF I**

Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any 8080 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

**DEBRIEF II**

This is an expanded debugging which has all of the features of Debug I. 76 more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur. Also, instructions may be added to and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And it's RUBBER function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

**APPLE**

A Z80 executive and debug monitor. 26 commands, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! $160.00.

**ORDERING INFORMATION**

Visa, Master Charge and C.O.D. OK. To order call or write with the following information:

1. Name of Product (e.g. Macro I)
2. Name, Address, and Phone number.
3. Price and method of payment (e.g. C.O.D.)
4. Credit card info. if applicable.

**DEALER INQUIRIES INVITED.**

OEMS

Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

- 8080 is a trademark of Zilog
- TRS-80 is a trademark for Radio Shack
- CP/M is a trademark of Computer Design Labs. It is not CP/M
- CPM is a trademark of Digital Research

**DEALER INVITATIONS.**

342 Columbus Avenue
Trenton, N.J. 08629
Dear Mr. Glueck:

While the S-100 bus is the hardware bus, it seems that Apple and TRS-80 have most of the game software. I'd really like to read their disks, but my North Star DOS and hard-sectored North Star drive won't (as far as I know) let me. I am in the market for an S-100 system (drive, controller, software patches for DOS) that would let me read these disks. Unfortunately, I can't seem to find anything like this on the market. Even if I were to do the software patching, all I find is advertisements that hint but do not promise anything. At $400 to $1000 prices, I can't afford to buy a pig in a poke—I must have promises.

It would seem to me that there is a market for this sort of thing. Could you see if any manufacturers are interested in selling to it? If not, then please just publish this letter. Maybe they can take a hint.

Also, if someone out there knows a DOS-patch to let me read those disks, I'd be very happy to hear from them.

Bill Gates
Bellevue, WA

Response from the manufacturer:

Dear Editor:

While the S-100 bus is the hardware bus, it seems that Apple and TRS-80 have most of the game software. I'd really like to read their disks, but my North Star DOS and hard-sectored North Star drive won't (as far as I know) let me. I am in the market for an S-100 system (drive, controller, software patches for DOS) that would let me read these disks. Unfortunately, I can't seem to find anything like this on the market. Even if I were to do the software patching, all I find is advertisements that hint but do not promise anything. At $400 to $1000 prices, I can't afford to buy a pig in a poke—I must have promises.

It would seem to me that there is a market for this sort of thing. Could you see if any manufacturers are interested in selling to it? If not, then please just publish this letter. Maybe they can take a hint.

Also, if someone out there knows a DOS-patch to let me read those disks, I'd be very happy to hear from them.

Bill Gates
Bellevue, WA
HOW'S YOUR MEMORY?

Our MEMORIES Have Made US Famous!

WHY?

Reliability!

That's the secret of our success. We are dedicated to producing the most reliable memories possible for the S-100 bus and the new IEEE S-100 bus.

Flexibility!

No matter what CPU you use, we have the memory for you.

SUPERAM™ 2 — We've delivered over 5,000 of these 64K dynamic RAM boards to OEMs worldwide. This board sets the standard for reliable operation.

SUPERAM™ 4 — Now available for 8085A users, with bank select and optional parity bit.

SUPERAM™ 5 — Our forthcoming IEEE-696 (S-100) compatible RAM will provide 64 Kbytes of dynamic RAM as either 32K x 16 or 64K x 8.

With ECC, too!

Announcing the SUPERMEM™ ECC! This is the ultimate in memory data protection: built-in automatic error correction. The SUPERMEM transparently corrects single bit errors and detects double bit errors.

We'll Design One for You!

If you have a special situation that calls for a custom design involving memory, processors, controllers, or even complete systems, give PIICEON a call. We might have the answer to your product need.

And That's Not All!

We also supply the following Microcomputer Boards:

- C-86 — 16-bit 8086 processor board with multi-processor arbitration logic, on-board PROM and RS-232 port.
- V-100 — 80x24 high-speed video output board.
- D-100 — Coming soon, the answer to 5½ and 8-inch Winchester control.

WRITE FOR OUR NEW BROCHURE

OEM Computer Products
2350 Bering Drive
San Jose, CA 95131
(408) 946-8030

©PIICEON, INC. 1980
CP/M & S-100 Sessions At Computer Faire

Sol Libes, Editor of Microsystems, will chair two sessions at the West Coast Computer Faire (see ad in this issue). The panelists and topics are:

Session #1—CP/M & MP/M User Meeting
- Dr. Gary Kildall, Digital Research, topic: New CP/M & MP/M developments
- Tony Gold, Lifeboat Associates, topic: CP/M applications software
- Dr. Bruce Kendall, Corvus Systems Inc., topic: A user’s perspective of CP/M

Session #2—The IEEE 696/S-100 Standard
- Howard Fullmer, Parasitic Engineering Chairperson, IEEE 696/S-100 Committee, topic: Status of standard’s approval
- Kels Elmquist, Ithaca Intersystems Inc., topic: S-100 standard’s DMA protocol
- Mark Garetz, Godbout Electronics, topic: IEEE-696 compatibility

A 5-MByte Disk System For Under $2K

Shugart Associates is offering a special deal on their SA1002 5-MByte Winchester disk system. For $1,950 you can get the drive, an SA1400 intelligent controller card, cables and documentation. The controller provides for back-up on single or double-sided floppy disks. You will have to do some minimal interfacing to get the system running.

This special deal is really intended as an evaluation offer to OEM’s considering the design of the SA1002 into their computer systems. But Shugart has placed no restrictions on the offer, which is called the “Success Kit,” and anyone can buy just one unit. To get more information call (800) 852-7888, or in California call (800) 852-7777 (operator 12).

Give Your Printer Ribbon New Life

Printer ribbons wear out pretty quickly and replacement is not cheap, particularly on the closed-loop type ribbons. Therefore, any way to extend their life is a welcome blessing. Here is one method, as told to me by a computer hobbyist friend of mine.

Carefully pry off the lid on your closed-loop ribbon case. Do it slowly so that the ribbon does not pop out of the case. Then spray the ribbon with WD-40 all-purpose lubricant. This will cause the ink from the unused portions of the ribbon to flow down onto the dry, used areas of the ribbon. Be careful not to over-spray the ribbon as this leads to streaking. Close the lid and let the ribbon sit overnight for the lubricant to do its work. You should be able to do this as many as three or four times before the ink in the ribbon is completely spent.

Computer Consultants Association

If you are doing independent computer consulting you should check out the “Independent Computer Consulting Association,” Box 27412, St Louis, Missouri 63141, phone (314) 567-9708. They publish a newsletter with tips for consultants, an annual directory and provide useful resource materials (e.g. sample contracts). They also have several chapters which hold meetings with speakers.

8086 Version Of Cobol Announced

The 8086 software picture keeps improving. Seattle Computer Products, Seattle WA, has announced a version of Microsoft Basic for the 8086 and now Microsoft, Bellevue WA, has announced Cobol-86 which will run under the CP/M-86 operating system.

It is expected that the execution time of these packages will be three times as fast as the 8080/Z80 versions. Further, because of the global call capabilities of the 8086 the packages will be better suited to multiple user systems than the 8-bit versions.

Pascal/Z User Group Releases Disks

The Pascal/Z User Group is devoted to supporting users of the Ithaca Intersystems Z80 and Z8000 hardware and software products. They have been up and running for about 8 months now. They have already released four 8" disks of public domain software (disk #5 will be released shortly) and have published four copies of their newsletter. The work has been done primarily by Charlie Foster.

The group charges $6 (U.S. funds) to get on the mailing list. They charge $10 (postpaid) for each disk of software.

For more information call or write Charlie Foster c/o ZUG, 7962 Center Parkway, Sacramento, CA 95823. His telephone is: (916)392-2798 (home), or (916)447-6077 (work).

Zilog Z8000 Home Study Course

Zilog has announced a short home-study course for the Z8000. You can study at your own pace at home. The tests you take are sent to Zilog for individual grading and critiquing. The course costs $39 and is available from: Zilog Training and Education, 10460 Bubb Road, Cupertino, CA 95014.
Is there an alternative to CP/M—Try it—you'll like it say the developers.
—Westport, CT

InfoSoft Systems, Inc., the developers of I/05 2, think they have an answer to the complaints about CP/M (the trademarked product of Digital Research). InfoSoft's head of development revealed many of the advanced features that I/05 allows while still supporting all previous CP/M programs. He discussed what the development group learned from their previous development of Cromemco's CDOS, with over 15,000 installations. I/05 2 is geared, he stated, toward the turnkey system and the developer, yet presents a friendly face to the user. Such features as turnkey startup, local batches, no system on the 'A' disk, and terminal controls are integrated into the design.

Taking concepts from Unix, I/05 2 has full hard disk support, the ability to mix various sized disks; and a clean documentation format. For the developer, I/05 is the only system available with a full mix of devices and disks supported by the developer, and a question and answer system creation process. End user price is similar to CP/M, yet the dealer has a greater margin to work with, and far more support from the factory.

The spokesman refused to comment upon rumors that other, lesser developed programs were being released for the more sophisticated hacker.

Software Beasts released from the dungeons—Westport, CT

A spokesman for InfoSoft Systems today admitted that strange creatures were being released from their development dungeons below their software sales offices in beautiful downtown Westport. "These creatures are totally unrelated to our normal superb line of micro-processor software," stated the speaker.

Stressing that the beasts were not harmful, she admitted they did tend to bewitch hackers, forcing them to tinker till all hours. Under intense cross-examination she disclosed that these creatures, while only half-formed, were very useful as assistants or homunculi for would-be software wizards. Before dissolving into a puff of green smoke, she (?) intimated that full details on capturing the creatures were available from the sales desk.

I/05 is a trademark of InfoSoft Systems.

NS8ASIC is a trademark of Infosoft Systems.

I/05 2 is a registered trademark of Digital Research, CA.
DISK DRIVE WOES? PRINTER INTERACTION?
MEMORY LOSS? ERRATIC OPERATION?

Don’t Blame The Software!
Power Line Spikes, Surges & Hash could be the culprit!
Floppies, printers, memory & processor often interact!
Our unique ISOLATORS eliminate equipment interaction
AND curb damaging Power Line Spikes, Surges and Hash.

Clear up Software and System problems
with an ISOLATOR!

ALL ISOLATORS: • 125 VAC, Standard 3-prong plug
• 1875 W Max Load — 1 KW/Socket or socket bank
• Balanced Pi Filtered sockets or socket banks
• Spike/Surge Suppression — 1000 Amps, 8/20 usec
(SUPER ISOLATORS offer expanded filtering and
Spike/Surge Suppression capabilities)

ISO-1
ISO-2
ISO-3
ISO-5
ISO-6
ISO-7

*SWITCHABLE ISOLATORS —
ALL ISOLATOR advantages
combined with the versatility,
convenience and utility of
individually switched sockets.
Each switch has associated
pilot light.

ISO-1 — 3 individually filtered sockets — $62.95
ISO-4 — 6 individually filtered sockets — $106.95
ISO-2 — 2 filtered banks; 6 sockets — $62.95
ISO-5 — 3 filtered banks; 9 sockets — $87.95

ISO-6 — 3 switched, filtered sockets — $141.95
ISO-8 — 5 switched, filtered sockets — $178.95
ISO-3 — 3 super filtered sockets — $94.95
ISO-7 — 5 super filtered sockets — $154.95

*SUPER ISOLATORS —
Cure for severe interference
problems. Useful for Industrial
applications and heavy duty con-
trolled equipment or peripherals.

ISO-7

Dual Balanced Pi Filtered sockets
Spike/Surge Suppression —
2000 Amps, 8/20 usec

*CIRCUIT BREAKER any model (add-CB) — ADD 8.00
*CKT BK/R SWITCH/PILLOT any model (CBS) — ADD 16.00

CP/M is used in a wide variety of environments on
many types of machines. It provides a powerful operating
environment and allows portable software to be created.
This regular column will serve to enhance reader knowledge
of specific facets of CP/M and applications running on
various systems. It will attempt to answer your questions
regarding CP/M and related topics. Please mail all questions
to:

The CP/M Bus
c/o Microsystems
Box 1192
Mountainside, NJ 07092

Other topics of interest will also be presented from
time to time. I also expect to describe interesting software
from the CP/M Users’ Group as space permits.

I believe that software bugs should be documented as
discovered. I will pass on information as I discovered it; if
you discover bugs, please don’t hesitate to inform me.
This month I will discuss: 1) CP/M 2.2, the newest version of
CP/M; 2) A Bug report on a problem with Lifeboat’s COPY
utility and errors in The CP/M Handbook.

CP/M—The new release
CP/M 2.2 is the latest release of the operating system.
It provides new features which are upwardly compatible
with CP/M 1.4 and allows a greater flexibility in software
development. CP/M 2 was designed with the newer flexible
and non-flexible disk technologies in mind. It can handle
devices of up to 8 megabytes capacity. It also corrects
some of the annoying aspects of release 1, including
some nice changes to parameter passing which are more
natural for 8080 and Z80 assembly language programmers.
Furthermore, file extents are handled more logically to
provide more effective use of directory space and greater
efficiency in data access. Random access file I/O is also
supported. The purpose of this overview is to describe the
benefits of release 2. This is not intended to be
tutorial so I will not discuss the new BDOS commands,
but only discuss the capabilities of the operating system.
In a future column I plan to describe additional features
which might well be added to the CP/M environment.

Enhancements to CCP and line editing
The CCP contains a new command in CP/M2. This is
USER command which sets CP/M’s path to one of 16
possible ’logical’ user areas. Each file on every disk has
a user parameter set to a value from zero to fifteen that
indicates to which user area it belongs. Each user area is
separate from all the others. The enhanced PIP utility
permits reading from another user area—but not writing
to it from a distinct area. This feature permits compatibility
with MP/M which is designed for multitasking applications. I
find these user areas useful for organizing data on disk.
For hard disk systems where a large amount of data is involved, this becomes even more useful. Because of the restriction on writing to user areas, it is necessary to move PIP to the new area via a programming "trick." Digital Research recommends that PIP be loaded into the TPA (transient program area) with DDT and then saved in the new user area with the SAVE command at the CCP level. I present an alternative method below which makes use of the "zero program," @.COM. This transient allows programs already in the TPA to be re-executed, since it is of zero length (see my article in DDJ Vol 5 #1, January 1980). I use this technique since it does not require DDT or the less convenient SAVE procedure.

We start in USER x and will initialize USER y where x\not<\leq y.

A>PIP 'A
   ; control char causes
   ; PIP to be loaded but not
   ; executed.
A>USER y <CR>
   ; enter user area y
A>SAVE @.COM <CR>
   ; create zero program
A>@ <CR>
   ; execute zero program
   ; to effectively run PIP.
*PIP.COM=PIP.COM@.COM<CR>
   ; move PIP program into
   ; current area.

...you may now move additional programs as needed.

nb: x,y are integers on the closed
   interval zero to fifteen.

Another nice feature of release 2 is that backspaces are interpreted as a delete request in line input sequences. Actual backspacing is performed instead of echoing. I have found echoing very annoying, especially when running applications programs which use the line input function of CP/M. Also, DIR listings are four columns wide which is very convenient. One needn't use a special program to read the directory in a readable way.

File attributes, enhanced STAT and PIP

File attributes are a nice addition to CP/M. One may make a file read-only, read-write or set the SYS attribute which makes it invisible to directory commands. ED.COM will not allow SYS files to be edited either. The write-protect feature is clearly useful and the SYS attribute allows standard programs to be made "invisible." This makes directory listings more readable, since standard programs don't appear. This may simplify operation for experienced users also. STAT.COM reports the write attributes flag and parentheses SYS files. Attributes are set and cleared via STAT:

STAT $R/W
   ; set all files to read-write
STAT $COM $SYS
   ; add system attribute to all
   ; .COM files
STAT $COM $DIR
   ; remove system attributes
   ; (undoes previous example)

It is also possible to list file size via the $S option which is useful for files created with random access commands supported by CP/M2. This is not necessary for files created sequentially, since the values will be the same.

STAT also has several new "devices." "DSK:" displays the attributes of all active disks in the system. (Optionally, the user may specify "dn:DSK:" where dn is the disk name.) Remember, in release 2, disks may have very diverse hardware characteristics and capacities. This option

---

MICROSYSTEMS

---

...with tiny-c two — the compiler

Tiny-c two is ten times faster than tiny-c one, with many features, including long (32 bit) integers, lots of new operators, and re-directable and direct access input/output. VIable for professional work, either systems programming or business applications.

It comes with a UNIX® style command interpreter called the "tiny-shell."® Every compiled tiny-c program becomes a new shell command. Commands can have arguments, and dash (-) options, just as real UNIX shell commands do. The < and > input/output redirection operators are supported.

Fifty standard library functions, and readily extended. The input/output functions are UNIX styles, including fopen, fprintf, etc. Both ascii and raw (binary) input/output are supported. Package is portable. Bringing it up on a new processor or new operating system should take just days. And as usual with tiny-c products, all the source code is included.

Tiny-c two is available now on standard 8" CP/M.

$250.00 - Includes Owners Manual and Disk
Manual Only $50.00
(20% Discount to tiny-c one owners)

The original tiny-c ONE is still available on a wide variety of cassettes and diskettes. This version is an interpreter, complete with a Program Preparation System. Disk or cassette versions $100 (this price includes the Owners Manual, available separately at $50). Disks: CP/M Apple DOS 3.2, H8/89 HOS, PDP-11, Flex 2.0, Northstar, CDDOS, Cassettes: KIM, SYM, TRS-80, Tarbell, Cuts.

Call or write tiny-c associates, P.O. Box 269, Holmdel, N.J. 07733 (201) 671-2296. You'll discover tiny-c is flying higher and faster. New Jersey residents include 5% sales tax. Visa or Master Card accepted. Include charge plate number with order.

UNIX is a trademark of Bell Laboratories, Inc.
tiny-c and tiny-shell are trademarks of tiny-c assoc.
CP/M Bus cont'd. . .

displays vital statistics about those characteristics. The "USR:" device returns the active user area and reports which areas have files. By the way, this option cannot be re-executed with the zero program described above, as it reports active user areas incorrectly on the second and subsequent runs of the program.

Several additions have been made to PIP. As illustrated above, the "G" option (get) allows reading files from one user area into another. Because of the write-protect option, the "W" command option has been added. This allows PIP to overwrite read-only files. If this option is not specified, PIP asks for permission to overwrite read-only files. PIP will also ignore system files unless the new "R" option is specified. Here are some examples:

```plaintext
PIP B:=A:*.TXT[W]
PIP D:=A:*.COM[R]
```

**DDT, ED and ASM**

Unfortunately, these three transients have not been improved significantly from version 1.4. I do not use ED and rarely use ASM in favor of a macro assembler. DDT is useful for modifying program and patching but I prefer a ZX80 debugger to this transient which doesn't understand ZX80 instructions. In any case, ED refuses to access $SYS or $R/O files in its operation and has line numbering set as the default since this feature is one of its greatest benefits.

**SUBMIT and XSUB**

On many timesharing systems, batch processes permit console input to originate from the batch file. SUBMIT 1.4 would not permit this but through the use of XSUB, this feature is implemented through SUBMIT 2.2. By placing XSUB as a file to execute in the batch process (.SUB) file, all line input requests by transients will be read from the disk instead of the console.

**Summary**

CP/M2 provides nice features which make it superior to version 1.4 and the increase in capabilities and efficiency make it worth the cost of upgrading. CP/M2 is a superior environment for hard disks as it provides a more flexible interface to the hardware environment.

**II. Bug Reports**

In this and succeeding columns, various corrections to CP/M related software and documentation will be made.

**Problem with COPY**

Since installing CP/M2 on my system, I have noted that the COPY utility provided by Lifeboat Associates (for CP/M2 on Micropolis) has a serious bug. It does not seem to format the last track of the disk correctly. When filling a disk to capacity, bad sector errors appear. These errors do not occur when the disks are initialized with FORMAT under CP/M 1.4 and subsequently used under CP/M2. I have not had time to debug this program but thought that this should be disclosed.

**Errors in "The CP/M Handbook with MP/M"**

I recently purchased The CP/M Handbook with MP/M, by Rodnay Zaks. (This is a Sybex book.) It is intended to be an introductory text for novices and describes both CP/M and MP/M. I found several errors which should be noted:

Is:

```plaintext
page 70: A>STAT B: <CR>
BYTES REMAINING ON B: 192K
B:R/O
```

This line does not appear.

```plaintext
page 116:
A>A:=B:FILE.INT <CR>
A>PIP A:=B:FILE.INT <CR>
A>B:=A:FILE.INT <CR>
A>PIP B:=A:FILE.INT <CR>
```

I would recommend that beginners obtain a copy of this book and study it carefully. It has sound information about the general operation of a computer room as well as good description of the CP/M and MP/M operating systems.

**Conclusion**

I hope that this and future installments of "CP/M Bus" prove useful and I would again ask for your suggestions, questions, and ideas about CP/M software and programming.
Lifelines.
The serious publication for the serious software user.

If you care, really care, about choosing and using software, you need Lifelines.

February 1981. Vol. 1 Issue #9

• A comparison of VisiCalc* & T/MAKER.*
• Abstracts and catalog of a new software diskette available from The CP/M* Users Group.
• The first in a series of articles by Osborne/McGraw/Hill on the use of their accounting packages available from The CP/M Users Group.
• A review of MACRO-60 by Microsoft highlighting its compatibility with Digital Research’s MAC.*
• A tip on how to save substantial space in your CP/M BIOS.
• Current up-to-date information on serious software for CP/M compatible computers.

Lifelines is the publication dedicated to keeping you up-to-date on happenings in the explosive microcomputer world.

Lifelines specializes in news about software for CP/M* and similar operating systems.

Lifelines does it with a guarantee of high level, in-depth analysis of software uses and capabilities.

Lifelines does it with valuable information necessary to make intelligent software buying decisions.

Lifelines does it with the latest information on The CP/M Users Group.

Lifelines does it with thought provoking discussions on many of the more controversial issues facing computer users.

How can you live without Lifelines?

Subscribe Now!
$18.00 for twelve issues: U.S., Canada, and Mexico.
$40.00 for twelve issues: all other countries.
$2.50 for each back issue: U.S., Canada, and Mexico.
$3.60 for each back issue: all other countries.

All orders must be pre-paid by check to: LIFELINES, 1651 Third Avenue, New York, N.Y. 10028—Checks must be in U.S. $, drawn on a U.S. bank. Or use your VISA or MASTERCARD. Call (212) 722-1700

*VisiCalc is a trademark of Personal Software, Inc.
CP/M and MAC are trademarks of Digital Research, Inc.
The CP/M Users Group is not affiliated with Digital Research.
T/MAKER is a trademark of P. Rozier.
The CP/M Connection, Part 4

by Chris Terry

Part IV—How to use CP/M Facilities In Your Own Applications Programs

Previous articles in this series described how the CP/M file management system works, and how the I/O system can be enhanced to give a choice of peripherals. This article discusses how the facilities provided by CP/M can be used in your own application programs, with particular emphasis on portability.

PORTABILITY CONSIDERATIONS
Avoiding System Incompatibility

Let me say at the start that, for programs intended to be run only on one's own system, there is no reason not to make use of any and all the facilities of CP/M. Disk I/O primitives and peripheral drivers in the CBIOs may be invoked with subroutine calls—but at a price. The price paid is that the program may not run on any system that uses a different disk format. Prime examples of this are the SAP (Sort And Purge directory), WDIR (Wide DIRectory), and XDIR (extended attribute DIRectory) programs in Volumes 19, 4, and 24 of the CP/M Users' Group library. These work like a charm on systems using CP/M Version 1.4 with 8" IBM-compatible (soft-sectored) drives, and are extremely useful utilities. But they all use direct calls to disk primitives in the CBIOs that assume 77 tracks, 26 sectors per track, and 128 bytes per sector. They do not run on any other disk system, nor under CP/M Version 2. SAP, in particular, destroys the directory if run on a double-density system. All these utilities need extensive recoding for double-density or hard sectoring.

Point No. 2 may seem obvious, but sometimes gets forgotten when adapting a private program for publica­tion. I have a 5K ROM monitor that combines and enhances the most useful features of Roger Amidon's Apple monitor and the old Processor Technology Software Package; it contains excellent buffer scanning and code conversion routines that I frequently call from CP/M application programs. But these are peculiar to my system, and when someone would ask me for a copy of one of my utilities, I sometimes would forget to extract one of them. Nowadays I keep the most useful ones in a .L1B file for easy inclusion in programs to run on other systems.

Point No. 3 is important and is often overlooked. CP/M has a pretty good stack, but a program that does a lot of PUSHing and POPping when subroutines are already nested to considerable depth can exhaust the CP/M stack and crash. I have encountered this problem several times, chiefly because my programming style leans toward a separate subroutine for each logical
function; my main program loop is nothing but a series of subroutine calls to major functions, and those in turn call nested subroutines to perform subfunctions. My programs may run a little more slowly because of this—but then they are I/O limited rather than CPU limited, anyway. I certainly find that this style makes the logic easy to follow when I want to modify a program six months after it was written. Further functions can be added or algorithms changed with very little trouble.

Point No. 4 is also important. The BDOS uses all registers, and even a simple call for console input or output can destroy data in BC or HL if these registers are not saved. If a macro assembler is used, code Save and Restore macros and use them before and after every call to BDOS, remembering that single-byte results are returned in the A register, and double-byte results in BC.

**Avoiding Size Incompatibility**

CP/M systems come in all sizes from 16K (minimum) up to 64K (maximum without memory Bank switching). Regardless of system size, Page 0 (locations 0000 through 00FFH) and 22 pages at the top of memory are reserved for the use of CP/M. Thus, the area available for application programs and their work space varies from 40 pages (10K) in a minimum system to 248 pages (62K) in a maximum system. This variability has implications both for application programs which make only BDOS calls, and for those which make direct calls to CBIOS routines.

**Finding Available Memory Size.** Programs which perform extensive searches, such as text editors and formatters, and database update and retrieval programs, run faster if the data to be processed (or a substantial portion of it) is available for processing in memory. Also, telecommunication programs communicating with remote computers generally require large buffers to avoid interruption of data transfers on the line by disk accesses. Programs of this type need to know the last memory location available to them, so that they can define large buffers and workspace without encroaching upon CP/M. Locations 5, 6, and 7 in Page 0 always contain a jump instruction to the BDOS entry point; thus, the last available memory address can be found by the sequence:

```
LHLD 0006H
DCX H
SHLD MEMTOP
```

where MEMTOP is a 2-byte storage location within the application program.

**Finding The CBIOS Jump Vector.** Programs which make direct calls to CBIOS routines need to know the locations of these routines. One way of doing this is to include in the application source code equate statements that give the addresses contained in the CBIOS jump vector. This is an unsatisfactory method, however, since it ties the executable .COM program to one particular CP/M size; if the system is changed the equates must also be changed and the application program reassembled. It is better to define the CBIOS locations dynamically upon entry to the application program; the program will then run correctly on any system that uses a similar disk format.

There are numerous ways of locating the CBIOS routines at run time; all depend on the fact that CP/M places a jump instruction in location 0000, followed by the address of the Warm Boot routine, which is the second item in the CBIOS jump vector. This vector contains the following items:

```
JMP BOOT ;Arrive here from cold start load
JMP WBOOT ;Arrive here for warm start
JMP CONST ;Check for console input character ready
JMP CONIN ;Read character from console input
JMP CONOUT ;Write character to console display
JMP LIST ;Write a character to list device
JMP PUNCH ;Write a character to Punch device
JMP READER ;Read a character from Reader device
JMP HOME ;Move to track 00 on selected drive
JMP SELDISK ;Select a disk drive
JMP SETTRK ;Set track number on selected drive
JMP SETSEC ;Set sector number on selected track
JMP SETDMA ;Set address of disk I/O buffer
JMP READ ;Read selected sector
JMP WRITE ;Write selected sector
JMP WBOOT ;Arrive here from cold start load
JMP WBOOT ;Arrive here for warm start
```

Once the address of the jump to WBOOT is known from locations 0001-0001, the application program can compute the address of any other CBIOS routine. The simplest way of doing this is to create an identical jump vector within the application program, as follows:

```
GETVEC: EQUVECTRS+3 ;Do not remove any items
LSET: EQUVECTRS+6 ;from this list, or addresses
SREAD: EQUVECTRS+9 ;will not match those in the
CONIN: EQUVECTRS+12 ;CBIOS jump vector.
CONOUT: EQUVECTRS+15
LIST: EQUVECTRS+18
PUNCH: EQUVECTRS+21
READER: EQUVECTRS+24
SELDISK: EQUVECTRS+27
SETTRK: EQUVECTRS+30
SETSEC: EQUVECTRS+33
SETDMA: EQUVECTRS+36
READ: EQUVECTRS+39
WRITE: EQUVECTRS+42
```

```
GETVEC: LXI D,WBOOT ;Set Destination to start of local vector.
LHLD 1 ;Get start address of CBIOS vector.
MVZ 6,42 ;Set byte count (14 jumps X 3).
GETVE1: MOV A,M ;Get a byte from CBIOS vector
DCR ;and copy it to the local vector.
INX H ;Jump the
INX D ;pointers,
DCR B ;check the byte count.
JNZ GETVE1 ;and loop till done.

;When local vector complete, fall through
;into main body of application program.
```

**BUFFERED I/O**

**Buffered Console I/O**

CP/M provides facilities for buffered I/O as well as single-character I/O. Calls to BDOS for buffered I/O have the same form as other BDOS calls—that is, the appropriate function code is placed in the C register and the starting address of the buffer in the DE register pair. A subroutine call to BDOS at the standard entry point (0005H) then initiates the operation.

Buffered console input is used by CP/M for commands and their arguments, using the default I/O buffer at 80H. On input, characters are echoed to the console display device and then accumulated in the buffer, starting at TBUF+2, and the byte count at TBUF+1 is updated. TBUF contains a constant representing the
maximum buffer length (80 characters for the CCP).

Accumulation ends when a Carriage Return is entered, and control is returned to the calling program. Application programs can also use this facility; they can use the default buffer at TBUF, or can define a buffer of up to 256 characters (including the maximum length byte, the current byte count, and the Carriage Return terminator).

Buffered console output is used mainly for messages. The print buffer function is placed in the C register and the address of the string to be printed in the DE register pair. There is no limit on the string length; multiple lines can be printed in one operation by including Carriage Return/Line Feed (ODH, OAH) line separators. Printing stops when a Dollar sign ($, 24H) is encountered and control is returned to the calling program. This function is strictly for ASCII printable characters; there are no facilities for converting hexadecimal or BCD numbers to ASCII.

**Buffered Disk I/O**

All data transfers to and from successfully opened disk files take place via a 128-byte (single-density) or 256-byte (double-density) buffer. BDOS is told the starting address of this buffer by a SETDMA function call; all subsequent read and write operation use this one-sector buffer until the address is changed by another SETDMA call to BDOS. If an application program does not execute a SETDMA call to BDOS, reading and writing takes place through the default buffer (TBUF) at 80H. The application program is responsible for filling the buffer before issuing a Write, and for extracting the data from the buffer after a Read; routines to do this are discussed later in the commented listings. The read and write calls must supply the address of a File Control Block; if reading/writing is sequential, BDOS updates the NR (Next Record) byte in the FCB after each operation.

In many applications it is desirable to read more than one sector before processing the data. In such cases, the application must define a buffer of appropriate size (4K is common, and editors may create buffers of 20K or more). A SETDMA call to BDOS establishes the address for the first read, and a further SETDMA augments the buffer address by one sector length after each read. The same procedure is used for writing multiple sectors to the disk.

In the next installment I will present examples and a complete program listing to illustrate the techniques discussed in this article.

## Software Consultants Harken!

Do you do software customizing? Do you do software package installation? Do you provide turnkey systems? Do you run a programming shop? Full-time? Part-time?

Starting with a future issue *MICROSYSTEMS* will publish a directory of "software shops." Places where computer customers can go for software package customizing. Most purchases of computer software packages find that the package does not exactly meet their needs—customizing is required—sometimes this is minor and sometimes it is major. Our directory will provide low cost advertising for "software shops." If you wish to be listed write to *MICROSYSTEMS*, Box 1192, Mountainside, NJ 07902 for information.

---

**File-Oriented Winchester Back-Up? Take a ‘TIP’ from ALLOY**

ALLOY-CPD’s ‘TIP’ (Tape Interchange Package) provides S-100 users with either a 13.4 mB Cartridge or 45 mB IBM compatible 9-Track Magnetic Tape SUBSYSTEM and a software utility to permit efficient transfer of programs and files from your WINCHESTER DISK.

Simple to install, ‘TIP’ features comprehensive menu-driven software under both CP/M™ & MP/M™ OS’s.

Over 500 of these S-100 Tape Sub Systems have been installed.

To place your order or for further details contact: Nigel R. Spicer, Director of Marketing.

DEALER and OEM inquiries invited.

CP/M & MP/M are registered trademarks of Digital Research Inc.
To preserve the integrity of the products that we sell, we wish to clarify one ever increasing problem: technical expertise (and the accompanying reduction of support requirements) and to reward it with the lowest possible discount prices. FULL WARRANTY PROTECTION IS ASSURED!!!

Due to the difficulty in providing adequate product support cross-country, we cater primarily to the experienced individual or small quantity O.E.M. buyers. We wish to recognize your technical expertise and (with both our appologies and regrets) encourage the first-time-user to purchase these products from their LOCAL dealer at his asking price ... the premium is worth the added support that he has to offer. If no local supplier is available, write or give us a call and one of our staff will be glad to offer assistance. Kindly ask us to cooperate in this matter, than you.

- D. Scott Secor, General Manager

---

**SUPERIOR PRINTER VALUES**

We will not stock printers that print ILLEGAL output (7-wire matrix printers). . . all will support TRUE TRUe case decoders.

NEW! EPSOM MX-80 8x8 matrix impact printer

Serial $175.00

Parallel $185.00

Anadex 5800 with SUPER PRINTER! 132/176 column (150,000 char/Sec)

Serial, parallel & TTY interfaces ALL standard!

C. J. Thorntwight (25-350cps) daisywheel

(same as Vieco, Exide, etc.)

Add 600 to each of the above prices for 45-50cps versions

Quad Sprint (25cps daisywheel printers)

CALL

---

**$100 BUS CIRCUIT CARD ASSEMBLIES:**

Retail OEM (1)

Intersystems DPS Front Panel card

495.00 445.00

Intersystems MPU-8000-1 CPU card

1395.00 1255.00

Intersystems MPU-8002-2 CPU card

1245.00 1120.00

Intersystems MPU-8003 card

355.00 355.00

Intersystems 16KD 16K Static (250ns)

455.00 445.00

Intersystems 64KDR 64K Dynamic RAM card

955.00 950.00

Intersystems FDCC DD DMA data channel

455.00 445.00

Intersystems VIO-1 (4P + 250 I/O card)

445.00 400.00

Intersystems VIO-3 Same w/o interrupts

355.00 350.00

Intersystems ESGC Serial I/O card

955.00 950.00

Intersystems EPDA 64K EPROM card

295.00 265.00

Intersystems EPM-2 EPROM programmer

355.00 350.00

Intersystems ARDA 94 I/O card & I/O cable

395.00 395.00

Intersystems WW1 Prototyping card

35.00 35.00

---

**CIRCUIT CARD SPECIALS**

Entire line of Vector Graphic circuit cards

20% off

Entire line of Intersystems circuit cards

10% off

Other S100 and Multibus circuit cards available soon!

**VECToR GRAPHIC SYSTEMS**

The best complete systems packages on the market! Each model includes CP/M 2.2, Microsoft BASIC, all CP/M utilities plus -SCOPE screen editor (with much improved Warp technology), RAID debugger (much like the best Development Systems!), 204 assembly language instructions, disk space as listed below, very thorough documentation (supplied to 3-ring binders), and more! Vector prices are being raised, effective 2/1/73. (Personally, we feel that they would remain the value leader at even higher prices, but don’t let the "no frills" O.E.M. pricing mean real Savings! Prices listed do not include sales tax (Minnesota residents only) or shipping charges. Prepayment eliminates shipping charges on orders within North America. We require a minimum 25% down payment on all orders. Systems houses, computer clubs and O.E.M. are encouraged to inquire for further price reductions on quantity purchases — please contact our office for formal quotations. Yes, we will export to all points of the globe — purchase of an appropriate support/repair/diagnostics kit is suggested."

---

**INTERSYSTEMS SPECIALS**

The company which helped bring us the blessing of the IEEE, now allows us to offer a generous 15%, off list for all mainframes ordered during January!

---

**VECTOR GRAPHIC SYSTEMS**

The best complete systems packages on the market! Each model includes CP/M 2.2, Microsoft BASIC, all CP/M utilities plus -SCOPE screen editor (with much improved Warp technology), RAID debugger (much like the best Development Systems!), 204 assembly language instructions, disk space as listed below, very thorough documentation (supplied to 3-ring binders), and more! Vector prices are being raised, effective 2/1/73. (Personally, we feel that they would remain the value leader at even higher prices, but don’t let the "no frills" O.E.M. pricing mean real Savings! Prices listed do not include sales tax (Minnesota residents only) or shipping charges. Prepayment eliminates shipping charges on orders within North America. We require a minimum 25% down payment on all orders. Systems houses, computer clubs and O.E.M. are encouraged to inquire for further price reductions on quantity purchases — please contact our office for formal quotations. Yes, we will export to all points of the globe — purchase of an appropriate support/repair/diagnostics kit is suggested.

---

**INTERSYSTEMS SPECIALS**

The company which helped bring us the blessing of the IEEE, now allows us to offer a generous 15%, off list for all mainframes ordered during January!

---

**INTERSYSTEMS SPECIALS**

The company which helped bring us the blessing of the IEEE, now allows us to offer a generous 15%, off list for all mainframes ordered during January!
Much has happened in the S-100 field since the first Z-80 processor cards were introduced. Most significant is the adoption by most manufacturers of the proposed IEEE standard for the bus. The general availability of denser memories, shielded motherboards, octal bus drives and more sophisticated medium-scale ICs has made the designer's job easier, while more sophistication on the part of the software engineers has increased the efficiency and utility of systems.

We will look at three of the currently available S-100 CPU cards that use the Z-80 microprocessor for its high speed and extended instruction set. I refer to them as "new generation" in that they were all introduced after the IEEE S-100/696 spec was released. I won't attempt to classify this as the second or third generation, just a new one. The three cards reviewed are the Godbout CPU-Z, the Intersystems Series II and the SSM CB2. This selection of manufacturers does not suggest that these are the only viable Z-80 processor cards to use; there just isn't room or time enough to review them all.

The three selected processor cards share several features: They are true CPU cards, meaning that they do not have I/O ports, significant system memory or disk controllers. They provide 24 pin sockets for system monitors and bootstraps, and they provide a means of controlling interrupts. Each implements some form of extended addressing. They provide bus and system controls such as power-on jump, jump on reset, generation of MWRITE, front panel status signals, wait states and clock speed.

From there, the differences begin.

Godbout CPU-Z
The Godbout CPU is the newest entry to the Z-80 processor field. It is designed to replace that 8080 in your old system or to be the foundation for a new one, and it fully implements the IEEE S-100/696 specifications. Extended addressing is accomplished through OUT instructions to port OFDH. A feature of the board is a vectored interrupt mask register, addressable through port 0FEH. It permits the programmer to select which of the S-100 V10-7 lines will be able to reach the CPU at any point in time. This is a handy way to set the priority of interrupts, as each interrupt handler routine can begin with a mask instruction that determines what, if anything, is allowed to interrupt the routine being executed.

A lot of design time went into the implementation of the on-board ROM sockets. They are designed for two 2716's. 2732's can be accommodated with a few (well-documented) trace cuts and jumpers. Both the socket address and the jump address are selectable via dip-switches. Another switch decides whether the sockets will be present in all the extended pages of memory or just in the base 64K. The bus drivers float when addressing the ROM, so that it co-exists nicely with overlapping memory — without need for the PHANTOM line or an I/O port to disable it. This implementation also gets along well with bootstraps that grab the bus, such as the Tarbell double density controller. With a system monitor ROM in the sockets, the CPU never sees the bootstrap. Reset takes you into your monitor. With the sockets disabled, the bootstrap functions normally.
When you're playing around with the skimpiest of the lot. Their weight in gold. Furthermore, the quality of the options and swapping ROMs and the like they're worth the effort. Aside from the fact that this board came up vectored interrupt controller. In addition to masking it, extended addressing is not IEEE standard in that only four additional address lines are provided out of the eight specified. Thus you can address the first megabyte of the crystal and Z-80B on a normal assembled board gets it right up there at warp factor six. You doubled your processing speed when you went from 2 to 4 MHz. Are you ready for another 50 percent increase? It should be noted that 6 MHz operation is not to IEEE spec since the bus clock line runs at 3 MHz instead of the specified 2 MHz.

Timing is generous enough to permit slower, possibly marginal memories to operate at or near full speed.

A trivial thing can alter one's entire perception of a product. Aside from the fact that this board came up running the first time I plugged it in, the thing that sold me on it was the inclusion of card eject levers. In my opinion, they should be part of the physical specifications for S-100 boards. When you're playing around with options and swapping ROMs and the like they're worth their weight in gold. Furthermore, the quality of the sockets used on the boards was the best of the group.

The documentation provided, while adequate, is the skimpiest of the lot.

Ithaca Intersystems' Series II

Ithaca Intersystems' new Z80 is actually almost a year old, and has some interesting features worth studying. It only has one ROM socket, but it will accept anything from a 2708 to a 2732 and can be addressed on any 1K boundary in the address space. Like the Godbout board, it handles interrupts, but uses the sophisticated AMD vectored interrupt controller. In addition to masking it permits setting priorities, rotation and polling. The extended addressing is not IEEE standard in that only four additional address lines are provided out of the eight specified. Thus you can address the first megabyte of the IEEE 16 Mbyte address space, sufficient for most purposes. The unique thing about Intersystems' implementation is that the extended address bits are controlled by two four-bit registers. This, in conjunction with on-board address decoding, provides either one 8K window or two 4K windows into the extended address space. The approach, reminiscent of that used on some DEC LSI-11 implementations, is good for executing programs with small overlay sections. It is not suitable for "banking" multiple 64K pages nor for running large programs in alternate pages, unless specific calls to a kernel are made to supervise address translation. Detailed programming examples are given in the manual.

The manual also goes into extreme detail on bus timing and control signals. The board permits selection of either the partial or fully latched modes of operation, with fully latched mode being preferable for high-reliability or electrically noisy environments. Fully latched mode requires memory with a worst-case access time of 160 nanoseconds or the inclusion of a wait state. The card also has the necessary circuitry to properly manage a DMA transfer on the bus by a temporary master. The objective is to provide the control circuitry on the permanent bus master so that it need not be duplicated by other cards in the system.

Intersystems boards have a reputation for working best in the DPS-1 mainframe with front panel and with other Intersystems products. In actual use, the two copies of the Series II card did not quite live up to the expectations generated by the manual. Neither board worked properly at 4 MHz in an unterminated 8-slot mother board. Neither of them liked the terminated environment in my TEI mainframe, since they would only run while on an old unshielded extender card. They would not run when plugged directly into the bus. There was some conflict between the Tarbell controller bootstrap and the jump on the reset circuitry on the CPU. It was a tossup to see which would get the system.

Once running, however, the processor was quite reliable. DMA transfers by the Tarbell board were handled without incident. The CPU did have a distressing tendency to want to keep on running when I had something else in mind, as it largely ignored the reset button. When this happened, the only recourse was to shut it down and regain control through the power on jump. With the memory I was using (Godbout 200 nsec) there was no discernible difference in operation between partially and fully latched modes.

SSM Microcomputer Products' CB-2

The SSM processor is truly remarkable. It tries, and nearly succeeds, to be all things to all people. Like other SSM products, jumper options abound. No matter how non-standard your S-100 system may be, there is a very good chance that the SSM board will run in it and make it better at the same time. It came up and ran in my systems with the same alacrity as the Godbout board and exhibited no quirks—with the exception of an occasional dubious response to reset in a 20-slot TEI box. It probably needed a slightly larger capacitor to trigger the circuit properly.

In addition to full extended addressing, the SSM board provides all the other IEEE signals necessary to get along with other state of the art components. But SSM obviously remembers their roots; you can easily get the board up to be compatible with just about anything back to an Altair 8800B. This is accomplished through wire-wrap jumpers. In fact, there are so many of them that areas of the board resemble a bed of nails for
Before you buy your new microcomputer, chances are you have a lot of questions. Important questions that could mean the difference between a working system and a wasted system. The vital ingredient is expertise. The microcomputer people at Computer Mart are expert at answering your questions and helping you put together the best system for your application. Whether it's for business, the home, or the laboratory, come see the experts at Computer Mart of New Jersey. We have the vital ingredient.

CONCLUSIONS

In summary, these boards represent an advance in the sophistication of the S-100 bus. Each has its own personality and each is a viable choice in its own right. I recommend the Godbout board if you expect to do a minimum of hardware hacking. The implementation of the ROM disable is worth the price of admission, especially if you intend to go beyond a single 64K page of memory. Two ROM sockets will prove their value once the price of the 2K X 8 RAMS comes down. I don't know of anyone else at this writing who is offering 6 MHZ operation either.

The SSM CB2 is obviously the bit-fiddler's delight. Even the extended address lines come out through a dip header before going to the bus, so you can mix them around to suit old bank select boards and the like. It is also the one to have if your system is old or non-standard and you want a strong cornerstone for entry into the current S-100 world. Or maybe you simply disagree with those guys on the IEEE standards committee and you want to do it your own way.

The Intersystems Series II gave me some hardware hassles, but I know it's a tiger in an environment that it likes. Combined with the front panel in the DPS-1 mainframe and a scope, it is the next best thing to a commercial signature analyzer. It has two test modes that facilitate hardware checkout and debugging. And I dearly love that interrupt controller.

So there you have it; three of the new breed of CPU boards. I can hardly wait for the next generation.
THE BIGGEST
Don't miss the largest microcomputer-related conference and exposition in the United States.

BUSIEST
"...teeming with people, new products, conferences, and good parties..." — InfoWorld

MOST CAPTIVATING &

FASCINATING
Hear more than 100 speakers, on as many subjects. This is an idea-sharing conference, and you just might come back as a speaker next year.

MICRO SHOW & CONFERENCE
"...wild mix of exhibitors and attendees...some of the brightest microcomputer scientists and applications engineers, as well as a horde of personal and business computer buyers...a place to learn of the future as well as profit from the present." — Robert Lively, The Sizzle Sheet

IN THE UNITED STATES
"...the West Coast Computer Faire...draws people from all over the country, and around the world..." — Dr. Adam Osborne "From the Fountainhead"

...AND THE WORLD?
"...Computer Faire appears to be heading for international status, perhaps reaching the size of NCC and WESCON..." — Dr. Adam Osborne, "From the Fountainhead"

MORE THAN 100 CONFERENCE SPEAKERS
Take Your Pick of Conference Sessions
For Example:
- Tutorials for Novices
- Tech Talks for Experts
- Low-Cost Business Computing
- Public Information Utilities
- Legal Aspects of Computing
- Inexpensive Educational Computing
- Biomedical Applications
- Computer Graphics & Art
- Computer Music
- Unusual Applications
- Microcomputing for the Physically Impaired
- Social Implications of Computers
- Simulation Applications & Exotic Games
- Users Meetings, e.g.: Apple, TRS-80, Commodore, Forth, etc.

Conference Proceedings Will Be Published & Available at the Computer Faire

MORE THAN 400 EXHIBITS
SAN FRANCISCO
Civic Auditorium and Brooks Hall
(Largest Convention Facility in Northern California)
Friday, April 3rd 9am - 6pm
Saturday, April 4th 9am - 6pm
Sunday, April 5th Noon to 5pm
Registration Includes Conference Program & Exhibits All 3 Days
ON-SITE REGISTRATION: $10

FREE! SILICON GULCH GAZETTE SUBSCRIPTION
Get a free subscription to the Silicon Gulch Gazette. Keep in touch with the microcomputer industry and the 6th Computer Faire. Loaded with news, gossip and tidbits about Silicon Valley (just down the road from the office). Latest information on speakers, exhibitors, special features of the 6th Computer Faire.

COMPUTER FAIRE
333 Swett Road Woodside, CA 94062 (415) 851-7075
Dear Faire Folks: I don't want to miss anything. Please enter my name to receive the Silicon Gulch Gazette, for FREE!
Name ____________________________
Address __________________________
City ____________________________ State __________ Zip __________

6CF-5
The SSM VB3 Video Board

by Jon Bondy

The VB3 video board is a 'third-generation' memory-mapped character-oriented video board made by SSM Microcomputer Products. Unlike the first generation (16 by 64) and second generation (24 by 80) display boards, the VB3 offers a dense display (up to 51 lines by 80 characters) with a variety of character 'attributes' such as low intensity, reverse video, underscore, blinking, user-defined characters and graphics characters. In addition, unlike most of the previous generation boards, the VB3 provides a blank select option whereby the refresh memory can be removed from the address space of the computer when not in use (even in-between writing each character to the screen), allowing almost all of the computer's 64K byte address space to be used without interference.

The VB3 comes with a thick (¼ inch) manual which covers a lot of material, including unpacking/assembly instructions, board setup, theory of operation, schematics and board layout, a short section on troubleshooting, and voluminous listings of I/O drivers. Although I did not build the kit, I did look over the assembly instructions and they appear to take the builder through the process step-by-step, with more warnings and comments than are usually found in such manuals. Board checkout is a staged process, with each section of the board being tested before the IC's for the next stage are inserted. The VB3 uses the SMC CRT5037 and CRT8002 ICs which are programmable video display chips, and the SSM manual discusses how to use their flexibility for a variety of purposes. One of the chief features of the VB3 is the access which the user gets to these programmable IC's.

The VB3 board contains sockets for 8K bytes of memory, of which only 4K bytes were installed on my board. This allowed for storage of 2048 characters (enough for an 80 by 24 character display) and 2048 attribute bytes (to determine if each character is to be displayed normally or with reverse video or with underlining or the like). If one were interested in using the VB3 to display 51 lines by 80 columns, one could install the additional 4K bytes of display memory.

In order to use the board properly, one first must initialize the attribute memory for normal display and then clear the character memory to all blanks. One might think that the default value of the attribute memory for 'normal' display would be zero, but SSM implemented the hardware so that the value '3' must be used instead. There are two ways to set the board up, either as a normal memory board (so that it is always in the address space of the computer) or with the bank select option enabled so that the memory on the board can be removed from or brought back to the address space of the computer on command. I started running the board in the former mode for simplicity, and then tried to use the more complex memory management feature later on.

The VB3 video board is a 'third-generation' memory-mapped character-oriented video board.

The VB3 allows the user to configure it in a variety of ways. The user can determine whether the board uses horizontal and vertical sync signals generated on the board itself, or whether it uses externally supplied signals. One can change the character width in a range from 6 to 16 dots, and one can address the VB3's on-board keyboard ports to any port address pairs in the 256 port range. One can instruct the VB3 to latch the S-100 bus address lines during each bus cycle, or to accept the bus data directly (through receiver buffers, of course) without latching, by setting some jumpers. One can set the display memory address to start at any multiple of 8K bytes, and one can set the CRT5037 control register port addresses to start at any multiple of 16. Finally, the bank select option can be enabled or disabled.

The VB3 which I reviewed was sent to MICROSYSTEMS by SSM already assembled, so I did not get a chance to inspect it as closely as I might have had I performed the assembly myself. Upon inspecting the

Jon Bondy, Box 148, Ardmore, PA 19003.
board, however, I noticed a few jumper wires and cut wire runs on the back side, indicating that some sort of bugs in the board layout had been fixed at the factory; I found no documentation on the changes in the manual.

The first thing I tried to do with it was to plug it in and use it simply as a memory board. The board is densely packed, and there is a voltage regulator heat sink at each edge which got in the way of my card guides, making it difficult to insert into my computer; for testing purposes, I inserted the VB3 into an extender board. When I powered up my computer to test the VB3 as a RAM board, my ROM monitor would not work; when the VB3 was removed, and the monitor began to work again. The VB3 was interfering with the rest of my system in some way.

It turned out that the factory 'assembled and tested' unit had a jumper wire missing; it was clearly described in the manual, but I had assumed that the unit would work as delivered, without changing jumpers. With the jumper applied and the various switches correctly set up, I tried again. My ROM monitor worked and so did the VB3, at least as a memory board. It passed my memory tests perfectly, but nothing appeared on the screen.

After some thought I realized that the flexibility which the programmable CRT5037 IC offered was also going to be an inconvenience, since the board was not going to do a thing until I 'programmed' it; and I would have to program it each time I reset my computer. I had encountered a similar situation when I purchased and modified the S.D Sales VB-8024 (see MICROSYSTEMS, Vol 1 No 1, p. 11), so once I realized that the VB3 was not a simple plug in board like its first- and second-generation predecessors, I began to read the I/O driver listings to see what I had to do.

I keyed in the initialization routine which I found in the manual (about 40 bytes) and ran it, and the screen filled with the garbage which was resident in the memory after it powered up. I filled the attribute memory with '3's, the value which causes the board to display normal characters, and then ran a memory test on the character storage portion of the board. Patterns began to race across the screen as the memory test progressed. Unfortunately, although the patterns were correct, the characters themselves seemed to be shimmering, as if black 'snow' were all over the picture. Deciding that the shimmering was a minor problem, I proceeded to try to use the board with its bank select feature. I never managed to solve the shimmering problem, though, and the characters remained only marginally legible.

The tests which I had run up until now were simple ones run from my ROM monitor, but they were laborious, since I had to key them in each time I powered the computer down. I wanted to continue using the VB3 with my UCSD Pascal system, but to do that I needed all 64KB of my computer as RAM (not as a video board); I needed the bank select feature to work. Unfortunately, for quite a while I could not get that feature to work at all. The VB3 moves the on-board memory into the computer's address space when the keyboard data port is written to (since under normal circumstances this would never happen), and removes it from the address space when the keyboard status port is written to. After much muttering and peering about with my scope, it turned out that the VB3 would not accept just any old port number for the keyboard port value, but if I used a port address of 0FEH, it would work and the on-board memory could be moved into and out of my computer's memory at will. To test it, I had written a short Pascal program, which follows this article, and it finally began to work. In order to get it to do so, however, I had to play with the location where I placed the display memory, since it could neither lie on top of the Pascal program nor the p-machine interpreter. This took some finagling. Unfortunately, it did not work consistently, and it is not clear whether the problem is in the VB3 or in my computer system.

```pascal
program ssmb3test;
const
  vlaC = 2081; (video chip control registers -- 080DH)
  kbdata = 254; (0FEH -- write to remove VB3 from address space)
  kdbstat = 255; (0FFH -- write to put VB3 in address space)
  video = 16385; (04000H -- address of video ram)
var
  i: integer;
  ch: char;

procedure portwrite(addr: integer; data: integer); external;
procedure memwrite(addr: integer; data: integer); external;

begin
  portwrite(vlaC+14;0); (reset video chip)
  portwrite(vlaC+10;0);  
  portwrite(vlaC+103); (character times per line)
  portwrite(vlaC+189); (interlace and sync pulse time information)
  portwrite(vlaC+2109); (scans per row and characters per scan)
  portwrite(vlaC+23); (skew bits and data rows per frame)
  portwrite(vlaC+4); (scans per frame)
  portwrite(vlaC+49); (vertical margin)
  portwrite(vlaC+23); (row address of last row on screen (scrolling))
  portwrite(vlaC+14+23); (restart timing chain)
  portwrite(kdbstat+0); (place VB3 in address space)
  (fill attribute memory with '3')
  for i := 1 to 2048 do
    memwrite(video + 4096 + i;3);
  (fill video memory with stuff)
  for i := 1 to 2048 do
    memwrite(video + i;i)
  portwrite(kbdata+0); (remove VB3 from address space)
end.
```

MICROSYSTEMS 27
Why buy used or obsolete boards, when you can purchase new IEEE standard current designs at Spectacular Savings!

ALL GODBOUT ELECTRONICS FACTORY ASSEMBLED & TESTED BOARDS

20% OFF LIST

We carry the largest variety of S-100 products in the world. For a special cash price please call.

For our 1981 Catalog, please write

Subject to available quantities. Shipping and insurance extra. VISA and Master Charge accepted at no additional charge for orders over $500.00.

S-100, inc.
7 White Place
Clark, N.J. 07066
201-382-1318

S-100

Cat-100 FULL COLOR GRAPHICS
The original 256-color imaging system with high resolution video FRAME GRABBER for the S-100 bus.

Capture and digitize a video frame in 1/60 of a second. Select the best resolution for your application, from 256 to 1280 pixels per TV line. Display your digitized or computer processed image with 256 gray levels or 256 colors on standard B&W, NTSC or RGB color TV monitors.

Compact two-board basic system

Features:
- Highest possible quality 480x512 x 8 digital video image presently available on the market
- Input capability from TV camera or other sources
- Variety of synchronization choices
- 2 selectable video A/D conversion circuits
- Choice of 1, 2, 4, 8, 16 or 32 bits per pixel
- 32K-byte image memory on the basic system
- 32, 64, 128 & 256K byte system capacity
- Up/down input
- Photographic trigger control input
- Software selectable system parameters
- Interfaces for TRS-80 and other processors
- Comprehensive line of accessories, monitors and support software

Send for free catalog

DIGITAL GRAPHIC SYSTEMS
441 California Ave., Palo Alto, CA 94306 415/494-6088

SSM VB3 cont'd...

Despite the fact that I have not yet gotten the VB3 to work satisfactorily, I think that I can make some comments about it. My initial reaction to the board (even before I received it) was that it was unwise to put the character memory in the computer's address space, since at this point in the development of S-100 bus computers the software requires more memory resources than it did only a few years ago. SSM solved that problem by allowing the user to remove the VB3 from the address space of the computer except for the times when things are actually being moved into that memory. This allows boards like the VB3 to be used with software systems like UCSD Pascal, as long as the feature works reliably. I have every confidence that my problems with this feature were due to my computer system rather than to a problem with the SSM design.

When I agreed to review the VB3, I had figured that it was just another memory-mapped video board, and that with an afternoon of playing with it I could make some useful observations about it. Unfortunately, it's too smart. I couldn't just plug it in and test it; rather, I had to program it and coddle it in order even to see a few characters on the screen. The complexity of the board is such that one can't just use it for video output. You must initialize it carefully, and then provide a non-trivial driver program to use it.

If you have a requirement for more than 24 lines of text or for a 'terminal' with complex features which you cannot find elsewhere, you should consider the VB3.

If you compare the VB3 with the S.D. Sales VB-8024, you discover that the two boards have similar capabilities (although the VB3 has more memory, more attributes, and more flexibility). The chief differences lie in the fact that the VB-8024 has its own on-board processor and driver ROM; you turn it on and use it, with no tricks or problems. The on-board processor on the VB-8024 also makes complex manipulations of screen data less of a processing burden for the main processor. The VB3 offers more flexibility and capability while at the same time requiring more care on the part of the user. The VB-8024 is a 'black box' which the user plugs in and uses, while the VB3 is a complex but capable peripheral which the user must customize carefully.

In summary, if you have a requirement for more than 24 lines of text or for a 'terminal' with complex features which you cannot find elsewhere, you should consider purchasing the VB3. I think that you will find the VB3 to be more complex and troublesome than it is worth if your application is a simple 80 by 24 terminal; in this case, the VB-8024 will give you the same functions for the same price and with a lot less software headaches.

The SSM VB3 is $375 (kit) and $440 (assembled) and is manufactured by SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131.
Choosing Between CRT Output and Printer Output

by Bob Kowitt

Some versions of Basic allow you to specify while running your program whether you want to output to your CRT terminal or to your printer. Unfortunately, one of the most widely used and powerful Basics, Microsoft Basic, does not. If you use the methods proposed in the user’s manual, you are told to use the command PRINT to go to the CRT terminal and the command LPRINT when you want to output to your printer.

There is, however, a way you can bypass this deficiency if you are using Microsoft Basic Rel. 5.0 or later, under CP/M. Locations 0000, 0001, and 0002 contain the jump to the BIOS in CP/M. Microsoft Basic uses the data stored at these locations to direct your output as you have chosen with the commands PRINT or LPRINT in your program. Using this same information, you can locate the point in memory that contains your routine to write to the terminal or to the printer.

You can bypass the use of LPRINT by fooling the Microsoft interpreter. In Microsoft BASIC 5.0 and higher, this data is stored at a location between 16000 and 18000 (decimal), depending on which release you are using. The location changed during modification of Microsoft Basic to eliminate bugs that were discovered after the original release. By including within your program the following routine, you can at any time while your program is running choose the output direction at runtime rather than being forced to duplicate the code when writing your program. You cannot poke the data directly into the jump table of CP/M because Microsoft Basic does not use this jump table after finding its location.

Lines 60 to 100 define your variables and prepare your program for further input during your program.

Poke F, OT (line 160) should be inserted before each point at which you may want to change the output. Poke F, C (line 180) should be inserted to get output back to your CRT terminal. You must put a copy of line 180 at the end of your program. If you don’t, you will be locked into your printer and not your CRT at the end of the program. Your keyboard will still be entering data to your computer but there will be output to the printer and not the CRT.

Should you get trapped in printer mode, simply type:

POKE F, C (cr) to regain control and printout at the console.
Dot Graphics on the IMSAI-VIO

by Cary Sabot

Run TRS-80, Apple or PET graphics programs on your IMSAI-VIO and similar video boards with this program.

Several of the personal computers in wide use today are capable of displaying low resolution graphics. This means that they are able to display dots, lines, pictures, or even animated characters on their screens. The TRS-80 and the Apple have this capability. Because of the widespread use of these two computers, there are many programs available which make use of their graphics capability. This article presents a program that will enable owners of the Imsai VIO (or similar memory-mapped displays, such as the Polymorphics VTI) to utilize these programs. By making a few simple changes, all TRS-80 graphics programs, most Apple graphics programs (those programs which do not make extensive use of color), and some PET graphics programs will run on your machine.

The Imsai VIO is capable of displaying special "graphics characters." (See Figure 1.) Each graphics character contains six squares. By using the proper graphics character, it is possible to turn each of these six squares on (white), or off (black) independently. The problem is: how can a single square be turned on, without disturbing the five squares that surround it?

My solution to this problem is in the form of a machine language program. (See listing #1.) It allows a Basic program to quickly and easily plot points using the VIO. It can be modified to work with other memory-mapped display boards, such as the Polymorphics VTI. The program is designed to be used in conjunction with Microsoft Basic and CP/M. (Of course, it can also be used by a machine language program.)

To plot a point, the proper graphics character must be selected; then this character must be placed in the correct memory location. If this process were to be implemented as a Basic program, it would take approximately one-half second to plot each point. If a program that plots several hundred points were run, however, those one-half seconds would add up, delaying the program. I have implemented the plotting program in machine language, because a machine language program is considerably faster than an equivalent program written in Basic. If a Basic program needs to turn a certain square "on," it simply passes its X-Y coordinates to this routine (see Figure 2) and calls it, using the USR function. The routine then turns the square "on," and subsequently returns to the Basic program. Analogous procedures may be used to determine the square's present color (black or white), or to turn it off (black).

Utilizing The Program

If you have a 30K CP/M system using the Imsai VIO, you can employ the program just as I assembled it. To use the routine (after you have POKE'd it into memory—see the Basic listing), first POKE the Y coordinate of the desired pixel into location 6889H, then POKE the X coordinate into 688AH, and POKE the function number into 688BH. The function number would be a 1 to set the pixel white. This is equivalent to the TRS-80's SET(X,Y) command. The function number would be a 0 to set the pixel black. This is identical to the TRS-80's RESET(X,Y) command. If the function number is a two, the plotting routine will determine the present status of the pixel, without disturbing it. This is similar to the TRS-80's POINT(X,Y) command. The status of the pixel is retrieved by a PEEK to 688BH. A 0 will be found there if...
Figure 1. Some of the Imsai VIO’s “graphic characters.” The number of the desired character can be placed in the VIO’s refresh memory; subsequently, the character will appear on the screen.

The pixel is black and a 1 will be found there if the pixel is white. Once the proper values for the X,Y coordinates, and the function number have been POKED into memory, jump to 6800H using the USR function to execute the plotting routine.

I have provided a sample Basic program which uses my point plotting routine. (See listing #2.) It is a modification of program on page 33 of Introduction to Low Resolution Graphics, by Nat Wadsworth. The program draws lines between random points on the screen.

This plotting routine enables you to adapt to your computer the many TRS-80, Apple, and PET programs employing graphics which are in the public domain.

Because the plotting program is written in machine language, it is necessary to reserve memory for it when MBasic is run. To do this, instead of simply typing “MBasic” to run MBasic, type “MBasic /M:&H67FF”. This sets 67FF as the highest address available for use by MBasic. The prefix “&H” indicates that a hex number will follow.

If your system is larger than 30K, you might want to reassemble the routine at a higher address, in order to fully utilize your memory. For every kilobyte of memory that you have above 30K, add 400H to 67FFH. Then reassemble the routine at the resulting address. Finally, substitute the new origin for the 67FFH in “MBasic /M:&H67FF”. Of course, the locations for POKING and PEEKING change each time a different version of this program is assembled.

If you are using a video board other than the VIO, you will probably have to reassemble the program, making one or more of the following changes (most involve the “EQUates” in the beginning of the program):

1) If your video board is addressed at a location other than 61440 (F000H), change the value in the line beginning “SCREENAD EQU…” in the listing to the correct screen address.

2) If the line length of your display is different than 80 characters per line, change the line length in the listing (“LINE EQU…” ) to the proper value, and reassemble it.

3) If a black pixel is represented by a 1, and not a 0, substitute 1 for 0 in the listing where it now reads “BLK EQU 0”.

4) Determine the value of a blank (all black) character cell and substitute this for the 80H in the line “BLKCHR EQU 80H”.

5) If the progression from black to white is different than that shown in Figure 1, you will have to modify the portion of the program (6843H thru 6863H) that calculates the bit mask. This can be a very involved process!

Now that your computer has dot graphics capability, make it work for you. Programs can be written with output in the form of a graph, instead of in numbers and letters. Programs can even be designed to imitate arcade games. One of the most exciting possibilities of this plotting routine is that it enables you to adapt to your computer the many TRS-80, Apple, and PET programs employing graphics which are in public domain.

—PROGRAM BEGINS NEXT PAGE—
POINT PLOT LISTING

LISTING #1

;******** POINT PLOT ROUTINE FOR MEMORY MAPPED DISPLAYS ********
;******** BY GARY SABOT 11/27/79 ********

F000 =
SCREENAD EQU 61440 ;ADDRESS OF VIDEO BOARD
0050 =
LINE EQU 80 ;LINE LENGTH
0000 =
BLK EQU 0 ;BIT THAT REPRESENTS A PIXEL THAT IS "OFF" (BLACK)
0800 =
BLKCHR EQU 80H ;CONTENTS OF A BLANK (BLACK)
0C00 =
CHRAND EQU 0C0H
0800 =
CHRCPI EQU 80H

;CHRAND IS "ANDED" WITH THE CONTENTS OF A CHARACTER CELL. IF
;THE CELL CONTAINS A VALID GRAPHICS CHARACTER (NON-ALPHABETIC)
;THE RESULT SHOULD EQUAL CHRCPI. IF IT DOES NOT, A BLANK
;CHARACTER WILL BE PLACED IN THE CELL.

ORG 6800H

;DATA SHOULD BE STORED IN FORM Y, X, FUNCTION
;FUNCTION #-- 0 MEANS SET BLACK, 1 MEANS SET WHITE,
;ALL ELSE MEANS RETURN DOT STATUS (0=BLACK, 1=WHITE)

6800 218958
6803 0E00
6805 7E
6806 D03
DIV: SUI 3 ;DIVIDE Y BY 3,
DIVX:
JC DIVX
INR A
ANA J!>IZ
STA INX
RRC ANI LXI MOV
MVI DAD
6810 328C68
DIVX: STA ;SAVE REMAINDER FROM
YREM ;DIVISION OF Y
;
6813 23 INX H ;READ IN VALUE OF X
6814 7E MOV A,M
6815 0F RRC
6816 E67F
6818 2100F0
681B 5F
681C 1600
681E 19
681F 115000
6822 79 MOV A,C
6823 A7
6824 CA3058
6827 3D
MULT: DCR A ;ADD Y*LINE LENGTH TO
ADFPND ;BASE ADDRESS
;
6828 CA2F68
682B 19
682C C32768
682F 19
ONEMOR: JMP MULT

;ADFPND:

;FIRST CHECK IF A GRAPHICS CHARACTER IS ALREADY IN CHARACTER
;CELL. IF NOT, STORE A BLACK CHARACTER THERE.
;THEN CALCULATE BIT MASK

6830 7E MOV A,M ;LOAD CHARACTER
6831 E6C0 ANI CHRAND ;CHECK IF GRAPHICS CHARACTER
6833 F800 CPI CHRCPI
6835 CA3B68 JZ FNDBIT ;YES, NOW CONTINUE
6838 3E00 MVI A,BLKCHR ;NO, STORE AN ALL

32 MICROSYSTEMS
683A 77  MOV  M, A  ;BLACK CHARACTER
683B 3A8A68  FNDBIT: LDA DATA+1  ;LOAD VALUE OF X
683C E601  ANI 1  ;GET REMAINDER FROM
683D 3E01  JZ XRZER  ;DIVISION BY 2
683E C34A68  MVI A, 1  ;REMAINDER IS ZERO
683F 3E01  JMP CONT  ;BEGIN TO FORM BIT MASK IN A
6840 CA4868  CONT: MOV B, A  ;CONTINUE
6841 3E08  LDA YREM  ;SAVE PARTIALLY FORMED MASK
6842 3A8C68  ;LOAD REMAINDER FROM
6843 CA64  ;DIVISION OF Y
6844 A7  ;SET FLAGS
6845 CA6068  ANA A  ;NEEDS TO BE SHIFTED TWICE
6846 3C  JZ SHIFT2  ;DOES NOT NEED TO BE SHIFTED
6847 CA6468  INR A  ;MUST BE SHIFTED TWICE
6848 3C  JZ MSKFND  ;NEEDS TO BE SHIFTED TWICE
6849 C26068  INR A  ;DOES NOT NEED TO BE SHIFTED
684A A7  JNZ SHIFT2  ;MUST BE SHIFTED TWICE
684B 78  DATA+2  ;LOAD BIT MASK
684C 07  A, B  ;SHIFT LEFT ONCE
684D 47  MOV B, A  ;SAVE IN B
684E C36468  JMP MSKFND  ;MUST BE SHIFTED TWICE
684F 78  1  ;THE BIT MASK IS IN B AND THE CHARACTER ADDRESS IS IN HL.
6850 0707  A  ;NOW COMPUTE AND PUT PROPER VALUE ON THE SCREEN.
6851 47  MOV B, A  ;SHIFT LEFT TWICE
6852 3A8B68  LDA DATA+2  ;SAVE IN B
6853 A7  ANA A  ;THE BIT MASK IS IN B AND THE CHARACTER ADDRESS IS IN HL.
6854 CA6068  JZ SETBLK  ;SET CALLER WANTS DOT STATUS
6855 3D  DCR A  ;0 MEANS SET DOT BLACK
6856 CA8568  JZ SETWHT  ;1 MEANS SET DOT WHITE
6857 C26068  JNZ SHIFT2  ;THE FOLLOWING ROUTINES ALL LOAD THE CHARACTER CELL,
6858 78  DATA+2  ;MODIFY IT IN THE DESIRED WAY, SAVE IT, AND RETURN.
6859 0707  SETBLK: MOV A, M  ;SET DOT BLACK
685A 47  ANA B  ;LOAD CHARACTER CELL
685B 7E  IF BLK  ;AND WITH BIT MASK
685C A0  JNZ RETBLK  ;CONDITIONAL ASSEMBLY
685D 3A8B68  ENDIF  WHEN BLACK=1
685E A7  JNZ RETBLK  ;RETURN A 1 FOR "WHITE"
685F 7E  IF NOT BLK  ;RETURN A 0 FOR "BLACK"
6860 3A8B68  ENDIF
6861 CA7A68  MVI A, 1  ;RETURN A 0 FOR "BLACK"
6862 E01  STA DATA+2  ;THE FOLLOWING ROUTINES ALL LOAD THE CHARACTER CELL,
6863 328B68  RETBLK: STA DATA+2  ;MODIFY IT IN THE DESIRED WAY, SAVE IT, AND RETURN.
6864 C9  ;SET DOT BLACK
6865 RET  ;SETBLK:
6866 3E01  IF BLK  ;CONDITIONAL ASSEMBLY
6867 328B68  MOV A, M  WHEN BLACK=1
6868 C9  ORA B  ;RETURN A 1 FOR "WHITE"
6869 RET  MOV M, A  ;RETURN A 0 FOR "BLACK"
686A C9  RET

MICROSYS

MICROSYS 33
A “Daisy” Of A Terminal!!

ALPHA DATA SERVICES offers the small businessman and serious computer enthusiast a complete word processing terminal at an unheard of price!!

CHECK THESE FEATURES:
- Reliable Diablo Hytype-I print mechanism.
- Completely refurbished and tested.
- Unconditional 30 day warranty
- Functions as electronic typewriter in local mode.
- Standard RS-232 and 20 mil. current loop interface at 110, 150 or 300 Baud.

$1249.00 Prepaid orders crated free of charge
Shipped via motor freight

ALPHA DATA SERVICES
810 Daleview Place
Greensboro, North Carolina 27406
(919) 373-1726
An Ultra-Simple Index For Sequential Files

by Fred Gohlke

File indexing routines are generally complex. There are applications, however, where comprehensive file indexing is overkill. These are characterized by sequential structure, relatively static size, slow access time, and a programmer who'd like to access his data faster without programmed or purchased complexity.

The approach suggested here covers such cases. It isn't intended as a definitive solution to file indexing problems, but as an aid to finding data in certain types of files. It requires that the files be maintained in alphabetical order and that they be randomly accessible. It doesn't index an individual record directly, instead it finds a small group of records which contain the record sought. The fact that the files must be maintained alphabetically means that adding and deleting records is slow. So, it works best when such changes are infrequent, as in name and address, inventory, and similar files which are frequently read and/or updated.

The method is to maintain the number of entries in the file beginning with each letter of the alphabet, and to use these amounts to find the first entry in the file beginning with the first letter of the entry sought. Implementation will vary with the language and dialect used. The following description uses CBASIC statements and assumes:

1 - A randomly accessible alphabetized file called STOCK.FIL.
2 - A randomly accessible file called INDEX.FIL containing the number of entries in STOCK.FIL for each letter of the alphabet.
3 - The data on which STOCK.FIL is alphabetized are the leftmost characters of each record.
4 - The key for the record sought is in a variable called KEY.DATA.
5 - The length of the key is in a variable called KEY.SIZE.

The first step is to create a pointer into INDEX.FIL by isolating the first letter of the data sought and removing its ASCII bias:

```
KEY.LETTER$ = LEFT$(KEY.DATA$,1)
POINTER% = ASC(KEY.LETTER$) - 64
```

The first instruction isolates the first letter of the data sought, and the second creates POINTER equal to the KEY.LETTER's position in the alphabet, i.e. A=1, B=2, ... Z=26.

Next, POINTER is used to calculate the offset from the beginning of STOCK.FILE for the entries starting with KEY.LETTER:

```
FILE.KEY$ = ""
SIZE% = 0
FOR I% = 1 TO POINTER%
    FILE.KEY$ = FILE.KEY$ + " "
    READ #INDEX.FILE%,I%;SIZE%
NEXT I%
```

FILE.KEY and SIZE are cleared. A loop adds the SIZE for each letter to FILE.KEY until the SIZE of KEY.LETTER is read. When these instructions have executed, FILE.KEY is equal to the number of entries in STOCK.FIL which are alphabetically lower than KEY.LETTER.

Processing beyond this point is sequential:

```
FILE.KEY% = FILE.KEY% + 1
READ #STOCK.FILE%,FILE.KEY%;FILE.DATA$
IF KEY.DATA$ = LEFT$(FILE.DATA$,KEY.SIZE)
    THEN GO TO xxx
SIZE% = SIZE% - 1
IF SIZE% GT 0
    GO TO 100.
```

If there are no entries for this KEY.LETTER, handle that condition. Otherwise, read the file data sequentially until the entry sought is found or all entries are exhausted. If found, the routine exits with FILE.KEY pointing to the correct record. If updated, it can be rewritten using the same key.

Of course, each time a new record is added to STOCK.FIL the INDEX.FILE must be updated. That's easily done:

```
KEY.LETTER$ = LEFT$(KEY.DATA$,1)
POINTER% = ASC(KEY.LETTER$) - 64
READ #INDEX.FILE%,POINTER%;SIZE%
SIZE% = SIZE% + 1
PRINT #INDEX.FILE%,POINTER%;SIZE%
```

handles the job nicely. For deletions, reduce SIZE by 1.

Fred L. Golke, 1000 Blair Road, Carteret, NJ 07008.

MICROSYSTEMS
Patch a CP/M Diskette on a North Star System

by Tom Wiens

Every computer system manual at some point includes a pointed warning—always back up your diskettes! But after logging months of work without a fatal error, I became lax and not a little bit careless, until the inevitable happened: late one night I finished off an additional ten pages of writing on a word processor running under CP/M, and began to write to disk—ERROR: DISK FULL was the program’s response. All my attempts to salvage the situation only got me in more trouble. Finally, in desperation, I rebooted without ending the run or closing the file in the hope that some temporary files could still be found on the diskette and used to reconstruct the text. But the CP/M DIR command produced no evidence that any part of my creation was alive or well in any form.

Having found in the past that erased CP/M files could easily be salvaged (see The CP/M Connection, Part II, in Vol. 1/No. 5), and groaning at the prospect of many hours wasted recreating lost ideas, I turned immediately to North Star DOS and the North Star monitor to have a look at the CP/M directory. On my version (Lifeboat Associates’ single-density CP/M 1.4 for North Star diskettes), the latter is stored as 8 blocks beginning at disk address 30, which can be read into memory using the DOS RD command, and then examined and modified with the aid of the monitor. In this instance I found that the directory contained one “live” old version of the document I was working on (minus the ten pages added that evening) and one “erased” new version. Since CP/M erases by placing an E5H in the first byte of the file entry in the directory, I used the CP/M monitor to replace the E5 with a 00H, modified the filename to prevent confusion with the old version, and used the DOS WR command to write the directory back to diskette. Re-booting CP/M and running the word processor, I was further dismayed to find that the restored file included little of the lost text.

Not ready to give up, I returned to DOS and again examined the CP/M directory entries. I was particularly interested in the group numbers—since the word processor has crashed while trying to write to disk, perhaps it had not written the file entry back to disk. Perhaps some of the “unused” groups not included in the directory entries for the old or restored versions of the file contained parts of the new text.

To check out this possibility, I first had to be able to translate the group numbering system used by this version of CP/M into the corresponding North Star disk addresses, so that I could RD or WR particular groups at will. Since the groups were numbered 2-79 and I knew that my CP/M files began at North Star disk address 38 and could run through disk address 349, it was easy to deduce that if E were the group number, the corresponding North Star disk address would be (E-2)*4+38. With this knowledge, I was able to RD in the unused sectors and then examine the text with the monitor. A slow process to be sure, but within an hour I had found most of my missing text, determined the ordering of the groups (which was sequential with some jumps), and then “created” a new file entry into which I placed the series of groups. When I returned to CP/M and the word processor and tried to read this text, I found that it would only read a small piece of the reconstructed file. Returning to DOS and examining the end of this piece, I found a series of 1AH’s—Control-Z’s, the CP/M end-of-file mark. Zeroing these and rewriting the affected block to disk solved the problem. I went to bed reassured that the total loss of text had been reduced to one paragraph instead of ten pages!

Let a Program Do the Dirty Work

While the above procedure will work in a pinch, it is rather tedious—why not automate the whole process, taking advantage of what I now understand about CP/M diskette structure? This notion led to the North Star Basic program CP/MDP given in the listing. The program is written for the Lifeboat single-density version of CP/M 1.4, Release 5 single-density DOS and North Star Basic, and furthermore exploits the display control features of the IMSAI VIO-C memory-mapped video board. Obviously a bit of careful patching is required before the program will run correctly on other systems.

What does it do? First, it automatically provides a listing of a CP/M diskette directory, both “live” and “erased” entries, giving filenames (exactly as written on disk) and group numbers (translated into North Star disk addresses). Those files which are “live” are highlighted on the screen; if any groups of “erased” files have not been overwritten by “live” files, the first appearances of those groups are also highlighted to indicate that they may contain salvageable information (though not necessarily the information you might expect!). After this listing is displayed on the screen, the user may also request a hardcopy printout.
Following the above, the directory information is used to display a "diskette map," which simply lists the contents of the diskette in order of disk address, indicating also where empty sectors occur.

A menu is then displayed, which allows the user to choose to:

1. REPEAT SECTOR MAPPING
2. MODIFY DIRECTORY
3. READ A SECTOR (HEX)
4. READ A SECTOR (ALPHA)
5. JUMP TO CP/M
6. JUMP TO DOS

The first option allows the user to either list the directory of another CP/M diskette or relist the previous one, perhaps to confirm that changes have been made as desired.

The second option requests the number of the directory entry to be changed (as previously listed), then allows the user to restore or erase the file, change the filename, change the number of records and/or change the group of numbers. One can, for example, take an existing erased entry, restore it to life, put in a new name, number of records (128 bytes per record; 8 records per group; and 128 records per entry) and set of disk addresses, thereby creating an entirely new file from groups which have not been committed to another live file (as I did when salvaging my text as described above). The program does not currently allow you to modify the extent numbers, although that could easily be added—they lie in the directory byte immediately following the filename.

The third option requests a North Star disk address and reads in 4 blocks of hex data. The user may view this by moving the cursor forward or backward on the screen with keyboard left and right arrows (or some other control keys, if statements 1040 and 1050 are modified). Typing in any two hex characters (0-9,A-F) replaces the pair under the cursor. Typing ESCAPE writes the modified data back to diskette; typing RETURN returns one to the menu. The most frequent use of this option is in patching .COM files. It can be used, for example, to examine and patch your BIOS I-O routines, the location of which on diskette is usually described in the implementation manual for any version of CP/M.

The fourth option also requests a disk address, reads in 4 blocks of ASCII text and displays it all on the screen (with ? marks for non-printable characters). This option can be used to search for missing or garbed pieces of files. Unlike CP/M, it is not stopped by Control-Z's in the text, and so it can be used to pinpoint their locations. However, the third option must be used if the user wishes to modify the text (e.g., to remove garbage).

Options five and six provide jumps to a cold boot of CP/M, assuming that a CP/M diskette has been placed in drive 1, or to North Star DOS (the entry points used are given in the CALLs at statements 795-800).

**What's the Gimmick?**

North Star Basic, like CP/M, won't let you read or write from disk independent of the file structuring system. So how do we accomplish the above? The gimmick is to exploit North Star DOS' auto-start facility (see North Star System Software Manual, Rev. 2.1, pp. F-1 to F-2). To use CP/MDP, the DOS byte which initiates this

---

**Load TRS-80® software on your S-100 Z-80 or your money back!!!**

Of the 500,000 home computers in this country more than 200,000 are TRS-80®. Look through your magazines and you will see that there is more software available for the TRS-80® than all other computers combined. Here is what we offer:

1. An assembled hardware interface and software drive which will enable you to load data from TRS-80® cassette tapes into your S-100 memory.
2. Complete documentation telling you how to relocate the program at its correct address, find the entry point to the program, and link the program to your keyboard input and video output routines.
3. Includes examples of how we interfaced TRS-80® Level II basic and SARGON II® with our system.

**NOTE:** Knowledge of Z-80® Machine Code is required or FREE with purchase of Assembled and Tested Compurprism or Super Compurprism Unit.

**ONLY** $30.00

**PLUS** Expandoram (4MHz) MOD. KIT

**ONLY** $10.00

**PLUS** 16 A-D 8 D-A

This S-100 board has 16 channels of analog to digital input and 8 channels of digital to analog output. Enough for most burglar alarm or home energy monitoring systems!! It uses National Semiconductor's ADC0816 sixteen channel analog to digital converter, which is available from DIGI KEY and other mail order houses for about thirty dollars. The total cost of construction including the board and parts should not exceed a hundred dollars. All inputs and outputs are 5 volts. Dual or split power supplies are not required. There is a on board kluge area for construction of custom circuits.

**COMPURPRISM & SUPER COMPURPRISM COLOR GRAPHICS**

Compurprism is a color graphics interface for S-100 Systems, with 16K of on board dynamic memory. Refresh of the dynamic memory is accomplished on board compurprism. (super compurprism has 32K of on board dynamic memory) The resolution for compurprism is 144 horizontal by 192 vertical pixels. (super compurprism resolution is 288 horizontal by 192 vertical pixels). Each byte of memory controls only two pixels of the matrix. Four bits of memory are dedicated to the exclusive control of every single pixel. Therefore, every pixel may always be programmed in any one of sixteen colors or sixteen shades of grey, completely independent of all other pixels in the matrix. (Please compare this to any other color graphics interface in our price range.) From the upper left hand corner to the lower right hand corner of the matrix, the pixels are mapped to consecutive memory bytes. This greatly simplifies the programming of compurprism.

**COMPURPRISM SOFTWARE PACKAGE**

Includes for both compurprism and super compurprism, alpha numerics, TRS-80® graphics simulation, and point plot and line draw. The price of the software package is **$45.00**

**ONLY** $30.00

**OR** FREE with the purchase of an assembled and tested compurprism or super compurprism unit. The TRS-80® cassette interface described above is also **FREE** with the purchase of an assembled and tested compurprism or super compurprism unit.

**NOTE:** Although we are happy to sell compurprism as a bare board we strongly urge the novice or person who feels that they do not have a strong hardware background to purchase an assembled and tested unit. Compurprism Bare Board with documentation **$45.00**

**Board with documentation only** $45.00

**Kit - $300.00** Assembled and Tested - $350.00

**Super Compurprism Bare Board with Documentation** **$50.00**

**Kit - $350.00**, Assembled and Tested - $400.00

Add $3.00 to bare board price for hard to find I.C.'s / Add $20.00 to assembled and tested price for memory management port. / Add $20.00 to assembled and tested price for 16 level grey scale option.

**J.E.S. GRAPHICS**

Box 2752 Tulsa, Ok. 74101 (918) 742-7104

**TRS-80® is a trademark of TANDY CORPORATION**

**SARGON II® is a trademark of HAYDEN BOOK COMPANY (CHESS program written by DAN and KATHE SPACKLEN)**

**MICROSYSYEMS**

37
facility must be set to zero (it is address 2030H; a FILL statement may be added to the beginning of the program to initialize this byte). One must also locate the DOS command input buffer (its location is stored at 2031-2032H; in the single-density version 5.0, the buffer begins at address 10109 decimal). A number of statements in the program stuff commands into this buffer, and may need modification for different version of DOS (statements 50, 980, and 1235-1245). Finally, we need an entry point into DOS which, if called, initiates an auto-start. This entry point is an "LDA 2030H" statement which, in my DOS, is found at address 8379 decimal; it may be located in other versions using the monitor command, SM 2000-2CF 3A,30,20. Statements 70, 1090, and 1100 should be modified if the address is not the same. With these adjustments, when a read or write of the CP/M diskette is necessary, the program will stuff the appropriate command into the DOS buffer and call the auto-start entry point, which completes the command, leaving the user in DOS. To return to the program, it is necessary to type "BP 2004" (2004 in double-density versions) and "RUN n", where n is the desired CP/M entry point (all this is prompted by the program before it enters DOS.)

When the program reads blocks of text into memory, it stores them beginning at address 01. Since every RUN command re-initializes all variables, the program uses its own buffer at address 4091 decimal as temporary storage. Both these locations are arbitrary and can be changed if necessary.

As the program is written for my system, it exploits the power of the IMSAI VIO-C video board for an attractive display which highlights significant data (using character-by-character reverse video). The relevant video control codes, which will need to be replaced for other terminals/displays, are as follows:

CHRS(26) = protect/unprotect reverse video fields;
CHRS(22) = turn on/off individual character reverse video;
CHRS(16) = protect/unprotect reverse video fields;
CHRS(27)+ = "+CHR$(31+Y)+CHR$(31+X) positions the cursor at line Y and column X.

Also, for keyboards without positional keys (arrows), the ASCII numbers "9" and "8" in statements 1040 and 1050 will need to be replaced.

The formulas converting CP/M group numbers to North Star disk addresses are found in statements 235, 730, 925, and 935. For different versions of CP/M or DOS, these formulas may need to be modified. The first part of the article suggests how to deduce the required conversion formula.

In reading the CP/M directory, CP/MDP determines the number of file entries by examining the first byte of each potential entry to see if it is a 0 or E5H; if it is not, the program assumes that the last entry has been reached. Should the directory itself get garbaged, this assumption may be incorrect, and the program may fail to read portions of the directory. In that case, use option 5 to jump to DOS and any of the monitors other than MOOOO to examine the directory (DA 1.90 will do for the first block) and place a zero or E5H in the garbaged byte which is blocking the full directory read. After returning to DOS, use the WR command to write the directory back to diskette (8 blocks).

CP/MDP maps the diskette by setting up a vector X with 79 entries corresponding to groups 2-79 (the dimensions may need to be changed for other version of CP/M—see statements 140 and 720). If a group appears in "live" file, the entry number for that file is written as a positive number in X; if it appears in an "erased" file, it is written as a negative number. If the group appears in no directory entry, it is treated as "empty."

Saved Again

Ironically, I had written two-thirds of the text for this article when my word processor again crashed—this time because I had mistakenly directed the new text file to a crowded diskette. With CP/MDP available, no worry: I moved the injured diskette to disk 2, booted DOS on disk 1 and ran CP/MDP. The directory entry to the text file contained no group information, but I noted the disk addresses where there were blocks which were potentially "alive" but uncommitted to an unerased file. Using the fourth finding, I checked each of these, finding that they contained all but a small part of the lost text (of course, had the diskette been completely full, nothing could have been saved). Returning to option 2, I entered 8 addresses in the file directory entry, and computed and entered the number of records as 8x8 = 64. After having the directory written back to diskette, I moved the CP/M diskette back to drive 1, chose option 5 to boot CP/M, and was back in business.

*****************************************************************
# LISTING: CP/MDP  AUTHOR: T. WIENS
*****************************************************************

1REM THIS PROGRAM USES SOME VIDEO CONTROL CHAR$ WHICH WILL NEED MODIFICATION
2REM TION FOR OTHER SYSTEMS--CHECK ALL 1CHR$() STATEMENTS. DOS 5.0 SINGLE
3REM DENSITY AND CP/M 1.4 SINGLE DENSITY (LIFEBOAT ASSOC. VERSION) IS ASSUMED.
4REM FOR OTHER VERSIONS, CHECK CAREFULLY THE DOS LOCATIONS CALLED OR FILLED,
5REM AND THE LAYOUT OF CP/M DISKETTES. SOME KEY PARAMETERS:
6REM 10109+ -- DOS COMMAND ENTRY POINT, 8379 -- DOS AUTOSTART ENTRY POINT
7REM 2A04H -- BASIC CONTINUE ENTRY POINT, 8232 -- DOS REENTRY POINT
8REM 59648 -- COLD BOOT ENTRY, 4091+ -- BUFFER USER BY PROGRAM
9REM 22146 -- TERMINAL CHAR.P X Y, 30 -- SECTOR WHERE CP/M DIR. BEGINS
REM 38 -- FIRST SECTOR FOR CP/M FILES, 349 -- LAST SECTOR
REM 2 -- FIRST CP/M GROUP NO., 79 -- LAST CP/M GROUP NUMBER
LINE#1,84
REM TO DEFEAT AUTOMATIC CARRIAGE RETURNS
!CHR$(26),"CP/M DISKETTE PROCESSOR by Thomas B. Wiens"
! "This program provides a means of reading & modifying CP/M diskettes,"
! "including editing CP/M directories to recover erased files or destroy"
! "garbage (even to piece together new files from pieces scattered around"
! "the diskette), reading individual blocks of text or hex files, patching"
! "COM files, and screen or hardcopy mapping of diskette structure."
! " RAM starting at 01H is used as buffer space. References to sector"
! "numbers are based on NS sector format, converted back and forth from"
! "CP/M's group numbering scheme. For some commands, the program stuffs"
! "commands into DOS' input buffer and calls DOS to execute; to return"
! "to the program, follow the printed instructions EXACTLY."
! " BEWARE OF MODIFYING THE CP/M DIRECTORY UNTIL YOU KNOW WHAT YOU"
! " ARE DOING--PRACTICE ON A DUPLICATE DISKETTE."
! " NOTE: CP/M AND ITS PROGRAMS TAKES CONTROL-Z (1AH) AS AN END-OF-FILE"
! " MARK, AND WILL NOT READ PAST SAME, WHEREAS THIS PROGRAM WILL. IF "
! " DESIRED, YOU CAN USE THE ROUTINES HEREIN TO ELIMINATE THE CONTROL-Z'S."
! "PRESS ANY CHARACTER TO CONTINUE..."Y$=INCHAR$(O)
10 !CHR$(26),"INSERT YOUR CP/M DISKETTE IN DRIVE 2, THEN PRESS ANY KEY"
11 Y$=INCHAR$(O)
40 DATA 82,68,32,51,48,44,50,32,49,32,56,13
41 DIM Z(12) REM CONTAINS ASCII CODES FOR DOS RD COMMAND
50 FOR I=1 TO 12 READ Z(I)
51 FILL 108+I,Z(I)
60 !"NOW, TYPE: JP 2A04"
65 !" RUN 100"
70 CALL(8379) REM JUMP TO DOS AUTOSTART
100 !CHR$(26),"******* DIRECTORY LISTING FOLLOWS *******"
101 !"ACTIVE FILES ARE LISTED IN REVERSE VIDEO; ERASED FILES IN NORMAL VIDEO"
102 !"ERASED BUT PERHAPS SALVAGEABLE BLOCKS ARE ALSO IN REVERSE VIDEO!"
105 B=0 L1=32 L2=64 P1=0
106 ERRSET 4000,E1,E2
110 FOR I=1 TO L2*32 STEP 32
115 X=EXAM(I)
120 IF X=229=ERASED FILE EXTANT
122 X=EXAM(I+1)
125 NEXT I
130 L2=(I-1)/32
140 DIM D(L2),R(L2),NS(11*L2),M(L2,16),A$(11),X(79)
150 FOR I=1 TO L2
152 K=(I-1)*32+1
155 D(I)=EXAM(K) IF D(I)=229 THEND(I)=1
160 FOR J=K+1 TO K+31
162 L=L+1
164 X=EXAM(J)
165 IF X=229=ERASED FILE EXTANT
167 R(I)=EXAM(K+15)
169 NEXT J
170 NEXT I
175 GOSUB 400
200 "NO. FILENAME DISK SECTORS"
210 FOR I=1 TO L2
215 IF D(I)=0 THEN !CHR$(22), REM REVERSE VIDEO IF "LIVE" ENTRY
220 IF P1=1 AND D(I)=1 THEN !CHR$(91), REM BRACKET DEAD FILES
225 A$=NS((I-1)*11+1),(I-1)*11+111
227 !P1,%2I,1,$#3I," ",A$," ",
230 FOR I=1 TO 2O IF M(I,J)<>0 THEN !CHR$(22),
235 X=(ABS(M(I,J))DIV 2)*4+38\!P1,X,"
237 IFM(I,J)<0 THEN !CHR$(22),
238 NEXT J
240 IF P1=1 AND D(I)=1 THEN !CHR$(93), REM RIGHT BRACKET
245 IFD(I)=0 THEN !CHR$(22), !#P1 REM TURN OFF REV VIDEO
Patching cont’d...

242 IF I=20 OR I=40 THEN GOSUB 280

245 NEXTI

247 "PRESS ANY KEY FOR MEMORYMAP:"

249 Y$=INCHAR$(0)

250 GOTO 700

250 "PRESS ANY KEY FOR NEXT DISPLAY; 'P' FOR HARD COPY:"

290 Y$=INCHAR$(0)

295 IF Y$="P" THEN 305

300 P1=0

305 P1=1

310 IF I="NO. FILENAME DISK SECTORS"

320 RETURN

340 FOR I=1 TO L2

345 IF D(I)<>0 THEN 350

350 FOR J=1 TO 16

355 X(M(I,J))=I

360 NEXT J

370 NEXT I

380 FOR I=1 TO L2

390 IF D(I)<>0 THEN 400

400 FOR J=1 TO 16

405 X(M(I,J))=-I

410 M(I,J)=-M(I,J)

420 END IF

430 NEXT J

440 NEXT I

450 RETURN

460 NEXT I

470 IF D(I)<>0 THEN NEXT500

480 FOR J=1 TO 16

485 X(M(I,J))=-I

490 END IF

500 NEXT I

510 RETURN

520 REM MEMORY MAP

530 REM USES PROTECTION FOR INVERSE VIDEO FIELDS TO FORMAT OUTPUT

540 CHR$(26),CHR$(16),CHR$(22),CHR$(13),CHR$(26)

550 "********** DISK MAP (BY NS SECTOR) **********"

560 A$=" 

570 FOR I=2 TO 79

575 IF X(I)=X(I-1) THEN 750

580 IF X(I)<>0 THEN 570

585 A$=ABS(X(I))-1,ABS(X(I))-1

590 (I-2)*4+38,"-",A$," 

595 NEXT I

600 CHR$(22),CHR$(16)

605 "PRESS ANY KEY TO CONTINUE:"

610 Y$=INCHAR$(0)

615 "CHOOSE ANY OF THE FOLLOWING:"

620 "1 -- REPEAT SECTOR MAPPING"

625 "2 -- MODIFY DIRECTORY"

630 "3 -- READ A SECTOR (HEX)"

635 "4 -- READ A SECTOR (ALPHA)"

640 "5 -- JUMP TO CP/M"

645 "6 -- JUMP TO DOS"

650 "YOUR CHOICE?"

655 X=VAL(Y$)

660 IF X=1 THEN 40

665 IF X=6 THEN 850

670 IF X=5 THEN 700

675 IF X=2 THEN 1000

680 IF X=3 THEN 1100

685 IF X=4 THEN 1200

690 GOTO 775

695 X=VAL(Y$)

700 IF X<>6 THEN CALL(8232)

705 CALL(6)"FILE GROUPS (4 SECTORS EACH):"

710 FOR J=1 TO 16

715 X=(ABS(M(N1,J))-2)*4+38

720 IF X<>30 THEN 745

725 IF X<>80 AND X>1 THEN 745

730 FILL K+15,0, \M(N1,J)=X

735 NEXT J

740 "WRITE DIRECT. TO DISK (Y/N) OR CR TO REVIEW ENTRY:""Y$=INCHAR$(0)

745 IF Y$="Y" THEN 975

750 IF Y$="N" THEN 775

755 ELSE 855
970 DATA 87,82,32,51,48,44,50,32,49,32,56,13
975 RESTORE 970
980 FOR I=1 TO 7: READ Z(I): FILL 10108+I, Z(I): NEXT
990 GOTO 60
1000 GOSUB 1200: "RUN 1010": Z=CALL(8379)
1010 "CHR$(26), "USE LEFT OR RIGHT ARROWS TO VIEW OR POSITION CURSOR"
1015 "TYPE TWO HEX CHAR TO REPLACE THOSE UNDER CURSOR; TYPE AN "
1020 "ESCAPE TO WRITE MODIFIED FILE BACK TO DISK; CR TO RETURN TO"
1025 "MENU." "CHR$(27)+"=""CHR$(Y)+CHR$(X)", "REM PLACE CURSOR ON SCREEN
1030 Y$(1,1)=INCHAR$(0): X=EXAM(I): FILL 22146: 0X2=ASC(Y$)
1040 IF X2<>9 THEN 1050 ELSE I1=I+1; X1=X+1: GOTO 1030
1045 IF X2<>8 THEN 1060 ELSE I1=I-1; X1=X-1: GOTO 1030
1050 IF Y1>36 THEN 1060 ELSE Y1=Y1-1; GOTO 1030
1055 IF Y1>36 THEN 1060 ELSE Y1=Y1+1; GOTO 1030
1060 IF X2<>13 THEN 1065 THEN 1065 ELSE I1=I-1; X1=X1+1: GOTO 1030
1065 IF X2<>13 THEN 1065 ELSE IF X2<48 THEN 1035
1070 Y$(2,2)=INCHAR$(0): X2=FNH$(Y$): IF X2>=10 THEN FILL I1, X2
1075 "CHR$(27)+"=""CHR$(Y$), I1=I+1; X1=X1+3: GOTO 1045
1080 RESTORE 1077: "CHR$(22)
1082 B$=""X=EXAM(090): FOR I=1 TO X: B$=B$+CHR$(EXAM(090+I))
1084 "WRITING SECTOR ": B$
1085 FOR I=1 TO 10: READ Z(I)
1090 GOSUB 1235: "RUN 775": Z=CALL(8379)
1100 GOSUB 1200: "RUN 1110": Z=CALL(8379)
1110 "CHR$(26), "SECTOR READS:"
1120 FOR I=1 TO 10: STR$(X)
1130 IF X>=32 AND X<128 OR X>9 AND X<14 THEN 1060 ELSE "?"
1140 GOTO 770: REM NON-ALPHA CHAR PRINT AS '?'
1150 INPUT "SECTOR NO. FOR 4-SECTOR READ? ", X
1160 DATA 82,68,32,44,50,32,49,32,52,13
1170 RESTORE 1210
1180 FOR I=1 TO 10: B$=STR$(X)
1190 D2=INT(D/(16^I))
1200 IF D2>=10 THEN H1$=CHR$(ASC("A")+D2-10)
1210 IF D2<10 THEN H1$=CHR$(ASC("0")+D2)
1220 D=D-(D2*(16^I))
1230 NEXT
1240 IF D=0 THEN 1010
1250 IF N=0 THEN "NOW, TYPE: JP 2A04"
1260 RETURN
1260 IF D=0 THEN "NOW, TYPE: JP 2A04"
1270 RETURN
1280 FN END
1290 DEF FNH$(D)
1300 H1$=""
1310 FOR I=1 TO 10: STR$(X)
1320 IF I=10 THEN H1$=CHR$(ASC("A")+D2-10)
1330 IF I=10 THEN H1$=CHR$(ASC("0")+D2)
1340 D=D-(D2*(16^I))
1350 NEXT
1360 IF E1=100 THEN 150: REM TO PREVENT REDIMENSIONING ERROR
Selecting Functions Via Tables
In 8080 Assembly Language

by Fred Gohlke

It is often necessary to write a routine which will perform one of several functions, depending on a code character. The beginning programmer may first notice this need when providing for selections from the keyboard. For this reason, these routines are frequently written to take the first character of a mnemonic command:

'E' from Execute
'P' from Print
'T' from Tape

and use it as the code character. Execution of the routine selected is usually accomplished by putting the code character in the 'A' register and proceeding something like this:

```
cpi Code-for-execute
jz Execute-routine

Code-for-print

Code-for-Tape
jz Tape-Routine
```

While effective for selecting from a few alternatives, this style becomes cumbersome and slow as the system grows in size: The selection of suitable mnemonics becomes increasingly difficult; there is no logical relationship between the code characters and the routines they invoke (in the sense that the 4th code invokes the 4th routine, rather than the letter indicating the routine name); each new set of codes and routines requires a new set of decodings and invocations; and, it fails to recognize that selecting between alternative routines is a central part of the programmer's task which should be handled systematically.

There is another approach to this problem which allows great flexibility and lends itself to modular program design. The first step is to design a module which can be used from anywhere in the system to distinguish between alternative functions. The second step is to set up the means of invoking this module when necessary.

The module will require 2 parameters: the address of a table of function addresses, and a function pointer. The SELECTOR routine which follows expects the table address in the 'HL' register pair and the function pointer in the 'A' register:

```
SELECTOR
add A double the function pointer
add L add low table address byte
mov LA restore to 'L'
jnc load no carry, no overflow
inh h overflow, increment 'g'
load mov A,m pick up low function addr
mov H,A point to high function addr
mov H,N set high function address
mov L,A set low function address
pchl
```

Note that the routine ends by jumping directly to the selected function. This requires that SELECTOR be CALLed, and that each function end with a RETurn instruction, which tends to maintain modularity. Also note that a function pointer of zero points to the first function. The routine simply uses the function pointer as an offset from the beginning of the table of routine addresses. Finally, note that the 'BC' and 'DE' register pairs are available for passing parameters to functions.

With the SELECTOR routine as a base, all that's left is to provide for its invocation by table construction and code selection. Table construction is neatly handled by assemblers. Assuming the appropriate routines exist somewhere in the system:

```
TABLE1 DW RTN1 Routine invoked by zero
                              DW RTN2 Routine invoked by 1
                              DW RTN3 Routine invoked by 2

TABLE2 DW XCHRA Routine invoked by zero
                              DW XCHRB Routine invoked by 1
                              DW XCHRC Routine invoked by 2
```

will create the necessary tables.

Code selection can be handled several ways. One way is direct conversion of the code character to a function pointer. This can be illustrated by selecting a sequence of ASCII characters, such as 'J' through 'V'. If the code character is between 'J' and 'V', and the ASCII value for 'J' (4A hex) is subtracted from it, the result will be between zero and 12 (decimal). Thus, the letter 'J' causes execution of the first routine, 'K' the...
with the code character in the 'A' register:

```assembly
CODES1 DB 'E' Code to be converted to zero
DB 'P' Code to be converted to 1
DB 'G' Code to be converted to 2
DB 'D' Code to be converted to 3
      DB 0 Table terminator
```

performs the conversion and validates the result. This leaves the offset in the 'A' register, so SELECTOR can be called as soon as the 'HL' register pair is set to the appropriate table.

If a code sequence isn't practical, the assembler will create a code table:

```assembly
D64A SUI FIRSTCH
F25D CPI LIMIT + 1
D2nnn JNC Error Routine
```

In this case, with the code character in the 'A' register, and the 'DE' register pair holding the code table address, the following instructions will set up the call to SELECTOR:

```assembly
CHECK MOV B,A Save for comparison
XRA A Clear 'A' register
MOV C,A Clear 'C' register
CHECK1 LDX D Pick up code letter
ORA A Check for terminator
STC Use carry as error flag
RS Error return, no match
CMP B Compare code character
MOV A,C Prepare for SELECTOR
RS Valid, offset in 'A'
INR C Increment the offset
INX D Increment code pointer
JMP CHECK1
```

Variations of this routine will accommodate multi-character codes for tasks. They must simply insure an unambiguous error flag for invalid codes, and a zero offset for the first routine.

The procedures recommended here seem more complex than a simple CPI followed by JZ. But don't be put off by the verbosity of this presentation. What it boils down to is that, after getting the code character into the 'A' register, either:

```assembly
SUI FIRSTCH
CPI LIMIT + 1
JNC Invalid-code-routine
LXI H,TABLEn CALL SELECTOR
```

or:

```assembly
LXI D,CODELISTn CALL CHECK JC Invalid-code-routine
LXI H,TABLEn CALL SELECTOR
```

handles all the processing. As confidence in this method grows, the selection of codes and creation of tables becomes almost trivial.
New Life For
An Old Text Editor

by Bill Machrone

How to use ZTEL (Z-80 Text Editing Language) for letter and address merging.
Also, how ZTEL can be used with other text editors such as "Word Star".

Since I fired up my first microcomputer in 1977, the majority of my time has been spent editing text of one kind of another. That first system was a TDL Alpha box with Z-80, 16K, the VDB terminal board and 1200 baud cassette. It came with the TDL software package, which included the excellent macro assembler, Basic, the Text Output Processor and a complicated text editor called ZTEL.

ZTEL supports a potentially bewildering array of macro instructions, conditional branches, iterations and options. I say complicated because ZTEL (Z-80 Text Editing Language) supports a potentially bewildering array of macro instructions, conditional branches, iterations and options in addition to the normal text editing functions. These additional features permit you to actually write programs that manipulate text, prompt the console and perform logical operations. I eventually mastered it and came to regard it as a close friend even in the slow tape environment. As time passed I added more memory, then disk. Desiring the maximum software compatibility, I selected the TDL disk utilities package to supplement those that came with CP/M. The package included a new version of ZTEL, with some useful additions for the CP/M environment.

ZTEL is an editor that encourages proficiency on the part of the user. It is neither simple nor 'friendly.' While its basic command structure resembles Digital Research's ED, it was modelled after DEC's TECO, long considered to be one of the best 'programmer's' text editors. In fact, ZTEL supports most of all of the useful features of TECO, including text registers, value registers and macro capability. It uses virtually all the same commands, so that an experienced TECO user could move right over to ZTEL and be comfortable with it. ZTEL's syntax consists primarily of one-or two-letter commands that are fairly mnemonic in terms of what they do. For instance, 'i' puts the editor in insert mode, 's' searches for a text string, 'd' deletes a character and 'k' kills a line of text. Most of the commands accept a numeric parameter, so that '5d' deletes 5 characters. This is fairly standard nomenclature for a text editor of this type, but where ZTEL goes beyond is in the ability to modify some of the parameters in more complex ways and to store whole strings of commands in one of ten macro registers which can be invoked with just two keystrokes, such as 'm5'.

The contents of macro 5 could possibly tell the editor to move the pointer back 40 lines, then display 20 lines. Another macro might move the pointer ahead 20 lines, then display the next 20 lines. Thus, you can implement your own "page forward" and "page back" commands. As I mentioned above, testing and conditional branching is supported. This means that you can write macros that do things like find all occurrences of the word "boat" and change them to "ship." With a little ingenuity, you can construct the macro so that it automatically capitalizes...
An experienced TECO user could move right over to ZTEL and be comfortable with it.

The biggest problem with ZTEL, as with Digital Research's ED, is that it is oriented towards a hard copy terminal such as a teletype instead of a CRT. Thus you have to end every command with an additional command that displays a few lines and the pointer, so you can see the results of your editing. This can be built into a macro, but the pointer and the cursor are in two different places on the screen. This makes text insertion somewhat difficult.

So, when I saw a demonstration of WordStar, I couldn't live without it. WordStar seemed to overcome all of the operating disadvantages of ZTEL, primarily through its full screen edit, real-time justification and margination and concurrent printing. Of course, it takes three times as much memory as ZTEL, but everything has its price.

As I began to use WordStar more, I realized that it, too, was less than perfect. I missed being able to construct macros that would move large pieces of text around dynamically. WordStar is limited to one marked block of text that can be moved, deleted, copied or written out. ZTEL offers up to ten text registers which can hold a macro, a single character or an entire file's worth of text. When you delete a string with ZTEL, you often delete to a text register so that you can get the deleted text back if you change your mind. It also sports ten value registers which may hold the decimal equivalents of ASCII characters or the results of calculations. ZTEL performs the four arithmetic functions, so that calculation is possible. It is my understanding that other text editors, such as Magic Wand and Wordmaster, have some of the features that ZTEL offers, but none of them seem to do everything you want.

As time went on I found myself using WordStar more and more, using ZTEL infrequently, perhaps for an intensive search-and-replace job such as converting a Basic program to run on another interpreter. ZTEL is tremendously handy here because it can be programmed to create line numbers if desired, so that a Basic-E program, which does not require a number on every program line, can be easily converted to run with an interpreter that requires that every line be numbered (as most do).

As a hobbyist who was able to turn microcomputing into a business, I have maintained a vigil on most new hardware and software releases so that I could ride the crest of constantly improving technology and offer customers the best product at a good price with minimum effort. Faced with a need to do a letter and address merge program, I eagerly awaited the release of the mail/merge version of WordStar. As with most new software products, its release date came and went and we had an eager customer who wanted the mail/merge function as soon as possible. Of course, all the then commercially available mailing list packages were incompatible with their needs. We had to choose between writing an assembly language program to do the job, with its attendant long debug and development times, or doing it in compiled Microsoft Basic. Or did we? Out came the old ZTEL manual, and a quick perusal confirmed my suspicion: ZTEL could do the job.

Our requirements were straightforward: Read a fixed-length random access file with names, addresses and other data, compress extra spaces out of the city, state and zip fields, merge the output with a WordStar-generated one-page letter.

The CP/M version of ZTEL reads a file called "INIT.TEL" from the logged in disk, loads it into text register zero and executes it. This is normally used to initialize value registers that set the width of the screen, TTY or CRT mode or to automatically load your favorite macros into the appropriate registers. The need was for the mailing list merge to begin without intervention from the (non-programmer) user, who definitely would not understand the complexities of ZTEL.

By using the CP/M Submit function, we could dynamically rename a macro specifically constructed to merge addresses or type envelopes as INIT.TEL. Further, the user could use the familiar WordStar to create the letter that would receive the addresses. Better still, the macros would prompt the user so that he would know when to insert a new envelope or sheet or letterhead into the printer.

Out came the old ZTEL manual, and a quick perusal confirmed my suspicion: ZTEL could do the job

The system is described below, with appropriate listings for those of you who would like to implement a similar function on your own system. The program that extracts the addresses from the mailing list master file outputs the addresses as three or four lines terminated by carriage return/line feed sequences. Since part of the objective in a high-quality direct mailing campaign is to eliminate blank lines and fixed-length fields, the first character or of the address is the number '3' or '4', which fixed to variable length conversion is done at extract time. The output file, EXTRACT.ADR, looks like this:

```
3ABC Corporation
2150 Topanga Canyon Road
Torrance, CA 94503

4Tweedle Instruments, Inc.
Building 5
345 Maple Street
White Plains, NY 10607

3Underwood Manufacturing Co.
Merritt Industrial Park
Union, NJ 07083
```
The application is set up on a two drive system so that the file EXTRACT.ADR and the letter for merging, which can be any name, are on drive B. Drive A is the logged-in disk and contains the following files:

- MRGMACRO.TEL
- ENVMACRO.TEL
- ZTEL.COM
- SUBMIT.COM
- MERGE.SUB
- ENVELOPE.SUB
- MARGIN.COM

Below is the listing for MERGE.SUB, the submit file that controls the merging of addresses with a letter specified by the user. It is invoked by typing,

```
SUBMIT MERGE FILENAME.TYP nn
```

where the filename.typ is the name and type of the letter file created with WordStar on drive B:

- margin $2
- ren init.tel=mrgmacro.tel
- ztel
- ren b:$1=b:letter.mas
- ren mrgmacro.tel=init.tel
- margin 0

A brief narrative of what happens is as follows: A short program, MARGIN.COM, is invoked. This assembly language program takes an ASCII numeric parameter from the CP/M default buffer and converts it into an escape sequence which positions the daisywheel printhead at the specified column and sets the margin there, too. SUBMIT then renames MRGMACRO.TEL to INIT.TEL, so that ZTEL will find, load and execute it when it is invoked. The letter specified by the user is renamed "LETTER.MAS" for the duration of the job so that ZTEL can find it easily. It then performs the merge, quits upon detecting an out-of-date condition and allows SUBMIT to restore everything back the way it was. Moving the margin is necessary because ZTEL is printing a raw WordStar file, which has not been indented or otherwise manipulated by the WS print module.

The file below, ENVELOPE.SUB, does essentially the same thing, except that the user does not specify a letter file:

```
SUBMIT ENVELOPE nn
```

For a 10-pitch typewriter, the user normally enters an indentation of 30 or 35. The ability to input this parameter is necessary so that the user can align 12-pitch addresses properly on the envelope.

- margin $1
- ren init.tel=envmacro.tel
- ztel
- ren envmacro.tel=init.tel
- margin 0

Below is a listing of MRGMACRO.TEL. If you are not familiar with ZTEL it might be a little confusing, because the expressions are so terse. I’ve done some interpreting to aid in comprehension.

```
@!/MRGMACRO.TEL/
80wl
124wy
@er/b:letter.mas/
0a
hipl
@b/READ/
@1/MESSG/84/
?? Type RETURN, "R" or "C". /
@0/KEY/
@1/READ/
@er/b:extract.adr/
@1/LOOP/8x/
Align paper in printer and type "RETURN"/
@1/KEY/6k
81/Get char from kypd./
v4-1*e0/DONE/81/C gets you out./
v4-10*e0/REPT/81/R repeats letter./
v4-13*e0/EVAL/81/CR merges next addr. /
@0/MESS/
@1/EVAL/
6k
81/Clean slate./
6a
81/Get a line, but/
ve-1*e0/DONE/81/Branch if EOF./
req1
81/Put first char. in vr 1/
vl-51*e0/W0/81/If vr 3 = 2 lines, *4 = 3 lines./
vl-52*e0/W0/81/If vr 4 = 2 lines, *5 = 3 lines./
@1/TWO/
2a
6b/MERGE/
@1/THREE/
3a
@1/MERGE/5hpi2
81/Put address in tr 2/
5g
81/Get the letter back/
5j
81/And get to top. /
5q/ADR/
81/Find the place. /
5k
81/Kill the line. /
5g
81/Get the address. /
5f
81/Go to the bottom. /
5e
81/Insert a form feed. /
@1/REPT/
1wa
81/Interprets form feed. /
5b
81/List it out. /
5a
81/Put off A flag. /
80/LOOP/
81/Next address. /
@1/ERROR/6X/
81/If not 3 or 4...
```

Here’s a synopsis of what’s happening in the above program: After some initialization, ZTEL reads a file called LETTER.MAS and appends its contents. The letter is then deleted from the console and ZTEL accepts a character of input, which is stored in value register 4. The contents of that register are evaluated for a return, which causes the merge process to begin, or either of two control characters which force and end of job or cause the letter to be reprinted. Input of other than the three recognized characters causes the error message at label MESS to be displayed. ZTEL then reads the EXTRACT.ADR file and appends one line. Before doing anything else, it checks to see if the append operation set an end-of-file flag. If it did, the macro branches to the label “DONE” and tells ZTEL to quit. If not, it extracts the first character from the file and puts it into a value register where it can be evaluated. Similar to an assembler CPI instruction, ZTEL subtracts the decimal value of the character ‘3’ or ‘4’ from the character in the value register. If the result is zero, ZTEL branches to the appropriate label. If neither value is found, there is an error condition and a branch is made to the error message routine, which then falls through to "DONE".
At this point the editor has either three or four lines of an address in the file buffer. Beginning at the "MERGE" label, the editor places the address in text register 2 and gets a copy of the letter back into the file buffer from text register 1. It then searches for the string "ADR", which is the location at which to replace the string with the address. It does this by killing the line that contains the "ADR" string and getting the address from text register 2. It then inserts a formfeed at the bottom of the text so that the printer will eject the letter when it is done.

The merged letter is actually typed by using ZTEL's 'el' (extended list) function which can send a copy of the file buffer or any portion thereof to the list device. The macro then branches back to "LOOP" to begin again. Although not illustrated in the listing above, any branch to the end of a macro takes the user back to the command mode of the editor. This is not particularly useful for the mailing list program, since the user has no idea how to operate ZTEL, but is normal procedure for most ZTEL macros.

```
@!/ENVMACRO.TEL/
80V1
124wy
hk
13ei 10ei @!/Insert CR LF and/
hrp2 @!/put it in tr 2. /
@er/b:extract adr/
@!/ENVMACRO.TEL/
@!/REPT/
?? Type RETURN, "N" or "C".
@b/KEY/ ek4 @!/Get echar from kybd./
@b/PRINT/ 92 @!/Print from reg./
@b/STOP/ 85 @!/End this program./
```

The ENVELOPE macro is very similar to the MERGE macro, except that it does not read the letter file. As part of the initialization, the macro places a carriage return and a line feed in text register 2. Once it inserts a formfeed after the address, it jumps back to the top of the file and uses ZTEL's iteration feature to get the CRLF sequence ten times. Thus, the envelope is spaced up ten lines before the address is printed, an alignment convenience for the user.

While ZTEL provides an interesting array of conditional branch tests that are well suited to text processing (only the branch if zero and branch if alphabetic tests are used here), it does not support subroutine constructs. Therefore, you definitely won't be able to write structured-code macros! Be that as it may, the old fall-through constructs have served programmers well for a couple of decades now, and they still aren't obsolete. You might even consider ZTEL to be a sort of APL for text editing. The expressions are economical, powerful and all but incomprehensible to the uninitiated.

The above examples, while indicative of ZTEL's capabilities, do not really tell the whole story. I'm sure there are TECO and ZTEL macros out there that would make these look paltry by comparison. You may also be wondering why I'm bothering to praise an editor offered by a company that is out of business. Well, the software lives on. Computer Design Labs, (342 Columbus Ave., Trenton, NJ 08629) has the rights to ZTEL and is still selling it. And the price, the last time I checked, was only $79.95, which is quite reasonable for what it can do.

In conclusion, ZTEL is not everybody's ideal text editor. This example, though, shows that you can accomplish a lot without spending hundreds of dollars for a mailing list management system without tedious programming. It is a unique utility for unique applications. Maybe it belongs in your personal arsenal of systems software.
An 88-S4K Design Update

by W. Howard Adams

As a matter of economy, the least expensive memory available is the memory already owned. As the S-100 bus has evolved, some older pioneering equipment has lost its ability to function due to increasing bus expectations. Such is the 88-S4K, a dynamic memory manufactured by MITS for their Altair system. In its original design, this card will not self-refresh during multiple WAIT states such as North Star disk systems utilize during disk reads and writes.

This update can render some memory useful again, providing the 88-S4Ks began dropping bits ironically after a new peripheral was added to the system. Be advised, however, to judge personal technical ability by reading this article completely. While dynamic memory is not the evil thing static ram board manufacturers would have believed, they are more difficult to understand and troubleshoot.

Before correcting the design for multiple WAIT state refresh interference problems, the REV level must be assured. An addendum written by MITS and dated November 1976 relates a factory directive modifying how the T4 state, or the so-called ‘transparent refresh’ state is detected. (Yes, it was pioneering MITS who probably found the ‘transparent refresh’ capability that the modern dynamic memory boards brag about!) The original 88-S4K simply counted the free-running Phase 1 clock pulses after PSYNC to find T4; the MITS’ modification calls for using the trailing edge of PDBIN to clock a flip flop, thereby providing an error free way of finding T4.

Table 1 relates the necessary MITS modification. Whether the multiple WAIT state modification to be discussed later is used or not, the MITS modification should be incorporated! There is at least one S-100 processor card on the market which allows WAIT state selections to be made on the processor board. Without the MITS mod, the WAIT state pulled by the processor card while an 88-S4K is selected will do a very nice job of completely ignoring the refresh cycle. From all indications the board will seem to have refreshed. Upon investigating the 4060 memory element it is readily seen that addresses are latched into the memory chip on the rising leading edge of chip enable. Assume the 88-S4K is selected, so the appropriate address bus signals are latched into the 4060s with chip enable. The processor card then pulls its own WAIT, but the memory card does not have the MITS mod.

When the old method of detecting T4 is used, the refresh addresses are presented to the 4060s, a refresh cycle is pulled and then cleared. Refresh is not seen by the 4060s because its chip enable is still asserted from the memory cycle still in progress; the 4060 never saw the refresh addresses. The MITS mod allows for the WAIT state (the instruction fetch cycle) and with the trailing edge of PDBIN allows a reassertion of chip enable which latches up the refresh addresses. A bona fide refresh cycle can therefore take place at a very safe place (i.e., after an instruction fetch, but only if IC X-3 has just previously gone high).

<table>
<thead>
<tr>
<th>TRACE RUN</th>
<th>CUT AT</th>
<th>SIDE OF BOARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-3 to N-10</td>
<td>N-3</td>
<td>Component Side</td>
</tr>
<tr>
<td>V-10 to L-10</td>
<td>V-10</td>
<td>Component Side</td>
</tr>
<tr>
<td>V-8 to M-1</td>
<td>V-8</td>
<td>Component Side</td>
</tr>
<tr>
<td>V-13 to N-8</td>
<td>V-13</td>
<td>Component Side</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADD JUMPERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-10 to V-13 to V-14</td>
</tr>
<tr>
<td>K-3 to G-11</td>
</tr>
<tr>
<td>N-8 to G-12</td>
</tr>
<tr>
<td>V-8 to N-10</td>
</tr>
<tr>
<td>G-8 to M-1</td>
</tr>
<tr>
<td>G-13 to F-8</td>
</tr>
</tbody>
</table>

Simply ohm out the jumper list in Table 1 on every 88-S4K. If the jumper list ohms out, the MITS modification is already in the board’s artwork, and the board is of a later REV level. No further consideration of Table 1 need be made.

Should the jumper list not ohm out, then the cuts in Table 1 should be made, then the jumper list must be
incorporated. Please read the modification considerations in Table 3 before attempting to do any work.

After the MITS modification is made, the boards should be checked for refresh retention. With a system monitor, write the entire board with 0's. Do not access the board for several minutes. Now read the memory and assure that no bits have toggled to a logic 1 state. Repeat this process by writing all 1's, waiting a few minutes, and reading the memory assuring no bit has toggled to a zero. This check must be made of each board modified.

If a problem does occur, check the rework carefully—but go no further into the next modification until the MITS modification is working. The problem will only become more complicated. Please refer to Figure 1 for the necessary documentation to update a schematic for the MITS mod.

Figure 1: The MITS modification changes necessary to update the 88-S4K schematic.

With the MITS' mod now functioning, it's evident that the 88-S4K is a fine memory card, except that the stricter bus requirements imposed upon it reveals two major design flaws.

First, the refresh address counter is incremented by the free running Phase 2 system clock. Assume a refresh cycle is pending, but the processor is undergoing multiple WAIT states as the refresh address counter is being constantly incremented. The refresh is finally granted, but is pulled not at the address needing the refresh, but at an address higher up in the domain. (For information purposes, consider that modern design criteria would have the refresh address counter incremented as a result of a refresh cycle being pulled, and never by a free-running system clock!)

The second flaw is that PWAIT, the processor signal indicating a WAIT state is being pulled, is not a part of the 88-S4K's refresh arbitration scheme. The modification to be described now corrects both these flaws.

The modification criteria includes these five very important attributes necessary for reliable operation:
1. The MITS mod must have been incorporated. This mod frees up a gate necessary for this WAIT mod's use.
2. The memory card should be responsive to all signals designed into it by MITS. Namely, RUN will still be active, and the 88-S4K will still operate in an Altair system.
3. The board should not refresh during PWAIT if the board is currently selected. As pointed out earlier, there is at least one Z80 processor card which has wait state selectability on board. This criteria point is necessary. If the board were allowed to refresh during PWAIT, and the 88-S4K were selected, the memory elements would not respond. The same discussion for the MITS mod earlier is also appropriate here.
4. Only those ICs already on board will be used. No "dead bugs" or "piggy backing" is necessary nor is it desirable.
5. The modification should allow refreshing during PWAITs at a 32 micro-second rate, as per 4060 memory element specifications, and as per the existing design of the 88-S4K. (As a point of interest, dynamic memory is quite desirable because of its high density and low power. These elements should not be allowed to refresh in an
Design Update cont’d...

S-100 bus every T4 cycle because access rate determines power dissipation. Refresh should occur only during T4 AND at the memory element’s specified refresh rate. There is only ‘modern’ S-100 64K dynamic now marketed which refreshes every T4; from a power dissipation point-of-view, this short cut refresh scheme is not as modern as the ‘old’ 88-S4K!

Table 2: Multiple WAIT refresh modification.

<table>
<thead>
<tr>
<th>CUTS:</th>
<th>(to be made at IC pin noted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-12 to T-10 to GND</td>
<td>Component Side, under IC T</td>
</tr>
<tr>
<td>N-1</td>
<td>Component Side</td>
</tr>
<tr>
<td>N-2</td>
<td>Component Side</td>
</tr>
<tr>
<td>M-2</td>
<td>Solder Side</td>
</tr>
</tbody>
</table>

ADD JUMPERS: (Signals defined and active state)

| I/O-27 to N-1 | (PWAIT-H) |
| V-6 to N-2    | (BOARD SELECT -L) |
| N-3 to T-13 to T-12 | (INVERTER) |
| T-11 to T-10  | |
| S-3 to T-9    | (REFRESH-H) |
| T-8 to M-2    | (REFRESH DURING PWAIT-L) |

NOTE: Observe N-3 was cut in the MITS MOD earlier, and should be open!

See Table 2 for the cuts and jumper list necessary for this modification. Please be familiar with Table 3 for modification considerations. Figure 2 shows the necessary documentation to update the schematic.

Table 3: Rework Considerations

1. Jumpers should be 26-30 gauge single stranded wire. Wire wrap wire would be a good selection.
2. Jumpers should be as short as possible while maintaining a good workmanship standard.
3. Jumpers should never be installed on the solder side of a board. These can easily snag when installing and removing boards in the system.
4. Jumpers should be run on the component side of the board, and under caps and resistors. This assures jumpers can’t lift up and become a snag problem.
5. The jumper ends should be stripped just enough so a shallow ‘U’ can be formed to hook about the IC pin without shorting to the adjacent pin.
6. Watch soldering technique. Solder bridges are readily formed when soldering jumpers to ICs.
7. Make trace cuts cleanly and well defined by inserting upon a sharp X-ACTO blade.
8. Check all rework before reapplying power in a system.

NOTE: Observe N-3 was cut in the MITS MOD earlier, and should be open!

The modification is now complete, and the 88-S4K should function like its modern designed counterparts in every way. Put it to the refresh retention test spoken about earlier. Be absolutely sure that the board is not dropping bits. If there are problems, check the rework carefully. See the appendix for the theory of operation for the multiple WAIT state modification.

Properly functioning memory is the key paramount issue in computing. Eagles Computer Works offers a comprehensive memory diagnostic which was used to detect the 88-S4K problem and help find the solution. MEMDOC contains 14 tests and an oscilloscope loop which can turn your processor into a digital signal generator. It is available assembled at five locations on North Star single density diskette, or as a hex image dump ready for entry for $34.95. A manual is included. The non-hardware oriented computer user can qualify computer memory periodically, and allow troubleshooting by board swapping to assure good computing.

APPENDIX — Theory of Operation for the multiple WAIT state modification. (Refer to Figure 2).

PWAIT is the processor signal relating that a wait state is in progress. Gate N-1 allows PWAIT to become part of the refresh arbiter logic. N-2 is low if the memory board is selected, and this disables PWAIT from asserting N-3 low. Gate T-13, 12, and 11 serves only as an inverter for T-10. T-9 is wired to S-3, and this line is active high at a 32 micro-second rate. This enables the PWAIT refresh cycles only at the appropriate time, assuring the board uses the least amount of power necessary to maintain memory element specifications. The output at T-8 is then active low for a PWAIT refresh, and this drives M-2. Any low on M-2, 1, or 13 is a refresh request command, and the modification is seen to work readily and that all text criteria points are met.
Digital Research's CP/M operating system is the standard for the microcomputer industry. It was introduced in 1975 and has since been adapted to almost every 8080, 8085 and Z80 based microcomputer available. CP/M will control the operation of your computer if you can figure out how to give the proper commands.

Until now, the only documentation available to acquaint you with the operation of CP/M has been the Digital Research manuals. Unfortunately, these manuals assume that you know quite a bit about computers, operating systems and other associated subjects. The organization of these manuals is sometimes quite difficult to follow; it can take a long time to find the section you need.

Six years later, books are appearing that will help you to understand the CP/M operating system and use it. Two of these are *Using CP/M: A Self-Teaching Guide* by Judi N. Fernandez and Ruth Ashley, and *The CP/M Handbook With MP/M* by Rodnay Zaks. I bought these books in a last attempt to understand CP/M. I succeeded.


The title is accurate; this book will really help you teach yourself how to use CP/M. The authors are specialists in writing computer-related education materials, and have published more than 100 courses. Their book on CP/M is a very readable tutorial in all of the facets of the CP/M operating system and utilities.

Each chapter of *Using CP/M* is written in the same format. A chapter starts with an introduction explaining what it will cover. This is followed by a list of objectives. The main body of each chapter is divided into "frames" composed of one or two paragraphs, followed by a series of questions about the material just covered. Chapters end with a self-test and a list of machine exercises to try. I found this format delightful after the usual computer user's diet of dry, cryptic and poorly written software manuals.

There are ten chapters in *Using CP/M*. The first is an introduction not just to CP/M but to computers, disks and buzz words as well. The basic structure of CP/M is covered. The FDOS, which is composed of the BIOS and BDOS, is explained so that the reader understands what parts of CP/M perform which tasks. A simple memory map is shown. This is the first time I have seen such a map presented so clearly. The chapter ends with a self test and machine exercises that concentrate on getting your CP/M system up and running.

Subsequent chapters cover command formats and control characters, Built-In (Intrinsic) Commands, and Transient Programs. Separate chapters are devoted to using the STAT, PIP, ED and SUBMIT utility programs. Each chapter is well written; explanations are clear. You may pace yourself, stopping whenever you need to. I found that I could set the book down and come back to it without problem. This is not always true of technical books.

ED, the CP/M text editor is key to most of the things you might do with CP/M. It is used for editing text to be printed, and for assembly language and high-level language source programs. Since this utility is so important, *Using CP/M* devotes three chapters to it: an introduction, instructions in editing existing files and advanced features. As with the rest of the book, the reader is led slowly at first, then more complex topics are covered until the entire subject has been explained.

The chapter on SUBMIT explains both the SUBMIT utility and the XSUB program supplied in CP/M versions 2.0 and later. SUBMIT allows you to create a file of commands using the text editor. This can save a lot of repetitive typing if there is a group of commands you use frequently. XSUB allows you to type information into the CP/M system in the middle of executing a SUBMIT file. Thus if you have a submit file that assembles a program, lists it and then runs it, XSUB could be used to accept the names of the source and object files. That allows you to write a general purpose submit file that can be varied when you run it.

There are three rather short appendices at the end of the book. The first, "Changing Device Assignments," is too superficial to be of use to the novice. Unless you can determine which drivers have been installed in your CP/M, appendix A won't help you. Appendix B summarizes the PIP parameters, including H,I, O, V and...
Book Reviews cont'd...

Z—which aren’t explained in the book. Appendix C summarizes all CP/M commands. I feel that Using CP/M: A Self-Teaching Guide is the best book on CP/M I have ever seen. Though I am not a novice, I got a truly thorough understanding of the operating system without feeling that I was being talked down to. Novices will find the book easy to read and all concepts used in the book are first explained. I highly recommend Using CP/M to anyone with a CP/M operating system.


Dr. Rodnay Zaks is one of the most prolific authors writing about microcomputers today. It was to be expected that he would make a contribution towards filling the gap in CP/M documentation. The CP/M Handbook With MP/M is a very thorough user’s manual on the CP/M operating system.

The first four chapters cover much the same material as Using CP/M. The two computer systems used as examples are the Processor Technology SOL and the Cromemco System 3. The SOL is defunct for the time being and the Cromemco is at the high end of the microcomputer product family. I don’t feel these two systems were a good choice for teaching you how to get CP/M up and running.

The writing in The CP/M Handbook is less readable than that in Using CP/M. It’s not poorly written, but the style used requires more concentration. The real value of the book for me is contained in the later chapters.

Chapter 5 is “Inside CP/M (and MP/M).” It gives a thorough explanation of how the operating system is organized, how it works and how to modify it for your needs. A turn-key system is offered as one reason you might want to modify CP/M. Turn-key systems assume very little knowledge on the part of the user. Turn the computer on and it asks you questions using menus until your input is no longer needed. The procedure is thoroughly documented. Examples such as this give the reader the confidence to go into the system and do some fiddling around.

A complete command reference for CP/M and MP/M is contained in chapter 6. The commands are listed alphabetically, with one command per page. Each page has a large print heading of the command and the upper right corner of the page shows whether the command is in CP/M version 1.4, version 2.2 or in MP/M. Each listing gives the format, arguments used with the command, a description, and explanation of how and why to use the command and some examples. A collection of CP/M commands, thoroughly explained and all in one place is very helpful to me since I don’t use the commands frequently enough to have memorized them.

Chapter 7 is called “Practical Hints.” It could have been called “Good Procedures for Maintaining Sanity in the Computer Room.” Helpful suggestions for operating your computer, disk system and printer are given. These have nothing to do specifically with CP/M, but could come in handy if you want to get along with your computer.

The last chapter devotes three pages to conjecture about the direction of microcomputer operating systems in the future. It is followed by fifteen appendices. Some are useful and some aren’t. They are:

Appendix A: Common CP/M Error Messages
Appendix B: Hexadecimal Conversion Table
Appendix C: ASCII Conversion Table
Appendix D: ED Control Characters
Appendix E: ED Commands
Appendix F: PIP Device Names
Appendix G: PIP Keywords
Appendix H: PIP Parameters
Appendix I: CP/M (and MP/M) Commands
Appendix J: Command Editing Controls
Appendix K: CP/M Extension Types
Appendix L: Supplies (Checklist)
Appendix M: Computer Room Organization (Checklist)
Appendix N: Failure Checklist
Appendix O: Basic Troubleshooting Rules

I felt that some of these appendices serve only to make The CP/M Handbook a little thicker, but others may find them more helpful.

I conclude that both Using CP/M and The CP/M Handbook belong in your library if CP/M is in your computer. I used the former to learn how to use the operating system, and the latter as a reference book. If you have to pick only one, Using CP/M is definitely the better written and, I feel, the most helpful.

RESEARCH GRADE

is how we describe the quality built into our S-100 microcomputer systems. We use only the finest quality components available and subject them to exhaustive testing. The result - systems of unparalleled reliability and performance.

Systems:

Graphic-80 ................................................................. $700
15" High Resolution Scion Corp. Display, 512 x 484 pixels, Two 4Mhz Z-80 processors, 64K system RAM, 32K display RAM, Detachable 72 key keyboard with graphics function keys, Two Double Density 8" Floppy Disk Drives, SDOS operating system (CP/M compatible), ROM monitor and diagnostics, RS-229 serial port and Centronics parallel port, Rugged TEI packaging.

Signal-80, Image-80, and Image-80H systems built upon the Graphic-80 and used in scientific data acquisition and image analysis. Please call for information.

Component Specials:

Shugart SA-801R Single Side, Double Density ........................................ $410
SD Systems ExpandoRAM II ........................................... $628
4Mhz, Bank Select, 64K RAM
TEI DFD-0 8" Floppy Disk Enclosure ........................................ $525

Laboratory Computer Systems, Inc.
139 Main St.
Cambridge MA 02142
(617) 547-4738
All prices include Cash or CDO discount, freight charges extra.
**SOFTWARE DIRECTORY**

Program Name: Z-80 FORTH
Hardware System: CP/M 2.0 or MP/M 1.0
Minimum Memory Size: 32K, may be reconfigured by user to take advantage of larger memory sizes.
Language: Z-80 Assembler
Description: Optimized fig-FORTH for Z-80 microcomputers. Uses standard CP/M random access disk files for screen storage. Extensions allow use of all CP/M functions. Distribution diskette includes: interpreter, line editor, screen editor, decompiler, debugging aids, utilities, several demonstration programs and over 100 Kbytes of documentation. Source code provided to extend vocabulary to meet FORTH-79 Standard.
Release: Currently available
Price: $50.00
Included with price: Eight inch soft­s­sectored, single-density diskette, 55 page user manual
Where to purchase it:
Laboratory Microsystems
4147 Beethoven Street
Los Angeles, CA 90066

Program Name: Wiremaster
Hardware System: Any Z-80 CP/M system
Minimum Memory Size: 48 Kbytes
Language: Written in ZSPL (Pete Ridley Software)
Description: Wiremaster is a software tool to aid in the design, layout, and construction of electronic hardware. Its inputs are easily derived from the schematic diagram and fed to Wiremaster in a CP/M text file. Outputs include a network map graphically showing all pins and wires, a wirelist sorted by lengths and levels, a parts list, and check lists that detect all wiring errors. The resulting information is then used for layout, error checking, wiring, component stuffing, and system debugging. Together with the schematic, this forms a complete and easily updated documentation package for an electronic product, and results in substantial savings of time.
Release: November 1980
Price: $75; manual $4
 Included with price: Disk and user's manual
Where to purchase it:
Afterthought Engineering
7266 Courtney Dr.
San Diego, CA 92111

Program Name: Comstar
Hardware System: Double or quad density North Star System
Minimum Memory Size: 32K
Language: Machine Language
Description: Full compiler system for North Star Basic (type 2) programs. Compiled programs run faster and original source is protected. Variable dimensions and disk file numbers must be decimal constants.
Release: December 1980
Price: $400
Included with price: Documentation and disk
Author: Allen Ashley
Where to purchase it:
Allen Ashley
395 Sierra Madre Villa
Pasadena, CA 91107

Program Name: Alpha APL Version 2.0
Hardware System: Alpha Micro
Language: Assembler
Description: Implementation of APL language functionally compatible with large mainframes. Runs as a multi-user system. Has system variables, system functions, I-beam, component I/O and other features. Runs under Alpha Micro operating system. Can be used with either ASCII or APL terminals. Assembler subroutines can be called directly. Source code for many external subroutines and assembler subroutine development aids are included.
Release: November 1980
Price: $500; manual $25
Included with price: Disk and user's manual
Where to purchase it:
Softworks Limited
607 W. Wellington
Chicago, IL 60657

Program Name: D80
Hardware System: CP/M with 8" drive
Minimum Memory Size: 24K (CP/M System Size)
Language: 8080, 8085, and Z80 compatible machine code
Description: D80 is a flexible and powerful disassembler for 8080, 8085, and Z80 machine code programs. It accepts a machine code program from disk or memory and produces a disk file of the disassembled code using either the Intel or Zilog mnemonics. In the created source file, D80 will produce a map of the object file, symbol table, up to four types of symbolic labels, and uses the ORG, EQU, and ND pseudo-­opcodes. The created source code is not held in memory, therefore very large programs can be disassembled. Also, all information about a disassembly can be stored in a disk file and then loaded at a later time to pick-up where you left off. Also runs in interrupt and MP/M environments.
Release: November 1980
Price: $85.00(disk and manual); $75.00 (disk only); $10.00(manual only)
Included with price: Disk contains D80 with sample disassemblies and original source code of an included utility
Author: Dennis Gallagher
Where to purchase it:
DG Software
P.O. Box 1035
Iowa City, IA 52244
TOUGHEST BOARDS
IN TOWN
FOR S-100's

Monitor and control in wicked environments.

Want to put your S-100 system to work in the world of computerized monitoring and process control? Dual Systems has all the boards it takes to do the job in the toughest factory environments. All are designed to function dependably in the real world of industrial control. All operate with Cromemco, North Star, and other S-100 systems.

**A/D board.** 12-bit precision, 32 single-ended inputs. Or 16 differential inputs, 25 MHz conversion time. Vectored interrupt, $635. Or $725 with 1 to 1000 gain transducer amplifier. Works with our thermocouple compensation board and our 4-20 mA input boards as well.

**D/A board.** Four independent channels, 12-bit precision. Input is binary or 2s complement. Compatible with all existing I/O mapped software, $495. Drives our amplifier board which outputs 4-20 mA.

**CMOS RAM board.** On-board battery back-up preserves data a year. 200 ns read/write time. Runs at 4 MHz. 8K bytes $590. 16K bytes $990.

**CMOS clock board.** On-board battery back-up keeps clock running a year. New LSI chip carries date, hours, minutes and seconds. Read or write directly from I/O port. Vectored interrupt $250. We also provide complete mainframe systems. OEM and dealer inquiries are invited.


---

**TOUGHEST BOARDS IN TOWN FOR S-100's**

**Program Name:** ENTRY

**Hardware System:** CP/M

**Minimum Memory Size:** 52K bytes

**Language:** Microsoft Basic

**Description:** The UNIVERSAL DATA ENTRY System provides an interactive editing of data files complete with CRT format and prompts. Once defined, data entry becomes a 'fill in the blank' operation. A wide range of validity checks reduces error to a minimum.

**Release:** Currently available

**Price:** $195; License Agreement Required

**Included with price:** Diskette, manual, examples, support

**Author:** The Software Store

**Where to purchase:** The Software Store

706 Chipewa Square
Marquette, MI 49855

---

**Toughest Boards in Town for S-100's**

**Program Name:** Z-80 DES

**Hardware System:** Z-80 based

**Minimum Memory Size:** 16K

**Language:** Z-80 assembler

**Description:** High Speed Implementation of the NBS data encryption algorithm. Modular and user oriented. Fully documented source code supplied for the algorithm. Special run-time package for TRS-80. Database protection, password scrambling, telecommunications security from remotely-accessed data files. Protect sensitive or proprietary files. Easily adapted by user for custom uses.

**Release:** April 1979

**Price:** $34.95 + $9.95

**Included with price:** Documented source code listing

**Author:** ITM

**Where to purchase:** Interface Technology of Maryland

P.O. Box 745
College Park, MD 20740

---

**TOUGHEST BOARDS IN TOWN FOR S-100's**

**Program Name:** Infomedia System (IMS) Ver 1.1

**Hardware System:** S-100 (Vector MZ), Micropolis Drives, CRT with cursor controls

**Minimum Memory Size:** 48K

**Language:** MDOs Micropolis Basic of CP/M-CPBasic2

**Description:** A menu driven data base and file management system, plus report writer. All user created data base formats, data files, and report formats are listed on a system directory. Allows up to 20 user defined data formats and reports per disk. The user can define up to 24 numeric or alphanumeric fields for file records. File functions include: Create, Delete, Duplicate, Add/Modify, and List. Record functions include: Add, Update, Delete, Scan, List, Sort, Compact, and Create, Delete, Add/Modify, List and Print. User selectable column or label format, titles, fields, subtotals, total, and printing of selectable records. Fields may be specified as mathematical calculations of other fields (+, -, "", &). All are supported.

**Release:** January 1980

**Price:** $195

**Included with price:** Eight programs and manual send $3 for brochure and sample print-outs, or $195 plus $2 shipping. CA residents add 6% sales tax.

**Where to purchase:** Investment Analysis Systems

P.O. Box 282
Palos Verdes Estates, CA 90274

---

**TOUGHEST BOARDS IN TOWN FOR S-100's**

**Program Name:** Property Analysis System (PAS) Ver 2.11

**Hardware System:** S-100 (Vector MZ), Micropolis Drives, CRT with Cursor controls

**Minimum Memory Size:** 48K

**Language:** MDOs-Micropolis Basic or CP/M-CPBasic2

**Description:** Property Analysis System, for both residential and income property. Analyzes the effects of financing, income, expenses, depreciation, taxes, and inflation on the return on investment for nine years. PAS produces a three-page report consisting of initial conditions and a nine year projection of property value, liabilities, equity, gross income, expenses, net income and capital and percentage return on investment before and after taxes. PAS also provides percentage return on equity.
**DIGITAL SOUND SYNTHESIZER**

**BY CASHEAB**

- 32 CHANNELS
- S-100
- FREQUENCY MODULATION
- UP TO 16 WAVEFORM STORAGE

Casheab has designed and developed a 32 channel digital sound synthesizer for the S-100 bus. The synthesizer consists of two cards: a synthesizer card (SYN-10) and a controller card (CTR-10). The S-100 host processor programs the waveforms (1024 by 12 bits) into the synthesizer. Either 4 waveforms (SYN-10/4) or 16 waveforms (SYN-10/16) can be stored. Any of the channels can use any of the waveforms. In addition attack, steady state and decay envelopes can be implemented by the host processor controlling each channel's amplitude. The synthesizer also incorporates frequency modulation which can be used for vibrato or FM synthesis.

- PROGRAMMABLE TIMBRE WAVEFORMS
- INDIVIDUAL CHANNEL AMPLITUDE CONTROL
- INDIVIDUAL CHANNEL FREQUENCY CONTROL
- INDIVIDUAL CHANNEL TIMBRE SELECTION

Software on a CP/M compatible floppy disk is provided free with the purchase of the synthesizer. This software includes a waveform creation, music compiling and a real time operating program. The waveform creation software generates waveforms from user supplied data. This program, written in BASIC utilizes a FFT algorithm. A music compiler program converts music data, entered using data statements to and executable format. The real time operating program, written in 8080 assembly language loads the waveforms and plays the music generated from the compiler.

*CP/M is a trademark of Digital Research*

**CASHEAB**
5737 AVENIDA SANCHEZ
SAN DIEGO, CA 92124
(714) 277-2547

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYN-10/4 &amp; CTR-10</td>
<td>$1095.00</td>
</tr>
<tr>
<td>SYN-10/16 &amp; CTR-10</td>
<td>$1245.00</td>
</tr>
<tr>
<td>MANUAL</td>
<td>$5.00</td>
</tr>
<tr>
<td>DEMO CASSETTE</td>
<td>$3.00</td>
</tr>
</tbody>
</table>

Overseas order for air mail add $20.00 for synthesizer and $3.00 for MANUAL purchase.

---

**Worth A Fortune**

Past issues of Creative Computing. What are they worth today? It varies. To a collector, Vol. 1, No. 1 is worth $7 or $8. To a scrap dealer, less than two cents.

But we're not selling old back issues. We're all out.

On the other hand, you know that much of the content of Creative Computing is timeless. The Depth Charge program in Vol. 1, No. 1 is just as challenging today as the day it was written. Walter Koetke's series of five articles on using computers in the classroom are as valid today as the day they first appeared in print. And scores of people have written about obtaining reprints of Don Piele's classic problem-solving series.

**Our Solution**

We guarantee you'll never find a better value in computer applications reading matter. On average you're getting 128 pages of solid information for each $1.00. If you're not completely satisfied after you've read them, send the books or magazines back to us and we'll refund your full purchase price plus the return postage.

**Our Offer**

You select the best material from the three years of Creative Computing in book form and the next three years (minus four issues) in the original magazine form.

**Best of Creative Computing-Vol 1**  $8.95
**Best of Creative Computing-Vol 2**  8.95
**Best of Creative Computing-Vol 3**  8.95

Volume 4 (Four issues)  6.00
Volume 5 (Ten issues)  15.00
Volume 6 (Twelve issues)  18.00

Less discount (5% for two items, 10% for three, 15%, for four, 20% for five, 25% for all six) Shipping ($2.00 USA, $5.00 foreign)

---

This isn't really an order form. Cut it out or copy the items you want onto a sheet of paper. Send it with your name and address and payment or credit card number and expiration date (Visa, MasterCard or American Express) to Creative Computing, Morris Plains, NJ 07950. Or call toll-free to 800-631-8112 (in NJ, 201-540-0445).

We guarantee you'll never find a better value in computer applications reading matter. On average you're getting 128 pages of solid information for each $1.00. If you're not completely satisfied after you've read them, send the books or magazines back to us and we'll refund your full purchase price plus the return postage.

---

**creative computing**
Morris Plains, NJ 07950
Toll-free 800-631-8112
(In NJ 201-540-0445)
New Products cont’d...

for before and after taxes. PAS was designed for ease of use without prior computer experience—field tested since mid 1979. All data can be changed at any time with the effects immediately displayed for review, allowing the user to model an investment property and ask “What if?” PAS stores up to 20 different properties per disk with full file management capability (create, delete, read, & write/update).

Price: $195
Included With price: Five programs and 60 pg. users manual, send $3 for brochure and sample printouts, or $196 plus $2 shipping. CA residents add 6% sales tax.
Where to purchase it: Investment Analysis Systems
P.O. Box 282
Palos Verdes Estates, CA 90274

---------------------------

Program Name: HayesSys
Hardware System: PolyMorphic Systems 8813 single density
Minimum Memory Size: 8K (uses system RAM)
Language: 8080A Machine Language
Description: HayesSys is a D.C. Hayes MICROMODEM 100 terminal operating system consisting of two programs and a modified version of the operating system which allows operation of the D.C. Hayes MICROMODEM 100 board. The system includes complete software control of all parameters, auto dial, and all other features of the D.C. Hayes board except auto answer. The system includes the ability to download to disk files and to send files from disk, as well as the ability to log the incoming text to a printer. Available for Exec/78 and for Exec/83 (specify which).
Release: September 1980
Price: $85 postpaid
Included with price: Installation instructions & disk.
Where to purchase it:
Ralph E. Kenyon Jr.
145-103 S. Budding Ave
Virginia Beach, VA 23452

---------------------------

Program Name: DF
Hardware System: CP/M
Minimum Memory Size: 30K CP/M Configuration
Language: Compiled from C (BDS version)
Description: Shows all differences between two versions of a printable file, such as a source program. Re-synchronizes and continues after reporting deletions and insertions. Comparisons are on a line basis, but you can specify the line delimiter and a character to be ignored so that PCL and other text block files can be compared. As compiled, it can handle differences as large as 8K in files of any length.
Release: Available now
Price: $20 (check or money order)
Included with price: CP/M COM and C files on CUTS cassette or a paper listing. Or send a Micropolis Mod II diskette. Can arrange for conversion to standard 8” diskette for $5 extra. Modem?
Where to purchase:
Richard Greenlaw
261 Colney Ct.
Gahanna, Ohio 43230
The Sixth
Trenton Computer Festival
TCF-81
April 25 & 26, 1981
10 AM to 6 PM
at TRENTON STATE COLLEGE
Trenton, New Jersey
10 AM to 4 PM
Saturday, 25th
Sunday, 26th

Super Outdoor Flea Market
Surplus computer gear, bargains galore, over 5 acres of space (5/spot, no electricity).

Indoor Commercial Exhibit Area
90 exhibitor booths showing newest products; special discounts; funky games to play.

Forums, Talks & Seminars
Meet the leading experts and hear sessions on robots, computer music, amateur radio, etc.

Convenient to NY, PA, MD & DEL
Easy to get to; free parking for over 6,000 cars.

Free Short Courses on Sunday
Hundreds of Door Prizes • Banquet Saturday Night

For additional information call 609-771-2487
Admission $5 - Students $2

Banquet $10. Avoid disappointment – pre-register for the banquet! Send your check for $10 per person to: TCF 81, Trenton State College, Trenton, NJ 08625

Sponsored by:
Amateur Computer Group of New Jersey
Philadelphia Area Computer Society
Trenton State College Digital Computer Society
Dept. of Engineering Technology, Trenton State College
I.E.E.E., Princeton Section
New 8/16-Bit Memory Board By SSM Microcomputer Products

SSM Microcomputer Products has announced the MB10—a 4-MHz, 16Kbyte static RAM memory board that supports the proposed IEEE S-100 bus standard. For those applications requiring bank switching, the MB10 is jumper selectable to any I/O address with enable/disable on power-up or reset (Cromemco-compatible). The board uses the popular, low power 2114L memory chip, but the on-board logic is designed to upgrade easily to the use of faster memory chips as they become available in the future. Option selection is via wire-wrap jumpers.

Suggested retail price of $399. SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131. (408) 946-7400.

ADA Manual Available

A manual on ADA, the defense department's (DOD) new standard programming language is now available from Pony Express Computer Services, 100 West 57th Street, New York, N.Y. 10019.

In addition to the ADA manual, Pony Express offers a complete update service which keeps subscribers fully informed of ADA developments and program availability.

The ADA manual costs $100 exclusive of shipping. The complete update service is extra. Call (212) 265-7446 for more information.

New Cromemco Serial Communications Interface Card

Cromemco announces the new QUADART serial communications S-100 interface card, with four serial channels. Any channel can support Asynchronous or Synchronous Byte mode (Bisync) or Bit mode (SDLC) communication protocols under software control, with handshaking provided for modems. Baud rates for each may be software-selected from 0 to 300K baud (asynchronous to 19,200 baud).

The QUADART also supports the powerful internally-prioritized vectored interrupt structure of the Z-80A microprocessor.

The price is $654. SSM Microcomputer Products at 2190 Paragon Drive, San Jose, CA 95131, (408) 946-7400.

64 Channel D/A Converter Card

The Model SB-64 D/A converter board has 64 output channels accomplished by multiplexing a single 8-bit converter. Each output drives a 1K ohm load, and has ±4% absolute accuracy from 0V to 10.625V. Address decoding allows memory-mapped I/O to any 256 byte sector in the 65K. IEEE specifications for S-100 Bus interface devices are fully met. Options available are: 10 digital outputs, fast 12-bit DAC with 3 microsecond settling, ±15V tracking regulator with 600ma output, and 24-bit address decoding.

The price is $514; Digital Multi-Media Control, 2338 Patterson #12, Eugene, OR 97405.

A/C Power Noise Suppressor

Electronic Specialists recently announced the Model ISC-3 Super Isolator, an isolator designed to control electrical pollution. Incorporating heavy duty spike/surge suppression, the Super Isolator features 3 individually dual-Pi filtered AC sockets. Equipment interacations are eliminated and disruptive, damaging power line pollution is controlled. The Super Isolator will control pollution for an 1875 watt load. Each socket can handle a 1000 watt load. Price is $94.95. Electronic Specialists, Inc., 171 South Main Street, Natick, MA 01760. Phone: (617) 655-1532.

SSM Unveils New Z-80 CPU Board

The CB2 Z-80 CPU Board for S-100 based systems is capable of operating at 2MHz or 4MHz. The CB2 features firmware vector jumps and an output port to control eight extended address lines. These permit the use of more than 64K of additional memory with the CPU board. Separate run/stop and single/step switches are also included. Jumper options on the board will also generate the industry's new proposed IEEE S-100 signals. The on-board EPROMs have a capacity of 2048/4096x8 bits, are DIP switch addressable or disabled and have one wait state added. The on-board RAM has a capacity of 2048x8 bits and is also DIP switch addressable or disabled.

The price is $595. Quasar, Inc., 280 Bernardo Avenue, Mountain View, CA 94043. (415) 964-7400.
In which we sell a 386 page, $11.95 book for just 2¢.

Liquidation Giveaway

Byte magazine. You’ve seen it. It's the fat technical one.

Back when Byte was first publishing independently, Creative Computing and Byte cooperated in many areas. We ran joint promotions, directed articles to each other and the like.

In 1976, Creative published The Best of Creative Computing, Volume 1. I proposed to Virginia Londoner, publisher of Byte, that we also publish articles from Byte in book form. She agreed, and so we published The Best of Byte, Volume 1. It’s a huge book of 386 pages with articles on hardware, software, technical tutorials, how-to materials and even some philosophy.

Although some of the technical material in The Best of Byte is out of date today, it nevertheless provides a good historical framework for the personal computing field.

Not at all out of date are most of the software articles and tutorials. Similar books of other publishers are selling for $20 and up, so at $11.95, this one is quite a bargain.

Big Hearted

About the same time we were preparing The Best of Byte for publication, Nat Wadsworth of Scelbi approached Byte about doing a similar book. Virginia wanted to be nice to everyone, so she gave permission. Thus was born the Scelbi-Byte Primer.

Unfortunately, about half of the content of the two books was identical. Thus Byte was faced with a dilemma of which book to endorse and sell through their magazine. Inexlicably, they chose the Scelbi book. Thus we were left with twelve skids of The Best of Byte.

Hidden Away

In the next three years we sold a lot of these books. In fact, after we ran a special in 1979, we thought we had sold out.

However, we just moved to new quarters.

In the move we found, lurking away in the back of our old garage, four skids of The Best of Byte. After some fitting words, the boss said “for 2¢, I'll give them away.” So that's what we're doing.

Our Ridiculous Offer

The original price of The Best of Byte was $11.95. If you order $11.95 worth of any of our other books or records, we'll throw in The Best of Byte for 2¢.

Thus you could order The Best of Creative Computing, Vol. 3 ($8.95) and Computer Coin Games ($3.95). The total price is $12.90. For $12.92 you also get The Best of Byte. Shipping and handling on all book orders is $2.00.

Here are the books you can use to come up with an $11.95 or greater total:

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best of Creative Computing, Vol. 1</td>
<td>$6.95</td>
</tr>
<tr>
<td>Best of Creative Computing, Vol. 2</td>
<td>8.95</td>
</tr>
<tr>
<td>Best of Creative Computing, Vol. 3</td>
<td>8.95</td>
</tr>
<tr>
<td>Basic Computer Games</td>
<td>7.50</td>
</tr>
<tr>
<td>More Basic Games (Microsoft)</td>
<td>7.95</td>
</tr>
<tr>
<td>More Basic Games (TRS-80)</td>
<td>7.95</td>
</tr>
<tr>
<td>Computer Coin Games</td>
<td>3.95</td>
</tr>
<tr>
<td>Be A Computer Literate</td>
<td>3.95</td>
</tr>
<tr>
<td>Computers in Mathematics</td>
<td>15.95</td>
</tr>
<tr>
<td>Problems for Computer Solution (Student)</td>
<td>4.95</td>
</tr>
<tr>
<td>Problems for Computer Solution (Teacher)</td>
<td>9.95</td>
</tr>
<tr>
<td>Computers in Society Bibliography</td>
<td>17.95</td>
</tr>
<tr>
<td>Katie and the Computer</td>
<td>6.95</td>
</tr>
<tr>
<td>Computers For Kids (TRS-80)</td>
<td>3.95</td>
</tr>
<tr>
<td>Computers For Kids (Apple)</td>
<td>3.95</td>
</tr>
<tr>
<td>Tales of the Marvelous Machine</td>
<td>8.95</td>
</tr>
<tr>
<td>Colossal Computer Cartoon Book</td>
<td>4.95</td>
</tr>
<tr>
<td>Computer Rage Game</td>
<td>8.95</td>
</tr>
<tr>
<td>Computer Music Record</td>
<td>6.00</td>
</tr>
</tbody>
</table>

Limited Supply

We expect a heavy response to this offer, so order today to be sure of getting The Best of Byte for just 2¢.

Send us your order for books of $11.95 or greater plus 2¢ for The Best of Byte and $2.00 postage handling. Send payment or Visa, MasterCard or American Express number and expiration date to the address below or call our toll-free number.

Don’t delay; order today.

Creative computing

Attn: Terry
P.O. Box 789-M
Morristown, NJ 07960
Toll-free 800-631-8112
(In NJ 201-540-0445)
New QT System Offers Up to 16 Megabyte Storage Capacity

QT Computer Systems has introduced their new QT System™ which provides standard two Megabyte storage, plus availability of up to 16 Megabytes storage. The new system includes mainframe, two 8" disk drives, double-sided double-density 5 1/4" & 8" floppy disk controller, power supply, fan and Televideo 920C terminal. This 4MHz/Z80/CPU System includes such features as: 48K Dynamic Memory (expandable to 64K); 2K Monitor program and Disk BIOS on 2716 EPROM; AM/RAM/ PROM in combination up to 8K, two RS-232C Serial I/O ports; two Parallel I/O ports; Hard disk compatible; Real time clock; CP/M 2.2 Operating System; Power-on/Reset jump to monitor program.

The one Megabyte single-sided, double-density system sells for $4,285, while the two Megabyte double-sided, double-density unit is priced at $4,995. The units are assembled, tested and burned into at the factory. Each system includes documentation and has a one year warranty against defects in material and workmanship.

For additional information, contact QT Computer Systems, Inc., 15620 South Inglewood Avenue, Lawndale, CA 90260. Call (213) 970-0952. The toll-free number (except California, Alaska and Hawaii) is 800-421-5150.

Konan Unveils New Controller

Konan's KNX-500™ is a new hard disk controller which is hardware compatible with the Alpha Micro AM-500™. KNX-500™ Interfaces with the S-100 Bus, plugs in the same as AM-500 and uses the same software drivers. It will control four drives from one board and is designed to work with most 5440 drives, typically 10 to 20 megabyte, such as Western Dynex, CDC Hawk, Diablo, and Pertec. Bundled with a Western Dynex drive, KNX-500™ will be offered as subsystem—including cable, disk pack, drive, and controller—for $4500 O.E.M., quantity one. (Additional disk packs are $95 each.) Alone, KNX-500™ is priced at $1200 O.E.M., quantity one, with quantities of 200 or more (rackmount) and $2200 each (table mount) in lots of 25. Contact Konan Corporation, 1448 N. 27th Avenue, Phoenix, Arizona; 800-526-4563. *trademark of Alpha Micro Systems.

S-100 Controller Supports IBM-Compatible 9-Track Tape

The Alloy Engineering TS-100 Controller permits any IBM-compatible 9-track formatted tape drive, including the new Cipher Microstreamer™, to be interfaced with the S-100 Bus. TS-100 Controller is $600, including software and dual 8 ft. data cables with 50 pin connectors. Contact: Alloy Engineering Company, Inc., Computer Products Division, 85 Spen Street, Framingham, MA 01701 (617) 620-1710.

Winchester Disk Backup Cartridge Tape Subsystem

The Alloy Engineering TIP (Tape Interface Package) is an S-100 compatible cartridge tape subsystem and software utility that permits transfer of programs and data files from a Winchester disk to an easy-to-handle 13.4 MB 1/4" tape cartridge. Facilitating off-premises data base storage and shipping, the subsystem features comprehensive menu driven software, and links to the Winchester disk under Digital Research CP/M and MP/M Operating Systems.

The TS-100 Controller is $600. Including software, and dual 8 ft. data cables with 50 pin connectors. Contact: Alloy Engineering Company, Inc., Computer Products Division, 85 Spen Street, Framingham, MA 01701 (617) 620-1710.

MICROSYSTEMS
You might think the term ‘creative computing’ is a contradiction. How can something as precise and logical as electronic computing possibly be creative? We think it can be. Consider the way computers are being used to create special effects in movies—image generation, coloring, and computer-driven cameras and props. Or an electronic “sketchpad” for your home computer that adds animation, coloring and shading at your direction. How about a computer simulation of an invasion of killer bees with you trying to find a way of keeping them under control?

Beyond Our Dreams

Computers are not creative per se. But the way in which they are used can be highly creative and imaginative. Five years ago when Creative Computing magazine first billed itself as “The number 1 magazine of computer applications and software,” we had no idea how far that idea would take us. Today, these applications are becoming so broad, so all-encompassing that the computer field will soon include virtually everything.

In light of this generality, we take “application” to mean whatever can be done with computers, ought to be done with computers or might be done with computers. That is the meat of Creative Computing.

Alvin Toffler, author of Future Shock and The Third Wave says, “I read Creative Computing not only for information about how to make the most of my own equipment but to keep an eye on how the whole field is emerging. Creative Computing, the company as well as the magazine, is uniquely light-hearted but also seriously interested in all aspects of computing. Ours is the magazine of software, graphics, games and simulations for beginners and relaxing professionals. We try to present the new and important ideas of the field in a way that a 14-year-old or a Cobol programmer can understand them. Things like text editing, social simulations, control of household devices, animation and graphics, and communications networks.

Understandable Yet Challenging

As the premier magazine for beginners, it is our solemn responsibility to make what we publish comprehensible to the newcomer. That does not mean easy; our readers like to be challenged. It means providing the reader who has no preparation with every possible means to seize the subject matter and make it his own.

However, we don’t want the experts in our audience to be bored. So we try to publish articles of interest to beginners and experts at the same time. Ideally, we would like every piece to have instructional or informative content—and some depth—even when communicated humorously or playfully. Thus, our favorite kind of piece is accessible to the beginner, theoretically non-trivial, interesting on more than one level, and perhaps even humorous.

David Gerroll of Star Trek fame says, “Creative Computing with its unpretentious, down-to-earth lucidity encourages the computer user to have fun. Creative Computing makes it possible for me to learn basic programming skills and use the computer better than any other source.

Hard-hitting Evaluations

At Creative Computing we obtain new computer systems, peripherals, and software as soon as they are announced. We put them through their paces in our Software Development Center and also in the environment for which they are intended—home, business, laboratory, or school.

Our evaluations are unbiased and accurate. We compared word processing printers and found two losers among highly promoted makes. Conversely, we found one computer had far more than its advertised capability. Of 16 educational packages, only seven offered solid learning value.

When we say unbiased reviews we mean it. More than once, our honesty has cost us an advertiser—temporarily. But we feel that our first obligation is to our readers and that editorial excellence and integrity are our highest goals.

Karl Zinn at the University of Michigan feels we are meeting these goals when he writes, “Creative Computing consistently provides value in articles, product reviews and systems comparisons … in a magazine that is fun to read.”

Order Today

To order your subscription to Creative Computing, send $20 for one year (12 issues), $37 for two years (24 issues) or $53 for three years (36 issues). If you prefer, call our toll-free number, 800-631-8112 (in NJ 201-540-0445) to put your subscription on your MasterCard, Visa or American Express card. Canadian and other foreign surface subscriptions are $29 per year, and must be prepaid. We guarantee that you will be completely satisfied or we will refund the entire amount of your subscription.

Join over 80,000 subscribers like Ann Lewin, Director of the Capital Children’s Museum who says, “I am very much impressed with Creative Computing. It is helping to demystify the computer. Its articles are helpful, humorous and humane. The world needs Creative Computing.”
**North Star**

**Basic Utility Set**
- Editor: Create & edit a Basic program using 26 commands, including GLOBAL locate & change.
- BpRt: Print & cross reference a Basic program.
- BpAk: Pack a Basic program.
- Re: Rename a disk file.

**Prices**
- Editor: $69.00
- BpRt: $69.00
- BpAk: $69.00
- Re: $69.00
- Plus $1.50 shipping.

**Software Systems**
1265 Rubio Vista Road, Altadena, Calif. 91001
(213) 791-3202

---

**LETTER QUALITY**

<table>
<thead>
<tr>
<th>Type</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCl</td>
<td>$1600</td>
</tr>
<tr>
<td>KSR TERNIMALS</td>
<td></td>
</tr>
</tbody>
</table>

**NEW PRODUCTS**

- Digital Research announces BT-80™, a comprehensive single user record retrieval system. Based on the B-Tree index organization technique, BT-80 is designed for use in PL/I-80 applications where single or multi-keyed access to data records is required.

---

**Amateur Radio Interface Card**

The Snow Micro Systems AR.01 is a versatile S-110 Bus card designed to facilitate the merging of amateur radio and the personal computer. It contains a digital and an analog section. The digital section may be used as a serial I/O port for RS-232 levels at 45.5, 50, 56, 74, 75 and 110 bauds using ASCII or Baudot codes. The analog section comprises a modem and can be set up for any commonly used tone pairs. The AR.01 is normally supplied set up for the standard 2125-2295 tone pairs commonly used in the amateur bands.

The board can thus be used as a serial I/O port driving an ASCII or Baudot printer, as an interface to existing RTTY equipment such as an ST-5, as a full RTTY terminal unit, as an interface between a radio receiver and a computer, for morse code reception software, and as an RTTY modulator/demodulator (TU) for an existing printer in a non-computer environment.

The AR.01 is available as follows: Bare board and documentation for $45, kit for $245, assembled and tested for $349. For information contact: Snow Micro Systems, Inc., P.O. Box 1704, Silver Spring, MD 20902, Phone: (301) 622-2194.

---

**Station Controller Card**

The Snow Micro Systems SC.01 Station Controller card is a versatile S-100 bus card designed for process control applications. It can be used in almost any control type of environment such as Energy Management, Alarm Monitoring and Amateur Radio.

The specifications of the card include: S-100 bus compatible, hardware time of day 24 hour clock, 6 Reed relays with isolated contacts, 7 Analog input channels (5 channels have a maximum input level of 2 Volts, 2 channels are scaled via 100K potentiometers. The Analog channels are multiplexed by a 4051 CMOS switch, and converted to digital levels by a 31/2 digit ADC device (14433). The time of day clock uses a low cost 3.579 MHz color TV crystal and has provision for battery backup. The PC board may be used with several kinds of relays.

The SC.01 is available as a bare board for $45, as a kit for $245 and assembled and tested for $349. For information contact: Snow Micro Systems, Inc., P.O. Box 1704, Silver Spring, MD 20902, Phone: (301) 622-2194.

---

**MicroSystems**

---

**Letter Quality**

<table>
<thead>
<tr>
<th>Type</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCl</td>
<td>$1600</td>
</tr>
<tr>
<td>KSR TERNIMALS</td>
<td></td>
</tr>
</tbody>
</table>

**NEW PRODUCTS**

- Digital Research announces BT-80™, a comprehensive single user record retrieval system. Based on the B-Tree index organization technique, BT-80 is designed for use in PL/I-80 applications where single or multi-keyed access to data records is required.

BT-80 facilities may be accessed from PL/I-80 or assembly language application programs via two BT-80 system procedures which handle all data file and index maintenance. The BT-80 system also includes utilities that provide access to commonly needed facilities at the command level.

BT-80 runs on CP/M version 2, MP/M™ and CP/NET™ operating systems, and requires the PL/I-80 Run-time Library and LINK-80 Linkage Editor to operate.

For information contact: Digital Research, Post Office Box 579, 801 Lighthouse Avenue, Pacific Grove, CA 93950: (408) 649-5898.

---

**Amateur Radio Interface Card**

The Snow Micro Systems AR.01 is a versatile S-110 Bus card designed to facilitate the merging of amateur radio and the personal computer. It contains a digital and an analog section. The digital section may be used as a serial I/O port at RS-232 levels at 45.5, 50, 56, 74, 75 and 110 bauds using ASCII or Baudot codes. The analog section comprises a modem and can be set up for any commonly used tone pairs. The AR.01 is normally supplied set up for the standard 2125-2295 tone pairs commonly used in the amateur bands.

The board can thus be used as a serial I/O port driving an ASCII or Baudot printer, as an interface to existing RTTY equipment such as an ST-5, as a full RTTY terminal unit, as an interface between a radio receiver and a computer, for morse code reception software, and as an RTTY modulator/demodulator (TU) for an existing printer in a non-computer environment.

The AR.01 is available as follows: Bare board and documentation for $45, kit for $245, assembled and tested for $349. For information contact: Snow Micro Systems, Inc., P.O. Box 1704, Silver Spring, MD 20902, Phone: (301) 622-2194.

---

**Station Controller Card**

The Snow Micro Systems SC.01 Station Controller card is a versatile S-100 bus card designed for process control applications. It can be used in almost any control type of environment such as Energy Management, Alarm Monitoring and Amateur Radio.

The specifications of the card include: S-100 bus compatible, hardware time of day 24 hour clock, 6 Reed relays with isolated contacts, 7 Analog input channels (5 channels have a maximum input level of 2 Volts, 2 channels are scaled via 100K potentiometers. The Analog channels are multiplexed by a 4051 CMOS switch, and converted to digital levels by a 31/2 digit ADC device (14433). The time of day clock uses a low cost 3.579 MHz color TV crystal and has provision for battery backup. The PC board may be used with several kinds of relays.

The SC.01 is available as a bare board for $45, as a kit for $245 and assembled and tested for $349. For information contact: Snow Micro Systems, Inc., P.O. Box 1704, Silver Spring, MD 20902, Phone: (301) 622-2194.
At last there is a magazine written exclusively for 5-100 system users. No other publication is devoted to supporting 5-100 system users. No longer will you have to hunt through other magazines for an occasional 5-100, CP/M* or PASCAL article. Now find it all in one publication. Find it in MICROSYSTEMS.

Every issue of MICROSYSTEMS brings you the latest in the 5-100 world. Articles on applications, tutorials, software development, letters to the editor, newsletter columns, book reviews, new products, etc. Material to keep you on top of the ever changing microcomputer scene.

SOFTWARE
CP/M*
Assembler
BASIC
PASCAL
applications
and lots more

SYSTEMS
Cromemco
Intersystems
North Star
IMSAI
SOL
and lots more

HARDWARE
8 bit & 16 bit CPUs
interfacing
hardware mods
bulletin board systems
multiprocessors
and lots more

© 1980 Digital Research

USA Canada, Mexico Other Foreign (Air)
THREE YEARS (18 issues) $24 $38 $69
TWO YEARS (12 issues) $18 $27 $48
ONE YEAR (6 issues) $10 $15 $25

New Renewal Payment Enclosed
Visa MasterCard American Express
Signature Card No.
Expiration date

Please bill me ($1.00 billing fee will be added; foreign orders must be prepaid)

MICROSYSTEMS ORDER FORM
Name
Address
City State Zip

MICROSYSTEMS BACK ISSUES
1-1 Jan/Feb 1980 $5.00 
1-2 Mar/Apr 1980 $2.50 
1-3 May/June 1980 $2.50 Sold out!

Postpaid in USA; add $1.00 per issue foreign postage. Subscriptions start the month following receipt of order. Subscriptions cannot start with earlier issues.
SAVE MORE THAN 20%!

NORTH STAR — INTERTUBE — MICROTEK
ZENITH — RCA-COSMAC — ITHACA
THINKER TOYS — GODBOUT — SUPERBRAIN
The smartest computers at the smartest price!

FACTORY ASSEMBLED & TESTED LIST ONLY

HORIZON-32K DOUBLE DEN 32K=$469 3199
HORIZON-32K QUAD DEN 32K=$579 3299
HORIZON 2-32K QUAD HARD DISK 32K=$632 3329 7149
HORIZON RAM ASM SALE! 16K=$1240 32K=$1672
HORIZON RAM KIT SALE! 16K=$1424 32K=$1848
HORIZON DISK DRIVE SALE DOUB DEN SALE 1
HORIZON RAM KIT SALEI 16K=$1495 32K=$1996
HORIZON RAM ASSM SALE! 16K=$2095 32K=$2596
HORIZON-2·32K·00UBLE DEN 3095
HORIZON PORTS, PARITY, EDGE CONNECTORS, ETC.
PASCAL FOR NORTH STAR ON DISK 199 190
NORTH STAR HARO DISK 16 Mb 4999
PASCAL PLUS 14, 18, or 36 DIGIT PRECISION 249
EZ·CODER II to BASIC 79
A BASIC PROGRAM TRACER! + FANCY RENUMBERING
ASSEMBLER PLUS DISASSEMBLER
Z·6002 CPU CARD II·bit ITHACA S·1 200 1059
ITHACA FRONT PANEL COMPUTER 64K 3195 2195
RCA·COSMAC VP·lll 99 GODBOUT SPECTRUM 269
NORTH WORD 294 MAILMAN 234 INFOMAN 364
NSSE 1·22
BOX OF DISKETTES 29 EZ·80 Z80 TUTORIAL
North Star Documentation refundable w/HRZ
Which Computer, BEST? BROCHURE ..
THINKER TOYS — GODBOUT — SUPERBRAIN
CAT NOVATION MODEM 179
ZENITH·HEATH SMART TERMINAL 995
EMULATOR 4 IN 1 TERMINAL 895
INTERTUBE III SMART TERMINAL 895
CENTRAL DATA 64K RAM 665 599
SYSTEMS GROUP RAM 64K A & T 4mHz 599
SEATTl 8086 CPU 16 bit 556 RAM 16K 816 356
PASCALIZ+ THE SPEED KING 395 375
ITHACA MEMORY 8116·bit 64K 995 845
Mi.roAngolo HI·RES GRAPHICS 1095 985
ZENITH·HEATH Z·89 48K 2895
DISCUSII+2 1.2 Mbyte. A & T 1545 1259
DISCUSI2D A & T + CPIM 1199 938
SYSTEMS GROUP RAM 64K BANK SELECT 789
SUPERBRAIN QUAD DENSITY 3995 2995

YES WE WILL BEAT OUR COMPETITION'S PRICE!

HORIZON DISK DRIVE SALE DOUB DEN *2695
FACTORY ASSEMBLED" TESTED LIST
ORDER 2 or more COMPUTERS .... BIGGER DISCOUNTS

SSM Introduces Eprom Board

SSM Microcomputer Products has introduced the MBBA* 1K—16K EPROM Board. The new MBBA provides sockets to support up to sixteen 2708 EPROMS. Simply by removing the 1K EPROM, the board can be disabled in 1K increments. The user can overlay RAM and ROM at the same address in any desired increment. This provides a marked flexibility when the board is used with RAM boards equipped with Phantom Disable.

For complete information and specifications on the MBBA EPROM Board, contact your local computer store or call or write SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131, (408) 946-7400.

8" Floppy Subsystem for North Star

John D. Owens Associates, Inc. announces an 8 inch floppy subsystem for North Star users. In response to the need of North Star users for greater auxiliary storage and their need to transfer files from 5 1/4 inch media to 8 inch media or vice versa, a software-hardware package, fully integrated, is offered for a total cost of $1,910. The hardware requires simple, plug-in installation and is completely compatible with North Star hardware. The package is designed to operate in any density or double density on 8 inch media in IBM compatible format.

The hardware includes a Tarbell double density controller, dual Shugart 800R drives housed in a cabinet with power supply, fan and all by all necessary cabling. The software, produced by Dynamic Microprocessor Associates, is accompanied by a detailed, comprehensive 80 page manual. An export model strapped for 50 Hz, 220 volts is also offered for an additional $100.

The software alone is $150 from John D. Owens Associates, Inc., 12 Schubert Street, Staten Island, NY 10305. Telephone: (212) 448-6283 or (212) 448-2913. Overseas callers can reach the export desk only at (212) 448-6296 or TWX 710 5882844.

IS YOUR North Star OUT OF SORTS?

INCREASE YOUR BASIC'S SORTING POWER OVER 1800%!

N=SORT is easy to use and will perform sorts on one and two dimensional or string arrays using optional sort keys. For example, to alphabetize A:

10 A$ = "ZYXWVUST" REM Define String 20 SORT A$ LET(NA).

N=SORT interfaces to any release 4 or later North Star Basic and can be yours for only $89 plus $15.00 shipping.

CALL RES. 6% tax. Send check VISA or M/C

North Star Documentation refundable w/HRZ

Software Shops

Systems And Services: Custom software in assembly language & Basic for S100 Z80/8080 CP/M systems. Contact Buz Koenig C/O Systems And Services, P.O.B. 961 Hurst, TX 76053 (817) 268-2938 Eve & W.E.

Theta Labs., Inc.: Software in Basic, Pascal and 8060/8028 assembly. Process control, automatic test equipment, factory timekeeping systems. Consulting or turn-key. Box 20337, Dallas, TX 75220; (214) 241-1090.

Provar Inc.: Specialists in multi-user and networking systems. Currently marketing an MP/M utility package. Experienced in Assembler, Pascal, and Basic. We also design and build process control systems. Contact Mark Winkler, 6217 Kennedy Ave., Hammond, Ind. 46329; (219) 844-0370.

AMERICAN SQUARE COMPUTERS
KIVETT DR • JAMESTOWN NC 27282
(919) 889-4577

SQUARE COMPUTERS
You've asked for an innovative OEM manufacturer of S-100 Modules. Well, this is what we've to say about that.

When all is said and done, the fact remains that we at IDS make some of the most advanced and unique S-100 Modules on the OEM market today.

Our wide range of plug-compatible modules means we can help provide most of the computer applications you can think of. And, perhaps, even some applications you haven't thought of yet. Our manufacturing philosophy is to combine rugged-reliability with high performance to produce some of the most efficient and innovative S-100 Modules you can buy. Anywhere.

But don't just take our word for it: ask for our free 1981 OEM Products Catalogue and see for yourself. In it you'll find specifications for everything from MODEMS-to-Frequency Counters.

So whatever your S-100 computer needs, consider IDS your single problem-solving source. Write or call (703) 661-8442 today for your free 1981 OEM Products Catalogue.

P. O. Box 17269
Dulles International Airport
Washington D.C. 20041

"We're your single source."

IDS INTERNATIONAL DATA SYSTEMS, INC.
If you ever need to crunch numbers, time intervals or events, create sequenced programs, check for power interruption, shuttle information through an RS-232C serial port at up to 19,200 Baud, tell your computer to do something at 02:47:37 AM on March 24th of 1982, service 15 levels of interrupts (8 vectored interrupts from the bus and 7 on-board interrupts), or need EPROM or battery-backup RAM to accomplish any of the above...and want to do all this while fully conforming to all IEEE 696/S-100 standards...System Support 1 is here.

Call (415) 562-0636, 9 AM to 5 PM PST, for the name of a CompuPro retailer near you.

System Support 1 prices: $295 Unkit, $395 A/T, $495 qualified under the Certified System Component high-reliability program. Math processor and ROM/RAM optional at extra cost. Prices shown do not include dealer installation and support services.