For the Advanced Computer User

Micro/Systems Journal™

Structured Programming With The Microsoft M80 Assembler

see pages 26-36

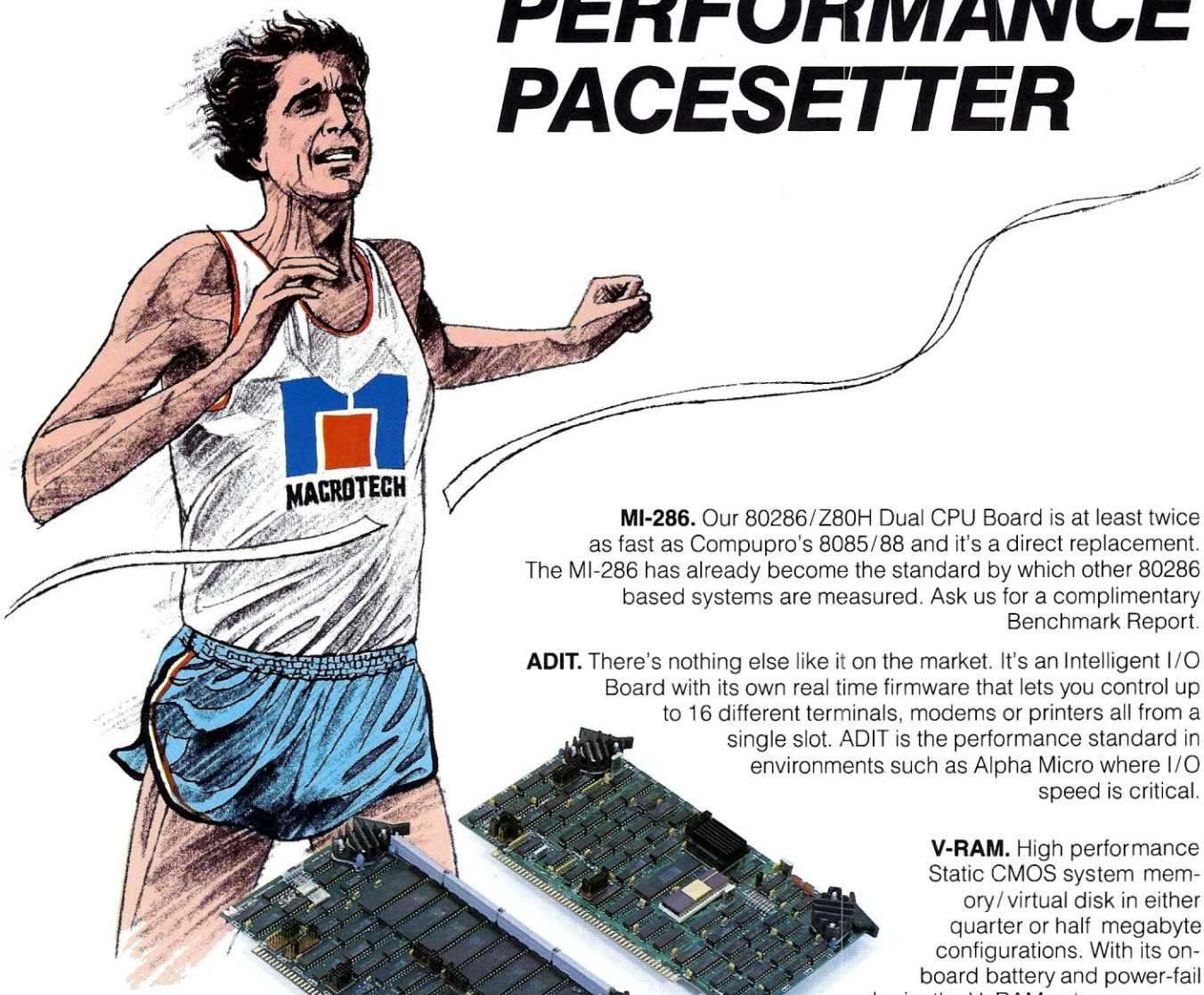
Also in this Issue

Local Variables In Forth 38
Interfacing to MS-DOS, Part II 62
Scientific & Technical Word Processors 40
Concurrent PC-DOS 66
Turbo Pascal V3.0 72

Complete Table of Contents on Page 3

July/August 1985 Vol. 1/No. 3 $4.00 U.S.A.

MACROTECH—STILL THE S-100 PERFORMANCE PACESETTER



MI-286. Our 80286/Z80H Dual CPU Board is at least twice as fast as Compupro's 8085/88 and it's a direct replacement. The MI-286 has already become the standard by which other 80286 based systems are measured. Ask us for a complimentary Benchmark Report.

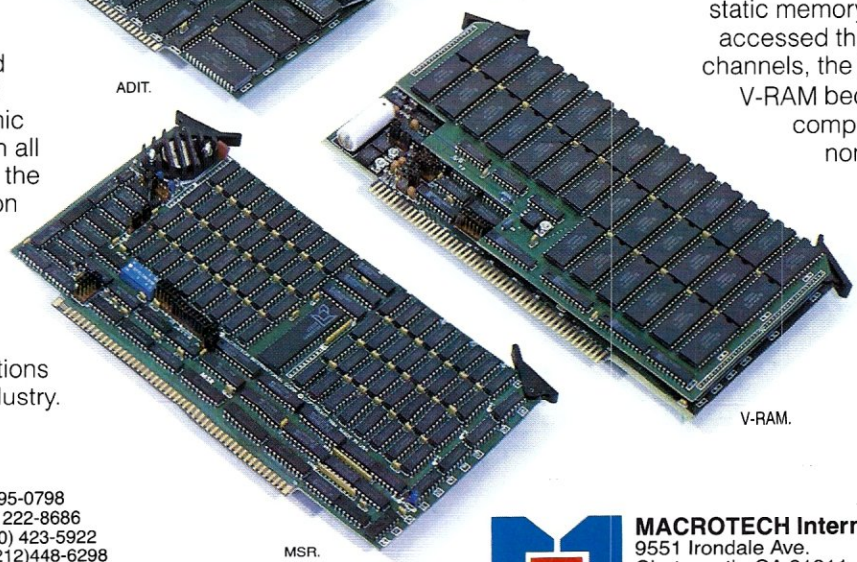
ADIT. There's nothing else like it on the market. It's an Intelligent I/O Board with its own real time firmware that lets you control up to 16 different terminals, modems or printers all from a single slot. ADIT is the performance standard in environments such as Alpha Micro where I/O speed is critical.

V-RAM. High performance Static CMOS system memory/virtual disk in either quarter or half megabyte configurations. With its on-board battery and power-fail logic, the V- RAM sets a new performance standard at conventional static memory prices. When accessed through I/O port channels, the half megabyte V-RAM becomes M Drive compatible with true non-volatile solid-state disk capability.

MI-286.

ADIT.

MSR. High performance and reliability in a memory so fast you won't believe it's a dynamic ram product. Compatible with all popular S-100 environments, the MSR's low power consumption and 120 nanosecond ram devices set a new standard for dynamic memory products. The MSR is available in quarter, half, one and two megabyte configurations at the lowest prices in the industry.



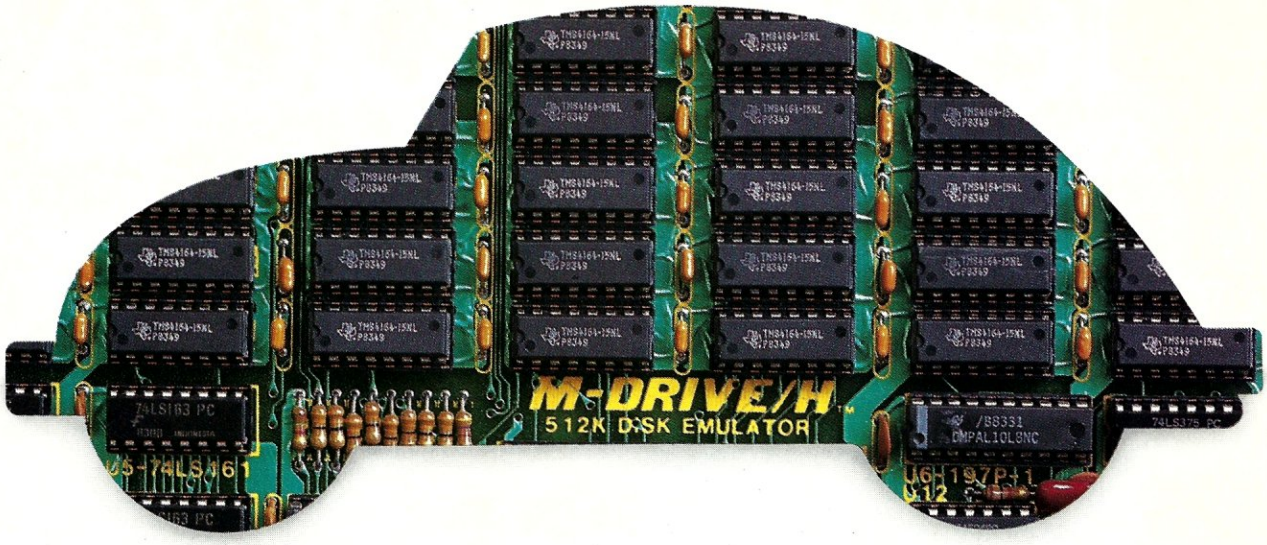
V-RAM.

MSR.

Dealers:
Gifford Computer Systems (415) 895-0798
Custom Computer Technology (800) 222-8686
Priority One Electronics (800) 423-5922
John D. Owens & Associates (212)448-6298
In England; Fulcrum (Europe) Ltd. (0621) 828763

Macrotech dealers also include most Compupro Systems Centers, Heathkit Electronic Centers and Alpha Micro Dealers.

MACROTECH International Corp.
9551 Irondale Ave.
Chatsworth, CA 91311
(800) 824-3181 • in Calif. (818) 700-1501
Telex: 9109970653



You can still buy quality and dependability at a reasonable price.

That's exactly what we drive home at Viasyn. We offer you a whole line of CompuPro® IEEE 696/S-100 Bus boards. Along with single- and multi-user systems compatible with over 3,000 standard business applications, plus a wide variety of scientific and industrial programs. Each one's a value our competition finds hard to beat.

You can choose from a broad range of CPU boards, memory boards, disk controllers, network and interface boards, plus PC Video, in color or black and white. You can mix or match 8-bit and 16-bit software on the same machine or on different machines. And pick just the power and memory you need. Object: to grow and multiply in capabilities, at optimum cost efficiencies. Without sacrificing quality or dependability.

If that's what you're driving for, write us for our new short-form catalog or simply call our toll-free number.

VIASYN™

The CompuPro People

Where Computers Grow

3506 Breakwater Court, Hayward, CA 94545
Call 800/VIASYN-1. In CA, 800/VIASYN-2. TWX: 510-100-3288 VIASYN CORP

CompuPro is a registered trademark of Viasyn Corporation.

Teletek**nology**

One Success After Another.

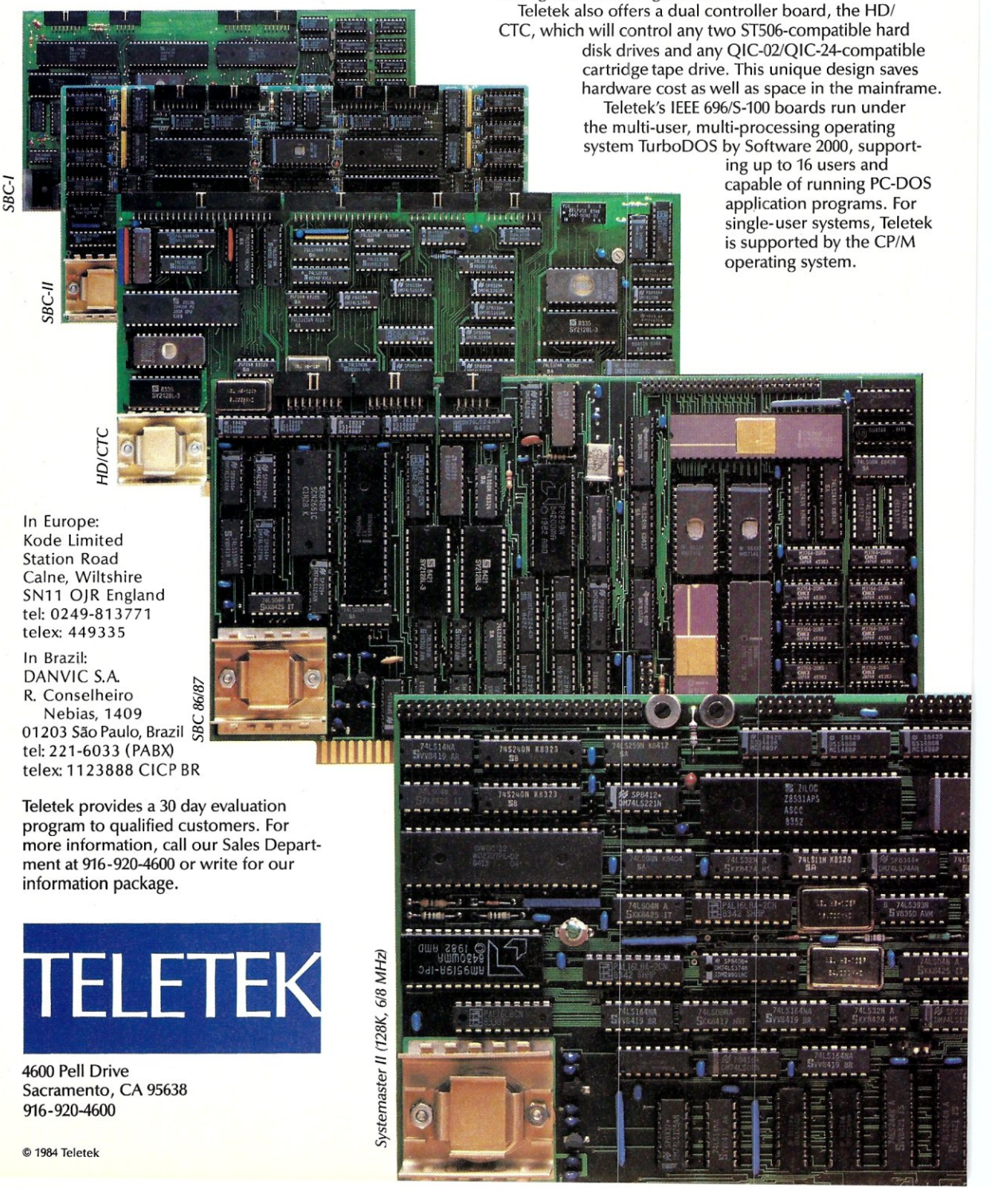
Since 1968 Teletek has been a leader in the design and manufacture of single board computers, controllers, memory boards and interface boards.

Teletek offers five distinct single board computers (SBCs), each with its own unique features, to meet the varied needs of the system integrator. Based on the 8086 16-bit and Z80 8-bit microprocessors, Teletek's SBCs will run at 4, 5, 6, or 8MHz and are available with up to 512K of onboard dynamic RAM. The SBC 86/87 also offers an optional 8087 math coprocessor for numeric intensive applications.

Teletek's Systemaster II provides two RS232C serial ports and two Centronics-compatible parallel ports or may be optionally configured to provide a SCSI interface or an IEEE-488 interface to support many laboratory testing and measuring instruments.

Teletek also offers a dual controller board, the HD/CTC, which will control any two ST506-compatible hard disk drives and any QIC-02/QIC-24-compatible cartridge tape drive. This unique design saves hardware cost as well as space in the mainframe.

Teletek's IEEE 696/S-100 boards run under the multi-user, multi-processing operating system TurboDOS by Software 2000, supporting up to 16 users and capable of running PC-DOS application programs. For single-user systems, Teletek is supported by the CP/M operating system.



In Europe:
Kode Limited
Station Road
Calne, Wiltshire
SN11 OJR England
tel: 0249-813771
telex: 449335

In Brazil:
DANVIC S.A.
R. Conselheiro
Nebias, 1409
01203 São Paulo, Brazil
tel: 221-6033 (PABX)
telex: 1123888 CICP BR

Teletek provides a 30 day evaluation program to qualified customers. For more information, call our Sales Department at 916-920-4600 or write for our information package.

TELETEK

4600 Pell Drive
Sacramento, CA 95638
916-920-4600

© 1984 Teletek

For the Advanced Computer User

Micro/Systems Journal™

July/August 1985
Vol. 1 No. 3

STAFF

Publishers
Sol & Lennie Libes

Editors
Sol & Don Libes

Associate Editor
Susan Libes

Contributing Editors
Andrew Bender
David Carroll
Ian Darwin
Dave Hardy
Ken Jackson
Henry Kee
Steve Leon
Bruce Ratoff
William Wong

Production
Neil Sanford

Advertising
Lennie Libes

Circulation/Administration
Lennie Libes
Diane Overholser

Editorial Assistant
Lenore Melillo

SUBSCRIPTION RATES

1 year	$18
2 years	$32
1 year (Canada & Mexico)	$24
2 years (Canada & Mexico)	$44
1 year (other foreign)	$32
2 years (other foreign)	$58

Make all checks payable in U.S. funds on a U.S. bank, please.

ADVERTISING RATES: Available on request. Call (201)522-9347.

CHANGE OF ADDRESS: Please send old label and new address to: Micro/Systems Journal, Box 1192, Mountainside, NJ 07092.

Micro/Systems Journal is published bi-monthly by Libes Inc., 995 Chimney Ridge, Springfield NJ 07081. Application to mail at Second Class Rates is pending at Springfield, New Jersey and additional mailing offices.

POSTMASTER: Please send address changes to Micro/Systems Journal, P.O.Box 1192, Mountainside NJ 07092.

Copyright © Micro/Systems Journal, a subsidiary of Libes Inc. All rights reserved, reproduction prohibited without permission.

Micro/Systems Journal is a trademark of Libes, Inc.

IN THIS ISSUE

FEATURE ARTICLES

Structured Programming With The Microsoft M80 Assembler 26
Dennis N. Quinn

Local Variables in Forth 38
Thomas W. Reno

Program Interfacing To MS-DOS 62
William G. Wong

PRODUCT REVIEWS

Scientific & Technical Word Processors—Part I 40
Steven Bosak

The Macrotech MI-286 48
Charles Strom

The Slicer 56
William Earnest

Concurrent PC-DOS 66
Michael Guttman & Vincent Mills

Turbo Pascal V3.0 72
David W. Carroll

The Coherent System 74
Les Hancock

DEPARTMENTS

Publishers Page 4
Lennie & Sol Libes

News, Views & Gossip 8
Sol Libes

Mail 10

The SIG/M Public Domain 12
Steve Leon

The PC/Blue Report 14
Hank Kee

Turbo Pascal Corner 18
David W. Carroll

The C Forum 22
Don Libes

The Software Directory 54

The UNIX File 81
Ian F. Darwin

The PC Bus 79
Dave Hardy

AUTHORS: Micro/Systems Journal is always seeking good articles. Please write or call first to see if we are interested in the subject. Please do not send the article unless we ask for it.

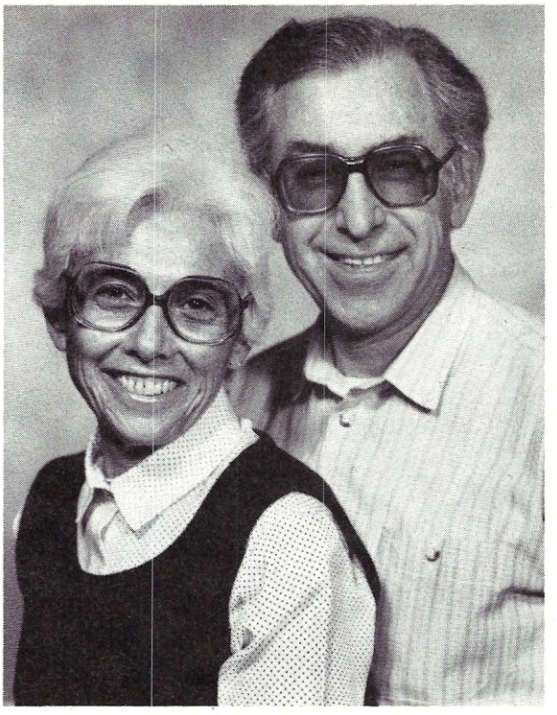
If you are interested in reviewing hardware or software please write telling us your interests, your background and include a sample of your writings.

Send a stamped self-addressed business size envelope for a copy of our Author's Guide.

TRADEMARK ACKNOWLEDGEMENTS

Macintosh	Apple Computer Company
UNIX	AT&T
GEM, CP/M	Digital Research Inc.
Concurrent DOS	Digital Research Inc.
8088, 8086, 80286	Intel Corporation
IBM PC, PC/XT, AT	International Business Machines
MS-DOS, XENIX	Microsoft
GW Basic	Microsoft
68000, 68010,68020	Motorola

PUBLISHERS PAGE



A Report To The Subscriber

We have been in operation now for six months and have put out three issues. It is therefore an opportune time for us to take stock of how well we are doing.

In some ways we are doing well and in others not so well.

WHERE ARE WE DOING WELL?

It is apparent from the reader response that, with few exceptions, our readers are delighted to have the magazine back again. And they like the content.

We have also found that authors are eager to have their articles appear here. There are more articles submitted to us then we have the space to publish and thus we are able to pick the best.

The feedback from our advertisers is also very positive. Virtually every advertiser has signed up for repeat advertising. The rates are low, quality is high and several of our advertisers have told us that advertising in Micro/Systems Journal pays the best returns for them of any publication they advertise in.

We have no advertising sales staff. Yet our advertising has increased with each issue. We have 68 advertisers in this issue, up from 62 and 57 in our previous issues. This increase in advertising has enabled us to increase the size of the magazine from 80 to 88 pages.

We are maintaining an advertising/editorial ratio of 40/60 while most other computer magazines are just the reverse... 60% advertising and only 40% editorial. Further, we attempt to pack in as much editorial as possible by omitting space consuming artwork and white space that convey no information whatsoever. Thus, we feel that we provide as much technical content as magazines twice our physical size ... and more important, we feel we provide a much higher quality of technical content.

WHERE ARE WE NOT DOING WELL?

We now have six months of operation under our belts. And, the hectic time of signing up over 8,000 subscribers in three months is over. We have thus had the opportunity to assess how we are doing financially. The situation is not encouraging.

We are operating a real low-budget operation. We have no rental space (we operate out of our home). We have no full time employees; only a little bit of part-time help. We have not drawn any salary. We are resigned to a labor-of-love philosophy.

But there are expenses that must be incurred ... printing, typesetting, postage, telephone, etc. It is very expensive getting this operation up off the ground.

It is apparent to us that we are a long way from breaking even. The basic problem appears to be in reaching the "economy of scale", the "critical mass", necessary to make this a possible endeavor. Our circulation is still only 17% of the old Microsystems and advertising less than 40%. Most of our expenses are overhead expenses. Hence, we must grow in circulation and advertising income to cover these expenses.

ASSESSING THE SITUATION

We are committed to investing a year's worth of our labors to this magazine. We are cutting expenses wherever we can, without sacrificing quality.

Therefore, we are appealing to you, our readers, to help us make this magazine live again. Please tell your friends to subscribe. If you belong to a User Group please promote it to the group's members (call or write us as to how you can help).

Please tell your local computer dealers, college and technical bookstores to carry the magazine on their racks.

Please do not ask to be billed for subscriptions — we have found that less than half of the bill-me's actually pay and the mailing of issues and billing expense makes this a losing proposition.

You can help in other ways as well. If you move, please notify us promptly. You would not belive what it costs us for returned torn off covers from the Post Office. The P.O. will not forward second class mail. We then have the additional expense of mailing a second issue to the subscriber. So please, if you are going to move notify us well before hand and tell your P.O that you want to have your magazines forwarded.

Also, when renewal comes around, please renew promptly so that we don't have to send you additional notices. Or better yet, renew early. You will notice the date of your last issue in the upper right corner of your mail label. Incidentally, one of our advertisers, Competitive Edge, is offering a free subscription to Micro/Systems Journal, or an extension of an existing subscription, when you purchase selected products from them. Check their ad for details.

You can also help us in another and very important way. Please tell advertisers that you saw their ads in Micro/Systems Journal and urge others to advertise.

With your help, we think there is a good chance we can make it.

—Lennie & Sol Libes

Quality S-100 Products

Now Available! FULCRUM'S New...

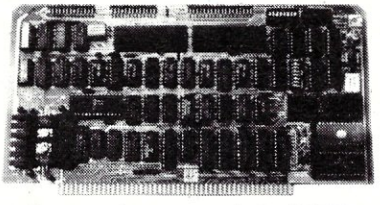
256K RAM CARD

- 8/16 Bit Transfers
- 24 Bit Extended Addressing

$595

Designed For Speed & Flexibility.

MPUZ CPU

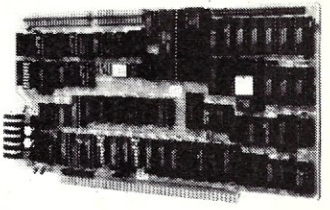


This NEW MPUZ CPU utilizes the Z-80 8MHz uPas a basis for its 8MHz CPU for S-100 systems, and has been carefully designed to meet the requirements of the IEEE - 696 standard. The quality and performance this CPU provides is rarely found in S-100 products, and you can see why... ***only $299***

- 4 or 8MHz clock rate
- Two RS-232 serial ports
- Centronics printer ports
- Real time clock with battery back-up
- Vectored interrups to any block location in memory
- Programmable timer
- ROM monitor
- Power on Jump
- On board wait states
- 2K of RAM space
- 24-bit extended addressing
- Latched Status
- Front panel compatibility
- MPM support

Best Value In Disk Controllers.

OMNIDISK



Now the FULCRUM OMNIDISK offers S-100 systems users a unique marriage of component compatability and technological innovation. These together produce features not found in any conventional disk controllers made today. See for yourself what tomorrow looks like... ***only $299***

- Simultaneous support of both 5¼'' and 8'' floppy disks and hard disks
- Complete 24 bit DMA
- Power on boot for 5¼'' and 8'' floppy and hard disks
- Power on boot PROM
- On board de-blocking to save RAM space over BIOS
- Interfaces with the WD 1001® hard disk controller
- Supports 13 devices simulatneously
- Full track buffer allows controller to recall entire track
- DMA'S at 10 MHz
- Supports MS DOS
- 10K on board buffer saves two K of TPA

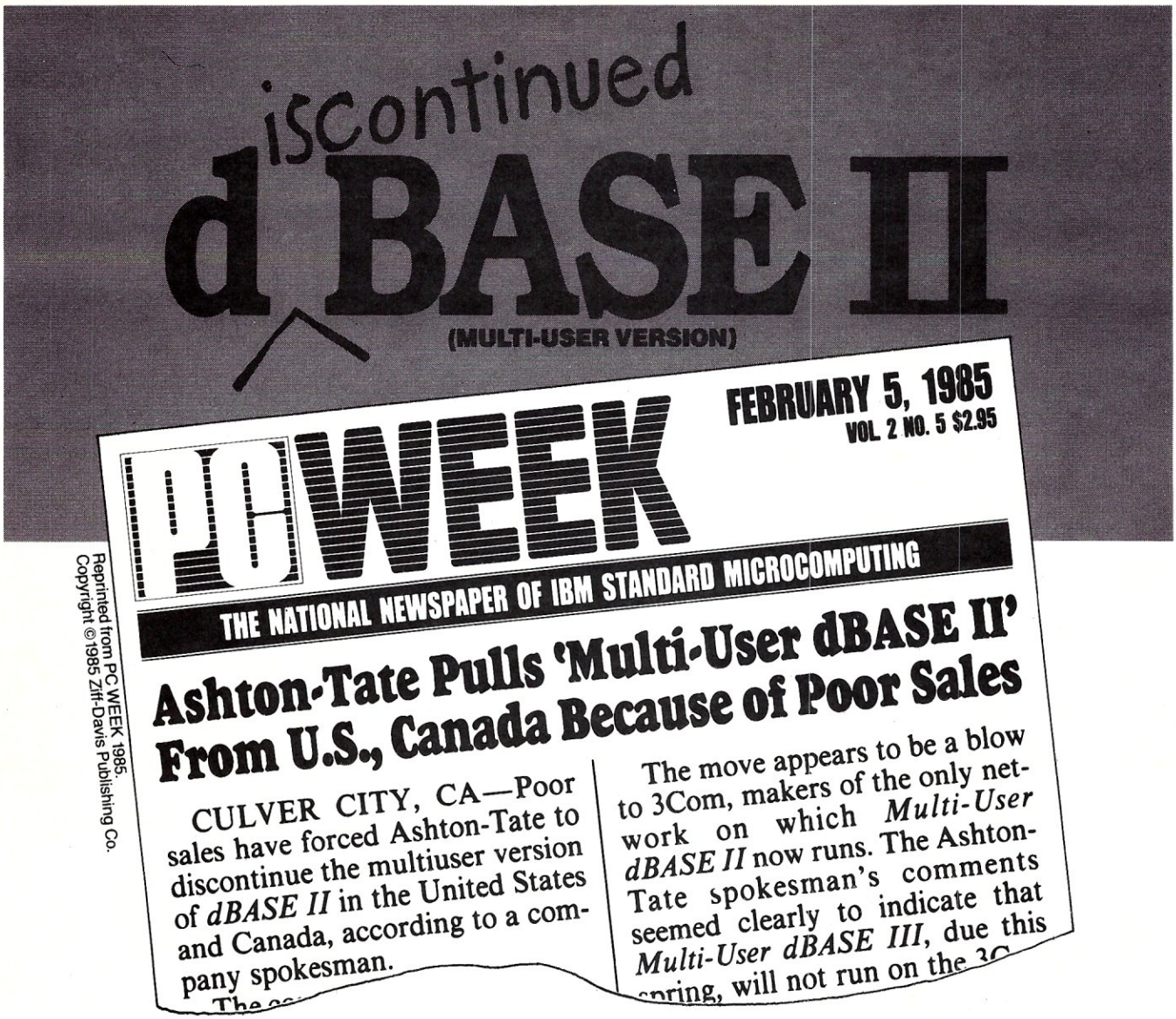
So before you buy another S-100 component, call or write for our *FREE* catalog. And see how your system can benefit from the FULCRUM difference.

*CP/M *2.2 configured for OMNIDISK $60. *Trade mark of Digital Research. FREE U.P.S. ground shipping on prepaid orders. Shipping is added to VISA, M/C, and C.O.D. orders. CA residents, please add sales tax.

FULCRUM COMPUTER PRODUCTS

707/433-0202 459 Allan Court, Healdsburg, CA 95448,

Also in FULCRUM'S Family: OMNIRAM 64K memory board. Serial I/O 2-2 & Video I/O Interface boards, Relay board, I-8080, 8015 and 8035 main frames with 21 slot mother boards, CPA front panel and A/D board.



NOW, THE RIGHT CHOICE IS EASIER THAN EVER.

DATAFLEX™

THE TRUE MULTI-USER APPLICATION DEVELOPMENT DATA BASE

DATA ACCESS CORPORATION
8525 SW 129 Terrace, Miami, FL 33156-6565 (305) 238-0012
Telex 469021 DATA ACCESS CI

Compatible with MSDOS, PC-DOS, CP/M, CP/M-86, MP/M-86, TurboDOS, Novell Sharenet, PC-Net, Molecular N-Star, Micromation M/Net, Action DPC/OS, OMNINET, IBM PC w/Corvus and OSM Muse.

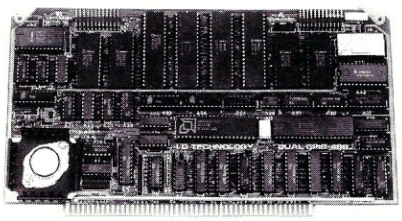
MSDOS is a trademark of Microsoft. CP/M and MP/M are trademarks of Digital Research. DataFlex and FlexKeys are trademarks of Data Access Corp.

dBASE II is a trademark of Ashton-Tate

INPUT/OUTPUT TECHNOLOGY, INC.

25327 Avenue Stanford, Unit 113, Valencia, CA 91355 • (805) 257-1000

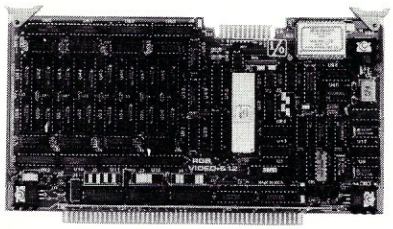
Uncompromising Additions to your S-100/IEEE-696 BUS



DUAL GPIB-488 INTERFACE BOARD

A Stand-Alone, Independently Controlled Dual Channel IEEE-488 I/O Processor. Interface Activity Modes for Controller-in-Charge, Controller Assigned or Terminal Bus Slave, and all Interface Functions are handled transparent to Host System CPU through an on-board CPU and DMA controller. User Friendly operation.

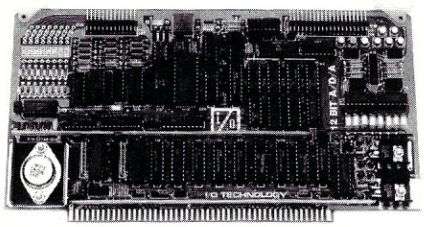
A&T, P/N 52748-800-102



RGB COLOR GRAPHICS BOARD

Programmable resolution up to 512 x 512 pixels with 4 local video planes and on-board graphics processor. Color mapper allows 16 colors from a palette of 4096. Light pen input. Plus more ...

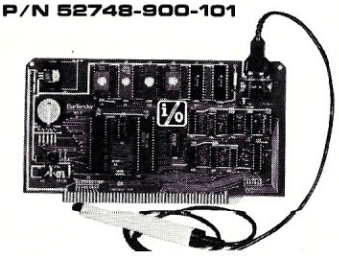
A&T, P/N 52748-300-101



12-BIT A-D-A CONVERTER BOARD

8 Channel A-D: 12 microsec. Conversion, 50KHz Sample Rate, Programmable Gains, Offset and Diff./Single Modes.
8 Channel D-A: 2 microsec. Settling, Bipolar V or Unipolar I Output. Programmable Reference levels, Dual-Ported Channel Refresh RAM. **16/8-Bit Data Transfers** via I/O or Memory Mapped

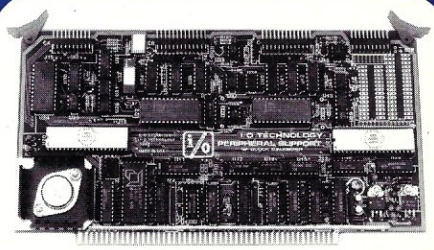
A&T, P/N 52748-900-101



BAR CODE PROCESSOR BOARD

The BarTender is a stand-alone I/O Processor that reads and prints most common Bar Codes. Includes bi-directional reading, wand interface, clock/calendar with battery. Extensive documentation and software.

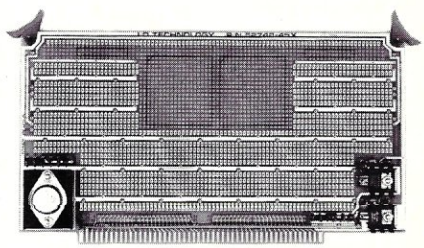
A&T,52748-500-101 Without Wand
A&T,52748-500-201 With Wand



PERIPHERAL SUPPORT BOARD

Two Serial SYNC/ASYNC Ports with RS-232, TTL or Current Loop Outputs, three 8-Bit Parallel Ports, three Timers, Real Time Clock/Calendar and Response Programmable Interrupt Controller. Small Proto Area with +5 and ±12v.

A&T, P/N 52748-150-101



MULTI-PURPOSE PROTOTYPING KIT

Industrial Quality with Plated-Thru holes for Wire-Wrap or Solder projects. Complete with +5, ±12v Regulators, Bus Bar, Filter Capacitors, and Manual.

P/N 52748-450

ALSO AVAILABLE: MULTI-FUNCTION I/O BOARD, SMART PROTOTYPING KIT, 128Kx8/64Kx16 STATIC RAM MODULE

CIRCLE 10 ON READER SERVICE CARD

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

News, Gossip Views &

by Sol Libes

GOSSIP & RUMORS

Kudos to **Altos**, the first company to introduce a Motorola 68020-based system, a true 32-bitter. Their Altos 3068 runs Unix System V serving up to 30 users.

Ashton-Tate has stopped selling its multi-user version of DBase II in the U.S. and Canada although it is still shipping it in international markets. The reason.....low sales. The software, introduced in May '84 was designed to work with the 3/Com LAN. A-T also disclosed it has dropped development work on a Turbo-DOS version.

Western Digital and **Xebec**, the two leading makers of hard disk controller systems are sampling new controllers for 3.5 drives that reduce the parts count to one quarter of current controller cards and require only one connecting cable to the drive instead of the current two. This is expected to signicantly reduce the cost of hard disk systems.

ATARI, who had promised to show a prototype of their new 32-bit personal computer running Unix, at the Hanover Fair in April, failed to do so. In the meantime, **Mark Williams Co.** disclosed that **Commodore** will use their Unix-like Coherent operating system on their new C900 personal computer. The entire operating system has been squeezed into 2Mbytes of disk space and supports up to seven users. The C900 will have a 1024 by 800 pixel bit mapped display.

Clone makers are switching from the PC to the AT marketplace. Expect **Tandy, Hewlett-Packard, Texas Instruments, Zenith, Wang, Honeywell, Philips, Siemens, Ericsson**, and **AT&T** to shortly introduce AT-compatibles. There are even rumors that Apple is seriously considering producing one. Most are expected to make their display circuitry compatible with IBM's Enhanced Graphics Adapter (EGA). Chips and Technologies, a custom IC maker in San Jose CA is reported attempting to integrate the EGA from 150 down to 23 ICs.

AT&T has reportedly sold only 30,000 6300 systems last year. This puts them in the same category as Victor Technology, Columbia Data and Vector Graphics!

Sun Microsystems is rumored to be developing an under $5,000 version of their popular graphics workstation. Their current unit is just over $10,000.

MICROSOFT SET TO INTRODUCE MS-DOS 4.0

Microsoft is expected, late this year, to release MS-DOS 4.0, adding multitasking and the ability to access a virtual memory in excess of 640K. The addition of multitasking is expected to improve the operating speed of Microsoft's WINDOWS environment. Currently, the only way to get multitasking on PC-compatible machines is with one of the Unix implementions (e.g. Xenix, Coherent, etc.) or Digital Research's Concurrent DOS whose latest version (4.1) also includes the GEM Mac-like user interface.

Version 5.0, designed specifically for the 80286 is expected early next year. It should execute programs in the 80286 protected virtual-address mode. DRI has already begun shipping its 286 version of Concurrent-DOS to OEMs. However, it has been reported that current 80286 production devices contain a bug which prevents Concurrent-DOS 286 from implementing the protected virtual-address mode while maintaining MS-DOS compatibility. Intel is aware of the problem and expects to correct it shortly.

INTEL UNWRAPS 386

Intel will shortly formally announce its new, long rumored, 32-bit microprocessor, the 80386. Intel currently finds itself running behind National Semiconductor (already shipping 32032 in production quantities) and Motorola (in limited 68020 production) and AT&T (shipping their Bellmac-32 chip set to OEMs). Intel is not expected to start samples until this fall with production to start next year.

The 386 will have more than twice the devices housed in the 286 and be two to three times faster. It should have on-chip memory management with both protection and paging features to work with up to a 4-gigabyte physical memory and 64-terabytes virtual memory.

Intel has halted all development work on its previous 32-bit micro chip set, the 432 (although High Integrity Systems, in England is continuing a 432 development effort). Intel is reported working on a new high-end 32-bit chip set, code name P-7, believed to be a less complex implementation of the 432, optimized for real-time work and easier to use than the 432.

The 432 was actually the first 32-bit microprocessor to be introduced (way back in '81). It contained a very innovative architecture but its performance proved disappointing. Intel introduced an improved version in '83 and added fault-tolerant features. But by that time Motorola and National had entered the market.

Look for at least one S-100 manufacturer to release an S-100 CPU card using one of the new 32-bit microprocessors before year-end. Expect more next year. We would be interested in hearing from readers as to how they feel the S-100/IEEE-696 bus standard should be upgraded to accommodate these 32-bit CPUs.

Z80K & Z800, WHERE ARTH THOU?

Zilog has again pushed back introducing its Z80,000, 32-bit and Z800 super 8/16-bit (Z80 compatible) microprocessors, to the spring of '86. If they do manage to ship samples, then it will be a full six years since Zilog announced they were developing the devices.

Zilog, owned by Exxon, has shown a profit in only one year of its 11 year life and recently cut close to 400 people from its payroll. Although a microprocessor pioneer with its Z80 and Z8000 8- and 16-bit micrprocessors, it has suffered from a reliance on microprocessor manufacturing while competitors, such as Intel, Motorola and National, have used microprocessors as

loss leaders to sell memory and other types of ICs.

USER GROUP & PUBLIC DOMAIN NEWS

The Apple II CP/M Library, Box 477, Kulpsville PA 19443 is furnishing copies of the entire SIG/M Public Domain Software library on 5.25 Apple disks. Send a stamped self-addressed business size envelope to receive an abbreviated catalog. Send $3 for a detailed printed catalog or $10 for catalog on disk.

CLERGY KUG is a user group serving Ministers, Priests and Rabbis using KayPro equipment. They have a 1200 program library for the KayPro which they share and also publish a small newsletter. For info contact Elmer F. Little Jr, 3868 Centorbi Court, Florissant MO 63034.

Users wishing SIG/M public domain software on Morrow MD-3 disks (single or double sided) can obtain them ($6 volume) from G.L. Harris, Box 797, Metuchen NJ 08840.

The Amateur Computer Group of New Jersey, the club that sponsors the SIG/M and PC/Blue public domain software libraries has started a public domain software library for the Apple Macintosh computer. The library already contains several volumes. For information send $2 and a stamped self-addressed business size envelope to: MAC-UG, c/o Keith Sproul, 698 Magnolia Rd, No. Brunswick NJ 08902.

RANDOM BITS

Novix, Cupertino CA has introduced a 16-bit microprocessor that directly executes the FORTH language..... **Tarbell Electronics**, Carson CA, is shipping an S-100 CPU card with 80186 and Z-80 processors that can directly boot PC-DOS.....Computer Language magazine will conduct a C-Language seminar/workshop, September 16-18 in Cambridge MA. For more info, call (415)957-9353 or write CL Publications, 131 Townsend St, San Francisco, CA 94107.....**Digital Research** has established a special User Group Forum on CompuServe (type Go PCS-13) and has a number of DRI engineers online as well as extensive data libraries and technical info.....**Intel** is offering a system (called the Intel Personal Super Computer, or IPSC) consisting of up to 128 IBM AT boards, each containing 512K of ram and an 80287 math coprocessor. Total memory is 16Mbytes and the CPUs operate in parallel. They promise performance equal to that of a Cray-1 and are recommending the system for artifical applications at a fraction of the cost and size of a Cray-1.

Z sets you FREE!

Z — yes! Synergistic combination of ZCPR3 and ZRDOS2 produces flexible state-of-the-art Z80 operating system with tremendous productivity features.

Z-System consists of software modules, dynamic loading segments, and tools permitting optimum computer usage ranging from production program development to turnkey, password-controlled, end-user installations. Facilities include: multiple commands per line, file search paths, named directories, I/O redirection, command flow control, screen-oriented menu generators, complete housekeeping file and directory management, shells, alias (scripts) and nested-alias generation, and complete online help.

Seventy-six support utilities, five tool packages, and two application programs available now! Fully upward compatible with CP/M-80.

Z can now be purchased as auto-install program (Z-Com) or as manual-install ZCPR3 with semi-auto install ZRDOS package (Z-System). Our latest versions, to be released this year, support Zilog Z800 and Hitachi HD62801/64180 high-technology chips, chips run existing 8080 and Z80 programs!

Echelon eight-bit operating systems written in Assembly Language, using linkable macro subroutine libraries, offer performance paralleling best single-user 16/32-bit microcomputer systems.

1. **Z-Com** Full-up Z Operating System with input/output redirection running under CP/M-80, online command and utility documentation and help system **$219.95**

2. **Z-System** Manual-install ZCPR3 and ZRDOS2, easily tailored by programmer to custom needs; source code to core and utilities; similar to Item 1 ..... **$199.95**

3. **Z-Tools** Four software development system packages permitting advanced, structured program design, macro relocating assembler, linking loader, librarian, cross-reference generator, debugger, mnemonic and pseudo-op translators, and interactive disassembler. Super $315.00 package value **$200.00**

4. **DSD** Dynamic Screen Debugger offers high-level features never before found in microcomputers; simultaneous display of dual-memory segments, stack, cpu states, and flags, with software In-Circuit-Emulation **$149.00**

5. **The Libraries** Linkable ZCPR3 libraries (Vlib, Z3lib, and Syslib3) of over 400 subroutines used for Assembly Language program writing. Simplifies structured, efficient code production; online help system and full source code provided **$45.00**
Syslib3 alone **$29.00**

6. **Term3** New generation communication program permits menu control of computer/modem operations between operator and time-share services, bulletin-boards and other remote computer systems; auto-answer to command-line prompt **$99.00**

7. **Discat** Fancy file and disk catalog program running under Z-System, menu driven and easily customized by operator **$49.00**

Fortnighter newsletter, 24-hour BBS Z-Node System keep Z users informed of microcomputer happenings. Write or call for brochure or order now! State disk format desired; add $3.00 shipping & handling; Californians please add 6-1/2% sales tax. Visa/MC, check, money or purchase order accepted. (Program names are trademarks of their respective owners.)

Echelon, Inc.

101 First Street • Los Altos, California 94022 • 415/948-3820

there is mail.....

Dear Lennie:

Since you are dealing with MICROSYSTEMS previous subscribers, I thought you might be interested in passing along the following information.

I had a 3 year sub to MICROSYSTEMS when Z-D folded it, and sent me a PC TECH JOURNAL instead. I have a North Star system; what would I want with PC TECH JOURNAL?

I called and wrote with no success. Finally, I got angry enough and located a way of stopping PC TECH JOURNAL *and* getting a full refund from the last issue of MICROSYSTEMS.

First, Call 1-800-525-0643 and request they instantly stop PC TECH JOURNAL.

Second, Call Ro Carbone at (212)503-5396 and request a full refund.

It worked for me.

Good luck to your readers.

T. L. F.

Dear Mr. Libes:

We wish you success in this venture. Along with many other long suffering users we are fed up with the machinations of so called professional management that ends up in the demise of good ventures because it didn't make them instant millionaires. It is a signficiant factor in the economic malaise in this country as professional management seems to think that they are the owners instead of the stockholders.

Charles C Wright
San Ysidro CA

Dear Sol:

Glad to hear you are back again and all the very best to you and the new venture. PLEASE lets have a publication that does more than blab incesantly about MSDOS and the IBM backward step standard and looks after us die hards with our *old* or *Ancient* S100 systems.

John T Linnell
Etobicoke Canada

The S-100 marketplace is actually doing quite well as it keeps at the forefront of the changing technology with high speed 16-bit systems and powerful hardware that enable users to build multi-user and multi-processing systems. One can pick from a wide variety of 8- and 16-bit processors, and in the near future 32-bit processors as well. Also, keep in mind that one can run many up-to-date operating systems on S-100 equipment; systems such as CP/M, Turbo-DOS, Concurrent-DOS, MS-DOS and Unix. Thus, I expect S-100 to be with us for quite some time to come.

CP/M-80, however, I think is another matter. By today's standards it is an ''old'' operating system and lacking the power and features needed to take the best advantage of today's powerful hardware systems. ZCPR3, as well as some other utilities have brought some of the newer features to CP/M-80. However, the fact that so few software developers are bringing out new software for CP/M-80 indicates that users should start looking at other operating systems that allow them to run the current generation of applications programs on their S-100 systems. We are attempting to publish articles to help our readers in this effort.

the editor

Dear Lennie & Sol:

It was a pleasure to receive the premiere issue of Micro/Systems Journal. I was pleased to see both the superb editorial content and a healthy compliment of advertising.

I hope the return to *garage* publishing is as pleasant for you both as the return of Microsystems was to me. You will shortly recieve both a company and a personal subscription check from me. Microsystems was the only 'micro' book I was subscribing to when I joined ZD and Micro/Systems Journal is the only one I pay for myself.

I wish you both the best of luck.

Jonathan D. Lazarus, Vice President
Ziff-Davis Publishing Company

Dear Sol:

I'm hesitantly backing your play. As a hardware kit-builder and software hacker, I feel that I really got the shaft on your previous venture.

I'm looking for a good S-100, hardware-oriented magazine. Somewhere along the line, the idea was foisted upon the world that S-100 and CP/M were synonomous. They AIN'T! I have recently purchased a Digital Research Corp. Big Board.

Your front-page headline proclaiming ''Bringing Up CP/M-Plus'' really turns me off. What I want to see are articles about putting other CPU's on the S-100 bus, or else redefining the dammed thing to get a 32-bit data path to accommodate either the NS32032 or MC68020 (along with 2-4Mbytes of unsegmented and unmapped linear memory. Intel and Zilog can go soak their collective heads!

Peter A Thiessen
Mercer Island WA

Dear Sol:

I do not want to read any publication that discusses, addresses, considers, reviews or otherwise mentions IBM, IBM-PC, XT, AT, etc. While I do not believe in censorship, I do not wish to contribute in any way to the acknowledgement of IBM except to damm them in every way.

Bernard S Gorman
Eatontown NJ

Readers will find the name of IBM in several places in this issue, in previous issues and no doubt in future issues. There is no doubt that IBM has a very dominant position in the microcomputer marketplace. This fact cannot be ignored. To do so is to act like an ostrich with the resulting consequences. The PC, XT and AT hardware configurations and bus systems have become defacto standards as dozens of other manufacturers have adopted this architecture.

We also feel that IBM is getting more than enough exposure in other publications and therefore we do not plan to promote IBM products here. However, we feel that many of the IBM-compatible products are being ignored in the computer press and deserve to be brought to the attention of our readers and therefore we will be carrying articles on these products.

We feel that the PC-bus, MS-DOS and Concurrent-DOS are today generic to the industry and we expect to treat them as such. We do not intend to review any IBM products or products that run exclusively on IBM systems. However, products that are compatible with defacto standard PC hardware and MS-DOS will be discussed here. To not do so would be hiding our heads in the sand and trying to ignore many very worthwhile products that are out there.

Dear Sol:

I would like to report an error in David Brewer's article in issue #1. Please refer to subroutine ZIP, in listing 1 on page 48 (FINDEM program). The subroutine works fine for digits 0 thru 9 but is in error for digits A-F Hex. The

binary number 1010 (decimal 10) is transformed to 40H, rather than 41H (ASCII for "A"), so the final value must be incremented to produce digits A-F.
I have altered the subroutine as follows:

```
ZIP:
        AND 0FH
        ADD A,90H
        DAA
        ADC A,40H
        DAA
;FOLLOWING THREE
 INSTRUCTIONS ADDED
        CP 40H
        JR C,Z1
        INC A

;LABEL ADDED TO NEXT
 INSTRUCTION
Z1
        CALL SENDA
        RET
```

Charles Prohaska
Dingman's Ferry PA

Dear Mr. Libes:

I received my first copy of your new journal in the mail today, and I was happy to note that your journal has maintained high standards of both quality and usefullness. I own an S-100 system, and for a number of reasons, I'm glad I got aboard the S-100 bus, IBM popularity et al!

One statement you made in the "Editor's Page" really got my attention in one big hurry! You said that "...four largest suppliers of S-100 systems are using Unix." I guess you meant Cromenco (?), Altos, and ?? Who were you referring to? I have an MS-DOS system (Seattle CPU, Tarbell controller, etc.) and I'm very interested in migrating to Unix. I'd like to keep my Microsoft compiler and other MS-DOS tools and run them in a Unix environment!

The closest I've come so far is: Dynacomp Computer Systems, Vancouver B.C., (604)872-7737 and Mississauga, ONT. (416)826-8002, which runs on a CompuPro 8/16E.

Anthony B Price
Burbank CA 91501

We expect to publish a review of the Dynacomp Unix implementation of the CompuPro 8/16 in the near future. The S-100 vendors who currently are shipping Unix systems include Cromemco, Morrow Designs and Dual Systems. Also, I expect that CompuPro and Macrotech will begin shipping S-100 Unix implemenations before year-end.

CP/M-80 C Programmers . . .

Save time

. . . with the BDS C Compiler. Compile, link and execute *faster* than you ever thought possible!

If you're a C language programmer whose patience is wearing thin, who wants to spend your valuable time *programming* instead of twiddling your thumbs waiting for slow compilers, who just wants to work *fast,* then it's time you programmed with the BDS C Compiler.

BDS C is designed for CP/M-80 and provides users with quick, clean software development with emphasis on systems programming. BDS C features include:

- Ultra-fast compilation, linkage and execution that produce directly executable 8080/Z80 CP/M command files.
- A comprehensive debugger that traces program execution and interactively displays both local and external variables by name and proper type.
- Dynamic overlays that allow for run-time segmentation of programs too large to fit into memory.
- A 120-function library written in both C and assembly language with full source code.

Plus . . .

- A thorough, easy-to-read, 181-page user's manual complete with tutorials, hints, error messages and an easy-to-use index — it's the perfect manual for the beginner *and* the seasoned professional.
- An attractive selection of sample programs, including MODEM-compatible telecommunications, CP/M system utilities, games and more.
- A nationwide BDS C User's Group ($10 membership fee — application included with package) that offers a newsletter, BDS C updates and access to public domain C utilities.

Reviewers everywhere have praised BDS C for its elegant operation and optimal use of CP/M resources. Above all, BDS C has been hailed for it's remarkable *speed.*

"I recommend both the language and the implementation by BDS very highly."
Tim Pugh, Jr.
in *Infoworld*

"Performance: *Excellent.*
Documentation: *Excellent.*
Ease of Use: *Excellent.*"
InfoWorld
Software Report Card

". . . a superior buy . . ."
Van Court Hare
in *Lifelines/The Software Magazine*

BYTE Magazine placed BDS C ahead of all other 8080/Z80 C compilers tested for fastest object-code execution with all available speed-up options in use. In addition, BDS C's speed of compilation was almost *twice* as fast as its closet competitor (benchmark for this test was the Sieve of Eratosthenes).

Don't waste another minute on a slow language processor. Order your BDS C Compiler today!

Complete Package (two 8" SSDD disks, 181-page manual): **$150**
Free shipping on prepaid orders inside USA.
VISA/MC, COD's, rush orders accepted.
Call for information on other disk formats.

BDS Software, Inc.

BDS C is designed for use with CP/M-80 operating systems, version 2.2. or higher. It is not currently available for CP/M-86 or MS-DOS.

BD Software, Inc.
P.O. Box 2368
Cambridge, MA 02238
(617) 576-3828

In the SIG/M Public Domain

by Stephen M. Leon

Steve Leon is the SIG/M Disk Editor. In other words, he is the person who assembles, compiles and edits all the of the SIG/M public domain software disks. Thus, he speaks with the greatest authority as to what is going on in the SIG/M public domain software area.

Is it true that only people on the wrong side of sanity spend weeks writing and debugging software only to donate it to the public domain? What normal person tolerates telephone calls in the middle of the night from strangers with questions that range from exceedingly clever to patently absurd? At least some of the local "breathers" and cranks take a couple of hours off at night for rest - but when its 3 AM your time and 6 PM at the caller's location, he gets annoyed if you don't respond promptly! Besides that, the person on the other end of the phone is not a crank caller. He or she is someone YOU promised to help when you put your name and phone number in the documentation.

If there is a word to describe such a person, that word might be *teacher*. Over the past ten years personal computing has been transformed from a tool of the hackers to a near appliance for the non-technical user. Moreover, while many of the former supporters of public domain have gone *commercial*, there is still a significant group of people who continue to support the public and its needs, without compensation, often at personal sacrifice.

Nowhere are these teachers more evident than in the group that continues to support CP/M. Much of the PC/MS-DOS public domain software support comes not from people who are earnestly trying to help, but from the vultures of 'freeware' whose motto is *If you like it - send me money*.

Personal computing is worldwide - and SIG/M functions on an international basis as a rallying point for the dedicated supporters of CP/M. In addition, there is a worldwide forum where much of this group meets. The CP/M SIG on CompuServe is the classroom of CP/M personal computing. Each of us probably gets six invitations a week to spend hundreds of dollars at a seminar to learn the truth about something in computing, but reading the messages on the CompuServe CP/M SIG will teach you more than any seminar. Where else can one get together with a faculty of Bill Bolton, Ward Christensen, Irv Hoff, Pete Holsberg, Sigi Kluger, Gary Novosielski, Harry Van Tassell, etc. etc., and get answers that are not hems and haws, but the right stuff. If the message board is a great way to get expert help fast, an even more interesting session is the on-line conference held Sunday evenings at 9:00 Eastern.

We persuaded one of the CompuServe regulars, Sigi Kluger, to donate a volume of his software to the SIG/M library. In the past, we have placed some of his works on miscellaneous volumes, but this time we were fortunate enough to obtain from him what is now Volume 226 - the Sigi Kluger Utilities.

Utilities are programs or modules you never notice until you *discover* them. A week after Sigi's disk arrived, it became necessary to cut the baud rate on my CompuPro system from 19,200 cps to 1,200. His SETBAUD went into the startup file and the problem was solved. Sure there were plenty of other ways to solve the problem, but with Sigi's utility handy, why not use it?

Get yourself a SIG/M catalog or order the catalog disk (Volume 00) and thumb through all of the utilities that are available. Do you use VFILER? You don't even know what it is? Then you use SWEEP? What about FINDBAD or SD? At least you use DU and UNERA!

If these names mean nothing then your CP/M-based computer is running at quarter speed. Rather than tell you what they are - get your checkbook and send a $10.00 check to SIG/M Box 97, Iselin, NJ 08830. Tell us whether you are running CP/M-80 or CP/M-86, and whether or not you can read an 8" single density disk. If you can't, notwithstanding the fact that SIG/M does not directly support anything but 8" SSSD, we will do our best to get you a 5" disk in the format you specify. SIG/M will send you a catalog and a sample disk as soon as possible. The extra dollar added to the charge is because this sampler volume is not a regular release volume.

Jim Van Zandt has done an update to Ron Cain's Small C. His version with floating point math is on newly released Volume 224. A word of caution, it runs only on a Z80. A CP/M-86 version of Small C is available on SIG/M Volume 149 (but without the floating point).

On newly released Volume 225, Paul Gans, a regular contributor to the SIG/M library, has come up with a program for MX-80 owners. It allows you to run Wordstar and have a second character set. Included in Paul's package is a scientific font for the MX-80. Also on 225 is a series of biochemical games and simulations from Henry R. Bungay of RPI. Professor Bungay's works may not have universal appeal, but that too is one of the virtues of the SIG/M library. We don't try to appeal to everyone on every volume. We do try to have something for everyone at some time or another. However, no commercial distributor could ever release an eight volume set of the Yale Catalog of Bright Stars (Volumes 31 through 38). Funny thing, we are constantly getting requests for this series. This is a clear demonstration of the need for public domain distribution of even limited appeal software.

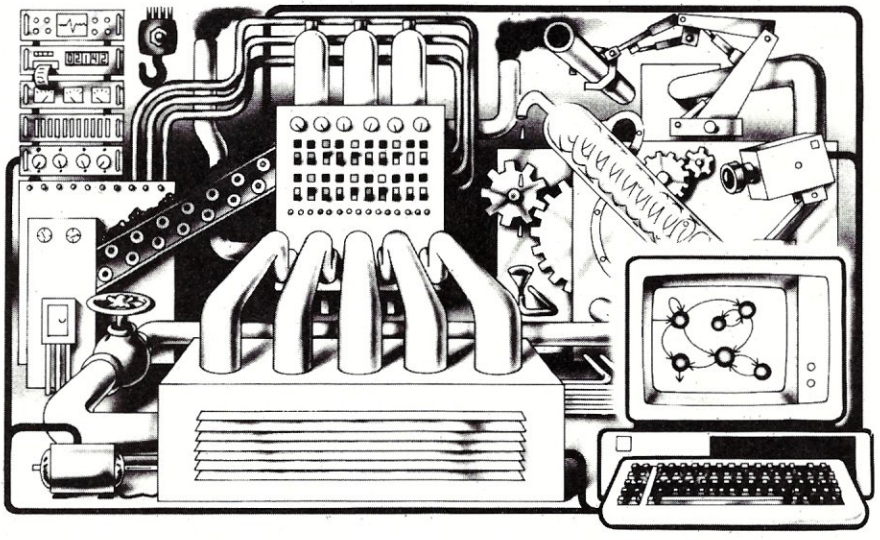
The Capital Osborne Users Group donated three dBASE applications to SIG/M from their library. One is a checkbook balancing program. Another is a genealogy program. The final program is a church management system. These applications are found on Volume 227.

SIG/M Volumes are available on 8" SS SD Disks for $6.00 each ($9.00 foreign) directly from SIG/M, Box 97, Iselin, NJ 08830. Printed catalogs are $3.00 each ($4.00 foreign). Disks in a variety of formats may be obtained through the worldwide SIG/M distribution network. The distributor list is included with the printed catalog. A disk version of the catalog (Volume 00) is available for $6.00

Volume 224	Small C Compiler with Floating Point (Z80) released April 19, 1985
CC.CQM	Z80 Small C compiler with
CC.DQC	floating point math
C80V.CQ	
C80V-2.CQ	
ARGS.CQ	I/O direction and command
ARGS.DQC	line parsing
ARGS.H	
ARGS.OQJ	
CLIBV.OBJ	alternate math & I/O library
FLOAT.CQ	floating point routines
FLOAT.DQC	
FLOAT.H	
FLOAT.OBJ	
IOLIB.CQ	integer math and I/O
IOLIB.DQC	
IOLIB.H	
IOLIB.OQJ	
PRINTF.DQC	formatted I/O
PRINTF1.CQ	
PRINTF1.H	
PRINTF1.OQJ	
PRINTF2.CQ	
PRINTF2.H	
PRINTF2.OQJ	
PROFILE.AQM	execution profile and walkback
PROFILE.DQC	
PROFILE.H	
PROFILE.OBJ	
TEST.CQ	sample program
TRANSCEN.CQ	transcendental functions
TRANSCEN.DQC	
TRANSCEN.H	
TRANSCEN.OBJ	
ZLINK.CQM	linkage editor
ZLINK.DQC	
ZMAC.CQM	assembler (relocatable output)
ZMAC.DQC	

Volume 225	FX-80 Character Set Biochemical Engineering Games & Simulation released April 19, 1985
FX.COM	allows second character set
FXDRVR.COM	with Wordstar and FX-80 printer.
FXDRVR.DOC	
FXDRVR.MAC	
ALTER.FON	scientific font for MX-80
SIMULA.LBR	biochemical eng. games & simulation
Volume 226	The Sigi Kluger Utilities released April 19, 1985
BACKMW.DOC	hard disk backup for Morrow M16
BACKMW27.COM	
BACKMW27.MAC	
BAUDRATE.ASM	sets I/F 4 & TVI 950 baud
COLONS.DOC	add/delete colons to ASM
COLONS12.COM	source file where needed
COLONS12.UPD	
DATE.ASM	convert DRI date to MM/DD/YY
UNDATE.ASM	and MM-DD-YY to DRI format
DRI-DATE.DOC	
DELETE.COM	ERA utility
DELETE.MAC	
DUMPDIR.COM	directory dump with file
DUMPDIR.DOC	allocation display
DUMPDIR.MAC	
E5.A86	fills CP/M 86 disk with E5s
FINDU10.DOC	string find utility for
FINDU11.COM	squeezed and unsqueezed files
FINDU11.MAC	
FREESP.MAC	module to calculate disk space
HEXDEC.A86	hex to decimal module for CP/M
HEXDEC.ASM	80 and 86
HXDC24.MAC	
HXDC24.REL	
LCHEK.DOC	library CRC program in CHEK and
LCHEK11 .COM	CRC compatible formats
LCRCK11.COM	
LCRCK11.DOC	
LCRCK11.MAC	
LDIR22.COM	library directory with version for
LDIR22.DOC	original filenames of squeezed
LDIR22.MAC	member files and filematch utility
LDIRR.ASM	
LDIRR.COM	
MATCHF.REL	
MYSORT.REL	
NCRCK1.COM	new CRC program
NCRCK1.DOC	
NCRCK1.MAC	
PRUN12.COM	runs PRL files under CP/M
PRUN12.DOC	
PRUN12.MAC	
SETBAUD2.ASM	sets baud rate for I/F 4
SETBAUD2.COM	
SORT.ASM	Shell-Metzner in-memory sort
TESTHXD.MAC	tests HXDC24
TIMEDF4.ASM	elapsed time calculator
TIMEDIF.ASM	
TYPEL36.COM	super type program, squeezed and
TYPEL36.DOC	unsqueezed, in library, to printer,
TYPEL36.MAC	etc.
UNSQZ13.COM	super unsqueezer
UNSQZ13.DOC	
UNSQZ13.MAC	
USQB.REL	
WILDEX.A86	wildcard expansion routine for
WILDEX.MAC	both ASM86/RASM86 and
WILDEX.REL	RMAC/M80
Volume 227	dBASEII Programs CHECKS, CHURCH MANAGEMENT and GENEALOGY from the Capital Osborne UG released April 19, 1985
CHECKS.LBR	checkbook program
CHURCH.LBR	church management program
GENEOLGY.LBR	genealogy program

Csharp Realtime Toolkit



Realtime on MSDOS? Csharp can do it! Get the tools without operating system overhead. Cut development time with C source code for realtime data acquisition and control. **Csharp** includes: graphics, event handling, procedure scheduling, state system control, and interrupt handling. Processor, device, and operating system independent. **Csharp** runs standalone or with: MSDOS, PCDOS, or RT11. **Csharp** runs on: PDP-11 and IBM PC. **Csharp** includes drivers for Hercules and IBM graphics boards, Data Translation and Metrabyte IO boards, real time clock, and more. Inquire for Victor 9000, Unix, and other systems. Price: $600

SYSTEMS GUILD

Systems Guild, Inc., P.O. Box 1085, Cambridge, MA 02142
(617) 451-8479

PC/Blue Report

by Hank Kee

The contribution of material to the PC/Blue public domain library comes from many different sources. A major repository of public domain software happens to be on the vast multitude of Bulletin Board Systems. Since I am also the system operator for the New York Amateur Computer Club Bulletin Board System, many users upload files from other bulletin boards. That is in the best tradition of networking.

Many contributed programs are sent via mail to the Amateur Computer Group of New Jersey and to the New York Amateur Computer Club. All programs donated to the public domain are welcome and appreciated. A general screening of the submissions is made to insure that user-supported programs are not limiting in their condition for distribution. We recently received a very nice little system from an author who specifically stated that if you do not send him a "donation", you must destroy the material in your possession after 10 days or else you would be subject to liabilities.

The collected material is then reviewed for prior duplication, latest revision update, reasonable working operation and freedom from violation of copyright restrictions. Afterwards, the programs and related files are, loaded en masse, onto a fixed disk. Subcatalogs by topics are created or carried over for the purposes of classification of materials. Topics include communications, games, utilities, applications, and system functions.

Programs are then grouped together into preliminary volumes. To insure some compatibility between IBM clones, we have chosen to use the lowest common denominator which is an 8-sectored-per- track, double-sided format. If there are single-sided users, they probably have "buddies" who can make the simple conversion for them. The trial volumes are then cataloged into a table of contents known as the -CATALOG.nnn file. As a general rule, the diskette will contain from about 250 to 310KB. Occasionally a diskette may use the 9-sectored format. PC/Blue Volume 117 is a case in point. It is a wholly self-contained system that just barely went over the 320KB capacity.

In the latest releases, it should be noted that Lotus Development should be commended for placing certain materials into the public domain. This represents a major change in attitude by key software vendors. They had always been wary that the public may confuse all of their materials as being public domain and as such available for use and distribution. This has not happened. The ready availability of applications dependent on a proprietary package serves the interests of the public, as well as the authors.

A topic of major concern is the distribution of *user- supported* software in the public domain network. In the IBM world, user-supported software comprises the major volume of what is available in public domain programs. It has become increasingly difficult for the beginning cottage industry author to get his software to the marketplace. Most, if not all, of the major publishing sources will not deal with the untested new author who wishes to strike out on his own.

Many would-be start-up software entrepreneurs now begin with the user-supported concept. The tenet is simply to have the program available to be freely distributed. If you, the recipient, like it please don't feel embarrassed to send in the suggested donation. This eliminates expensive overhead for advertising. As a general rule, people tend to build libraries, so the released program will have an ever-expanding audience.

The following are the newly-released PC/Blue public domain software volumes.

Copies of the PC/Blue printed software directory can be ordered from Micro/Systems Journal, Box 1192, Mountainside NJ 07092 ($4 U.S, Canada & Mexico; $6 foreign).

Editor's Note: Hank Kee is the librarian for the PC/Blue public domain software library. He is the person who collects, assembles, and checks all the software issued by PC/Blue and then compiles and edits them into the released volumes.

117 Genealogy On Display v3.0
by Melvin O. Duke

118 PC-Talk (.EXE) v2.1
Host-III v1.1
QModem v104
QModem Background Dialer

119 NewKey v2.13
System Reset
Kermit v2.27

120 Investment Record System
Check Reconciliation Program
New Fonts
Menu Generator

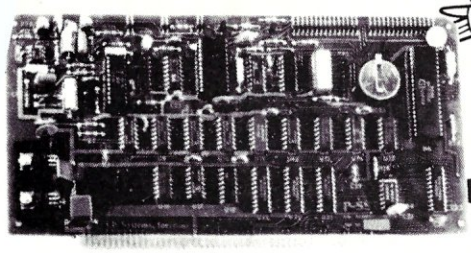
121 General Ledger Accounting
MergePlan - MultiPlan Consolidator
miscellaneous utilities
-Bi Directional Scrolling
-Diskette Jacket Listing
-Screen Page Dump
-Directory Sort
-Sort Directory List
-Displays No./Size of BACKUP
-"Undeleting" a File
-File Utility v1.70
-Tax Log for IRS

122 Symphony Applications
-Medical Worksheets
-Banking Worksheets

123 Lotus Graphics Printer Library II
Symphony Insurance Worksheets

124 Lotus 123 Worksheets
Lotus Macro Conversion Aid

If you would like to share some of your treasures with other PC hobbyists, please send your submissions to the New York Amateur Computer Club, Box 106, Church Street Station, New York, NY 10008.



PSST

Super Multifunction Card! *Zenith Z-100 • IEEE-696*

- *REAL-TIME CLOCK CALENDAR*
  - *Alarm, Heartbeat, Standby Interrupts*
- *TWO PARALLEL I/O PORTS*
  - *Joystick Compatible*
- *THREE-VOICE SOUND/MUSIC SYNTHESIZER*
- *SPEECH SYNTHESIZER*
- *EXTENSIVE SOFTWARE SUPPORT*
  - *MS-DOS, ZDOS, CP/M*
- *AUDIO OUTPUT*
  - *Pre-amp and Power amp*

$395!

P-SST

The Programmable Speech/Sound/Time Card by LP Systems, Inc. is a multifunction peripheral card designed to IEEE-696 (S-100) bus standards. The best selling accessory card for the H/Z-100 series of computers from Heath Co. and Zenith Data Systems, it is also adaptable, with appropriate software, to most systems which meet the IEEE-696 standard.

The **P-SST** combines those functions and features most requested by H/Z-100 users and programmers. Complete schematics and documentation, as well as extensive software support, make the P-SST useful to the novice as well as the advanced programmer. The P-SST is an "open" system, and user development of applications software is encouraged.

SOFTWARE

Current distribution and development software supports the use of the H/Z-100 series of computers under MS-DOS/Z-DOS, and CP/M-85/86 and any CP/M 2.2 IEEE-696 system.

Distribution software for the H/Z-100 includes automatic demonstration batch files, and demonstration programs for speech, music, graphics, sound, and joystick use.

Utility programs include a hardware diagnostic program for the P-SST, background music interrupt, and a clock read program to automatically set the time and date in MS-DOS/Z-DOS on boot-up.

Three-voice ASCII music scores are provided for use with the demonstration programs, and as examples of how to prepare scores for input to the MUSIC program. Notation used is similar to that for the IBM-PC BASICA "PLAY" command, for which many scores are available.

The SPEECH program will output words through the speech synthesizer from a specified ASCII text file.

Optional software includes the **P-SST Development Libraries** ($49.95), for use with most high level language compilers, and the **MAESTRO Music Editor** ($59.95) for music scoring and composition. The Development Libraries include over 50 subroutines, functions, and source code for most of the distribution demonstration and utility programs. MAESTRO allows graphic display and editing of music scores, background music utility programs, and an additional 50 three-voice music scores.

P-SST is also supported by the following **Software Wizardry** programs:

CHRONOLOGIC clock program (clock/calendar)
PALETTE color graphics editor (joystick)
REACTOR-100 Nuclear reactor simulation (sound/voice)
ZLYNK/II smart modem communications (clock/calendar)
ESP Bulletin Board/Dial in (clock/calendar)

a trademark of IBM, Inc.
d H-100 are trademarks of Heath Co.
100, Z-DOS, and Z-BASIC are trademarks of
Data Systems.
s a trademark of Microsoft, Inc.
rademark of Digital Research, Inc.
pyright 1984 by LP Systems, Inc.
are, MAESTRO, PALETTE, REACTOR-100,
d ESP are copyright by Software Wizardry, Inc.
ective authors.

Software Wizardry
THE MAGIC TOUCH

Dealer Inquiries Invited

1106 First Capitol
St. Charles, MO 63301
(314) 946-1968

ZENITH | data systems

Firepowe

Winchester Systems for the Zenith Z-150

- *11, 20, & 36 Megabytes internal*
- *Includes controller, drive, cables and instructions*
- *Specially engineered for the Z-150 by Software Wizardry*
- *No software mods-boot from hard disk*
- *11 Meg drive system $ 895*
- *20 Meg drive system $1195*
- *36 Meg drive system $1595*

System Specials!

Z-150 with:
640K RAM *$3495.00*
1 floppy
36 Meg hard disk
Green or Amber Zenith Monitor

Same system, but with 20 Meg hard disk

$2995.00

Call for Complete Catalog!

1-800-TO-BUY-IT (orders only)
1-314-946-1968 (other info)

First Capitol Computer™
1106 First Capitol Drive
St. Charles, MO 63301

Please add 2% for shipping in USA

First Capitol Computer

COMPETITIVE EDGE

P.O. BOX 556 — PLYMOUTH, MI 48170 — 313-451-0665

CompuPro®, LOMAS, EARTH COMPUTER, MACROTECH, TELETEK

S-100 CIRCUIT BOARDS

CompuPro 286 CPU™	$750	Lomas Lightning 286™	$821	MACROTECH 286/Z80H	$795	TELETEK SYSMASTER II	$899
CompuPro MDRIVE-H®	$417	Lomas THUNDER 186™	$1195	EARTH TURBO SLAVE I 8HMz	$395	ILLUMINATED TECH. 1024x1024 Graphics	$995

FREE MICRO/SYSTEMS JOURNAL SUBSCRIPTION WITH PURCHASE OF ANY OF ABOVE

DISK CONTROLLERS		CPU BOARDS		MEMORY BOARDS		I/O SERIAL/PARALLEL	
CompuPro Disk 1A™ A&T	$347	CompuPro 286™	$750	CompuPro RAM 22™ 120ns	$ 795	CompuPro System Support 1™	$245
CompuPro Disk 3™ A&T	$417	CompuPro SPU-Z™	$261	CompuPro Ram 23™ 64K 120ns	$ 277	CompuPro Interfacer 4™	$245
LOMAS LDP 72 Floppy Contr	$220	CompuPro 8085/88™	$245	CompuPro RAM 23 128K 120	$ 487	CompuPro Interfacer 3™	$346
Lomas WD1002 HD Contr	$325	CompuPro CPU Z™	$179	Lomas RAM 67 128K 100ns	$ 725	Lomas 2 ser, 2 par, clock Hazitall	$275
Teletek FDC II Flpy cont	$245	Lomas Lightning 286	$821	Lomas Megaram 256K 150 ns	$ 556	Lomas Octaport 8 serial	$316
Teletek HD H.D. Contr.	$375	Lomas 8086 8MHz	$420	Lomas Megaram 512K 150ns	$ 821	Lomas Octaport 4 serial	$200
80287 FOR LOMAS 286	$395	Lomas 8086 10MHz	$520	Lomas Megaram 1024K 150ns	$1096	Teletek PS10 2 par, 4 ser	$215
80287 FOR COMPUPRO 286	$325	Macrotech 286/Z80	$795	Lomas Megaram 2048K 150ns	$1595	Teletek I square 1 port	$195

SINGLE BOARD COMPUTERS		HARD DISKS		FLOPPY DRIVES		S-100 COLOR GRAPHICS	
THUNDER 186 256K QTY 1	$1195	Maxtor 2085 30 ms	$2295	8" Mitsubishi 2894-63	$389	Lomas Color Magic 16K	$476
THUNDER 186 256K Qty 3 up	$1095	Maxtor 1105 30 ms	$2795	5" Mitsubishi 4853 96 TPI for CompuPro	$125	Lomas Color Magic 32K	$556
TELETEK Systemaster®	$ 557	Rodime 53/40 55 ms	$1395	5" Mitsubishi 4851 48TPI	$150	CompuPro PC VIDEO 16K	$371
Teletek SBC-1 6MHz 128K	$ 695	Quantum Q540 45 ms	$1395	5" National JA551-2 48 TPI for Lomas	$135	Illuminated Tech. Color Graphics	$995
Teletek SBC-II Dual Slave	$ 854	Seagate 25/20 80 ms	$ 695	5" TEAC 55B for IBM PC®	$135	I.T. Board is 1024 x 1024 and	
TELETEK SYSTEMASTER II®	$ 899	Seagate 4026 40 ms	$ 995	5" Dbl. Sided Color Diskettes 10 Pk	$ 21	runs Auto-CAD-2 & other on Lomas	
EARTH TURBO SLAVE I 8MHZ Z80	$ 395	Maxtor 1140 30 ms	3595	5" Dbl. Sided Color Diskettes 100 Pk	$199		

Illuminated Technology 1024 x 1024 Color Graphics Board (for AutoCAD2)™ $995

Earth Computer Turbo Slave I 8mhz Z80H - 128K $395

HARD DRIVE KIT	IBM®PC™	AT
21 Megabyte	$895	$ 795
36 Megabyte		$1295
70 Megabyte		$2295
80 Megabyte		$3295

AUTOCAD 2® SYSTEM $6995

8 mhz 86/87, 512K, 10 MB Hard Disk, Terminal, MS DOS 2.11
H. Res. Color Monitor, Autocad 2 w/ADI & AD2 $6995
Many Options, 6 or 8MHZ 286, 1024K Memory, up to
16 MB Memory Drive, 19" Monitor, Plotters & Digitizer Pads

SYSTEMS READY TO RUN

NEW SYSTEMS WITH 15 SLOT MOTHER BOARD 30 amps of +8 Volts & Power for 2 Hard Disk Drives
CompuPro 286, SPUZ, 512K, SSI, I/04, I-96 TPI DR, 15 Slot, 40 MB - D3, 10 MB Tape — $6895

CompuPro 85/88, 256K, CDOS, SS1, I/O 4, 2-96TPI DRS, 15 Slot, 30 Amp, +8 Volt $2995
CompuPro 85/88, 256K, CDOS, SS1, I/O 4, 1-96TPI, 20MB, 15 Slot, 30 Amp +8 Volt $4095
286/Z80H, 1024K Static, CDOS, SS1, I/O 4, 1-96TPI, 40MB, 15 Slot, 30 Amp +8 Volt $7295
MDRIVE-H option $417 per 512K or $1495 per 2048K each
Lomas 286, 1024K, 20MB HD, 1-5", CDOS, 6 SERIAL, 2 PAR, 15 Slot, 30 Amp +8 Volt $4995
Lomas Thunder 186, 256K, 20MB HD, 1-5", CDOS, 4 Slot $2895
Teletek 8MHz Master, 4-8MHz 128K SLVS, 1-5", 20 MB HD, TDOS, 30 Amp +8 Volt $4495

LOMAS 24 MEGA BYTE TAPE BACKUP Sub System for MOST CP/M-86 & MSDOS S-100 Systems $1495

CompuPro 10 MB Tape with 40 MB Q540, 1-5" 96 TPI floppy & Disk 3 in Cabinet $2995

Independent, Custom, Scientific Programming Service Available

UPGRADE YOUR IBM PC!!

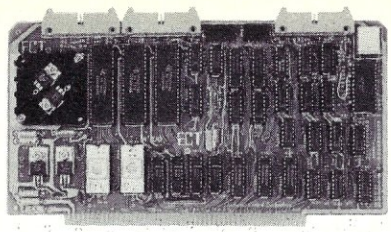
CABINETS		MONITORS		GRAPHIC BOARDS	
CompuPro Enclosure 2™ Desk	$715	Amdek 310A	$159	Hercules Monochrome	$299
CompuPro Enclosure 2 Rack	$750	Taxan Color 440	$549	Hercules Color Card	$159
Para Dynamics CVT DR Cab 2-8" Desk	$395	Princeton Color HR-12	$459	Tecmar Graphics Master	$449
Para Dyn CVT Rack DR Cab 2-8"	$495	Princeton Color SR-12	$649	Paradise Graphics	$279
Para Dyn CVT Desk DR Cab 1-8", 5" HD	$425	**MULTI-FUNCTION BOARDS**		STB Graphix + II	$279
Para Dyn Roll A Round CVT Pronto	$1195	AST 6 Pak 64K	$245	**FLOPPY DRIVES**	
Para Dyn 10 Slot Mini Pronto CVT	$795	Quadram Expanded Quadboard OK	$219	TEAC 1/2 HT FD55B	$119
Single 5" H.D. Cabinet	$150	TECMAR Captain 64K	$199	Mitsubishi 96 TPI	$125
Two 5" Hard Disk Cabinet	$195			5" DSDD Color Diskettes	$21

ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE AND STOCK ON HAND

CompuPro is a registered trademark of VIASYN Corporation, MDRIVE-H, 8-16, Disk 1A, Disk 3, CPU 286, CPU 8085/88, CPU Z, RAM 22, RAM 23, System Support 1, Interfacer 3, Interfacer 4, are either trademarks or registered trademarks of Viasyn, CP/M, Concurrent Dos are registered trademarks of Digital Research Inc., Turbodos is a trademark of Software 2000 Inc., Systemaster, Systemaster II are registered of Teletek Enterprises Inc., MSDOS is trademark of Microsoft, IBM PC is a registered trademark of International Business Machines. Thunder 186, Lightning 286, Lightning 1, LDP 72, Hazitall, Color Magic, are trademarks of Lomas Data Products. AutoCAD 2 is a trademark of AutoDesk Inc. IBM is a registered trademark of International Business Machines.

CUSTOM PRODUCTS

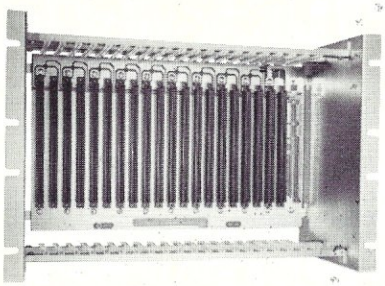
DESIGN • LAYOUT MANUFACTURING



R ²I/O

ROM/RAM & I/O

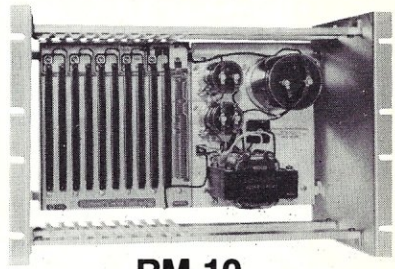
S-100 PRODUCTS



ECT-100-F

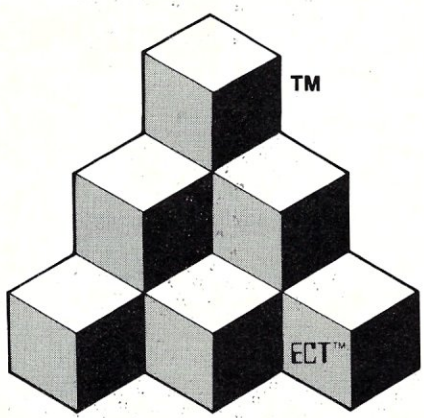
RACKMOUNT CARD CAGES

ECT™

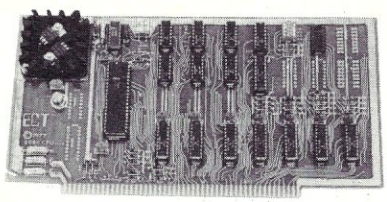


RM-10

CARD CAGE & POWER SUPPLY

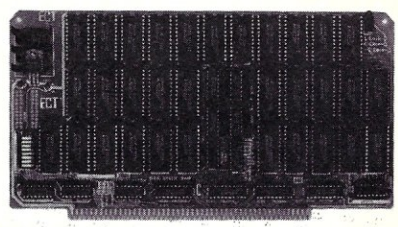


™



8080 CPU

CENTRAL PROCESSING UNITS

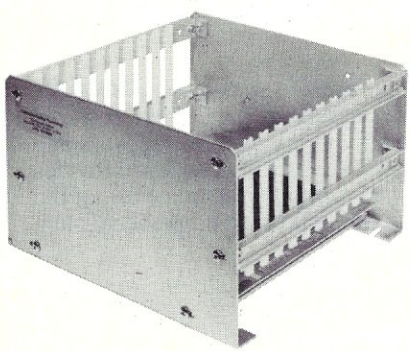


64K RAM

FULLY STATIC MEMORY

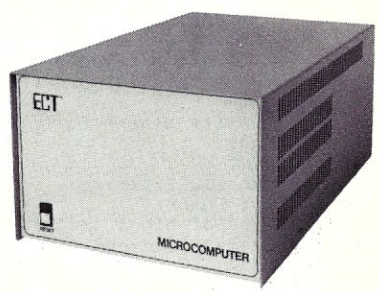
BUILDING BLOCKS FOR MICROCOMPUTER SYSTEMS, DEDICATED CONTROLLERS AND TEST EQUIPMENT

CARD CAGES, POWER SUPPLIES MAINFRAMES, CPU'S, MEMORY I/O, OEM VARIATIONS



CCMB-10-F MIN

6,10 OR 20 SLOT CARD CAGES



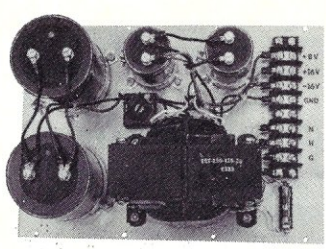
TT-10

TABLE TOP MAINFRAMES

ELECTRONIC CONTROL TECHNOLOGY, INC.

10 Cottage St., Berkeley Heights, NJ 07922 (201) 464-8086

SPECIALIZING IN QUALITY MICRO COMPUTER HARDWARE



PS-30 A

POWER SUPPLIES

MULTIBUS® PRODUCTS

MULTIBUS IS A TRADEMARK OF INTEL CORP.

Turbo Pascal Corner

by David W. Carroll

This column features tips and techniques for using Turbo Pascal productively on MS/PC-DOS and CP/M microcomputer systems. It discusses typical problems and their solutions. Reader's suggestions, comments, and questions are encouraged. Address them to:

Turbo Pascal Corner
Box 699
Pine Grove, CA 95665

In this issue, we will take a look at handling simple data translations in Turbo Pascal. We will develop a program to solve a sample problem that requires a byte-to-hex code translation. The task will be to design a file dump utility for MS-DOS (or CP/M with a few modifications) that will list a file's contents in both hex and ASCII on the screen.

The basic algorithm has to read the input file sequentially in byte format, translate each byte to its hex and ASCII equivalent (if printable) and display it on the screen. We will also display the data address in the file, and offset the starting address by 100 hex for .COM type files.

The pseudo-code for this is shown in listing 1.

LISTING 1

```
Get File Name
Open File
If fileext = .COM then address starts at 100h else 0
While not eof
  Translate address to hex
  Print address
  While not eol (on screen - 16 bytes)
    If not eof do
      Get byte
      Translate to 2 hex digits
      Print Hex digits and space
      If not printable ASCII character substitute '.'
      Add to ASCII print string
    Else
      Write blanks to fill line & string
    End;
  End;
  Print ASCII String
  Print CR-LF
  If end of block (address = F0h), print blank line
  Increment address by 16
End;
Close file
End.
```

The only sub-procedure required in the preceding example is a routine to translate a byte of data into its two digit hex equivalent and then in ASCII for display. Pascal provides a number of methods to code this translation, including using an array or a CASE/formula combination.

In both cases, the first step is to isolate the low and high order nibbles (4 bits) of the data byte to be converted. This is easily done with the code:

```
hbyte := databyte div 16;
lbyte := databyte mod 16;
```

ARRAY METHOD

Using the array method, we first set up an array [0..15] of characters. Then each member of the array is initialized with the correct ASCII character as determined by the hex value established by the first part of the main program. The procedure *trans()* accepts a byte of data and prints the two character hex code on the screen:

```
procedure trans(indat:byte);
var
   hexbyte  : string[3];
   hbyte    : byte;
   lbyte    : byte;

begin
   hbyte := indat div 16;
   lbyte := indat mod 16;
   hexbyte[0] := chr(2);
   hexbyte[1] := hex[hbyte];
   hexbyte[2] := hex[lbyte];
   write(hexbyte);
end;

begin  {main program}
   hex[0] := '0';     {initialize array}
   hex[1] := '1';     {with hex}
   hex[2] := '2';     {character set}
   hex[3] := '3';
   hex[4] := '4';
   hex[5] := '5';
   hex[6] := '6';
   hex[7] := '7';
   hex[8] := '8';
   hex[9] := '9';
   hex[10] := 'A';
   hex[11] := 'B';
   hex[12] := 'C';
   hex[13] := 'D';
   hex[14] := 'E';
   hex[15] := 'F';
   .....
```

and so on for the remainder of the main program.

CASE METHOD

Although arrays can be useful for complex, irregular coding plans, they are not a very elegant solution to the translation problem, especially where the data follows some regular pattern or scheme. The CASE statement provides an alternative in this particular problem with the use of the ordinal ASCII values for the formula based translation (listing 2):

LISTING 2

```
procedure trans(indat:byte);
var
   bdat     : byte;
   ch       : char;

procedure xlt;
begin
  case bdat of
  0..9   : ch := chr(bdat + ord('0'));
  10..15 : ch := chr(bdat + ord('A') - 10);
  else
  ch := 'X';                    {error on input value}
  end;
end;

begin
   bdat := indat div 16;
   xlt;
   write(ch);
   bdat := indat mod 16;
   xlt;
   write(ch);
end;
```

Finally, we can simplify even further by using a function in the translate procedure.

```
procedure trans(indat:byte);
var
  ch : char;

function xlt(bdat: byte) : char;
begin
  case bdat of
  0..9    :  xlt := chr(bdat + ord('0'));
  10..15  :  xlt := chr(bdat + ord('A') - 10);
  else
  ch := 'X';
  end;
end;


begin
   ch := xlt(indat div 16);
   write(ch);
   ch := xlt(indat mod 16);
   write(ch);
end;
```

CONVERSION OVERFLOWS

An interesting problem showed up during the development of the program when an integer type variable was used for the address counter. Guess what happens when a file is larger than 32K bytes? The answer is that the address will exceed the MaxInt value for Turbo and be displayed as a negative, or cause a run-time error. This problem can be avoided by using a real for the address and converting it, although the conversion can prove to be an interesting exercise in itself. And, what if you want to dump a file larger than 64K? This requires printing a 6 digit hex address instead of four digits.

The real to integer conversion process can be tricky, due to the automatic type conversions in Turbo. Intermediate results can exceed MaxInt, producing negative numbers unexpectedly or causing run-time errors, depending on the type of result variable. See if you can find the bug in this test program to convert a real number to a two integer byte value:

```
program test1;
var
  rec  : real;
  hrec : byte;
  lrec : byte;

begin
        rec := 65534.0;
        writeln('input value = ',rec:5:0);
        hrec := trunc(rec / 256);
        writeln('high order byte = ',hrec);
        lrec := trunc(rec - (hrec * 256));
        writeln('low order byte = ',lrec);
end.
```

This program generates a run-time error 92 at the statement:

```
lrec := trunc(rec - (hrec * 256));
```

because the intermediate result of (hrec * 256) is of type Integer causing a negative overflow (in the example shown, hrec = 255 and the intermediate integer result of (255 * 256) is -256 rather than 65280 as expected). In the incorrect example above, Turbo computes:

```
lrec := trunc(65534.0 - (255*256)) or
lrec := trunc(65534.0 - (-256))
```

and returns the run-time error because the trunc() value is greater than 32767.

To correct the problem, at least one of the operands must be of type Real to force the result to be of type Real and eliminate the overflow. Here is the corrected statement:

```
lrec := trunc(rec - (hrec * 256.0));
```

The Turbo 2.0 reference manual states, *If both the operands of the multiplying and adding [categories of] operators are of type Integer, then the result is of type Integer. If one (or both) of the operands is of type Real, then the result is also real.* What is not obvious is that this rule also applies to the INTERMEDIATE results in the evaluation of a complex expression.

The simple program, shown in listing 3, demonstrates the difference between the results than can be obtained, i.e. the integer result is -256 and the real result is 65280 - quite a difference.

LISTING 3

```
program test2;
var
  hrec : byte;
  xxx  : integer;
  yyy  : real;

begin
        hrec := 255;
        xxx := hrec * 256;
        writeln('integer intermediate value = ',xxx);
        yyy := hrec * 256.0;
        writeln('real intermediate value = ',yyy:5:0);
end.
```

FILEDUMP UTILITY

Now we can put these ideas together to create the final FILEDUMP utility program as shown in listing 4:

LISTING 4

```
{FileDump.pas
 Turbo Pascal Ver. 2.0
 PC-DOS Version
 Copyright 1985 by David W. Carroll
 Date:  4/23/85
 Version: 10
}

program filedump;
const
   bell      = 07;
   version   = '10';

type
   datstr        =   string[20];
   datafile      =   file of byte;

var
   infile        :  datafile;
   infname       :  string[20];
   asciitext     :  string[20];
   sec           :  real;
   sec1          :  real;
   col           :  byte;
   dat           :  byte;
   msec          :  byte;
   hsec          :  byte;
   lsec          :  byte;
   goodfile      :  boolean;
   quit          :  boolean;

procedure trans(indat:byte);
var
  ch : char;

function xlt(bdat: byte) : char;
begin
  case bdat of
  0..9   : xlt := chr(bdat + ord('0'));
  10..15 : xlt := chr(bdat + ord('A') - 10);
  else
  ch := 'X';
  end;
end;
```

```
begin
   ch := xlt(indat div 16);
   write(ch);
   ch := xlt(indat mod 16);
   write(ch);
end;

procedure Uppercase(var Str : datstr);
var
 indx,len    : Integer;

begin
  Len := length(Str);
  for Indx := 1 to len do
   Str[Indx] := UpCase(Str[Indx])
end;

begin
   asciitext[0] := chr(16);{set ASCII string length}
   sec := 256;             {set initial .COM address}
   quit := false;
   ClrScr;
   writeln;
   write ('File HEX and ASCII DUMP program Version');
   writeln ( version);
   writeln('Copyright 1985 by David W. Carroll');
   writeln;
   window(1,5,80,25);  {PC-DOS}
   repeat
     ClrScr;
     write ('Input filename  -->  ');
     readln (infname);
     if length(infname) > 0 then
     begin
       assign(infile,infname);
       {$I-} reset(infile) {$I+};
       goodfile := (IOresult = 0);
       if not goodfile then
       begin
         write (chr(bell));
         writeln ('FILE ',infname,' NOT FOUND');
         delay(3000)
       end;
     end
     else
     quit := true;
   until goodfile or quit;
   if not quit then
   begin

     {reset start address if not .COM}
     uppercase(infname);
     if (pos('.COM',infname)=0) then sec := 0;

     writeln;
     window(1,6,80,25);   {PC-DOS}
     write('Address  00 01 02 03 04 05 06 07  08 ');
     write(' 09 0A 0B 0C 0D 0E 0F   ');
     writeln('    ASCII TEXT');
     writeln;
     window(1,9,80,25);   {PC-DOS}
     while not eof(infile) do
       begin
         msec := trunc(sec / 65536.0);
         trans(msec);
         write(' ');
         secl := sec - (msec * 65536.0);
         hsec := trunc(secl / 256);
         lsec := trunc(secl - (hsec * 256.0));
         trans(hsec);
         trans(lsec);
         write('  ');
         for col := 1 to 16 do
         begin
           if not eof(infile) then
           begin
             read(infile,dat);
             trans(dat);
             if dat in [32..126] then
               asciitext[col] := chr(dat)
             else
               asciitext[col] := '.';
           end
           else
           begin
             write('  ');
             dat := 0;
             asciitext[col] := ' ';
           end;
           write(' ');
           if col = 8 then write(' ');
         end;
       writeln('  *',asciitext,'*');
       if lsec = 240 then writeln;
       sec := sec + 16;
       end;
     close(infile);
   end;
   if not quit then
   begin
     writeln;
     writeln(' - eof -');
   end;
end.
```

This version of FILEDUMP is designed to operate on the PC-DOS version of Turbo Pascal 2.0b which includes windowing procedures. If the code lines marked {PC-DOS} are removed, the program will work under other versions of Turbo. You are encouraged to modify the program to your own requirements. Some changes might include a printer output option, use of command line parameters, and display of one page of data (16 paragraphs of 16 bytes each) at a time.

This program and some 300 other Turbo Pascal public domain programs are available 24 hours a day for free by downloading from the *High Sierra RBBS* system at (209)296-3534.

David W. Carroll is a freelance writer and computer consultant living in the Sierra Nevada foothills near Sacramento, California. He is the author of "Telecommunications with the IBM PCjr" co-published by Microtext/Prentice Hall and "Programming with Turbo Pascal" to be co-published by Microtext and McGraw-Hill this summer.

SMARTS RAM DISK SOFTWARE
for the QT, Great Salt Lake
California Digital or Computime
S-100 256K DRAM Memory Boards
Only
$40.00
or $500 with A&T board

UFDC-I Floppy Disk Controller
Reads/Writes 54 Formats
$275.00

CP/M Football Prediction Program
available in many formats
$40.00

All S-100 HARDWARE AVAILABLE IN BARE BOARDS KITS OR A&T

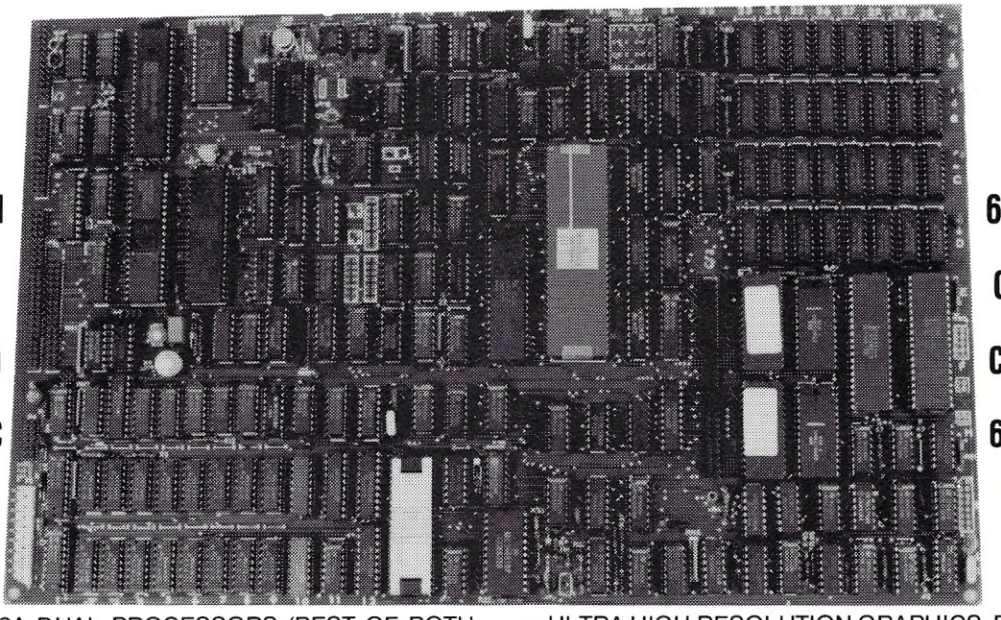
GSR COMPUTERS
60-10 69th St.
Maspeth, NY 11378
(718) 476-2091
NYS Residents Add Tax - $5 S&H

FOR THE **BEST** OF US . . .

THE CYPHER™

A COMPLETE 68000 & Z80A SINGLE BOARD COMPUTER SYSTEM WITH ULTRA-HIGH-RES GRAPHICS!!

FREE 68000 FORTH AND CYPHER-DOS (CPM-80 COMPATIBLE) AND 68000 BASIC



FREE 68000 FORTH AND CYPHER-DOS (CPM-80 COMPATIBLE) AND 68000 BASIC

- 68000 & Z80A DUAL PROCESSORS (BEST OF BOTH WORLDS! OPTIONAL Z80 H.)
- 1 MEGABYTE MEMORY. (41256 DRAM)
- DOUBLE DENSITY FLOPPY DISK CONTROLLER (8", 5¼" OR 3½" WD 2793).
- DMA CONTROLLER FOR FAST IMAGE TRANSFERS TO/FROM VIDEO MEMORY. (INT 8237).
- 2 RS232 SERIAL PORTS (ZSIO).
- 24 BIT ADDRESS MANAGEMENT FOR Z80.
- 4 LAYER P.C.B. (9¼" x 14¾").
- RUNS CP-M-80 2.2, CP-M-80 3.0, CP-M-68K, CYPHER DOS, RAM DISK, 68000 BASIC IN ROM, NEC 7220 TERMINAL EMULATION.
- ULTRA HIGH RESOLUTION GRAPHICS, PRIVATE 128K, PROGRAMMABLE UP TO 1024 x 1024 RESOLUTION (NEC 7220, GREAT FOR CAD SYSTEMS!).
- REAL TIME CLOCK (MULTITASKING CAPABILITY!).
- TWO CHANNELS OF D/A AND A/D, 12 BIT RESOLUTION (OPTIONAL MUSIC! ROBOTICS! LAB WORK!)
- 16K MONITOR EPROM. (EXPANDABLE TO 64K)
- 4K STATIC RAM. (EXPANDABLE TO 64K)
- PROGRAMMABLE BAUD RATE GENERATOR. (8253)
- PARALLEL ASCII KEYBOARD INPUT.
- FULL 68000 EXPANSION BUS (60 PIN HEADER, BUFFERED BUS)

New! LOWER PRICES! NOW 1 MEGABYTE CYPHER AT $1,299.95

- MANUAL AND COMPLETE SCHEMATICS **$20.00**
- BARE BOARD, EPROMS, Z80 BIOS, 68000 BIOS, Z80 MONITOR, 68000 MONITOR, CYPHER-DOS AND MUCH MUCH MORE **$399.95**
- KEYBOARD **$94.95**
- SWITCHING POWER SUPPLY **$154.95**
- CASE **$94.95**
- HARD DISC INTERFACE PLUG-IN CARD **$150.00**
- A/D D/A CYPHER OPTION **$100.00**

- **COMPLETE "CYPHER" WITH 68000, Z80, 1 MEGABYTE DRAM, 128 VIDEO DRAM, NEC 7220, REAL TIME CLOCK, DISC CONTROLLER AND SERIAL I/O, ASSEMBLED AND TESTED $1,299.95**

DEALER AND MANUFACTURING FRANCHISE INQUIRIES WELCOME.

VISA

master charge

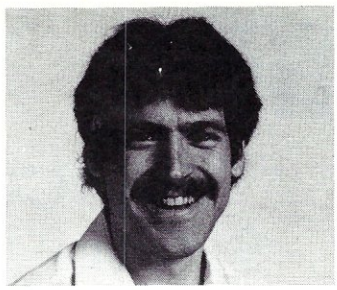
SHIPPING CHARGES:
ALL PRICES ARE IN U.S. DOLLARS
SHIPPING: 8% SHIPPING, EXCESS WILL BE REFUNDED

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.
CP-M IS A TRADEMARK OF DIGITAL RESEARCH.

MOTOROLA INTEL

MOTEL COMPUTERS LIMITED
174 BETTY ANN DRIVE, WILLOWDALE,
TORONTO, ONTARIO, CANADA M2N 1X6
(416) 229-4727

The C Forum



by Don Libes

IMPLEMENTING SETS WITH BIT OPERATIONS

Sets with a small number of elements are easily implemented by bit operations in most languages. Some C compilers have a declaration *enum* that allows the programmer to use sets directly without worrying about the implementation. Unfortunately, most C compilers don't have the *enum* construct. In any case, the implementations I will discuss allow more general uses then *enums* do, anyway.

Using bit operations we can implement sets, which in turn, specify such operations as: union, intersection, difference, assignment, membership, equal and size. The ordering relation can also be implemented, although, strictly speaking this is not a set operation.

SMALL SETS

A simple example of the use and implementation of a set is in the (BSD 4.2) UNIX select system call. select() takes parameters which represent sets of file descriptors. Since the number of files a process may have open is typically limited to 20, the set of file descriptors is easily represented by a 32-bit integer. Each bit then, designates a file descriptor. For example, if the set has an integer value of 25 decimal (or 11001 boolean), the file descriptors referred to are 4, 3 and 0 since the bits in the 4, 3 and 0 positions are on. In this example, the index of the bit is exactly the value of the file descriptor. However this need not be the case. What is important is the small number of elements.

For sets like the one above, which have no more elements than the number of bits in a single datum like an int, then we can use the following constructs for set operations:

```
int set1, set2, set3;
set3 = set1 | set2;     /* union */
set3 = set1 & set2;     /* intersection */
set3 = set1 & ~set2;    /* difference */
```

To declare a set, define it as an int (or whatever type you choose). I highly recommend using a typedef to make things more readable. Macros can also be used for the set operations themselves. For example:

```
#define UNION |
set3 = set1 UNION set2;
```

To define a one-element set, use the macro #elt with a small index representing the index of the element.

```
#define set(x)              (1<<x)
```

#member returns 1 or 0 if an element is in the set or not.

```
#define member(s,x)    (0 != \
                    (s INTERSECTION elt(x)))
```

min_elt() returns the smallest index in a set (or -1 if the set is empty). These examples should be enough for you to write any other set operations you need using this representation.

```
int min_elt(x)
int x;
{
    int i;
    /* first check if set is empty */
    if (x == 0) return(-1);
    for (i=0;~(x&(1<<i));i++) ;
    return(i);
}
```

BIG SETS

This is fine for sets with a small number of elements, but what about larger sized universes? For example, if we are writing a device driver for a disk, we must maintain the sets of disk cylinders that are queued to be read and written.

A list of cylinders is a prime candidate for representation by a bit vector. The only real difference is that such a set is probably bigger than the number of bits in any data type on your machine. For example, suppose we have 100 cylinders and our largest data type is a 32-bit long. Then we would need 4 longs to get at least 100 bits for the cylinders ($3*32 < 100 <= 4*32$). To get those 100 bits, we can declare an array of longs. #bigset is a macro that does exactly that .

```
#define bigset(name,bits)   struct {\
    int number;/* of subsets */\
    SUBSET data[1 + bits/BITS_PER_SUBSET];\
        } name = {1 + bits/BITS_PER_SUBSET};
```

This macro expands to a structure declaration! The structure defines an array of "SUBSET"s large enough to hold the set. We also define the number of SUBSETs in "number", so that we won't have to pass lengths as extra arguments into our set routines. SUBSET can be defined to be whatever is convenient for you. I always use "unsigned long" because the longer the datatype, the faster the routines will execute. For folks who watch every bit, though, shorter datatypes will waste less space (by leaving less unused bits at the end of the set array).

Finishing off #bigset, here are the macros needed.

```
#define BITS_PER_BYTE   8
#define SUBSET          unsigned long
#define BITS_PER_SUBSET (BITS_PER_BYTE\
                         * sizeof(SUBSET))
```

Now we can declare sets for 100 cylinders to be read and written as:

```
bigset(readcyls,100);
bigset(writecyls,100);
```

In order to pass these sets as parameters, we'll have to define a type for that. (Unfortunately, we can't use #bigset for this because it defines a class of structures.) We do this as follows:

```
struct bigset_param {
    int number;
    SUBSET data[1];
};
```

Most of the standard set operations (union, intersection, assignment, etc.) look very much the same. A single loop performs its respective operation on an entire SUBSET at a time.

Here is the code for union.

```
bigset_union(s1,s2,s3)/* s1 = s2 U s3 */
struct bigset_param *s1, *s2, *s3;
{
    int i;
    for (i=0; i<s1->number; i++)
        s1->data[i] = s2->data[i] UNION
                      s3->data[i];
}
```

To use this routine, of course, you must pass the address of the set.

bigset_print() requires an extra inner loop to print out each bit.

```
bigset_print(s)
struct bigset_param *s;
{
    int i, j;
    for (i=0; i<s->number; i++) {
        for (j=0; j<BITS_PER_SUBSET; j++) {
            putchar(s->data[i]&(1<<j)?'1':'0');
        }
    }
    putchar('\n');
}
```

Finally, here is some code for turning on an individual bit by its index. I chose to write it as a function only so that the parameter passing conventions in all the bigset routines would remain consistent. You should be able to produce its inverse, with a small change to the original version of min().

```
bigset_set(s,i)/* turn on element i in set s */
struct bigset_param *s;
int i;
{
    s->data[i/BITS_PER_SUBSET] |=
                        1<<(i%BITS_PER_SUBSET);
}
```

Finally, here is some code using these routines.

```
bigset(readcyls,100);
bigset(readcyls,100);

main()
{
    /* request cylinders 1 and 10 to be read */
    bigset_set(&readcyls,1);
    bigset_set(&readcyls,10);
    printf("cylinders to read: ");
    bigset_print(&readcyls);

    /* request cylinder 4 to be written */
    bigset_set(&writecyls,4);
    printf("cylinders to write: ");
    bigset_print(&writecyls);

    /* find out cylinders requiring action */
    bigset_union(&activecyls,&readcyls,
                                &writecyls);
    printf("cylinders requiring I/O: ");
    bigset_print(&activecyls);
}
```

OPTIMIZATIONS

You should now be able to finish the rest of the set routines. There are some obvious ways of improving the code that you should consider if you use these routines in production code.

1) Convert bigset__set() to a macro.

2) Use pointers instead of array references in the loops.

3) Convert divisions to shifts. Convert mods to bitwise ands. (This can be done only because we are working with powers of two.) For example, x/BITS__PER__SUBSET for BITS__PER__SUBSET == 32 can be written x<<5 (since 5 is log of 32 base 2).

C NEWS TIDBITS

In case you are wondering what I use as a reference book on C, it is *C - A Reference Manual* by Harbison and Steele. Published by Prentice-Hall. I encourage all C programmers to live by this book!

The C Users' Group publishes a newsletter for C news. The address is Box 97, 415 E. Euclid, McPherson, KS 67460. CUG also keeps a library of public-domain C software. CUG146 is a *Small C* for 6800's running FLEX.

Lattice is attempting to woo Microsoft C users by offering a package of the latest Lattice C compiler and symbolic debugger for $175. Phone (312) 858-7950 for more info.

I encourage readers to write to me about topics or problems that you want to know about. I want this column to be reader driven. Write to me care of M/SJ, Box 1192, Mountainside, NJ 07092.

Don Libes is a computer scientist working in the Washington, DC area. He works on artificial intelligence in robot control systems. He is also the son of Lennie and Sol Libes.

LIFE IN THE FAST LANE!

Experience NightOwl Software's cornucopia of ultra high-speed communications packages, featuring US Robotics' Courier 2400 modem, the NightOwl Connection and MEX-PC — the Cadillac of communications software.

Consider the elements:

• **U.S. ROBOTICS COURIER 2400**, *the* state-of-the-art auto dial/auto answer modem. Features: 300/1200/2400-baud operation; nine front-panel LEDs; line-busy detect; built-in help screens; status screen; volume-controlled speaker; full Hayes-type "AT" command set; a typeset 80-page User's Manual. List price $699.95.....Our price: $524.95

• **NIGHTOWL CONNECTION**, our multi-user software database. The Connection, with 2400-baud Couriers on all incoming lines, features an IBM area with 5 MB of public domain offerings, a Turbo-Pascal area, Kaypro- and Osborne-specific areas, plus directories crammed with compiler, language, word processing and communications programs for CP/M-80, CPM+ and TurboDOS — not to mention the most complete collection of MEX overlays and automated command files available anywhere. Allows full contact with other users and fast, personalized access to our technical support staff.
Our price: $100

• **MEX-PC**, the most flexible communications package available for the IBM-PC, puts more power at your fingertips than you ever dreamed possible. If you're struggling along with PC-TALK or Crosstalk XVI, experience the power of MEX-PC! Three protocols: Christensen XMODEM (CRC, Checksum & MODEM-7 "batch-file" protocols), CompuServe *A* protocol, XON/XOFF • - Multiple phone libraries • KEY editor for defining not only on IBM's function keys, but *any*alpha-numeric key on the IBM keyboard • Automatic list-dialing/redialing • Script file processor allows you to generate fully automated log-ons, log-offs, uploads & downloads — even conditionals • SHELL command lets you run any program or DOS command from within MEX without loss of text stored in memory. (Run your spreadsheet or word processor from within your communications program!) • Built-in HELP facility documents the entire program • 180-page User's Guide — ring-bound, typeset, fully-indexed, with complete tutorial.....Our price $59.95

Save even more with our packages:

Triple Play: Courier 2400 modem, MEX-PC and an *A*-class subscription to the NightOwl Connection (an $860 value) **$614.95**

Two Bagger: Courier 2400 and an *A*-class subscription to the NightOwl Connection (an $800 value) **$574.95**

Ground Rule Double: Courier 2400 and MEX-PC (a $760 value)..... **$534.95**

Call or write for information about other specials from NightOwl Software for CP/M, CP/M+ and TurboDOS operating systems.

To order by credit card, call us toll-free at 1-800-NITEOWL. In Wisconsin, call 1-414-563-4013. To order by mail, send certified check, cashier's check or money order to:

VISA

NightOwl Software, Inc.
Route 1 Box 7
Fort Atkinson, WI 53538

master charge

Personal cnecks accepted, but will delay shipment 14 days.

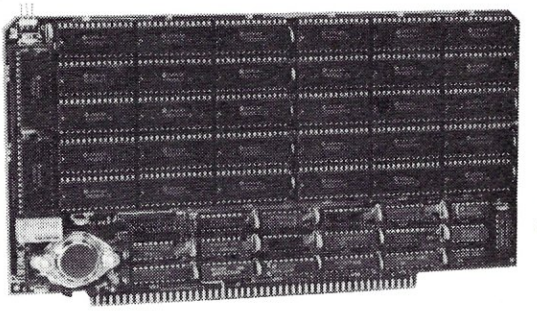
Prices do not include shipping. All items shipped UPS ground unless otherwise specified. For modems and modem packages, add $10; for software add $5. For UPS two-day air delivery (formerly UPS Blue) add $16 for modem packages or $13 for software products. Canadian orders add an extra $7.50 to shipping costs. Wisconsin residents add 5 percent sales tax.

Formats: 8" (SS,SD) format, and most 5¼" formats (except Apple and NorthStar). Please specify disk format when ordering.

YOU CAN'T BUY HIGHER QUALITY, MORE RELIABLE S-100 *STATIC MEMORY BOARDS* AT ANY PRICE!

256K BYTES
S-RAM 256™

$749* Qty. One



128K BYTES
S-RAM 128™

$449* Qty. One

STANDARD FEATURES:

- Operates in excess of 12 MHz
- Certified system ready (industrial grade dynamic burn in)
- Supports 8 and 16 bit data transfers
- IEEE 696/S-100 compatible
- 24 bit addressing
- Address strappable to any 128K block within the 16 meg address range
- Extremely low power consumption
- Single +5 volt operation
- One year warranty

OPTIONS

- Battery back-up capability
- Military MIL-STD-883B type burn in
- Faster ram chips available

***SAVE AN ADDITIONAL $100!** Buy one 256K board at $749 and receive a $100 certificate towards the purchase of your next 256K board. Buy one 128K board at $449 and receive a $50 certificate towards the purchase of either board.

NO HIDDEN CHARGES! Performics Inc., prices include dynamic burn in, insurance and UPS ground delivery within the continental U.S.

TERMS: Certified check, COD, Visa/Mastercard Volume discounts available.

Performics continues to deliver high performance products and excellent service at realistic prices.

S-RAM 128 and S-RAM 256 are trademarks of PERFORMICS INC.

MasterCard VISA

PERFORMICS INC.

ORDER NOW!
(603) 881-8334

11 Morning Dove Rd. • Kingston, New Hampshire 03848

TAKE THE PAIN OUT OF BACKUP

Famous Excuses For Not Doing Backups

Backing up a Winchester onto multiple floppies takes forever.

PIP makes you remember what you changed — too much trouble.

Disk failures only happen to the other guy.

Floppy-To-Floppy Incremental Backup For Only $40.

Like PIP but copies just the files that have changed.

Order Qbax1.

The Remedy: Copy Just What's New — Automatically!

Qbax2 will:
Split files bigger than one floppy.
Track all the pieces.
Tell you which floppy it needs and when.
Give you a built-in catalogue.
Reclaim wasted floppy space.

Qbax2 only $95 from:

Amanuensis, Inc.
R. D. 1, Box 236
Grindstone, PA 15442
(412) 785-2806

For CP/M 2.2 on 8" SSSD and popular 5¼" formats. Shipping: $2 U.S. and Canada, $4 overseas. Qbax TM Amanuensis, Inc. CP/M® Digital Research.

FOR EVERYONE WITHOUT A DISPENSATION FROM MURPHY'S LAW

THE PROGRAMMER'S SHOP™

helps save time, money and cut frustrations. Compare, evaluate, and find products.

SERVICES

- **Programmer's Referral List**
- **Compare Products**
- **Help find a Publisher**
- **Evaluation Literature free**
- **BULLETIN BOARD - 7 PM to 7 AM 617-826-4086**
- **Dealer's Inquire**
- **Newsletter**
- **Rush Order**
- **Over 700 products**

Free Literature - Compare Products

Evaluate products Compare competitors. Learn about new alternatives. One *free call* brings information on just about any programming need. Ask for any ***"Packet"*** or "Addon Packet": ☐ ADA, Modula ☐ "AI" ☐ BASIC ☐ "C" ☐ COBOL ☐ Editors ☐ FORTH ☐ FORTRAN ☐ PASCAL ☐ UNIX/PC or ☐ Debuggers, Linkers, etc.

RECENT DISCOVERIES

SMALL TALK for PCDOS - "Methods" has objects, windows, browser, inspector. PCDOS $239

"C" LANGUAGE

	OUR PRICE
MSDOS C86-8087, reliable	call
Instant C - Inter., fast, full	445
Lattice C - the standard	call
Microsoft C 3.0 - new	279
Williams - debugger, fast	call
Wizard C - full, fast	450
CPM80 - EcoPlus C - faster, SLR	275
BDS C - solid value	125
MACINTOSH: Hippo II	375
Megamax - optimizer, full	275
Consulair's MAC C, toolkit	395

Compare, evaluate, consider other Cs

BASIC

	RUNS ON	
BASCOM-86 - MicroSoft	8086	279
CB-86 - DRI	CPM86	419
Data Manager-full source	MSDOS	325
CADSAM-Full Btree, source	MSDOS	150
InfoREPORTER-multifile	PCDOS	115
Prof. Basic-Inter., debug	PCDOS	89
SCREEN SCULPTOR	PCDOS	115
TRUE BASIC - ANSI	PCDOS	125

Ask about ISAM, other addons for BASIC

SERVICE

ALL PRODUCTS - We carry 700 products for MSDOS, CP/M 86, CP/M 80. Macintosh and key products for other micros.

EDITORS Programming

	RUNS ON	OUR PRICE
BRIEF - Intuitive, flexible	PCDOS	195
C Screen with source	86/80	75
Epsilon - like EMACS	PCDOS	195
FINAL WORD-for manuals	86/80	215
PMATE - powerful	8086	185
VEDIT - full, liked	86/80	119
XTC - multitasking	PCDOS	95

COBOL

Dig. Res-decent	MSDOS	525
Macintosh COBOL - Full	MAC	1850
MBP-Lev II, native, screen	MSDOS	885
Micro Focus Prof. - Full	PCDOS	call
Microsoft - Lev II, no royal	MSDOS	500
Ryan McFarland - portable	MSDOS	695

Ask about program generators.

LANGUAGE LIBRARIES

GRAPHICS: Halo for Turbo Postal	MSDOS	95
GRAPHMATIC - 3D, FTN, PAS	PCDOS	125
MultiHALO - fast, full-all lang.	PCDOS	220
File MGNT: BTrieve-all lang.	MSDOS	215
CIndex + - source, no royal	86/80	369
CTree - source, no royal	ALL	369
dBC ISAM by Lattice	8086	229
dB VISTA - "Network" Structure	MSDOS	465
PHACT - up under UNIX, addons	MSDOS	225
OTHER: C Utilities by Essential	MSDOS	129
Greenleaf - 200 +	MSDOS	159
SOFT Horizons - Blocks I	PCDOS	139
SCREEN: CURSES by Lattice	PCDOS	125
MetaWINDOW - icons, clip	PCDOS	139
PANEL - many lang. term	MSDOS	249
ProScreen - windows, source	PCDOS	415
Turbo V - Greenleaf C, fast	PCDOS	159
Windows for C	MSDOS	175

FORTRAN

	RUNS ON	OUR PRICE
MS FORTRAN-86 - Impr.	MSDOS	239
DR Fortran-86 - full '77	8086	249
PolyFORTRAN-XREF, Xtract	PCDOS	165

OTHER PRODUCTS

Advanced Trace 86 - Symbolic	PCDOS	149
Assembler & Tools - DRI	8086	159
Atron Debugger for Lattice	PCDOS	395
C Helper: DIFF, xref, more	86/80	135
CODESMITH-86 - debug	PCDOS	129
MacASM - full, fast, tools	MAC	115
MBP Cobol-86 - fast	8086	885
MicroProlog - improved	MSDOS	185
Micro: SubMATH - FORTRAN full	86/80	250
Microsoft MASM-86	MSDOS	125
Multilink - Miltitasking	PCDOS	265
PC FORTH - well liked	MSDOS	219
Pfinish - Profile by routine	MSDOS	345
PFIX-86 Debugger	MSDOS	169
PL1-86	8086	495
Polylibrarian - thorough	MSDOS	95
PolyMAKE	PCDOS	95
TRACE86 debugger ASM	MSDOS	115
ZAP Communications-VT100, TEK 4010, full xfer	PCDOS	65

Call for a catalog, literature, and *solid value*

800-421-8006

THE PROGRAMMER'S SHOP™

128 MRockland Street, Hanover, MA 02339

Visa Mass: 800-442-8070 or 617-826-7531 MasterCard 8517

Note: All prices subject to change without notice
Mention this ad. Some prices are specials
Ask about COD and **POs**. All formats available
UNIX is a trademark of Bell Labs.

Structured Programming With The Microsoft M80 Assembler

by Dennis N. Quinn

A conscious decision to implement a computer program using assembly language must be tempered with the knowledge that the assembler contains no elementary control structures to help the developer. Those structures which are often taken for granted in high-level languages are usually absent at the assembly-language level. For example, the Pascal language permits the following construction:

```
if ((a => 'a') and (a <= 'z'))
then
a = chr (ord (a) - 32);
```

Examination of the above code will reveal that the contents of the string a is tested in order to determine if it is in the range 'a' through 'z' inclusive; if so, 32 is subtracted from the ordinal position of the character in the underlying character set; the result is converted back into a character and replaces the original value in variable a. The net effect of this instruction is to convert the 1-byte contents of a to upper-case (assuming that this instruction is operating on a machine which uses the ASCII character-set).

Use of certain language structures to control logic flow of assembly language programs

In assembly-language, this example becomes less easy to read (and certainly less easy to write). The same results as the above example can be obtained with the following assembly-language routine (Zilog mnemonics are used):

```
a:        defb   0
upshift:  ld     a, (a)
          cp     'a'
          jp     c, ok_as_is
          cp     'z'
          jr     c, shift_it
          jp     nz, ok_as_is
shift_it: sub    20h
ok_as_is: ld     (a), a
```

While the sequence written in assembly-language will undoubtedly execute faster than the Pascal example, most of the appeal of a structured language is its inherent readability. At times, though, the speed of assembly-language is required. What can be done?

Control Structures

Most languages which support structured programming techniques provide the following five control structures as basic elements of the language:

1) The *IF* structure, which permits the testing of a boolean (i.e, true or

false) value. If the boolean value is true, one set of instructions is performed; otherwise, a different set of instructions is performed.

2) The *REPEAT* structure, which causes a set of instructions to be performed until a certain boolean value becomes false. The set of instructions, called the *range* of the structure, is executed at least once because the boolean value is not tested until the end of the structure.
3) The *WHILE* structure, which causes a set of instructions to be performed until a boolean value becomes false. The range of the structure is not executed at all if the boolean value starts out with the value false, because the value is tested at the beginning of the structure.
4) The *CASE* structure, which is similar to the *IF* structure except that more than two sets of instructions are presented, each associated with a certain value of the controlling variable. The set of instructions which corresponds to the actual value of the control variable will be executed and the remainder of the sets will be skipped. Many implementations of the *CASE* structure allow the user to provide a default set of instructions which will be executed in the event that the value of the control variable fails to match any of the specified cases.
5) The *FOR* structure, which is similar to the *WHILE* structure except that a control variable is initialized to a specified starting value, tested against a specified termination value, and (provided the termination value has not been exceeded) the set of instructions (the "range") is executed. At the completion of the range, the control variable is incremented by a specified value and the test is again made against the termination value. This process continues until the termination value has been reached or exceeded. Like the *WHILE* structure, it is possible for the range not to be executed at all.

With these five control structures, considerable power can be realized in the coding of a computer program. I have used the Pascal language since 1975 for most of my development, and I cannot recall a single instance where I have felt the need for a *GOTO* verb.

When I purchased Microsoft's FORTRAN compiler for my Kaypro-IV, I did so because the first computer language I ever learned was FORTRAN and I felt a kind of nostalgic longing for the old language. Microsoft's F80 is very similar to the General Electric GE-635 Fortran-64 I learned in college many years ago, and although the language immediately came back to me, somehow I felt uncomfortable using it. I greatly missed the language structures of Pascal, and I thought perhaps that the not insignificant investment I had made was a bad one.

Fortunately for me, Microsoft included the M80 assembler with the F80 compiler (primarily because the FORTRAN compiler produces assembly-language source as its output). I had done quite a bit of development with the Intel-8080 assembler, and I welcomed the opportunity to learn the Zilog Z80 instruction set and do some really powerful and fast programs in assembly-language. The only problem was that assembly-language has the same limitations as does FORTRAN — no structures for controlling logic flow!

Possible Solutions

Given that I wanted to write programs in assembly-language but that I also wanted to use control structures, what could I do?

Buy a C compiler: Most *C* compilers output assembly-language source code, and everybody knows that *C* is a magnificent language for producing excellent structured programs which are fast and efficient. Unfortunately, good *C* compilers produce mountains of assembly-language source even for a simple program. Couple that with the runtime support subroutines required, and *C*'s desirability rapidly falls off.

Write a custom compiler: While this might be an interesting solution, developing the type of compiler I would require to produce efficient assembly-language would not be a simple project. Even rudimentary compilers are extremely difficult to write, and heaven knows people have been trying since the earliest days of computers to design a compiler which could produce code as efficiently as a mediocre assembly-language coder. Besides, creating your own custom language is somewhat limiting — nobody but you will use it, even if it is really good. I ruled this option out without even giving it much thought.

Write a custom assembler: This is certainly a better solution than a custom compiler, but good macro assemblers still take quite a lot of time to develop, though not as much time as a compiler. The return on my time investment would not be very good, however, since it would probably take me at least six months to finish the assembler and it would be as limiting as producing a custom compiler.

Buy a pre-processor: There are a number of products on the marketplace which serve as pre-processors for an assembler, permitting the program author to write assembly-language programs using certain of the five basic control structures I presented above. After writing the program, the programmer runs the program through the pre-processor, which translates the control structures into elementary assembly-language statements and includes them at the appropriate places in its output. The output, after the pre-process step, is then assembled using an ordinary assembler (such as ASM or M80).

There are a couple of disadvantages to this approach. First, you have to purchase the pre-processor, and even though they are not expensive, my funds are limited. Second, the pre-processor requires disk space, and that is at a premium on most microcomputers. Third, the output from the pre-processor is another source program, again requiring disk space. And finally, a non-trivial amount of time is spent passing the original source program through the pre-processor.

Write my own pre-processor: This is even a better solution, but the only advantage I gain over the off-the-shelf pre-processors is that my investment is not monetary. The disk space and processing time disadvantages are still there.

Write some macros for the M80 assembler: Now we're getting somewhere. I'm not against building tools, but the return on investment must be good. I'll write a program, even a non-trivial program, if I will use it often enough to recover the time it took me to write it. I figured that an evening or two would be sufficient to produce the macros I would need, and that investment was worth it.

Implementing the Solution

The Microsoft M80 macro-assembler has a pseudo-operation code* which associates a value with a symbol, **DEFL**. In this respect, it is identical to the common pseudo-operation code **EQU**; however, there is one small difference. Once the value of a symbol is set with **EQU**, it cannot be changed. The **DEFL** operation, however, allows the programmer to change the value associated with a symbol as often as needed.

*Actually, there are three synonymous pseudo-operation codes, **SET**, **DEFL**, and **ASET**. The **SET** code cannot, however, be used in the Z80 mode since there is an instruction with that mnemonic in the Z80 instruction-set. Since **DEFL** is the preferred Zilog pseudo-operation code, that name will be used throughout this document.

Why, then, would anybody use **EQU** if **DEFL** is so much more powerful? Well, for the same reason that programmers use constants in a program. Once a constant is identified as such, another programmer reading the program knows that the value will not be modified, while variables with a stated starting value could be changed, so the reader must be alert to possible changes to the value of the variable.

The **DEFL** pseudo-operation does for symbol values what variable-names in a program do for data-processing. And because the M80 assembler permits symbol names to be constructed by means of the concatenation operator "&," a pair of stacks could be incorporated into the assembly process which would permit the production of nested control structures.

Basically, there are two stacks, named ??STK.x and ??TYP.x (the **x**'s represent a decimal value 0 through 15). The question marks in the name were chosen so as to make the names of the stacks unique in the program. The programmer simply avoids coding any symbols beginning with a question mark, and he or she can be certain that no duplicate symbols will be created.

Another symbol, ??PTR, is used to keep track of the current place in the two stacks. ??PTR is initially assigned the value 0, pointing to the ??STK.0 and ??TYP.0 symbols. It is incremented when values are pushed into the stacks, and it is decremented when values are removed from the stacks. How these stacks are used will be made clear in a moment.

There are sixteen possible positions in each of the stacks, although this is an arbitrary limit. Since the implementation method used in generating code for the various control structures never uses more than two levels of the stacks, the programmer can be assured that he or she can nest the control structures at least eight levels deep. If that is not sufficient, a way ofincreasing the number of entries in the stacks will be presented later. It has been my experience that I have never used more than seven or eight of the available positions in the stacks.

The symbol ??SYM is used to produce custom symbols. The initial value of ??SYM is zero; it is incremented each time a symbol is needed. For example, the macros which implement the **IF** control structure need a symbol to mark the beginning of the true branch, the beginning of the false branch, and the end of the false branch.

As a structure macro begins processing, it increments ??SYM and pushes the resulting value into the ??STK stack. If more than one symbol is needed, ??SYM will be incremented again, and again the resulting value will be pushed into the ??STK stack. Each of the structures is associated with a specific control number according to the structure type. The structure macro pushes the control number onto the ??TYP stack as it pushes ??SYM values onto the ??STK stack. These two stacks are constantly kept in synchronization; if a structure macro pushes two symbols onto the ??STK stack, it pushes the same control number twice onto the ??TYP stack.

At least two macros are involved in implementing a control structure, because control structures have at the very minimum a starting point and an ending point. For example, the **IF** structure is implemented with five macros: **WHEN**, **IS**, **THENDO**, **ELSEDO**, and **EWHEN**. Each of these macros tests the entry at the current level of the ??TYP stack. If the value found there is not the proper control number for the **IF** structure, the programmer has failed to close a nested structure properly. No effort is made to patch the stack, since the macro has no way of knowing where the closure was omitted; rather, it generates an error message to inform the programmer of the problem.

Several macros, each beginning with "??," are used internally to manipulate the stacks. None of these macros actually generates any object code. Some of them simply test for error conditions, such as improper structure closure, stack underflow and overflow, etc. Other macros are included for the purpose of optimizing the "jp's" generated within the structures and will not ordinarily be used by the programmer (although they could be, if the programmer so desired; doing so, however, is tantamount to using a **GOTO** -- slightly gauche and to be avoided if possible).

The Control Structures

Before I present the actual macros, I would like to introduce the macros you will be using in your assembly-language programs. There are four sets, corresponding to the **IF**, **REPEAT**, **WHILE**, and **CASE** structures of Pascal. The **FOR** structure was not implemented, not because it was difficult to implement, but because the **FOR** loop is actually a variation of the **WHILE** structure and as such would be redundant.

We should probably also take a minute to discuss the way the Zilog Z80 microprocessor handles comparisons. Two flags are significant in comparison operations, the Carry flag and the Zero flag. All comparisons are performed with *unsigned 8-bit* operands. In any comparison, there are three possible outcomes:

1) The two operands are equal. In this case, the Zero flag will be set and the Carry flag will be reset.
2) The accumulator is less than the other operand. In this case, the Zero flag will be reset and the Carry flag will be set.
3) The accumulator is greater than the other operand. In this case, both the Zero flag and the Carry flag will be reset.

If it is your intention to treat the two operands as *signed 7-bit operands*, you must test the signs of the operands before making the comparison. If the signs are the same, the comparison may be performed and the flags will be set correctly. On the other hand, if the signs are different, the Carry flag must be complemented after the comparison. *You* are responsible for ensuring that the Carry flag properly reflects the type of comparison you intended, since none of the control structure macros can determine the original signs of the operands once the comparison has been performed.

The **IF** *structure*: The **IF** structure has four parts: first, the statement of the condition which is to be tested; second, the mark which identifies the beginning of the true-branch; third, the mark which identifies the beginning of the false-branch (and, by default, the ending of the true-branch); and fourth, the mark which identifies the ending of the false-branch.

The **IF** structure is complicated somewhat by the possibility that multiple conditions must be tested (see, for instance, the example I presented at the beginning of this article). Some means of continuing the condition testing over multiple statements must therefore be provided. Another complication is that the false-branch of the **IF** structure is optional; in this case, the mark which ordinarily identifies the beginning of the false-branch would be omitted and the ending mark must be taken to identify the end of the true-branch under these circumstances.

As it was mentioned earlier, there are five macros which together implement the **IF** structure. Each will be presented individually.

WHEN — In their infinite wisdom and 20-20 foresight, Microsoft snatched up the "IF" mnemonic for their own use. That's ok, though; WHEN works just fine and expresses the same thought. The WHEN macro introduces

the first condition test of the **IF** structure. It takes two operands; the first is required, and the second is optional.

The first operand expresses the actual condition desired. Rather than require the programmer to use the condition codes actually implemented in the Z80 hardware, the following condition mnemonics (in either upper or lower case) may be expressed:

eq or ze	equality or zero (interchangable)
ne or nz	inequality or nonzero (interchangable)
lt	less-than
le	less-than or equal
gt	greater-than
ge	greater-than or equal
cy	carry flag set
nc	carry flag not set
po or nv	parity odd or no overflow (interchangable)
pe or ov	parity even or overflow (interchangable)

The second operand is used to mark the fact that a compound conditional is desired. The value stated for the second operand must be in lower-case (an idiosyncrasy of the M80 assembler). Two values are permitted:

and	If the current condition is true **and** the next stated condition is also true, execute the true-branch; otherwise, execute the false-branch.
or	If the current condition is true **or** the next stated condition is true, execute the true-branch; if both stated conditions are false, execute the false-branch.

IS — The IS macro introduces the second (and subsequent) condi-tions in a compound condition of the **IF** structure. IS takes exactly the same operands as WHEN.

It should be pointed out here that if a compound conditional is expressed, the generated code takes its earliest possible exit to either the true or the false branch, testing no more conditions than are actually required to determine the truth of the conditional expression.

THENDO — The THENDO macro introduces the beginning of the true branch of the **IF** structure. This macro is always required and takes no operands.

ELSEDO — The ELSEDO macro marks the end of the true branch and the beginning of the false branch of the **IF** structure. This macro is optional; if omitted, the end of the true branch is marked by means of the EWHEN macro.

EWHEN — The EWHEN macro marks the end of the **IF** structure. If a false branch was included, the EWHEN marks the end of the false branch; otherwise, it marks the end of the true branch. The EWHEN macro is always required and takes no operands.

The **REPEAT** *structure*: The **REPEAT** structure has two parts: first, the beginning of the range which is to be executed repeatedly; and second, the statement of the condition which must be satisfied in order for the loop to be terminated.

There are two macros which together implement the **REPEAT** structure. Each will be presented individually.

REPEAT — The REPEAT macro introduces the beginning of the range of the **REPEAT** structure. This macro is always required and takes no operands.

UNTIL — The UNTIL macro marks the end of the range of the **REPEAT** structure. This macro is always required and takes one required operand. The operand expresses the actual condition desired for termination of the loop. The condition mnemonics (in either upper or lower case) are the same as for the **WHEN** macro. Compound conditionals are not implemented for terminating the **REPEAT** structure since this feature is seldom required.

The **WHILE** *Structure*: The **WHILE** structure has two parts: first, the statement of the condition which must be satisfied before the loop will be executed; and second, the end of the range of statements which are repeatedly executed.

LATTICE® WORKS

LATTICE, INC. OFFERS C COMPILERS

The industry's standard C compiler, Lattice C, is now published directly by Lattice. You can obtain editions of Lattice C from other publishers, but when you purchase our edition, you get support directly from the people who wrote the C compiler. And when you register your purchase with Lattice, you are notified of all updates and enhancements. You will also be notified of the new C programming tools as they become available from Lattice.

When you are ready to purchase a C compiler, consider the source. Then call Lattice, Inc.

LATTICE OFFERS C COMPILER UPGRADES TO MICROSOFT USERS

Lattice announces the opportunity for users of the Microsoft MS-DOS C compiler to upgrade to the standard Lattice C and obtain a free copy of Lattice's new C-SPRITE symbolic program debugger.

According to Dave Schmitt, Lattice, Inc. President, "Until recently, the Microsoft C compiler was an early version of our compiler. Now that Microsoft has switched to their own compiler, we feel that purchasers of the earlier product deserve a low-cost way to remain compatible with the latest Lattice technology and the extensive third-party support for the Lattice product."

The upgrade kit costs $150 and includes Version 2.20 of the Lattice C compiler with its new two-color, typeset manual and the C-SPRITE program debugger. (The regular retail prices are $500 for the compiler and $175 for C-SPRITE.)

This offer to Microsoft C users expires July 31. Upgrade orders must be accompanied by the original Microsoft diskettes.

LATTICE C USERS' GROUP FORMED

An independent Lattice C Users' Group has recently been formed, and is headed by Bill Hunt, author of the book, *The C Toolbox*. The group will serve users of Lattice C, but membership is open to any C compiler users.

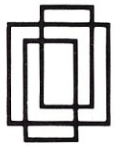
Members will receive a bi-monthly newsletter explaining C features and providing examples of their use, exploring new ways to use Lattice C, and will include a question and answer column. The newsletter will be supplemented by a disk containing source files, demo programs and library functions, and the best of the public domain C programs.

Lattice, Inc. will provide complete support for the new users' group by sharing new plans, discovered bugs, and "tech" tips.

Register your C compiler now to receive complete information on joining the Lattice C Users' Group.

ASK ABOUT OUR "TRADE UP TO LATTICE C POLICY"

After purchase, return registration cards for free subscription to the "Lattice Works" newsletter and important information about the Lattice Users Group.



Lattice, Inc.
P.O. Box 3072
Glen Ellyn, IL 60138
(312) 858-7950
TWX 910-291-2190

International Sales Offices
Belgium: Softshop. Phone: (32) 53-664875.
England: Round Hills. Phone: (0672) 54675.
Japan: Lifeboat Japan. Phone: (03) 293-2311.

Instant-C™: The Best Value In C Programming Tools

The edit-compile-link-test-debug cycle that takes tens of minutes with compilers and linkers is only seconds with the *Instant-C* interpreter. Yet it runs your programs **50 to 500 times faster** than conventional C interpreters! You get the best of both compilers **and** interpreters. Only *Instant-C* is a complete, integrated environment for creating, testing, and running your programs.

Instant-C gives you **all** of these proven capabilities in one tightly integrated package:

interpreter—*Instant-C* runs your programs faster than some compilers; has direct execution; full K&R

compiler—*Instant-C* can make stand-alone programs

full-screen language editor—shows syntax errors with cursor set to trouble spot

C source debugger—single-step, breakpoints, stack trace, more

run-time checker—validates pointer refs, array bounds, more

C source formatter—save editing time, find logical flaws

standard library with source—for best portability

linker—work with multiple source modules

Lint—extensive compile-time validation

The cheapest available examples of these tools would cost $800 (and they don't even work together). You could spend close to $3000 to get the best product of each kind, but you'd have ten times the complexity, filling megabytes of disk. *Instant-C* is faster: it performs these functions automatically. *Instant-C* is far more than the sum of its parts.

Instant-C is all of these capabilities in one package, fits on a single floppy disk, is full K&R, works on IBM PC's, compatibles, and others under DOS or CP/M-86. It costs only $495.

Instant-C is the best value in C programming tools. Guaranteed, or your money back for any reason in first 31 days.

Rational
Systems, Inc.

(617) 653-6194
P.O. Box 480
Natick, MA 01760

Instant-C is a trademark of Rational Systems, Inc.

There are two macros which together implement the **WHILE** structure. Each will be presented individually.

WHILE — The WHILE macro marks the beginning of the range of the **WHILE** structure. This macro is always required and takes one required operand. The operand expresses the actual condition which must be met in order for the range to be executed. The following condition mnemonics (in either upper or lower case) are the same for the **WHILE** macro. Compound conditionals are not implemented for the **WHILE** structure since this feature is seldom required.

EWHILE — The EWHILE macro marks the ending of the range of the **WHILE** structure. This macro is always required and takes no operands.

The **CASE** *Structure*: The **CASE** structure is a multi-way implementation of the **IF** structure and as such is somewhat redundant. Its usefulness, however, outweighs its redundancy, particularly where readability is important. There are four major parts in the **CASE** structure: first, the indication of the beginning of the **CASE** structure; second, the indication of the beginning of each of the select range groups and the associated value for that range; third, the indication of the beginning of the default range group; and fourth, the indication of the ending of the **CASE** structure.

There are four macros which together implement the **CASE** structure. The implementation of the **CASE** structure is unusual in that, while the other three structures assume that the programmer will perform the required comparisons prior to including the macro which tests the condition, the **CASE** implementation macros do all of the comparing for the programmer. Prior to coding the **CASE** structure, the programmer loads the accumulator register with the control variable; each of the select statements within the **CASE** structure then state a value which, if the accumulator contents matches the value stated, the range will be executed; otherwise, control will continue with the next select statement in the structure. Finally, if none of the specified values matches the value in the accumulator, the default range (if one is included) will be executed. The implementation of the **CASE** structure also automatically branches to the end of the structure after the code within the selected (or defaulted) range has been executed.

Graphically, this may be illustrated as follows:

```
beginning of CASE structure
  first selector
    statement-1
    statement-2
    statement-3
  second selector
    statement-4
    statement-5
    statement-6
  third selector
    :
    :
  n-th selector
    statement-i
    statement-j
    statement-k
  default selector
    statement-x
    statement-y
    statement-z
ending of CASE structure
```

Each of the four macros which implement the **CASE** structure will be presented individually.

CASE — The CASE macro introduces the **CASE** structure. This macro is always required and takes no operands.

SELECT — The SELECT macro terminates the previous select range

group and specifies the value which will select the following range group provided that value matches the value in the accumulator. There must be at least one SELECT macro for each **CASE** structure. The SELECT macro takes one required operand. The operand may be any of the following:

(HL) This expression indicates that the HL-register pair points to a memory byte containing the value which is to be compared against the value in the accumulator.

(IX+nn) This expression, where **nn** is a number in the range -128..127, indicates that the memory byte found at the address formed by the sum of the address in the IX-register added to the specified offset is to be compared against the value in the accumulator.

(IY+nn) This expression, where **nn** is a number in the range -128..127, indicates that the memory byte found at the address formed by the sum of the address in the IY-register added to the specified offset is to be compared against the value in the accumulator.

A This expression indicates that the contents of the A-register (the accumulator) are to be compared against the value in the accumulator. Because a match will always occur, this expression is trivial.

B This expression indicates that the contents of the B-register are to be compared against the value in the accumulator.

C This expression indicates that the contents of the C-register are to be compared against the value in the accumulator.

D This expression indicates that the contents of the D-register are to be compared against the value in the accumulator.

E This expression indicates that the contents of the E-register are to be compared against the value in the accumulator.

H This expression indicates that the contents of the H-register are to be compared against the value in the accumulator.

L This expression indicates that the contents of the L-register are to be compared against the value in the accumulator.

nn This expression, where **nn** is a decimal value between -128 and 255 inclusive, a hexadecimal value between 00 and FF, a symbol having a value in the same range, a single character enclosed in apostrophes, or a formula which will result in a value between -127 and 255 inclusive, indicates that the specified value is to be compared against the value in the accumulator.

If the expression coded as the operand of the SELECT macro contains any spaces, the Microsoft assembler requires that the operand be enclosed in angle brackets: '<' and '>.' The usual expression for this operand is a decimal or character constant, but the (IX+nn) and (IY+nn) expressions look like they could be used in interesting ways.

DEFAULT — The DEFAULT macro terminates the previous select range group and marks the beginning of the default range group. The DEFAULT macro is optional; if included, the statements in the default range will be executed provided none of the select ranges were executed. The DEFAULT macro takes no operands.

ECASE — The ECASE macro terminates the default range (if a DEFAULT macro was included) and marks the end of the **CASE** structure. This macro is always required and takes no operands.

Using the Structures

In order to use the structure macros described in this article, you should first build a file containing all of the macro prototypes shown in the accompanying listing. While that's a lot of typing, you will have to type them only once. You can omit the comments from the macros as you type them in if you want to, which will cut down on the amount of typing you have to do (the comments can be identified by the appearance of a pair of semicolons — everything following the semicolons is a comment); keep this article around, though, in case you need to find out how a particular macro works.

Next, when you write an assembly-language program in which you want to use these control structure macros, include the **.Z80** pseudo-operation code near the beginning of your module. (These macros generate Zilog mnemonics and use a few instructions which don't exist on the 8080. The macros could be modified to generate Intel 8080 instructions instead of Zilog instructions, but that is left as an exercise for the reader.)

After the **.Z80** pseudo-operation code, you use the **INCLUDE** pseudo-operation code* to bring the library into your module, then code the **EASIN** macro (no operands required) to produce the assembler symbols required for the other macros. (EASIN does not generate any executable code; all it does is declare the necessary symbols. It does not, therefore, have to be placed in an "executable" part of your program. Just put it somewhere after the **INCLUDE** of the macros but before you begin using them.)

Now we get to the good part — actually using the control structures. For practice, let's write our original example using the control structures, side-by-side with the assembly-language version:

Original version in assembly-language

```
a:        defb    0
upshift:  ld      a, (a)
          cp      'a'
          jp      c, ok_as_is
          cp      'z'
          jr      c, shift_it
          jp      nz, ok_as_is
shift_it: sub     20h

ok_as_is: ld      (a) a
```

Same thing using structures

```
a:        defb    0
upshift:  ld      a, (a)
          cp      'a'
          when ge, and
          cp      'z'
          is le
           thendo
            sub   20h
          ewhen
          ld      (a) a
```

Although the structured version is one line longer, the ability to use 'ge' and 'le' to express the conditions makes the code much more self-explanatory. Moreover, the proper use of indenting permits the subordinate nature of the subtraction to be clearly indicated. How about some other examples?

Here's a sequence which you can use to copy a block of data from one place in memory to another. While the **LDIR** instruction is much faster, this

*As is customary with Microsoft, you can use **$INCLUDE**, **MACLIB**, or **INCLUDE** interchangeably according to your personal preference.

exemplifies the use of a **REPEAT** structure:

```
d       bc, count;      BC := number of bytes.
ld      de, sending;    DE -> sending field.
ld      hl, receiving;  HL -> receiving field.
repeat
  ld      a, (de);        Get char to move.
  ld      (hl), a;        Put it away.
  inc     de;             Bump sending and
  inc     hl;             receiving addresses.
  dec     bc;             Decrement counter
  ld      a, h;           and test completion.
  or      l;
  until ze  ;
```

Here's an example of the **WHILE** structure, which sends a string to the CRT until a null (hex 00) character is encountered. Another macro, SVC, is used to invoke the CP/M BDOS function 6 for doing the actual character transmission. This example is written as a subroutine. Upon receiving control, the string to be transmitted is pointed to by HL. If a null string is received by the subroutine, nothing is sent to the CRT.

```
xmit:
  ld      a, (hl);        Pick up 1st char to xmit.
  or      a;              Prime condition codes.
  while ne
    ld      e, a;           Put char into E,
    push    hl;             Preserve the pointer,
    svc     6;              and invoke BDOS.
    pop     hl;             Get back the pointer
    inc     hl;             and bump it.
    ld      a, (hl);        Get next char,
    or      a;              and test it.
  ewhile;
  ret;                      All done — return.
```

For our last example, let's use the **CASE** structure to handle a menu. This would probably be the main processing loop of a program, so we'll throw in a **REPEAT** structure to handle the loop. In our example, four typeins are permitted: "A" (upper or lower case), "B," "C," or "D," corresponding to routine_A, routine_B, routine_C, and routine_D respectively. If the operator types anything else, an error message will be displayed. Finally, the program terminates when the operator types control-C:

Conclusion

Anyone who has used a high-level structured language, such as Pascal, Algol, PL/1, or C, but who wants the speed afforded by assembly-language programming, can benefit from these macros. As can be seen from an examination of the coding examples, their use can greatly improve the readability of an assembly-language program, and the provisions for using mnemonic condition abbreviations such as "gt" or "le" simplifies debugging by improving the self-documentation qualities of the program.

I mentioned earlier that I would reveal how to increase the number of entries in the stacks should sixteen be insufficient. There are only three changes that must be made, all to the EASIN macro. First, change the constant in the ??STKL DEFL 16 line to the desired value. Second, change the two IRP statements by including integers separated by commas after 16 and before the ">."

For example, if you want to increase the two stacks to twenty levels each, change the ??STKL DEFL line to ??STKL DEFL 20 and change the two IRP lines to " . . . ,15,16,17,18,19,20>."

Dennis Quinn is an independent consultant specializing in microcomputers and office automation. He has been a computer analyst/programmer since 1965 and lives at 4114 Berkshire, Royal Oak MI 48073.

```
mainline:
    repeat
        ld      hl, prompt_msg;     Send prompt
        call    xmit;                to CRT.
        repeat
            repeat
                svc     6, 0ffh;          Read keyboard
                or      a;
            until ne;                    until key rec'd.
            cp      'a';                 Make upper-case.
        when ge, and
        cp      'z';
        is le
          thendo
            sub     20h;
        ewhen;
        case
          select 'A';
            ld      (ok), a;              Set ok flag
            call    routine_a;
          select 'B';
            ld      (ok), a;
            call    routine_b;
          select 'C';
            ld      (ok), a;
            call    routine_c;
          select 'D';
            ld      (ok), a;
            call    routine_d;
          select 3;                      Control-C:
            ld      (terminate), a;       Set end flag,
            ld      (ok), a;              Set ok flag.
          default;                       Invalid response:
            ld      hl, no_good_msg;      HL -> message.
            call    xmit;                 Send to CRT.
            xor     a;                    Reset ok flag.
            ld      (ok), a;
        ecase;
        ld      a, (ok);                Test ok flag.
        or      a;                      Nonzero means ok.
    until nz;                          Repeat until valid.
    ld      a, (terminate);            Test terminate flag.
    or      a;                         Nonzero means quit.
until nz;                             Repeat until Control-C.
jp      0;                            All done--quit.
prompt_msg: defm 0Dh, 0Ah, 'Enter selection (A,B,C,D) =>', 0
no_good_msg:defm 0Dh, 0Ah, 'Invalid response -- try again', 0
ok:         defb 0
terminate:  defb 0
```


C Sick?

PLZ is the cure!

Introducing a native code PLZ compiler for the 68000, featuring:

- ☐ Complete PLZ language, including structure assignment and comparison
- ☐ Fully compatible with Zilog Z80, Z8000 PLZ
- ☐ Ideal for embedded, ROM based systems
- ☐ Strongly typed
- ☐ Data types include signed and unsigned byte, word and longword
- ☐ All of the protection of Pascal, with the flexibility of C
- ☐ Inherently more portable than either Pascal or C
- ☐ Easy for Pascal or C programmers to learn
- ☐ Fully compatible with the CP/M-68K C library

Requires CP/M-68K. Other systems and CPU's supported soon.

Package includes:
68000 Compiler-Code generator
User Manual
Springer-Verlag "Report on the Programming Language PLZ/SYS"
One Year free updates

All this for the low introductory price of $75

Add $3 S/H. NJ Residents include 6% sales tax.

KCSystems
20 Lamington Drive, Succasunna, NJ 07876
(201) 927-9104

```
 .LIST
 .COMMENT *
+--------------------------------+
| MACRO LIBRARY ver 2.0e (83.288) |
+--------------------------------+
 .XLIST
EASIN   MACRO;;
??SYM   DEFL 0;                         Next generated symbol count
??PTR   DEFL 0;                         Pointer to stack
 IRP    x, <0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15>
??STK.&x DEFL 0;;
 ENDM
??STKL  DEFL   16;                      Stack limit
 IRP    x, <0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15>
??TYP.&x DEFL 0;;
 ENDM
??STK   DEFL   0;;
??TYP   DEFL   0;;
??TRUE  DEFL   0;;
??FALSE DEFL   0;;
??SEL   DEFL   0;;
 ENDM

EASOUT MACRO
;;
;; See if all structures are properly closed to this point in program
;; (may be included multiple times)
;;
    IF      ??PTR
        ERROR '*** CLOSURE INCOMPLETE ***'
    ENDIF
 ENDM

??DEFL MACRO A,B
A       DEFL    B
 ENDM

??LABEL MACRO TYPE, LABEL
??&TYPE&LABEL:;;
 ENDM

??NEXT MACRO
    ??DEFL  ??SYM, <(??SYM + 1)>;;
 ENDM

??INCR MACRO
    IFE     (??PTR - ??STKL);;          Stack overflow?
        ERROR '*** STACK OVERFLOW ***'
        EXITM
    ENDIF
    ??DEFL  ??PTR, <(??PTR + 1)>;;
 ENDM

??DECR MACRO
    IFE     (??PTR);;                   Stack underflow?
        ERROR '*** STACK UNDERFLOW ***'
        EXITM
    ENDIF
    ??DEFL  ??PTR, <(??PTR - 1)>;;
 ENDM
```

```
??STACK MACRO TYPE,SYMBOL
    ??DEFL  ??TYP, TYPE;;
    IFB     <SYMBOL>;;                  Is <symbol> coded?
        ??DEFL  ??STK, ??SYM;;            no--use current value of ??SYM.
        ??DEFL  ??STK.&%??PTR, ??SYM;;    push it onto ??STK stack.
      ELSE
        ??DEFL  ??STK, SYMBOL;;           yes--use supplied value.
        ??DEFL  ??STK.&%??PTR, SYMBOL;;   push it onto ??STK stack.
    ENDIF
    ??DEFL  ??TYP.&%??PTR, TYPE;;       Push <type> onto ??TYP stack.
    ??INCR;;                            Advance stack ponter.
 ENDM

??CLOSE MACRO TYPE
    ??DECR;;                            Decrement stack pointer
    ??DEFL  ??STK, ??STK.&%??PTR;;      Get symbol and
    ??DEFL  ??TYP, ??TYP.&%??PTR;;      type from stack
    IF      (??TYP-TYPE);;              Test valid closure
        ERROR '*** IMPROPER CLOSURE ***'
    ENDIF
 ENDM

??GETSYM MACRO TYPE
    ??DECR;;                            Decrement stack pointer
    ??DEFL  ??STK, ??STK.&%??PTR;;      Get symbol and
    ??DEFL  ??TYP, ??TYP.&%??PTR;;      type from stack
    IF      (??TYP-TYPE);;              Test valid closure
        ERROR '*** IMPROPER CLOSURE ***'
    ENDIF
    ??INCR;;                            Fix stack pointer
 ENDM

BREQ MACRO ADDR;;                       Branch if equal
 JP Z,ADDR;;
 ENDM

BRNEQ MACRO ADDR;;                      Don't branch if equal
 JP NZ,ADDR;;
 ENDM

BRZE MACRO ADDR;;                       Branch if zero
 JP Z,ADDR;;
 ENDM

BRNZE MACRO ADDR;;                      Don't branch if zero
 JP NZ,ADDR;;
 ENDM

BRNE MACRO ADDR;;                       Branch if not equal
 JP NZ,ADDR;;
 ENDM

BRNNE MACRO ADDR;;                      Don't branch if not equal
 JP Z,ADDR;;
 ENDM

BRNZ MACRO ADDR;;                       Branch if nonzero
 JP NZ,ADDR;;
 ENDM
```

```
BRNNZ MACRO ADDR;;                    Don't branch if nonzero
 JP Z,ADDR;;
 ENDM

BRLT MACRO ADDR;;                     Branch if less than
 JP C,ADDR;;
 ENDM

BRNLT MACRO ADDR;;                    Don't branch if less than
 JP NC,ADDR;;
 ENDM

BRGT MACRO ADDR;;                     Branch if greater than
 LOCAL X;;
 JR C,X;;
 JP NZ,ADDR;;
X:;;
 ENDM

BRNGT MACRO ADDR;;                    Don't branch if greater than
 JP Z,ADDR;;
 JP C,ADDR;;
 ENDM

BRLE MACRO ADDR;;                     Branch if less or equal
 JP Z,ADDR;;
 JP C,ADDR;;
 ENDM

 BRNLE MACRO ADDR;;                   Don't branch if less or equal
  LOCAL X;;
  JR C,X;;
  JP NZ,ADDR;;
 X:;;
  ENDM

 BRGE MACRO ADDR;;                    Branch if greater or equal
  JP NC,ADDR;;
  ENDM

 BRNGE MACRO ADDR;;                   Don't branch if greater or equal
  JP C,ADDR;;
  ENDM

 BRCY MACRO ADDR;;                    Branch if carry flag set
  JP C,ADDR;;
  ENDM

 BRNCY MACRO ADDR;;                   Don't branch if carry flag set
  JP NC,ADDR;;
  ENDM

 BRNC MACRO ADDR;;                    Branch if carry flag reset
  JP NC,ADDR;;
  ENDM

 BRNNC MACRO ADDR;;                   Don't branch if carry flag reset
  JP C,ADDR;;
  ENDM

 BRPO MACRO ADDR;;                    Branch if parity odd
  JP PO,ADDR;;
  ENDM

BRNPO MACRO ADDR;;                    Don't branch if parity odd
 JP PE,ADDR;;
 ENDM

BRPE MACRO ADDR;;                     Branch if parity even
 JP PE,ADDR;;
 ENDM

BRNPE MACRO ADDR;;                    Don't branch if parity even
 JP PO,ADDR;;
 ENDM

BROV MACRO ADDR;;                     Branch if overflow
 JP PE,ADDR;;
 ENDM

BRNOV MACRO ADDR;;                    Don't branch if overflow
 JP PO,ADDR;;
 ENDM

BRNV MACRO ADDR;;                     Branch if no overflow
 JP PO,ADDR;;
 ENDM

BRNNV MACRO ADDR;;                    Don't branch if no overflow
 JP PE,ADDR;;
 ENDM

BRUN MACRO ADDR;;                     Always branch
 JP ADDR;;
 ENDM

BRNUN MACRO ADDR;;                    Never branch
 ENDM

BR MACRO CCODE,TYPE,LABEL;;           Generate appropriate branch command
 BR&CCODE ??&TYPE&LABEL;;
 ENDM

WHEN    MACRO   CCODE, ANDOR
    ??NEXT;;                          Generate FALSE branch name
    ??DEFL  ??FALSE, ??SYM;;           and save in ??FALSE
    ??STACK 1;;                       Push FALSE branch
    ??NEXT;;                          Generate TRUE branch name
    ??DEFL  ??TRUE, ??SYM;;            and save in ??TRUE
    ??STACK 1;;                       Push TRUE branch
    IFB     <ANDOR>;;                 If <ANDOR> is null
        BR      N&CCODE, 1, %??FALSE;;    Reverse-cond jump to false
      ELSE;;                           otherwise,
        IFIDN   <ANDOR>, <and>;;        If <ANDOR> = 'and'
            BR      N&CCODE, 1, %??FALSE;;  Reverse-cond jump to false
          ELSE;;                             otherwise,
            IFIDN   <ANDOR>, <or>;;         If <ANDOR> = 'or'
                BR      &CCODE, 1, %??TRUE;;  Normal-cond jump to true
              ELSE;;                           otherwise error
                ERROR '*** EXPECTED "and"|"or", ENCOUNTERED "&ANDOR"'
            ENDIF
        ENDIF
    ENDIF
ENDM
```

```
IS      MACRO   CCODE, ANDOR
    ??CLOSE 1;;                         Pop true-branch name
    ??DEFL  ??TRUE, %??STK;;             and save in ??TRUE
    ??CLOSE 1;;                         Pop false-branch name
    ??DEFL  ??FALSE, %??STK;;            and save in ??FALSE
    ??STACK 1, %??FALSE;;               Push false-branch back
    ??STACK 1, %??TRUE;;                Push true-branch back
    IFB     <ANDOR>;;                   If <ANDOR> is null,
        BR      N&CCODE, 1, %??FALSE;;    Reverse-cond jump to false
      ELSE;;                            otherwise,
        IFIDN   <ANDOR>, <and>;;          If <ANDOR> = 'and'
            BR      N&CCODE, 1, %??FALSE;;  Reverse-cond jump to false
          ELSE;;                          otherwise,
            IFIDN   <ANDOR>, <or>;;         If <ANDOR> = 'or'
                BR      &CCODE, 1, %??TRUE;;  Normal-cond jump to true
              ELSE;;                          otherwise error
                ERROR '*** EXPECTED "and"|"or", ENCOUNTERED "&ANDOR"'
            ENDIF
        ENDIF
    ENDIF
 ENDM

THENDO  MACRO
    ??CLOSE 1;;                         Pop true-branch name
    ??DEFL  ??TRUE, %??STK;;             and save in ??TRUE
    ??LABEL 1, %??TRUE;;                Emit true-branch label
 ENDM

ELSEDO  MACRO
    ??NEXT;;                            Create symbol for EWHEN
    BR      UN, 1, %??SYM;;             Emit jump to EWHEN
    ??CLOSE 1;;                         Pop false-branch name
    ??LABEL 1, %??STK;;                 Emit label for false-branch
    ??STACK 1;;                         Push EWHEN name
 ENDM

EWHEN   MACRO
    ??CLOSE 1;;                         Pop EWHEN name
    ??LABEL 1, %??STK;;                 Emit label for EWHEN
 ENDM

REPEAT  MACRO
    ??NEXT;;                            Create name for statement block
    ??STACK 3;;                          and push it
    ??LABEL 3, %??SYM;;                 Emit label for statement block
 ENDM

UNTIL   MACRO   CCODE
    ??CLOSE 3;;                         Pop statement block name
    BR      N&CCODE, 3, %??STK;;        Emit reverse-condition jump back
 ENDM


IS      MACRO   CCODE, ANDOR
    ??CLOSE 1;;                         Pop true-branch name
    ??DEFL  ??TRUE, %??STK;;             and save in ??TRUE
    ??CLOSE 1;;                         Pop false-branch name
    ??DEFL  ??FALSE, %??STK;;            and save in ??FALSE
    ??STACK 1, %??FALSE;;               Push false-branch back
    ??STACK 1, %??TRUE;;                Push true-branch back
    IFB     <ANDOR>;;                   If <ANDOR> is null,
        BR      N&CCODE, 1, %??FALSE;;    Reverse-cond jump to false
      ELSE;;                            otherwise,
        IFIDN   <ANDOR>, <and>;;          If <ANDOR> = 'and'
            BR      N&CCODE, 1, %??FALSE;;  Reverse-cond jump to false
          ELSE;;                          otherwise,
            IFIDN   <ANDOR>, <or>;;         If <ANDOR> = 'or'
                BR      &CCODE, 1, %??TRUE;;  Normal-cond jump to true
              ELSE;;                          otherwise error
                ERROR '*** EXPECTED "and"|"or", ENCOUNTERED "&ANDOR"'
            ENDIF
        ENDIF
    ENDIF
 ENDM

THENDO  MACRO
    ??CLOSE 1;;                         Pop true-branch name
    ??DEFL  ??TRUE, %??STK;;             and save in ??TRUE
    ??LABEL 1, %??TRUE;;                Emit true-branch label
 ENDM

ELSEDO  MACRO
    ??NEXT;;                            Create symbol for EWHEN
    BR      UN, 1, %??SYM;;             Emit jump to EWHEN
    ??CLOSE 1;;                         Pop false-branch name
    ??LABEL 1, %??STK;;                 Emit label for false-branch
    ??STACK 1;;                         Push EWHEN name
 ENDM

EWHEN   MACRO
    ??CLOSE 1;;                         Pop EWHEN name
    ??LABEL 1, %??STK;;                 Emit label for EWHEN
 ENDM

REPEAT  MACRO
    ??NEXT;;                            Create name for statement block
    ??STACK 3;;                          and push it
    ??LABEL 3, %??SYM;;                 Emit label for statement block
 ENDM

UNTIL   MACRO   CCODE
    ??CLOSE 3;;                         Pop statement block name
    BR      N&CCODE, 3, %??STK;;        Emit reverse-condition jump back
 ENDM

WHILE   MACRO   CCODE
    ??NEXT;;                            Create begin name
    ??STACK 4;;                          and push onto stack
    ??LABEL 4, %??SYM;;                 Emit begin name
    ??NEXT;;                            Create end name
    ??STACK 4;;                          and push onto stack
    BR      N&CCODE, 4, %??STK;;        Emit reverse-condition jump to end
 ENDM

EWHILE  MACRO
    ??CLOSE 4;;                         Pop end name
    ??DEFL  ??FALSE, %??STK;;            and save it in ??FALSE
    ??CLOSE 4;;                         Pop begin name
    BR      UN, 4, %??STK;;             Emit unconditional jump to begin
    ??LABEL 4, %??FALSE;;               Emit end label
 ENDM

CASE    MACRO
    ??NEXT;;                            Generate ECASE label
    ??STACK 5;;                         Push ECASE label
    ??STACK 5, 0;;                      Push zero to mark first time.
 ENDM
```

```
SELECT  MACRO   VAL
    ??CLOSE 5;;
    ??DEFL  ??FALSE, %??STK;;
    ??CLOSE 5;;
    ??DEFL  ??TRUE, %??STK;;
    ??STACK 5, %??TRUE;;
    IF      ??FALSE;;
        ??STACK 5, %??FALSE;;
        BR      UN, 5, %??TRUE;;
      ELSE;;
        ??NEXT;;
        ??STACK 5;;
    ENDIF;;
    ??CLOSE 5;;
    ??DEFL  ??FALSE, %??STK;;
    ??CLOSE 5;;
    ??DEFL  ??TRUE, %??STK;;
    ??NEXT;;
    ??DEFL  ??SEL, %??SYM;;
    ??STACK 5, %??TRUE;;
    ??STACK 5, %??SEL;;
    ??LABEL 5, %??FALSE;;
   CP       VAL;;
   BR       NE, 5, %??SEL;;
 ENDM
DEFAULT MACRO
    ??CLOSE 5;;
    ??DEFL  ??FALSE, %??STK;;
    ??CLOSE 5;;
    ??DEFL  ??TRUE, %??STK;;
    ??STACK 5, %??TRUE;;
    IF      ??FALSE;;
        ??STACK 5, %??FALSE;;
        BR      UN, 5, %??TRUE;;
      ELSE;;
        ??NEXT;;
        ??STACK 5;;
    ENDIF;;
    ??CLOSE 5;;
    ??DEFL  ??FALSE, %??STK;;
    ??CLOSE 5;;
    ??DEFL  ??TRUE, %??STK;;
    ??STACK 5, %??TRUE;;
    ??STACK 5, %??FALSE;;
    ??LABEL 5, %??FALSE;;
 ENDM

ECASE   MACRO
    ??CLOSE 5;;
    ??CLOSE 5;;
    ??LABEL 5, %??STK;;
 ENDM

SVC     MACRO   CREG, EREG
    LD      C, CREG;;
    IFNB    <EREG>;;
        IFIDN   <EREG>, <(hl)>;;
            LD      E, (HL);;
          ELSE
            LD      DE, EREG;;
        ENDIF
    ENDIF
    CALL    5;;
 ENDM
```

```
Pop next SELECT label
 and save it in ??FALSE
Pop ECASE label
 and save it in ??TRUE
Push ECASE label back
If not first time,
  push SELECT label back and
  emit jump unconditional to ECASE;
 Otherwise,
  generate label for next select
  and put it into stack.

Pop this SELECT label
 and save it in ??FALSE
Pop ECASE label
 and save it in ??TRUE
Generate next SELECT label
 and save it in ??SEL
Push ECASE label
Push next SELECT label
Emit label for this SELECT
Emit comparison against <val>
Emit jump not equal to next SELECT


Pop next SELECT label
 and save it in ??FALSE
Pop ECASE label
 and save it in ??TRUE
Push ECASE label back
If not first time,
  push SELECT label back and
  emit jump unconditional to ECASE;
 Otherwise,
  generate label for next select
  and put it into stack.

Pop next SELECT label
 and save it in ??FALSE
Pop ECASE label
 and save it in ??TRUE
Push back ECASE label
Push back SELECT label
Emit SELECT label


Pop previous SELECT label
Pop ECASE label
Emit ECASE label


Emit load of BDOS code
E-register operand supplied?
  Yes:  Is it "(hl)"?
    Yes:  Emit load through (hl)

    No:  Emit load into DE


Emit call of BDOS.
```


BUFFERED I/O BOARD — Introductory Price * $59.95
With despool functions, protocols supported: XON/XOFF, ETX: ETB/ACK

80 CHARACTER VIDEO BOARD — * $49.50
25 Lines with status, compatible with Wordstar & dBase

Includes Bareboard, Heatsink & Documentation Call or write for more information.

Simpliway Products Co.
P.O. Box 601
Hoffman Estates, IL 60195
(312) 359-7337

OEM dealer pricing available, $3.00 S/H, IL. Res. add 7% tax
dBase™ - of Ashton - Tate Corp. — Wordstar™ - of Micropro Int'l. Corp

MasterCard VISA

NEW PRODUCTS

SERIAL ADAPTER FOR THE VDB-A2-

- Plugs into keyboard socket
- Direct connect to Keytronic (IBM) serial ASCII keyboard
- Internal speaker for BEEPER
- Tx port for non VDB-A2 use
- Hardware programmed Rx & Tx
- Dual baud-rate generator
- Bare-board OR assembled

DOUBLE-HEIGHT S-100 PROTOTYPING-

- Oversized heatsinks for HI-CURRENT reg's: + 5, + / – 12 V.
- Use with wire-wrap sockets OR direct solder connections
- GOLD plated edge connector
- NO PLATING CUTS required
- NO wasted address decoding

WATCH OUR ADS FOR OTHER PRODUCTS COMING SOON

- MODEM ADAPTER FOR THE BIO-1
- 256K RAM SOFTWARE FOR BIO-1
- HIGH FEATURE DISC CONTROLLER
- Z80 C.P.U. / MEMORY MANAGER
- 1 MEGABYTE DRAM BOARD

Specifications subject to change without notice

YOU CAN C

for CP/M

C-BUNDLE.....$99

VIEW: CRT Based Disk Diagnostic
EZZAP: ROM Burning Utility
includes schematic
C-PACK: Utilities in C
C-GAMES: User Modifiable Maze Game
All are written in C, include Source Code, and are Available Separately.

C-COMPILERS

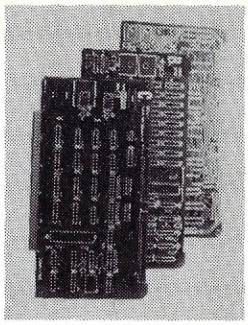
BDS C Compiler
with symbolic debugger.....$140

Aztec C Compiler
many operating systems
$499

CP/M®, Digital Research

Western Wares 303-327-4898
Box C • Norwood, CO 81423

S100
BARE BOARDS
ANY COMBINATION OF 3 BARE BOARDS
$150



	B/B	KIT	ACT
256K Dynamic Memory 64K to 256K to 1 Megabyte Capacity	$75	256K/510 64K/360	
FDC-1 Floppy Disk Controller 8" & 5¼" Drives	$70	$245	N/A
Z80 CPU 4 MHZ/2 MHZ Selectable	$65	$245	N/A
I.O. Board - 4 Serial Port Uses 8251 US ART	$65	N/A	N/A
S100 Clock/Calendar with On Board Battery Backup	$50	$99	$135
Source Code Monitor On Standard 8" Disk Used With CPM 2.2			$25
Monitor In PROM - 2716			$25

BUY ANY COMBINATION OF 4 BOARDS & RECEIVE FREE MONITOR PROM

All Manuals Included - All Parts Available - CA Residents Add Sales Tax
$4 Min. Shipping - Add $1.65 COD
We Accept Credit Cards

COMPUTIME
8614 HAMILTON AVE., HUNTINGTON BEACH, CA 92646
(714) 536-5000
Business Hours 10-5 Monday through Thursday

C Programmers: File System Utility Libraries

Source Code Included, No Royalties, Powerful & Portable.

BTree Library $75.00

- High speed **random** and **sequential** access.
- **Multiple keys** per data file.
- Up to **16 million** records per file.
- Full documentation and example programs included.

ISAM Driver $40.00

- Works with the BTree Library.
- Greatly speeds application development.
- Combines ease of use of **database manager** with flexbility of **programming language.**
- Supports **multi key files** and **dynamic index definition.**
- Very **easy to use; fully documented; example programs included.**

Both products

Are written entirely in **K&R C.**
Come with complete **source code.**
Are **free of any royalty charges.**

+ $3.00 Shipping & Handling Charge.

For more information call:

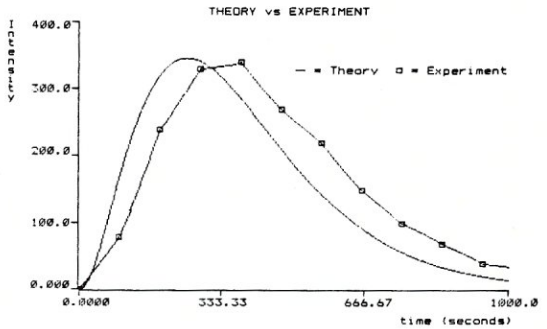
softfocus
1277 Pallatine Drive
Oakville, Ontario, Canada
L6H 1Z1
(416) 844-2610

Credit cards accepted. *Dealer inquiries invited.*

GRAF 2.0 *$29.95*

GRAF 2.0 allows you to create amazing graphics on your dot-matrix printer. Features include:

* Automatic bar chart and line graph generation
* Automatic scaling and labeling of both axes
* Ability to plot floating-point data obtained from most spreadsheets, word processors, or languages.
* Extensive, 60-page illustrated User's Manual includes examples showing how to do graphics from SuperCalc, dBASE II, MBASIC, and Turbo Pascal.



System Requirements Any Z80 computer running 60k CP/M and driving an Epson, Gemini, or C. Itoh dot-matrix printer.

MSC Microcomputer Systems Consultants

301 North Harrison Street CN 5279, Suite 228 Princeton, New Jersey 08540

Terms Send check or money order for $29.95 + $5.00 s/h to MSC at the address above. You MUST state your computer and printer make and model. NJ residents add 6% tax.

The following are trademarks or registered trademarks of the indicated companies: CP/M - Digital Research, MBASIC - MicroSoft, SuperCalc - Sorcim, Turbo Pascal - Borland International, dBASE II - Ashton-Tate, Z80 - Zilog, Keypro - Keypro Corporation

Local Variables In Forth

by Thomas W. Reno

One complaint that I hear frequently about FORTH is that it does not have local variables. These words were initially developed to prove that FORTH can indeed have local variables. Since they have been developed I find that they can be quite useful. When involved in the workings of some FORTH words the stack can be come quite cluttered and difficult to keep track of.

These simple words can do wonders for improving the readability of code. They also have the effect of saving memory required by FORTH standard variables for functions that have fleeting use.

These variables are dynamically allocated and deallocated from a predefined area. This area may be made only as large as the task requires. Probably the most unique thing about this implementation is the deallocation activation. One of the goals that I had was to make the use of these very simple and not require a programmer to do the cleanup. This goal was accomplished by extending the apparent end of the using word to perform the cleanup, by putting, on the return stack, the initial parameter field of the deallocation word. Thus when the word terminates, the local variables, are automatically deallocated. It also proved to be convenient to put the allocation value on the return stack.

The variables are referenced by number. This number is extracted from the referencing words name. This number then becomes the index into the variable stack to access the correct variable.

Screen 9 contains the deallocation word. Note that the first location of the stack is used as the stack pointer. The value for the number of variables to deallocate is received from the return stack.

Screen 10 contains the allocation word. The calling routines parameter field location is temporarily removed from the return stack. The allocation value and the parameter field address of the deallocation routine is then put on the return stack and the calling routines parameter field position is returned to the stack. This causes the deallocation word to be executed after the end of the user word.

Screen 11 contains the words used in accessing the specific variables in the user's program. The real tricky part, in this whole scheme, is in the setting of the WIDTH variable to one (see the reference). The WIDTH variable is used in many FORTH systems to control the number of significant characters to retain in the word header, as words are compiled. The length byte however is set to the number of characters in the name string at compile time. When the words are being compiled the length byte is used as the first comparison to decide if the names even need to be compared. If the length bytes match then the names are compared, character by character, until the last character is located.

Here is an example. The following sequence is executed:

```
: TEST1 1 . ;
```

The header of TEST1 will be:

```
HEX
05 54 45 53 54 E1
!   T  E  S  T  1-- high bit set
--> length byte
```

But if the following sequence is executed:

```
1 WIDTH !
: TEST1 1 . ;
31 WIDTH !
```

The header of TEST1 will be:

```
HEX
05 D4
!   T-- with the high bit set
--> length byte
```

In the first case TEST1 will only match with TEST1, but in the next case a match will be found for any 5-character name that starts with "T". This implementation allows for each user word to define and access up to nine local variables, 1..9. If more variables are needed adding the following definitions would allow for 99 variables.

```
: @XX (A) COMPILE @ ; IMMEDIATE
: !XX (A) COMPILE ! ; IMMEDIATE
```

Screen 12 contains some test conditions that show how the words maybe used in a program.

These routines may be used in any depth of nesting provided the stacks are sufficiently sized.

One note of caution for users of 6502-based systems, that use the processor's stack directly, the values (two 16-bit numbers) can quickly consume the full area.

REFERENCES

Long, Alden B. Numeric Conversion. Forth Dimensions, 6(2),8.

Thomas Reno is President of ONER SYSTEMS Inc., a software consulting company, located in Shoreham Vermont. He has been involved with FORTH for the last four years.

```
SCR # 9
  0 ( LOCAL VARIABLES -1- vars, VARDEALLOT          tr 03/23/85)
  1 FORTH DEFINITIONS DECIMAL
  2 0 VARIABLE LVSP          ( local variable stack )
  3 100 ALLOT                ( area for stack )
  4 LVSP 2 + LVSP !          ( init stack pointer is is first cell )
  5 : VARDEALLOT    ( --- )  ( deallocate previously allocated )
  6                          ( variables )
  7         LVSP @           ( get current stack position )
  8         R)               ( get the value for the allocation )
  9         -                ( deallocate the space )
 10         LVSP !           ( save updated stack position )
 11 ;
 12 --)
 13
 14
 15

SCR # 10
  0 ( LOCAL VARIABLES -2- VARALLOT                  tr 03/23/85)
  1 : VARALLOT      ( vars --- )     ( allocate vars number of )
  2                                  ( variables )
  3         2 * DUP          ( convert to words & copy )
  4         LVSP @           ( get current stack )
  5         +                ( allocate the space )
  6         LVSP !           ( save updated stack position )
  7         R)               ( get the pointer from whence we were )
  8                          ( called )
  9         SWAP             ( get alloc count back on top )
 10         )R               ( put it on the return stack )
 11         ' VARDEALLOT     ( get address of dealloc routine )
 12         )R               ( put it on the return stack )
 13         )R               ( put back our caller )
 14 ; --)
 15

SCR # 11
  0 ( LOCAL VARIABLES -3- (ADDR), (A), @X, !X       tr 03/23/85)
  1 : (ADDR)          ( var# --- varaddr) ( return address of the )
  2                                       ( the desired variable )
  3         2 *              ( convert to word offset )
  4         LVSP @           ( get current stack base )
  5         SWAP -  ;        ( address of variable )
  6 : (A)   0 0 HERE 1+ CONVERT DROP DROP  ( extract variable # )
  7         [COMPILE] LITERAL      ( make it a literal )
  8         COMPILE (ADDR) ;       ( dynamic part of address )
  9 1 WIDTH !  ( set significant characters in name )
 10 : @X               ( --- value ) ( fetch contents of variable X
 11      (A) COMPILE @ ;  IMMEDIATE
 12 : !X               ( value --- ) ( store value in variable X )
 13      (A) COMPILE ! ;  IMMEDIATE
 14 31 WIDTH ! ( restore significant characters in name )
 15 ;S

SCR # 12
  0 ( LOCAL VARIABLES -4- test cases                tr 03/23/85)
  1 : T1     ( test verb number 1 )
  2         1 VARALLOT       ( allocate 1 variable for this verb)
  3         1 !1             ( set it to a known value )
  4         @1               ( fetch the value back )
  5         CR U. CR         ( print it for operator to see )
  6 ;
  7 : T2     ( test verb number 2 )
  8         2 VARALLOT       ( allocate 2 variables for this verb)
  9         20 !2            ( set them to a known value )
 10         10 !1
 11         T1               ( execute test verb # 1)
 12         @1 U. CR         ( verify values to user )
 13         @2 U. CR
 14 ;
 15 --)

SCR # 13
  0 ( LOCAL VARIABLES -5- test cases                tr 03/23/85)
  1 : T3     ( test verb # 3 )
  2         3 VARALLOT       ( allocate 3 variables for this verb)
  3         100 !1           ( initialize them to known values )
  4         200 !2
  5         300 !3
  6         T2               ( execute test verb # 2)
  7         @1 U. CR         ( verify values to user )
  8         @2 U. CR
  9         @3 U. CR
 10 ;
 11 ;S
 12
 13 Execution of this test will cause the numbers 1,10,20,
 14   100,200, and 300 to be printed in a column.
 15
```


S-100 TM

ESTABLISHED 1977

Only 15% Of Our Products Listed--Call For Others!

Computer Systems

CROMEMCO *CS-100H50X20E* 50Mb H.D. **$13,849**
INTERCONTINENTAL MICRO SYSTEMS Call For *SYSTEM INTEGRATION* That Will Fit Your Needs!
VIASYN *816/C-H40* Dual Flpy & 40 H.D. **$6,795**

S-100 Bus Boards

If you purchased before calling us, you probably paid too much! We stock more than 20 brands!
A Few Of This Month's SPECIALS Are:
COMPUPRO/VIASYN *RAM 22, 256K* STATIC **$779**
COMPUPRO/VIASYN *RAM 23, 128K* STATIC **$415**
DUAL *NEW BOARD SPECIALS* **CALL**
MACROTECH *512-ST* STATIC RAM **$1,650**
MACROTECH *ADIT-4* **$699**
MACROTECH *MI-286* 80286/Z80H DUAL PROC. **$799**
MACROTECH *MSR-II* 1Mb DYNAMIC RAM **$1,165**

Floppy Disk Drives

MITSUBISHI *2894/2896* 8" DSDD **$395/$369**
TANDON *100-2A* IBM-PC Compatible **$115**
TEAC *FD55B* & **MITSUBISHI** *4851/4853* **$95/$139**

Hard Disk Subsystems

Integrated with *Concurrent Dos & Turbodos*
CALL FOR DLR &/OR QTY PRICES ON HARD DISKS:
QUANTUM **Amcodyne** **Tandon**
RODIME MICROPOLIS
MITSUBISHI **Maxtor** **Microscience**

Modems

PROMETHEUS *PROMODEM 1200* Hayes Compatible w/ Built-In PWR Supply-RS232 Stand Alone Unit **$289**
OPTIONS FOR PROMODEM 1200 ARE:
OPTION: *PROCESSOR/64K MEMORY* **$75/$35**
OPTION: *ALPHANUMERIC DISPLAY* **$79**
U.S. ROBOTICS *S-100 BOARD 300/1200* **$295**

Monitors & Terminals

TATUNG *CM-1322* 640X200 RGB-*SATISFACTION GUAR.* - IT PUTS COMPETITION TO SHAME! **$395**
TATUNG *CM-1360* 640X200 RGB w/GRN & AMB Switch. Ask about "HOT SPARES" Loaner Policy **$419**
LIBERTY *TERMINALS* w/Variable Scrolling, Green, Amber, 14", DEC Compatibility Extra Pages of Memory, and Graphics Options in Stock at Unbeatable Prices.
LINK *125* WYSE 50 Compat., 14" GRN/AMB **CALL**
WYSE *50/75 TERMINALS* in Stock **$469/$639**

Printers &

BROTHER *HR-15-XL* 17CPS Daisywheel S or P **$359**
CITIZEN *MSP-25* 200/50 CPS 15" ULTRA QUIET **$599**
EPSON *LX80, RX100+, FX80+, FX100+, LQ1500* **CALL**
OKIDATA *New 182* Quiet 120 CPS & 60 CPS **CALL**

Switch Boxes

GILTRONIX *MANUAL AND AUTOMATIC SWITCHING UNITS* to Fit all of Your *SHARED Printer, Terminal, Modem, and Other Peripheral Needs.* **From $79**

Software

We Have Access to all Well Known Brands - **ORDER CORRECTLY -- SOFTWARE IS NOT RETURNABLE!**
BD SOFTWARE *"C" COMPILER* 8 BIT **$99**
LATTICE *"C" COMPILER* 16 BIT **$299**

Graphics And CAD

VECTRIX *Graphics Systems* Utilize *9 Bit Planes* and Have 384K RAM to Operate Independently of Host Video Controller and RAM Memory
VECTRIX *VX384A* Stand Alone RS232 Co-proc. **$3,395**
VECTRIX *VX1301* 13" RGB Analog Monitor **$1,295**
VECTRIX *VXTB* 11"X11" Graphics Tablet **$895**

Power Solutions

SAFT *SPS1000VA/SINE* 1mSEC Switch **CALL**
TRIPPLITE *BC-425-FC 425 Watts 15-20 Minutes* **$449**

Mainframe & Drive Enclosures

INTEGRAND *1100* 7 Slots and 2 X 8" Drives **$475**
JMR *1H5* 5¼" H.D. Cabinet **$189**
PARA DYNAMICS *3820S* PRONTO **$1,150**

SALES 800-528-3138
CUST. SERV./TECH. 602-991-7870
TELEX 9103806778 SONEHUND
14455 N. 79th St., SCOTTSDALE, AZ 85260

Scientific & Technical Word Processors

By Steven Bosak

When I began my search for a scientific word processor just a few short years ago, I expected a glut of such software would be available. After all, my needs were tailor-made for a software solution: I wanted characters and fonts displayed on screen the way they would print, and an easy , fast method for entering technical and scientific characters and equations into the body of any text. I wanted, desperately wanted, all of this to work in one seamless package.

Needless to say, I needed, wanted, and expected too much. At the time little was available, and most solutions were piece meal at best...solutions that were, in all practicality, no more efficient a manner of preparing technical manuscripts than the old reliable method of inserting the symbols by switching elements on a printer or typewriter.

As an alternative to that cumbersome approach, a number of character and font generators hit the market, promising one-step processor-to-printer routines. But there is a serious problem with this technique: printer codes must be inserted into the equations to properly support fonts and scripting levels. The result on screen is a jumble of control characters and commands mixed with the objects of the controls. This may work fine with simple equations, but once the scripting goes beyond one or two levels, I defy any technophile to imagine what will ultimately roll out of the printer. It also makes format somewhat empirical, especially when overprinting is required.

When the average machine memory began to exceed the 64K barrier, possibilities suddenly presented themselves, giving rise to a new breed of technical word processors. The additional memory was used to generate technical, Greek, and math symbols on screen. High-speed dot matrix , ink-jet, and laser printers, capable of integrated letter AND high quality graphics , made possible fast and professional-looking technical and scientific word processing systems.

These new software products differ greatly in features, performance, and price. If you deal with technical or scientific manuscripts and want an overview of the latest and greatest available for microcomputers, the following evaluations will give you a good idea of what program might suit your particular needs. I have eliminated from review a great number of programs that I felt were confined to a narrow market, for example programs limited to one computer and printer or those which are simply a variation on the old thimble-juggling routines of the past as well as packages which are not fully integrated. As a technical editor and consultant in this arcane field of symbol and script scribbling, I've come to form very definite ideas about what the "perfect" technical and scientific word processor should do for the user.

The guidelines and requirements I used to examine these products are:

* The software should run under MS-DOS. Like it or not, this is the *defacto* DOS standard for office and personal computers.
* "What you see is what you get"....No surprises, please. If I have a five-level equation with large curly brackets, that's what I want to see on screen AND, of course, coming out of the printer.
* High-speed, letter-quality, dot-matrix, ink-jet, or laser printers should be supported. Yes, you CAN use double-thimble, daisy-wheel printers, but in my experience, they are noisy, slower, and no matter which thimble or wheel I choose, there is always another character on yet another thimble that I need. Why not let the software and versatile hardware do the job? Double-thimble, daisy-wheel printers just seem to be one cut above switching the elements on a Selectric III.
* The systems should be just that--systems. All the parts, if they are even discernible as such, should work in harmony.
* The processors should be easy to grasp. This is, perhaps, one of my most obvious but essential requests. A good technical and scientific word processor will be used by many people--from the head of the department to a member of the typing pool. Let's face it, most of the technical typing is done by the secretaries and administrative assistants. By asking that the packages be easy to use, I am not discounting their ability to grasp a complex program, but merely pointing out the reluctance on the part of most professional typists to put up with a grudging and cumbersome system that hampers their productivity. I cannot blame them in the least, and in my work as a consultant, their concerns are very often the bottom line.

In these evaluations I used a Zenith Z-151 PC with 640K RAM and two floppy disk drives. The Zenith video board displayed all the programs admirably--I didn't have a hitch with any graphics. For printing, I used Toshiba's new P351, an improved and faster version of their 1351 24-pin dot matrix printer. The print quality was letter perfect and the speed incredible--over 100 cps in letter-quality mode, and over 200 cps in the draft mode! All print samples in this article are from the P351.

VOLKSWRITER SCIENTIFIC

Lifetree Software, Inc. has been in the word-processing business a number of years. Their Volkswriter and Volkswriter Deluxe are solid and popular word processors. If you currently use either of them in your office or at home, then Volkswriter Scientific should be the first of the technical word processors to consider. You will find many familiar or very similar commands. In addition, Volkswriter Scientific imports Volkswriter Deluxe files for conversion to the VW Scientific format. (Other text files can be imported also.)

INSTALLATION: The program was simple to install-you merely execute a batch file for installation on either a floppy or hard-disk system, a process that took all of two minutes. VW Scientific was then ready to run.

FEATURES: The opening menu provides you with either print, retrieve, directory, create new document or exit options. After retrieving or opening a new document you are brought to the Format Menu. Here you have the option of setting defaults, modifying the existing parameters, or accepting

the existing defaults and continuing on to the actual processor. Among the format features that are set at the menu are line spacing, left and right margins, hyphenation, large or small font size, and tab stops.

Next is the processor. If you wish to edit an existing document, you are given the time-saving prompt "Which page to start?". The document comes up with a menu of menus along the bottom fourth of the screen. The F1 function key followed by numbers 1 through 9 bring up separate menus for cursor movement, microspacing and justification, footnoting, and use of Greek, script, mathematical, scientific, and engineering characters. Each menu pops up along the bottom fourth of the screen when called, so you always have a portion of your working text on screen.

Entry of text is straightforward. The processor has all the standard editing and formatting features of the common word processor, i.e. word wrap, reformatting, insertion, deletion, simple keystroke commands for rapid movement within the text, as well as commands to cut, paste, and unerase. Cursor movement is in full spaces, half spaces or even microspaces. This micropositioning is great for aligning subscripts, superscripts, or elements within an equation.

Technical symbols are entered with two keystroke commands that, in most cases, parallel their standard character cousins. For instance "\a" enters an alpha, "\b" beta, and so on. The menus for the symbols show "type this" in one column "to get this" shown in the neighboring column. The symbols are logically arranged and easy to choose from the screen. In addition to the 400-plus characters and symbols provided, Volkswriter Scientific has a macro-key feature--any of 26 (Alt.-A through Alt.-Z) keys can be used to define a series of keystrokes. The macro keys, in conjunction with VW's building-block font set, allow the creation of custom chemical chains, braces, brackets, squares, and parentheses, which can be easily recalled. A number of large bracket and brace sets are already defined with the initial package. I found creation and storage of new sets easy to accomplish, as long as cursor movement suggestions, as provided in the documentation , were closely followed.

Additional comforts provided in the VW Scientific font sets are true footnote-sized numerals, foreign symbols, and general characters such as pound, cent, paragraph and copyright. Boldface, underline and italics are just one keystroke away.

DOCUMENTATION: Provided with Volkswriter Scientific is an interactive tutorial disk which takes the user through the easiest to most complex aspects of the program. I strongly advise anyone, regardless of sophistication, to go through this tutorial. It is thorough and gives plenty of opportunity to use the features as they are being explained. In fact, after completing the disk tutorial, I merely used the documentation as a reference.

PRINTING: Printing a document with Volkswriter Scientific requires closing out the document, going back to the main menu, and choosing PRINT. Once in the printer menu, you are given the option of changing standard printing parameters such as model of printer, margins, number of copies. Alternatively you may simply enter a RETURN to use your pre-set defaults. Once I had the Toshiba P351 set in unidirectional mode, the processor and printer worked together perfectly.

FINAL CONSIDERATIONS: Despite VW Scientific's merits, it could stand some improvement. For one thing, the print speed, even with the P351, was slow. This is due, I'm sure, to VW Scientific's unidirectional printing as well as the method of operation; each page is graphically mapped. A Ctrl-F8 is needed to create each new page of text. The program tells you when the current page is filled. It then goes out and initiates that new page for mapping. On a floppy-disk system this is a bit annoying. A RAM disk and a print spooler alleviated these problems to a tolerable level. If you deal with longer manuscripts, I would strongly advise that you get a hard disk-mapping out the pages in this manner eats up disk space.

Still another drawback is the inability to add fonts or font sets or to modify the existing characters. If you deal with a number of irregular or customized characters, you should check the sets that are in the program before purchase to see if VW Scientific will meet your needs.

SUMMARY: VW Scientific was , without a doubt, easy to set up and use. If you want standard mathematical, Greek, engineering or scientific characters, and need to see your work displayed exactly (and I mean exactly!) on the screen as it will print out, this may be the package for you. The system runs well on a 256K double-disk drive PC, but as I said, if speed is a concern, or you are working with longer manuscripts, I would recommend use with a RAM disk and a small print buffer or a spooler and hard disk.

QUICK REFERENCE

PRODUCT: Volkswriter Scientific
Lifetree Software Inc.
411 Pacific Street
Monterey, CA 93940
(408) 373-4904
SYSTEM REQUIREMENTS: MS-DOS 2.x or higher on an IBM PC or compatibles. IBM color graphics adapter or any compatible video cards. A minimum of 256K RAM and two floppy disks. Additional RAM for spooler and RAM disk operation, as well as a hard disk, desirable for long manuscripts or heavy use.
PRINTERS SUPPORTED: HP Thinkjet; Epson MX, FX,; Toshiba 1340, 1350, 1351, P351; NEC 8023A; ProWriter; IBM Graphics Printer
PRICE: $495.00 includes system and tutorial disks with documentation.

T3

T3 by TRIAD Computing Inc. is a feature-loaded system. In fact, the entire program comes on three disks, all of which are needed for operation. Minimum RAM requirement is 512K—more is recommended. It can be run on a floppy system, as my experience has shown, but a hard disk is really essential to get any use out of this processor. Anything less does the program a disservice.

INSTALLATION: Installation, via one main batch file, was relatively painless. You'll experience some vertigo, however, when you attempt to install a printer not supported by the off-the-shelf standard product (IBM GRAPHICS Printer is the default, I believe). Not only must you install other printer drivers with their own versions on system disks, but once the program is supposedly "installed" for a particular printer, you must enter the program itself and "attach" the newly installed printer to the running program, otherwise the new printer driver is not recognized when you attempt to print a document and will default to the IBM GRAPHICS. The people at TRIAD assure me that this extra step to the printer installation will soon be eliminated.

When executed, the program loads an incredible amount of material--system volumes, font sets and drivers, as well as program overlays. On a floppy system this takes a great deal of time, a hard disk is definitely the way to go. T3 allows multiple users and password

protection. Users can have their access level set by the system operator. Once past the start-up, you are presented with a long opening menu from which you can set printer options, access levels (if you are the master user), and load other font sets, etc. Of course, all the expected options--print, create a new document and revise documents--are also present.

FEATURES: T3 has the typical word processing features, i.e. word wrap, reformatting, tab and margin settings, cut and paste, search and replace, footnotes and headers, easy text entry, and cursor movement. All text and equations are entered in the same manner, so there are no modes to switch. Functions are selected via pop-up menus or keystroke commands. Manuscript formats or templates can be set or reset and stored for easy retrieval. Key sequence macros can also be stored.

Full- and half-space cursor movement create true super and subscripts on screen. A nice feature of T3 is that it treats equations as single line entities, reformatting will not "dissolve" your work or break up lines, the equation will hold together. As a bonus, standard ASCII files can be imported to the T3 format.

T3 handles technical and scientific characters as well as other symbols through alternate keyboards. These keyboards--up to seven, in fact--can be attached to any given manuscript. When text is entered on the normal keyboard, one of the available keyboards is attached as the alternate, with the remainder attached but not usable until rotated into the alternate slot. If a keyboard, let's say the Greek, is designated as the alternate, all keys on the keyboard--both upper and lowercase--produce Greek characters when pressed with the Alt. key. Similar to VW Scientific, T3 has made such key assignments as logical as possible (Alt.- a for alpha, etc.). You can view the alternate keyboard at any time by pressing Shift-F8-- the alternate key-board, lower/uppercase, will pop-up on screen.

If you would rather avoid using the pop-up keyboard, you can print it out for reference by hitting the Shift-PrtSc keys. (See Figure 1) To get an alpha character into my text, I simply hit Alt.-a ,provided the Greek keyboard is my alternate. If I want to use a character from any other keyboard, it must be designated as the alternate. Better make sure the keyboard you choose as the alternate has the characters you are most likely to use.

One of T3's better character/font features is the ability to create your own

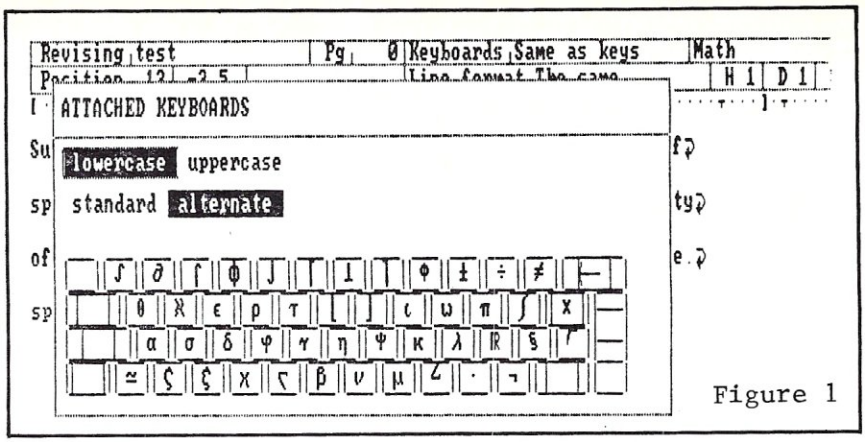


Figure 1

custom keyboards. If you rarely use a character and find that you need a different character from another keyboard, you can delete, through a menu-driven Revision program, the unwanted key and insert any other defined key in its place. Individual characters, while on a grid, can also be modified, or entirely new characters created!

PRINTING: Documents are printed after selecting "Send to printer" from a prompt at the end of an editing session or from the "Print" option under the DOCUMENTS menu. All the formatting is taken care of in your manuscript defaults and through the installation of the printer. T3, like VW Scientific, is bit-mapped. Exactly what is on the screen is printed. There are no surprises.

The only hassle with printing involves the unique manner in which T3 handles this bit-mapped printing. Each document is created inside a "shell". If pages are not determined and marked explicitly in the text, or if they are entered and the text is then reformatted, results can be interesting to say the least. Best to follow the documentation's advice and insert your page determinations or breaks only AFTER your text is ready for a FINAL print.

DOCUMENTATION: T3's documentation, like the program itself, is large. It could also stand for some thoughtful reorganization. After spending time on the program's installation, the documentation sends you to the end of the long tutorial. Sections on creating new document shells and formats are scattered throughout. The printer installation section is, similarly, off on its own, buried between KEY SEQUENCES and VOLUMES for some reason.

The on-disk tutorial was a better place to get acquainted with T3 than it's documentation. A variety of sample files, from simple letters to manuscripts containing multi-level equations and benzene ring chemical chains are provided to illustrate how to use T3's features.

FINAL CONSIDERATIONS: Despite T3's power and versatility, there are a number of things I dislike about the program. For one thing, even barring the password provisions, there are just too many features that need to be set. You have to set a "Face" key for either bold or underline—it has to be either/or—not both!). You have to set shells up. You must attach printers and manuscript formats. You even have to set your own "Profile" which carries some of the above preferences and defaults and attaches to you when you enter the program and sign on. Also, for some reason I'll never understand, on the command level all instructions are terminated with the "+" sign from the numeric keypad. Why not RETURN, like every other piece of software known to man?

Errors and error messages are not adequately explained on the screen or in the book. When I inadvertently booted T3 without the correct CONFIG.SYS file on disk, the resultant error message was anything but informative, and actually quite misleading. Nor did the documentation do anything to help extricate me from my misfortune. Fortunately, TRIAD has a toll-free support number to call.

SUMMARY: You'll have to spend time getting used to this system. T3 doesn't approach word processing in the same manner as most programs, so it takes quite awhile before you feel comfortable with T3's idiosyncrasies. (The "Face" key and "+" entry on command level were my biggest gripes.) The chemistry and font-editing facilities make it especially useful to chemists, chemical engineers and people that require customizable characters and keyboards.

This software is not effective on a two-floppy system, except for short manuscripts. I would set 640K RAM and a hard disk as the minimum system configuration. With its sign-on and password features, T3 would be well suited for a muti-user environment that

QUICK REFERENCE

PRODUCT: T3
TRIAD Computing Inc.
1190-B Foster Road
Las Cruces, NM 88001
1-800-874-2383

SYSTEM REQUIREMENTS: MS-DOS 2.x or higher on the IBM PC or compatibles. 512K RAM and IBM color/graphics card or Hercules graphics board. 640K and hard disk recommended highly. PRINTERS SUPPORTED: Standard system comes with support for IBM graphics; Epson FX, RX. Additional printers are supported, at additional cost ($60 - 90). Diablo 630 and 630 ECS; Epson LQ1500; Okidata 2410; QMS LaserGrafix 800; Santec S700; Toshiba 1340, 1350, 1351 and P351 Xerox 2700.

PRICE: $595.00 for three system disks. Printer driver addition at $60 to $90 (depending on printer). Chemistry disk is available for $30.

demands control over document access.

PROOFWRITER

Proofwriter is actually a family of products from Image Processing Systems. Proofwriter International is a program that focuses on foreign-language word processing. Proofwriter Graphics is the scientific and technical word processor which I reviewed. It provides extensive alternate scientific and character sets on disk in bit-mapped form for inclusion in the user's Proofwriter word processor. One version of the program keeps an alternate character set (like an extended scientific font) on a PROM, which replaces the regular character PROM on the main board of the Zenith Z-150. In essence, instead of installing a set of software fonts, the PROM set is resident. As I did not have time to review this hardware-supported version, whenever I mention "Proofwriter" in the following text, I am referring to Proofwriter Graphics.

This program not only provides full word-processing capabilities, but also had a feature not found on other programs: a spelling checker. In fact, the word-processing end of this program was remarkable--commands were elementary and logical: 'H' for Help, 'K' for Kill a Line. Text was held in memory, so the program was fast; changing from the editing to printing menu posed no problem for Proofwriter's memory--the last document edited was the default to send to the printer: a friendly touch.

INSTALLATION: Installation is moderately taxing, and involves a bit of disk swapping. To get the proper character set loaded and your printer installed, you have to go through at least four disks. But once installed, you are set: the system disk and alternate character sets need never be re-installed unless you have a need for entirely different sets. Printers were installed from a very complete menu, and was no more complicated than punching the correct number. Like T3, unusable characters could be replaced by characters from other font sets. Each character could also be edited on a cursor-addressable 24x24-pixel grid. Completely new characters can also be created in this manner.

FEATURES: I mentioned Proofwriter's word processing skill--it really is a nifty, little, FAST word processor. I only wish the creators had used as much care in endowing the rest of the program with such speed and ease-of-use. To start, those wondrous alternate character sets are buried in keyboard after keyboard. You have to be in an "alternate" mode to pick from these keyboards. The only other way around this "mode" is to enter the characters directly to the screen using their assigned ASCII value with the Alt. key. To get an alpha, you would need to know it had been assigned to ASCII number 155 and enter: Alt. 155. Luckily there is a provision for printing out this table, so you can look up the characters. (see Figure 2) It beats calling up the many pop-up keyboards to find your character, re-entering the text, switching to the correct alternate keyboard and THEN entering the (by now forgotten) character. Proofwriter should allow assignment of a first alternate character set during the installation rather than having the many alternate boards.

To further complicate these matters, the text processor itself has Insert, Change, and Command modes. Macros for the creation and storage of large braces, brackets or other multi-line symbols is hinted at on pages 3-44 to 3-45 of the manual. I am not sure if such construction and retrieval is possible. I ended up building each bracket I made from scratch.

Another feature glaringly absent from Proofwriter was the ability to see directly on the screen what would end up on the printer. You saw "kinda" what would end up on the page. The LEAST annoying omission in this regard, was the representation of half lines as full lines on screen. A line-space shift command would tell the printer to change the line spacing. Within the text, subs and superscripts were indicated by highlighting the "offending" script... unfortunately, the highlighting does not indicate if the character is sub or superscript.

More offensive to the typist was Proofwriter's inability to show--in the editing environment--the way that equations would end up once they were sent to the printer. To get such a preview, you have to leave the editor, go to the print menu and select "View" to see the actual results of all your editing. Not nice. The differences can

Alternate Character Set

128 - ∇	154 - ≅	180 - Ξ	206 - ∪	232 - ²
129 - ¡	155 - α	181 - Π	207 - ∩	233 - ³
130 - ¡	156 - β	182 - Σ	208 - ⊂	234 - ⁴
131 - ₖ	157 - γ	183 - Φ	209 - ⊃	235 - ⁵
132 - ₁	158 - δ	184 - Ψ	210 - ∂	236 - ⁶
133 - ₘ	159 - ∈	185 - Ω	211 - ¡	237 - ⁷
134 - ₙ	160 - ζ	186 - |	212 - ¡	238 - ⁸
135 - ₚ	161 - η	187 - ⌝	213 - ₖ	239 - ⁹
136 - ₑ	162 - θ	188 - ⌟	214 - ₁	240 - ₀
137 - ᵣ	163 - ι	189 - ∓	215 - ₘ	241 - ₁
138 - ₛ	164 - κ	190 - ⌈	216 - ₙ	242 - ₂
139 - ₜ	165 - λ	191 - ⌋	217 - ₚ	243 - ₃
140 - ᵤ	166 - μ	192 - ≥	218 - ₑ	244 - ₄
141 - ᵥ	167 - ν	193 - ≤	219 - ᵣ	245 - ₅
142 - ₓ	168 - ξ	194 - √	220 - ₛ	246 - ₆
143 - ₓ	169 - π	195 - ∪	221 - ₜ	247 - ₇
144 - ᵧ	170 - ρ	196 - ×	222 - ᵤ	248 - ₈
145 - ₂	171 - σ	197 - ⌝	223 - ᵥ	249 - ₉
146 - ₊	172 - τ	198 - ⌟	224 - ₓ	250 - ±
147 - ₋	173 - φ	199 - Δ	225 - ₊	251 - ≤
148 - ₌	174 - χ	200 - ⌞	226 - ₋	252 - ≥
149 - ₓ	175 - ψ	201 - ⌜	227 - ±	253 - $
150 - ¶	176 - ω	202 - ⌝	228 - ₓ	254 - ¢
151 - ⁄	177 - Γ	203 - ⊤	229 - ÷	255 -
152 - §	178 - Θ	204 - ⌟	230 - ₀	
153 - ∫	179 - Λ	205 - ⌐	231 - ₁	

Proofwriter scientific character set table Figure 2

Super assemblers plus the world's largest selection of cross assemblers!

Z-80 Macroassembler

Power for larger programs! This 2500AD macroassembler includes:

- Zilog Z-80 Macroassembler (with the same powerful features as all our assemblers)
- powerful linker that will link up to 128 files. Com files may start at any address
- Intel 8080 to Zilog Z-80 Source Code Converter (to convert all your Intel source to Zilog Syntax in one simple step)
- COM to Hex Converter (to convert your object files to Hex for PROM creation, etc.)
- 52 page User Manual

8086/88 Assembler with Translator

Available for MSDOS, PCDOS, or CPM/86! This fully relocatable macro-assembler will assemble and link code for MSDOS (PCDOS) AND CPM/86 on either a CPM/86 or MSDOS machine. This package also includes:

- An 8080 to 8086 source code translator (no limit on program size to translate)
- A Z-80 to 8086 translator
- 64 page user manual
- 4 linkers included:
  - –MSDOS produces .EXE file
  - –CPM/86 produces .CMD file
  - –Pure object code generation
  - –Object code and address information only

Linker features:

- Links up to 128 files
- Submit mode invocation
- Code, Data Stack and extra segments
- Handles complex overlays
- Written in assembly language for fast assemblies.

Z-8000 Cross Development Package

Instant Z-8000 Software! This package allows development and conversion of software for the Z8001, 8002, 8003 and 8004 based machines on a Z-80, Z-8000 or 8086 machine. This powerful package includes:

- a Z-80/8080 to Z-8000 Assembly Language Source Code Translator
- Z-8000 Macro Cross Assembler and Linker

The Translators provide Z-8000 source code from Intel 8080 or Zilog Z-80 source code. The Z-8000 source code used by these packages are the unique 2500AD syntax using Zilog mnemonics, designed to make the transition from Z-80 code writing to Z-8000 easy.

All 2500 AD Assemblers and Cross Assemblers support the following features:

Relocatable Code — the packages include a versatile Linker that will link up to 128 files together, or just be used for external reference resolution. Supports separate Code and Data space. The Linker allows Submit Mode or Command Invocation.

Large File Handling Capacity —the Assembler will process files as large as the disk storage device. All buffers including the symbol table buffer overflow to disk.

Powerful Macro Section—handles string comparisons during parameter substitutions. Recursion and nesting limited only by the amount of disk storage available.

Conditional Assembly—allows up to 248 levels of nesting.

Assembly Time Calculator—will perform calculations with up to 16 pending operands, using 16 or 32 Bit arithmetic (32 Bit only for 16 Bit products). The algebraic hierarchy may be changed through the use of parentheses.

Include files supported—Listing Control—allows listing of sections on the program with convenient assembly error detection overrides, along with assembly run time commands that may be used to dynamically change the listing mode during assembly.

Hex File Converter, included—for those who have special requirements, and need to generate object code in this format.

Cross reference table generated—

Plain English Error Messages—

System requirements for all programs: Z-80 CP/M 2.2 System with 54k TPA and at least a 96 column printer is recommended. Or 8086/88 256k CP/M-86 or MSDOS (PCDOS).

Cross Assembler Special Features

Z-8—User defined registers names, standard Zilog *and* Z-80 style support. Tec Hex output option.

8748—standard Intel *and* Z-80 style syntax supported.

8051—512 User defined register or addressable bit names.

6800 Family—absolute or relocatable modes, all addressing modes supported. Motorola syntax compatible. Intel Hex or S-Record format output.

6502—Standard syntax or Z-80 type syntax supported, all addressing modes supported.

	Z80 CP/M®	ZILOG SYSTEM 8000 UNIX	IBM PC MSDOS	IBM PC CP/M 86	OLIVETTI M-20 PCOS
Z8000™	$299.50	$750.00	$299.50	$299.50	$299.50
Z80	99.50	500.00	199.50	199.50	199.50
Z8	199.50	500.00	199.50	199.50	199.50
8086/88	199.50	750.00	99.50	99.50	199.50
80186	199.50	750.00	199.50	199.50	199.50
8748	199.50	500.00	199.50	199.50	199.50
8044/51	199.50	500.00	199.50	199.50	199.50
8080	199.50	500.00	199.50	199.50	199.50
8085	199.50	500.00	199.50	199.50	199.50
8096	199.50	500.00	199.50	199.50	199.50
68020	399.50	750.00	399.50	399.50	399.50
68000,08,10	299.50	750.00	299.50	299.50	299.50
6800,02,08	199.50	500.00	199.50	199.50	199.50
6801,03	199.50	500.00	199.50	199.50	199.50
6804	199.50	500.00	199.50	199.50	199.50
6805	199.50	500.00	199.50	199.50	199.50
6809	199.50	500.00	199.50	199.50	199.50
32000	399.50	750.00	399.50	399.50	399.50
COPS400	199.50	500.00	199.50	199.50	199.50
NSC800	199.50	500.00	199.50	199.50	199.50
6301	199.50	500.00	199.50	199.50	199.50
6501/11	199.50	500.00	199.50	199.50	199.50
6502	199.50	500.00	199.50	199.50	199.50
65C02	199.50	500.00	199.50	199.50	199.50
1802	199.50	500.00	199.50	199.50	199.50
F8/3870	199.50	500.00	199.50	199.50	199.50
NEC7500	199.50	500.00	199.50	199.50	199.50
NCR/32	399.50	750.00	399.50	399.50	399.50

Subtotal $ _____ $ _____ $ _____ $ _____ $ _____

Name _____
Company _____
Address _____
City _____ State _____ Zip _____
Phone _____
Make and model of computer system _____
☐ COD (2500AD pays COD charges)
☐ VISA or MasterCard
Number _____
Expiration Date _____

TO ORDER. **Simply circle the product or products you want in the price columns, and add up your order.**

Total $ _____

Check one:
☐ 8″ Single
☐ Osborne
☐ IBM PC
☐ Cartridge tape
☐ Apple (Softcard)
☐ Kaypro DSDD
other formats available, please call!

Shipping UPS Blue Label no charge, $15.00 International per unit $ _____

Total Order $ _____

Signature _____

220

2500AD SOFTWARE INC.

P.O. Box 4957, Englewood, CO 80155, (303) 790-2588 TELEX 752659/AD

QUICK REFERENCE

PRODUCT: Proofwriter Graphics
Image Processing Systems
6409 Appalachian Way
P.O. Box 5016
Madison, WI 53705
(608) 233-5033

SYSTEM REQUIREMENTS: MS-DOS 2.x or higher on the IBM PC, compatibles, or Texas Instruments PC. 256K RAM and at least two floppy disks.

PRINTERS SUPPORTED: Toshiba 1340, 1350, 1351, P351; Epson MX,RX,FX,LQ1500; IBM Matrix and Graphics; C.Itoh 8510A; Texas Instruments 850/856; NEC 8523A; IDS Prism Gemini 10 ProWriter; Okidata 84/92/93; Diablo 620, 630, 630 ECS; XEROX 1730, 1730 ECS, 1750; NEC 2250, 3550, 7700; Qume Sprint 5, 9, 11; Brothers HR1, HR15, HR25; Daisywriter.PRICE: $425.00

be astounding. An equation that looked great on screen came out as garbage when printed. Other users have experienced "phantom" characters in final print outs. As I said, you can check the equations, but this requires many extra steps--think of a document with dozens of equations!

DOCUMENTATION: The documentation was a joy to behold-- a full-page, 8x11 inch book that would lay flat, was easy to read and stuck to the step-by-step essentials. The only problem I had was with the mixture of Proofwriter Graphics and International documentation. On a number of occasions, I found myself reading sections intended only for the International user. Perhaps these sections could be better segregated or marked? All the sets of available fonts were listed in back, as was a genuinely useful index.

PRINTING: Printing is a painless and quick experience. Because Proofwriter does not depend on bit-mapping each page, it's speed was tremendous. It was the one program that fully utilized the Toshiba P351's speed and quality. What you pay for here is the speed, which you loose on the screen. Similar to WordStar, Proofwriter has a slew of printer codes for formatting and customizing each page, and even each paragraph.

FINAL CONSIDERATIONS: Proofwriter is a copy-protected program. I don't know how you feel about such matters, but I have a personal boycott in effect on such products. I don't allow any copy-protected program in the same system with my valuable text and data files. I have no idea why the folks at Image Processing Systems chose to go this route.

As I said earlier, this is a fast text processor with extensive character sets and optional character creation. If the on-screen-to-printer discrepancies and

This is an example of Proofwriter

$$\gamma_6 = \int \frac{e^{i|t|^{\alpha}} e^{-itx}(\alpha|t|^{\alpha-1} - x)}{(1+|t|)^{\beta+iy}(\alpha|t|^{\alpha-1} - x)}$$

The above is an example of what printed out when Proofwriter's editing screen showed a neat, tidy equation.

This is an example of PROOFWRITER text

This is underlined.

This is emphasized.

This is both emphasized and underlined.

This is Enlarged.

This is a superscript[1] and this is subscript[2] in text mode.

NOTE : supers and subs within text DO NOT appear as such on screen but are designated by highlighting. Sub and supers within equation mode appear on screen as full line above or below, not half line. The print formatter reduces the full lines to halves.

The following should be printed in the equation mode:

$$\gamma = \int \frac{e^{i|t|^{\alpha}} e^{-itx}(\alpha|t|^{\alpha-1} - x)}{(1+|t|)^{\beta+iy}(\alpha|t|^{\alpha-1} - x)}\,dt.$$

This is an example of T[3] text.

This is underlined

This is boldface ...*and script regular*

as well as script in boldface.

This is bold and underlined...

Superscripts[1] and subscripts[2] are shown on screen in true half space. If the superscripts or subscripts threaten the integrity of the lines above or below, T[3] automatically inserts half line spaces to accomodate. Text and equations are entered in the same manner, no shifting between modes.

$$\gamma = \int \frac{e^{i|t|^{\alpha}} e^{-tx}(\alpha|t|^{\alpha-1} - x)}{(1+|t|)^{\beta+iy}(\alpha|t|^{\alpha-1} - x)}\,dt.$$

This is an example of **Volkswriter Scientific** text

This is underlined.

This is boldface.

This is italic.

...and boldface italic

This is large typeface, **bold large type.**

Superscripts[1] and subscripts[2] show on the screen in true half space and are entered in the same mode as the text.

$$\tau = \int \frac{e^{i|t|^{\alpha}} e^{itx}(\alpha|t|^{\alpha-1} - x)}{(1+|t|)^{\beta+iy}(\alpha|t|^{\alpha-1} - x)}dt.$$

JOHN D. OWENS ASSOCIATES JOHN D. OWENS

LARGE DISCOUNTS FROM JOHN D. OWENS ASSOCIATES

MACROTECH MI-286: 80286/Z-80H DUAL PROCESSOR S-100 CPU BOARD: . . . **$876** MSR RAM: 120NS, high-speed dynamic RAM, Works with CompuPro 8085/8088; MI-286 and others: 256K . . **$556** 512K . . . **$716** STATIC RAM: Substitute for RAM 22 and RAM 23:
256K STATIC: **$960**
512K STATIC: **$1,800**

EMERALD SYSTEMS HARD DISK and TAPE SUBSYSTEMS
High capacity! Up to 280 MB! Emerald has overcome the 32MB DOS limitation! Multiple volumes per physical drive. Back up and restore utilities.

HOUSTON INSTRUMENTS: Plotters: DMP 41 OR 42: . . **$2,397,** DMP 29: . . . **$1,838** DIGITIZERS: DT11 . . **$694;** DT114 . . . . . **$750;** DT11AA **$714**
New! 14 pen DMP 51 and 52: . . . **$4,796**

ILLUMINATED TECHNOLOGY
S-100 COLOR GRAPHICS: **$1,116**

NEC APC III: 80816 MS-DOS system w/ spectacular graphics: . . . . . 20% off list

SEMIDISK 2MB DISK EMULATOR FOR IBM PC and EPSON QX 10: **$2,050**
S-100 IMB: . . **$1,472** S-100 2MB: . . **$2,091**

LOMAS: 80286, CCP/M and MS-DOS S-100 Systems with COLOR GRAPHICS for IBM-PC compatibility.

RACTER: Interactive conversational software for IBM-PC. Jan 1985 Scientific American called RACTER, ". . . extremely funny." RACTER is light years ahead of Eliza! . . . . . **$69.95.**

MIST + CONEXUS: Integrated software for IBM-PC combining telecommunications, text processing, data base management.

BIG DISCOUNTS on COMPUPRO, IMS, BYAD, many others. Prices & availability subject to change without notice.

WRITE OR CALL FOR PRODUCT LITERATURE.

WE EXPORT: Overseas Callers: TWX 710 588 2844 (OWENSASSOC NYK). EASY LINK MAILBOX ADDRESS: 62840 768.

ORDER BY PHONE, LETTER, TELEX OR EASY LINK. We accept VISA and Mastercard. Shipping $5 per board in continental USA.

JOHN D. OWENS ASSOCIATES
12 SCHUBERT STREET
STATEN ISLAND, N.Y. 10305
(718) 448 6283 (718) 448 6298
(718) 448 2913

JOHN D. OWENS ASSOCIATES JOHN D. OWENS

SCIENTIFIC WORD PROCESSORS

continued from page 46.

mode-shifting could be eradicated, this may well be a great product. Presently, it isn't very suitable for manuscripts with extensive sub and superscripting levels or numerous equations. There is no support for the creation and storage of chemical symbols.

SUMMARY: If you don't mind the extra pain and effort needed to "check up" on your equations between edit and print time, and you need a fast processor with a spelling checker, this is a good system. Because Proofwrit er doesn't map out each page, it can import any standard ASCII file without a "special utility" for doing so. This may be THE system for those that have relatively light technical and scientific references and who want a good word processor to boot. T he alternate character sets are among the most extensive I've seen--15 sets at approximately 128 characters per set! This may be the only system that would allow extensive foreign AND technical fonts to be easily inserted into text.

Steven Bosak lives in Evanston Illinois and is a freelance writer/editor who has worked on everything from technical writing to fiction. His novel, titled "Gammon" has just been published by St. Martins/Marek Press.

Disk Sale

dy/an Dysan CORPORATION

TYPE	BOX OF 10
5"-SS/DD-48 TPI	19.50
5"-DS/DD-48 TPI	25.50
5"-SS/DD-96 TPI	29.50
5"-DS/DD-96 TPI	37.50
5"-DS/DD-IBM/AT	52.95
8"-SS/SD-48 TPI	23.95
8"-SS/DD-48 TPI	25.50
8"-DS/DD-48 TPI	29.95

Available Soft or Hard Sector
For Plastic Case Add 1.25/Box
Plus Tax & Shipping
- Cash, Visa, Mastercard, COD -

Integral Systems Corp.
2900-H Longmire Drive
College Station, TX 77840
(409) 764-8017

Of course, POWER!™ saves your Bad Disk.

NOW! WINDOWS FOR IBM!

It also does 54 other things to keep your disk in line.

EVERYTHING YOU ALWAYS WANTED TO DO, BUT WERE AFRAID TO TRY

Unlike some utility programs that are a headache to use, POWER! is engineered to spoil you with 55 features, simple and uniform commands, and utter simplicity of use. POWER! automatically alphabetizes and numbers your files. You select by the number and never type file names again. Need to **[COPY]**, **[RENAME]**, **[ERASE]**, or **[RUN]** programs? Just type in their menu number! POWER! also locks out your disk's bad sectors **[TEST]** without destroying files–a critical difference from other utilities that search and destroy, without informing you what they've done, leaving you to wonder why your programs won't run. (And POWER! still has 50 commands to go!)

POWER! ONE PROGRAM DOES IT ALL!

You may own a few utility programs for your computer housekeeping, each with its own commands to memorize. POWER! has all the programs rolled into one 16K integrated package, so you do things you've never tried before–every day. Save sensitive data from prying eyes with **[PASS]** word protect, move a block of memory **[MOVE]**, look for data **[SEARCH]** or compare files **[CHECK]**. POWER! also makes easy work of patching, **[DISPLAY/SUBSTITUTE]**, customizing software **[LOAD/SAVE]**. Among the other commands are **[SIZE]**, **[STAT]** **[LOG]**, **[DUMP]**, **[TYPE]**, **[JUMP]**, **[FILL]**, **[SET]**, and the CP/M version lets you restore erased files–even when you don't remember the filename–at a flick of the POWER! **[RECLAIM]** command. (Still 31 commands to go!)

POWER! NOW FOR IBM's PC-DOS AS WELL AS CP/M

We first developed POWER! for CP/M two years ago, and a stack of testimonials from FORD to XEROX testify to its excellence. For IBM-PC™ users, special features like managing sub-directories, **[CHANGE]**, and a separate creation of up to 8 simultaneous, on-screen **[WINDOWS]** have been added.

MONEY-BACK GUARANTEE AND A 10 DAY TRIAL

POWER! has the Seal of Approval from the Professional Software Programmers Association, and you, too, must be happy with POWER!–or your money back! For only $169 you can now really be in control of your computer. Call Computing! at (415) 567-1634, or your local dealer. For IBM-PC or any CP/M machine. Please specify disk format.

The company that earns its exclamation point.

COMPUTING!

2519 Greenwich, San Francisco, CA 94123

TO ORDER CALL 800 TOLLFREE
800-428-7825 Extension 96C
In CA: 800-428-7824 Extension 96C

IBM and IBM-PC are registered trademarks of International Business Machines Corporation.

The Macrotech MI-286

A State of-the-art 80286/Z80 S-100 CPU Card

by Charles Strom

For several months, we have been working with several MI-286 CPU boards from Macrotech International. Always looking to improve throughput in leading-edge, multi-user, multi-tasking systems, Macrotech engineers have designed a board from the ground up to be plug-compatible with the CompuPro 8085/88 dual-processor board. This meant hardware and software compatibility, and therein lies a tale to be explored later in this article. The CompuPro 8085/88 board is a popular CPU board, due in large part to the reliability of hardware as well as the advanced technical features built into the CompuPro line allowing DMA transfers, interrupt-driven operation, etc.

A strong point in the design is the 8080-compatible processor on board. It is a simple matter for the user to insert a CP/M 2.2 disk, if desired, or change to a sixteen bit operating system such as CP/M-86, MP/M, MP/M 8-16, or Concurrent-CPM merely by inserting the appropriate boot disk. The beauty of MP/M 8-16 and other 8/16-bit operating systems, is that the user can transparently run virtually any CP/M-80 program. This is performed by automatic invocation of a utility which transfers control of the actual program to the eight-bit processor while funneling all operating system calls through to the sixteen bit chip. Macrotech recognized the value of this scheme and decided to bring the concept of the 8085/88 board up to date.

THE EIGHT-BIT PROCESSOR

The eight-bit chip is the Z80H running at 8MHz, as opposed to the *standard* Z80 4Mhz clock speed so common in the many eight bit machines produced today. There are extant a few proprietary programs that depend on the Z80 for operation, and many more public domain programs that use relative jumps, for example, and therefore crash on an 8085. Now we can run these utilities without incident; Gifford has even upgraded their SW.CMD (the "switch" program which is used to enable an eight-bit program to run under Gifford's MC-DOS) to support the extra registers of the Z80. What's more, the Z80H at 8MHz operates significantly faster than the 8085 at 5MHz on the older CompuPro board. There is 100% compatibility between the CompuPro and Macrotech boards in eight-bit operation as far as we can determine, except, of course, for the additional functionality we have gained through acquisition of a Z80 rather than an 8085. However, a program using the 8085 RIM and SIM (Read Interrupt Mask and Set Interrupt Mask) instructions, would not operate correctly (but we have never run across such a program).

There are several options available that configure Z80 operation. They include a choice of the default 8MHz operation or optional 2MHz operation for time-critical applications, Z80 reset on processor swap (when Z80 regains control after 80286 relinquishes control), and several wait-state generator options. We will discuss the latter along with the 80286 wait-state options below. Let it suffice to say that the Z80 side of the MI-286 operates as a perfectly domesticated eight-bit CPU board in all applications we have tested. Several representative benchmarks comparing the Z80H operation to equivalent 8085 tasks appear below.

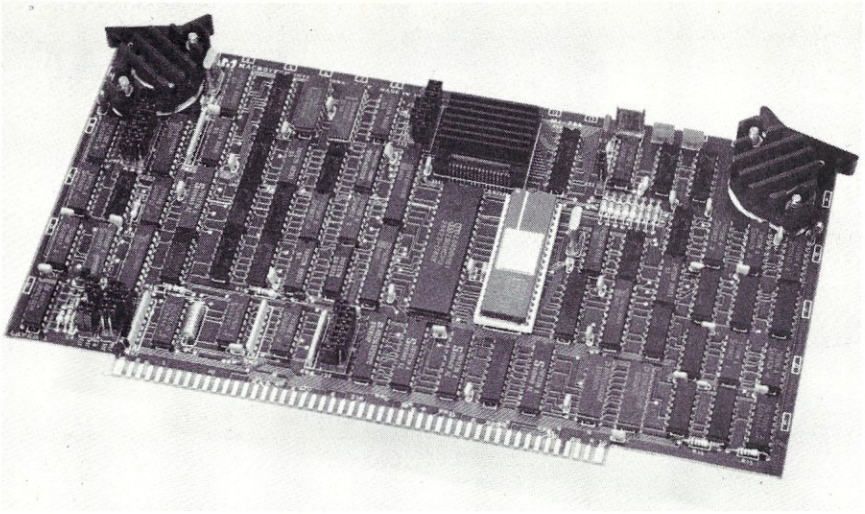
THE SIXTEEN-BIT PROCESSOR

I will devote considerably more space to a description of the 16-bit personality of the MI-286 processor board. The basic computational power is supplied by an Intel 80286 chip running at 6MHz. The 80286 is a state-of-the-art 16-bit processor. The basic set of registers, instructions and addressing modes are upward compatible with the 8086, 8088 and 80186 CPU's. Programs written for these CPU's will run unchanged on the 80286. Intel specifies clock rates of 4, 6 and 8MHz available for the 80286 in the preliminary datasheet, but 8MHz parts are not yet generally available and Macrotech is offering their board only with the 6MHz part.

The 80286 two basic operating modes, real address and protected virtual address. The chip's default mode is real, permitting direct addressing of one megabyte (a 20-bit address field). The details of the real addressing mode are nearly identical to that of the earlier Intel 16-bit chips so that all operating systems running on the 8086 or 8088 will operate without modification with the 80286. An exciting future growth path involves the protected virtual address mode. The protected mode offers one full one gigabyte of virtual address space mapped into 16 megabyte physical address blocks through a full 24 bit physical address field. In addition, the protected mode prevents critical instructions from affecting the CPU's execution, restricts writing to certain memory segments, and has a heirarchical system of task priorities.

Digital Research is reportedly designing a piece of software, called *Concurrent DOS-286*, which will operate with the 80286 and make full use of the protected mode of operation thus supporting a much-improved multiuser operating system. Unfortunately DRI seems to have a penchant for changing the name of their product on a monthly basis so this operating system will most likely be known by another monicker by the time it is released! The above-mentioned modes will allow a multiuser system to operate without crashing in the midst of a renegade application program. But not knowing anything concrete about this DRI project I cannot really make any significant pronouncements, though I do know our MI-286 is all ready to use it.

Some other 80286 features include 64K 8-bit (or 32K 16-bit) I/O ports, a six byte instruction prefetch queue, and pipelined addressing. The latter supports overlapping instruction fetches, instruction decoding and improved execution to increase processing throughput. Not being a microprocessor engineer, I do not feel competent to make a comparative analysis of the 80286 with respect to



other state of the art chips such as the National Semiconductor 16032 or Motorola 68000 families. However, it appears to me that the 80286 is an excellent choice of a processor that offers 100% upward compatibility with previous Intel chips, allowing me to run my current operating systems software without modification. In addition, I am looking forward to using the protected mode feature when the software becomes available.

MI-286 BOARD OPTIONS

There are a host of options on the board. Memory management control can be set as shown in table 1:

TABLE 1

Mode	Processor	A23-A20	A19-A16	A15-A0
0 (default)	Z80	latch	latch	direct
	286	latch	direct	direct
1 (virtual)	Z80	latch	latch	direct
	286	direct	direct	direct
2 (286 primary)	Z80	zero	latch	direct
	286	latch	direct	direct
3 (Z80 primary)	Z80	latch	latch	direct
	286	zero	direct	direct

Direct control of the upper address line refers to the existence of real address lines, while latched control involves writing data to the memory management port. The memory management port can be optionally cleared on reset. Mode zero sets up the process to be equivalent to the CompuPro 8085/88 addressing scheme, while mode one will presumably be used for Concurrent-286.

WAIT STATE SELECTION

Little did I realize when I received the first beta-test MI-286 board that the subject of wait state selection would consume so much of my attention. The Z80 wait state jumpers are flexible, permitting optional choice of one I/O, memory, and/or M1 wait. Our system operates properly with the factory default setting of no memory waits, having one I/O and one M1 wait state inserted for the Z80H. The 286 is a far more complicated proposition. Jumpers allow for zero through three wait states to be inserted for I/O and memory operations. Obviously, a primary design goal was to permit high speed operation (without 80286 waits) using CompuPro memory boards. Unfortunately, Macrotech developed their CPU board using their own 512K static memory board and had few problems in this area. They also tried some CompuPro boards from a nearby client's system, and true to Murphy's law, experienced no problems there either. But upon receipt of the original board, we were unable to operate our CompuPro RAM 16's or RAM 21's without inserting two memory waits. This slowed down the system to a speed which was unacceptable to Macrotech. After several weeks of of constant labor, Macrotech engineers discovered a bug in the current version of the 80286! To paraphrase Intel's 80286 (B-2/B-3) errata sheet of 22 May, 1984, if the 80286 executes a POPF (pop F) instruction, a pending maskable interrupt may be improperly recognized even if maskable interrupts were disabled before the POPF instruction and the value popped had IF=0. In an interrupt-driven operating system such as MP/M 8-16, this would be disastrous as indicated by randomly produced ''Panic! Interrupt'' messages. On the other hand, no such problem was encountered on non-interrupt-driven systems.

Intel's suggested fixes included recoding to eliminate POPF instructions. This is impossible since the MI-286 was designed to drop into an exisiting system with existing hardware and software. Another suggestion was insertion of two wait states. The latter fix, which we observed to operate effectively on our system, was also unacceptable because of the cost in speed. After considerable research, Macrotech determined that insertion of two wait states for every memory **data read** operation **only** would be the best compromise at this time. Thus, there is a special jumper for this operation on the current revision of the circuit board. The cost of this speed was determined by a **compute-bound** benchmark which decreased from 2.5 to 2.3 with use of the jumpers. Therefore, we are paying a worst-case price of 12% by inserting these waits.

MORE JUMPERS

There are a host of other jumpering options including reset options for both processors, 16/24 bit DMA addressing, MWRT enable, power-on-jump address selection, I/O base address selection, etc. There are four I/O ports defined for processor swap, memory manager address specification, and memory management mode control. Lastly, there are two jumpers (or ''sense switches'') available on the board which may be read as two bits when a specific port is read. These jumpers are not used by Macrotech (or CompuPro for that matter) and are available for custom programming purposes.

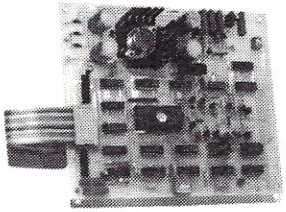
THE NUMERIC PROCESSOR

Macrotech offers an optional 80287 numeric processor. The 80287 is similar in instruction set to the 8087, but differs in that it may run asynchronously with the main processor. Thus, while the 80286 runs at 6MHz, the 80287 was initially set up to run at 4MHz. Macrotech engineers concluded that the numeric processor will run reliably at 5.33MHz. Thus, they supply a simple patch to change the clock frequency. There have been no problems whatsoever in several months of operation.

The 80287 chip is an optional extra with a list price of $375. It is also possible to purchase the chip from another source at a lower cost, but Macrotech points out that selection of this option with the initial board purchase will include complete testing of that 80287 chip in place as well as a one year warranty. Rumor has it that some competitors have designed their CPU boards in such a manner that it is impossible for a user to field-install his own numeric processor, but Macrotech advises that their design allows the owner to simply plug and go. Consequently, those of us with the ''hacker'' spirit can roll their own 80287 installation at any time.

CP/M EPROM PROGRAMMING SYSTEM

2708
2758
2716
2732
2764



2516
2732A
27128
27256
27CXX

-SOFTWARE SPECIFICALLY DESIGNED FOR CP/M COMPUTER SYSTEMS
- STAND ALONE BOARD - ELECTRONIC SWITCHING OF EPROM TYPES
- USES 24 VOLT XFMR FOR POWER - ALL SUPPLIES/TIMING ON BOARD
- DESIGNED WITH EASY TO GET PARTS - LARGE COMPREHENSIVE MANUAL

*** * CENTRONICS INTERFACE * ***

CONNECTS TO ANY CENTRONICS PRINTER INTERFACE - USES 8 OUTPUT DATA BITS AND ONE INPUT DATA BIT (BUSY LINE). BUSY LINE IS A HIGH SPEED SERIAL INPUT. FULL EPROM READING AND PROGRAMMING INSTALL PROGRAM PROVIDED FOR QUICK INSTALLATION TO HOST SYSTEM

*** * CONTROL PROGRAM COMMANDS * ***

- PROGRAM EPROM(S) FROM DISK
- READ DISK FILE INTO RAM
- READ EPROM(S) INTO RAM
- VERIFY EPROM IS ERASED
- SAVE EPROM(S) TO DISK
- PROGRAM EPROM(S) FROM RAM
- COMPARE EPROM WITH RAM
- COPY EPROM
- DISPLAY/MODIFY RAM - (MONITOR MODE) WITH 11 SUB COMMANDS
  FILL-DUMP-XFER-EXAMINE-MODIFY-BIAS-PROGRAM-VERIFY, ETC.)

BARE P.C. BOARD WITH COMPLETE DOCUMENTATION
AND SOFTWARE ON 8" SINGLE DENSITY DISKETTE $69

(ABOVE WITH COMPLETE PARTS KIT - $169)(A&T - $189)
SOFTWARE AVAILABLE FOR OSBORNE, KAYPRO AND OTHER 5 1/4 FMTS

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL
ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752-7218

CALL OR WRITE FOR MORE INFORMATION -- ADD $3.00 FOR SHIPPING
OHIO RES. ADD 5.5% TAX -- VISA/M.C. ACCEPTED -- $3.00 FOR COD

Still Searching For Files Without EUREKA!™ ??

You may not know it, but a disk cataloger can be a big help in managing your computer files. Why not go with the best? **EUREKA!** is a terrific time saver for ...

- Lawyers
- Software Developers
- Writers
- Teachers
- Project Managers
- Accountants
- Researchers
- Secretaries
- Consultants
- Journalists

People who try **EUREKA!** love it.....

"Just started cataloging with comments - Great Idea" GR-MI

"Great time saver in locating material on disks." WB-NY

"Your manual is the best written I have ever seen." MT-NS

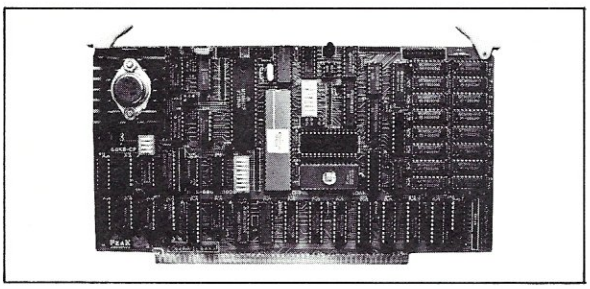
"We finally chose EUREKA! ... largely because it has the ability to read comments directly from a file ... EUREKA! is easy to learn and use, has more access and report choices, finds files by many different ways, and has an attractive price."
T. Bove & C. Rhodes, USER'S GUIDE No. 11

EUREKA!, the popular CP/M® disk cataloger
Still only $50

MENDOCINO SOFTWARE COMPANY, INC.
Dept. M -3
P.O. Box 1564
Willits, CA 95490
(707) 459-9130

add $2.50 shipping
Calif. residents add 6% sales tax
VISA & MasterCard accepted

A EUREKA! package is designed to run on only a single system.
Licenses for additional systems (for a single user) are $15 each.
EUREKA! is a trademark of Mendocino Software Company, Inc.
CP/M is a registered trademark of Digital Research, Inc.



68K8-CP

The World's First 68000 Coprocessor

Peak Electronics' 68K8-CP is a high performance processor card designed to function as one of several CPU elements within a multi-processor S-100 system. This card features the MC68008™ (8-bit object code compatible version of the industry renowned 68000), up to 512K bytes of RAM and 128K bytes of EPROM, two serial ports and a parallel printer port.

The true power of the 68K8-CP card is shown with its ability to be plugged into any existing S-100 system running CP/M®-2.2 and to be running CP/M-68K within minutes without any change in existing hardware or software. This card does not replace your current processor. All of the original system's devices (RAM, disks, and other peripherals) are immediately available to the user of CP/M-68K, files can be accessed by whichever operating system is currently active.

Features:
- Does not replace your current CPU card
- IEEE-696-1983, S-100 Compatible
- MC68008 8 or 10 MHz CPU
- 128K bytes of RAM expandable to 512K
- 8K bytes of EPROM expandable to 128K
- No wait state access to on board RAM and EPROM
- Two high speed serial ports (up to 38.4K BAUD)
- 8-bit parallel printer port
- Supports CP/M-68K and Concurrent-68K
- Onboard 16-bit counter-timer
- Includes printer buffer and RAM disk firmware

$995.00, includes CP/M-68K

PEAK electronics
P.O. Box 700112
San Jose, CA 95170-0112
(408) 253-5108

Now With Windowing!
$49.95 Basic Compiler

MTBASIC

Features:

Multitasking	Windowing
Handles interrupts	Interactive
Fast native code	Compiles quickly
Floating point	No runtime fee

MTBASIC is a true native code compiler. It runs *Byte's* Sept. '81 sieve in 26 seconds; interpreters take over 1400 seconds! Because MTBASIC is multitasking, it can run up to 10 Basic routines at the same time, while displaying ten separate windows. Pop-up/down menus are a snap to implement.

The MTBASIC package includes all the necessary software to run in interpreter or compiler mode, an installation program (so *any* system can use windowing), three demonstration programs and a comprehensive manual.

AVAILABLE for CP/M (Z-80), MS-DOS, and PC-DOS systems.

ORDERING: Specify format when ordering. We accept Visa, MC, checks and COD. Send $49.95 plus $3.50 shipping and handling ($10 overseas) to:

SOFTAID, Inc.

P.O. Box 2412 Columbia, MD 21045-1412
301/792-8096

INSTALLING THE BOARD

The MI-286 board was designed from the bottom-up specifically as a plug-and-go replacement for the CompuPro 8085/8088 CPU board. As such, great pains were taken to make the transition to the new board as simple as possible. Since the MI-286 is considerably faster than the older board, timing of the bus signals is a serious concern. Over months of beta testing by a small group followed by actual field experience with the real product, Macrotech engineers have compiled a detailed dossier of suitable solutions for problems encountered during the upgrade process. The manual provides detailed guidelines on the proper setting of wait states and other critical CompuPro board switch settings. There is also considerable information on hard disk controller boards of other manufacture, especially those offered by Gifford Engineering, as that firm has also taken part in beta testing of the MI-286, having adopted the CPU board as the primary offering in their S-100 systems. Hard disk controllers which are known to work satisfactorily, include the CompuPro Disk 2 and Disk 3 and the Konan DGC-100. There are definite problems with Morrow hard disk controllers. I would suggest that Macrotech be consulted before attempting to upgrade a system with a Morrow controller.

Both Macrotech's and CompuPro's static memories operate properly with the MI-286. All existing CompuPro I/O boards have been shown to operate correctly. If one intends to upgrade a non-standard system with the MI-286, I strongly recommend that the prospective purchaser contact Macrotech directly. It is my experience that the company is willing to work with system integrators to insure the best chance of success. In a worst case situation, the board might have to be returned (for full credit), but considering the substantial experience the troubleshooting engineers have gained over the past months, this is unlikely.

BENCHMARKS

In evaluating the CompuPro 8085/8088 and the Macrotech MI-286 processor boards, the bottom-line is found by comparing the execution of specific operations. Paul Homchick of Chimitt, Gilman, Homchick, Radnor, PA has done an exhaustive benchmarking and has graciously granted permission to present relevant details from his article. The full report is available directly from Macrotech or from Section DL9 of CP-MIG, the CP/M special interest group on Compuserve.

SIXTEEN BIT BENCHMARKS - First, a few simple machine instructions were executed 12,582,912 times. The running times(in seconds) are:

	8MHz 8088	6 Mhz 80286	ratio
add ax,1	18.9	7.4	2.55
mov ax, [30]	31.6	11.6	2.72
short JMP	28.3	21.1	1.34

The two arithmetic instructions were executed 4,194,304 times:

mul Word [30]	75.0	14.2	5.28
mul Byte [30]	45.6	8.6	5.30

The Sieve of Eratosthenes is a very widely known and oft-used benchmark. Homchick coded it in Digital Research CB-86. Using either 8-or 16-bit wide RAM, the 8088 ran the program in 8.1 sec. The Macrotech 80286, using 16-bit wide memory and one wait state for memory reads only (a typical setup for a system using CompuPro memory) executed the same program in 3.6 sec. Thus the Macrotech MI-286 represents a 2.25-fold improvement in speed over the 8088.

EIGHT-BIT BENCHMARKS

I compiled a C language version of the sieve program with the Software Toolworks' C80 compiler. CompuPro 8085 clocked in at 13 sec., while the Macrotech Z80H (using one M1 wait state, a typical requirement for CompuPro memory) ran the benchmark at 10.8 sec. Note that an additional advantage over the CompuPro board would be realized with a compiler that produced optimized Z80, rather than 8085 (8080), code.

BENCHMARK SUMMARY

Homchick went into further detail in comparing the 80287 processor to a variety of other systems, as well as timing several "real-life" operations such as file copying, spreadsheet operation, etc. To summarize, the MI-286 out-performed the 8085/8088 yielding 2.3 and 1.2-fold decreases in the speed of 16- and 8-bit operations, respectively, during compute-bound applications. Application programs will vary widely vary of course, but it is reasonable to expect a minimum improvement in speed of 1.2 to 1.7-fold. Needless to say, the addition of the 80287 coupled with software designed to use it, will improve the speed of math-intensive applications up to two orders of magnitude.

CONCLUSION

As a devoted S-100 bus enthusiast, I have felt privileged to work with the MI-286, a truly state-of-the art CPU board. Clearly, the S-100 bus has a promising future notwithstanding the tales of its demise so frequently reported in the trade press. Firms, such as Macrotech and CompuPro, are certainly in excellent technical health and I look forward to more advanced products from them in the months ahead.

MI-286 Z80/80286 S-100 CPU CARD
Macrotech International Corp.
9551 Irondale Avenue
Chatsworth CA 91311
(818)700-1501
Prices:

MI-286	$1,095
MI-287 (80287 Coprocessor)	375

Users' Group

Over 40 volumes of public domain software including:

- compilers
- editors
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details

The C Users' Group
415 E. Euclid • Box 97
McPherson, KS 67460
(316) 241-1065

BOBCAT 3
DISK CATALOG
The most versatile and powerful catalog program available

- creates, adds, updates, and deletes a filename catalog
- seven report formats
- hard disks, multiple drives, and user numbers
- individual file titles
- wildcard searches for filenames and file titles
- CP/M or PC/MS DOS

8" CP/M SSSD or Popular CP/M or PC/MS DOS 5¼"

U.S. residents ..... $49.95 U.S.$
Canadian residents ... $49.95 Cdn $ (Ont. residents add $3.50 pst)
Other countries ..... $54.95 U.S.$ plus $3.00 P & H

NAME, NUMBER EFFECTIVE & EXPIRY DATES (MC FOUR EXTRA DIGITS)
Bank drafts; certified checks; money orders; company checks

R&L MicroServices Inc.
Box 15955, Station F
Ottawa, Ont.
K2C 3S8 (613) 225-7904 THE HOME OF THE BOBCAT

LOMAS DATA PRODUCTS INVITES YOU TO:

SHARE THE THUNDER.

The S100-PC-TM offers the following standard features:

- ☐ High performance THUNDER186 8Mhz 80186 processor
- ☐ 512K bytes of RAM (expandable to 1Mbyte)
- ☐ 4 serial ports to support up to four users
- ☐ 3 Centronics compatible parallel ports
- ☐ Concurrent DOS operating system allows execution of both CP/M-86 and MS-DOS (PC-DOS) programs
- ☐ 5¼" IBM-PC compatible floppy drive
- ☐ 40 Mbyte high performance Winchester drive
- ☐ Attractive 10 slot desktop enclosure

In addition, a number of options are available including: larger Winchester drives, more user ports, 80286 processor, graphics support and additional operating systems (MS-DOS and CP/M-86).

S100 BUS boards products & support for the system integrator . . .

All of LDP boards are fully tested to exacting standards and carry a one year warranty. We specialize in 16-bit products & support the major operating systems for 16-bit processors: CP/M-86*, CONCURRENT CP/M-86*, and MS-DOS (PC-DOS).

■ **THUNDER186 — THE ONLY COMPLETE S100 BUS, 16 BIT SINGLE BOARD COMPUTER AVAILABLE TODAY.**
Concurrent CP/M-86, which in addition to running CP/M-86 programs, runs MS-DOS programs. Comes complete, ready to plug into an enclosure and run. 256K bytes of RAM only . . . . . **PRICE $1595.00**

■ **LIGHTNING ONE***8086/8088 CPU**
8086 or 8088, with 8087 and 8089 coprocessors. Up to 10 MHZ operation . . . . . **PRICES start at $425.00**

■ **HAZITALL SYSTEM SUPPORT BOARD**
2 serial, 2 parallel ports, battery protected clock calendar. Hard disk controller host interface . . . . . **PRICE $325.00**

■ **LDP 128/256K DYNAMIC RAM**
Advanced dynamic RAM with LSI controller for failsafe operation, parity . . **PRICE** 128K—**$395.00,** 256K**$649.00**

■ **RAM67 HIGH PERFORMANCE STATIC RAM**
High speed (100ns) low power CMOS static RAM. 128K bytes, extended addressing . . . . . **PRICE $995.00**

■ **LDP72 FLOPPY DISK CONTROLLER**
Single/double density, single/double sided disks, both 8" and 5¼" inch drives simultaneously . . . . . **PRICE $275.00**

■ **LIGHTNING 286—80286 CPU BOARD**
Offers 4 times the performance of a 5MHZ 8086 CPU while maintaining software compatibility . . . . . **PRICE $1095.00**

■ **OCTAPORT 8 PORT SERIAL BOARD**
0 to 19200 baud operation real time clock interrupt. Ideal for multi-user systems such as MP/M-86* **PRICE $395.00**

S100-PC-TM is a trademark of Lomas Data Products, Inc.
**CP/M-86, MP/M-86 and CONCURRENT CP/M-86 are trademarks of Digital Research.* ***MS-DOS is trademark of Microsoft.*
****Lightning One is trademark of Lomas Data Products, Inc.*

S100-PC-TM: The LDP Multi-user S100 Bus System offers high performance at a "low" price . . . plus, "our" system is expandable and upgradeable!

PRICE An unbelievable **$6995.00**

Call today!

LDP

LOMAS DATA PRODUCTS, INC.
66 Hopkinton road, Westboro, MA 01581
Tel: (617) 366-6434 ☐ Telex: 4996272

Dealer inquiries invited.

For orders outside the U.S., contact our exclusive dealers: ☐ **Australia** — LAMRON PTY. LTD., (02) 85-6228
☐ **Malaysia** — EXA COMPUTER (M) SENDIRIAN BERHAD, 795284

RP/M T.M.

By the author of Hayden's "CP/M Revealed."

New resident console processor RCP and new resident disk operating system RDOS replace CCP and BDOS without TPA size change.

User 0 files common to all users; user number visible in system prompt; file first extent size and user assignment displayed by DIR; cross-drive command file search; paged TYPE display with selectable page size. SUBMIT runs on any drive with multiple command files conditionally invoked by CALL. Automatic disk flaw processing isolates unuseable sectors. For high capacity disk systems RDOS can provide instantaneous directory access and delete redundant nondismountable disk logins. RPMPIP utility copies files, optionally prompts for confirmation during copy-all, compares files, archives large files to multiple floppy disks. RPMGEN and GETRPM self-install RP/M on any computer currently running CP/M®2.2. Source program assembly listings of RCP and RDOS appear in the RP/M user's manual.

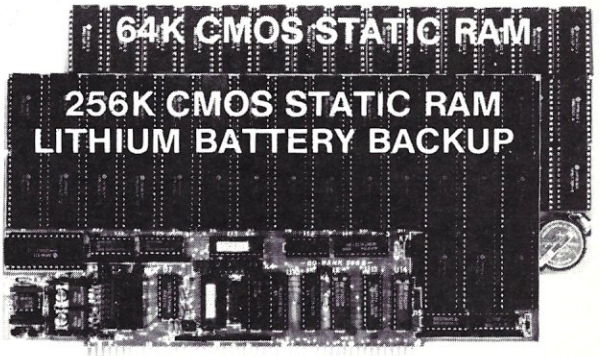
RP/M manual with RPMGEN.COM and GETRPM.COM plus our RPMPIP.COM and other RP/M utilities on 8" SSSD $75. Shipping $5 ($10 nonUS). MC, VISA.

118 SW First St. - Box G
Warrenton, OR. 97146

Micro Methods, Inc.

(503) 861-1765

PRAM PERMANENT RAM



GUARANTEED IN YOUR SYSTEM

CROMIX-D • MPM • CCS • OASIS • AMOS

✓PLUS: 8/16 BIT TRANSFERS • 24-BIT EX. ADDRESSING
8-12 MHZ • 2K DESELECTS • RAM-EPROM MIX
IEEE 696/S-100 • LOW POWER • FULLY STATIC

LITHIUM BATTERY BACKUP avoids power failure crashes intelligently. Unique POWER-FAIL-SENSE circuit allows processor to save register information and disable board before POWER FAILURE CRASHES memory.

CALL!

BG BANK 256S ....~~$1,299~~ **Battery Backup**.....**$79**
BG BANK 64S .....~~399~~ **Battery Backup** **39**

BG COMPUTER APPLICATIONS, 206 Brookside, Bryan, Texas 77801. International orders add 30%.

(409) 775-5009

NEW! Advanced Trace86™

Symbolic Debugger & Assembler Combo

- Full-screen trace with single stepping; Even backstepping!
- Write & Edit COM & EXE programs
- Conditional breakpoints (programmable)
- Switch between trace and output screen; Or set up two monitors
- 8087, 80186, 80286, 80287 support
- Write labels & comments on code
- Polish hex/decimal calculator
- and more . . . Priced at $175.00

To order or request more information contact:

Morgan Computing Co., Inc.

(214) 245-4763

P.O. Box 112730, Dallas, TX 75011

d/MULTI

MULTIUSER dBASE for TurboDOS

TRUE File and Record Locking as easy as dBASE II. Unlimited users can perform the magic of dBASE in the program or interactive mode

* TurboDOS 1.3 or 1.4
* No Peeks or Pokes
* System Date and Time Functions
* Printspooler Controls up to 16 printers

Martian Technologies...
.CREATEing Multi-users from
Single-users around the world

CALL FOR DETAILS

Martian Technologies
8348 Center Dr. Ste.-F La Mesa Ca. 92041
(619)464-2924

The Tools You Need To C You Thru.

Now the *WizardWare*™ Applications Programmers Toolkit provides everything you need to increase your C programming productivity.

APT™ features include:

- COMPLETE SOURCE CODE (over 5000 lines!)
- File handling with direct & keyed access
- Screen and Report Generators, with full screen handling for your programs
- Generic Terminal Driver for portable code
- String math functions, and string manipulation routines
- Reference Manual on Disk (over 50 pages)
- Tutorial Manual (over 25 pages) with Source for Mailing List Manager
- A host of useful Utilities, Database and File Editors
- Available for Lattice C, Mark Williams C, DeSmet C, BDS C, others.

Also Available: C-STARTER Toolkit, great for learning C!! Includes: Customized APT, DeSmet C Compiler, and "Programming in C on the IBM-PC" (200 pages)

APT/MS-DOS versions	$495
APT/DeSmet C version	$395
APT/BDS C version	$395
C-Starter (binary APT, DeSmet Compiler and Book)	$295
APT/Manual only	$ 50

Detailed Brochures on request

*Manual Cost will be applied if APT purchased within 30 days ($10 re-stocking charge.) U.S. funds only, please.

Trademarks: MS-DOS/Microsoft, Lattice C/Lattice, Inc., MWC86/Mark Williams Co., DeSmet C/C Ware, CI-C86/Computer Innovations, Inc., BDS C/BD Software, DR-C/Digital Research, WizardWare, APT, C-Starter/Shaw = American Technologies.

Call (502) 583-5527

Ask for APT™ or C-Starter, or Send Check to:

Shaw ☆ American Technologies

WizardWare™

830 South Second St. · Box 648
Louisville, KY 40201, USA

(C.O.D. and Foreign Orders - Add $5 Shipping/Handling)
References: Bank of Louisville, Citizens Fidelity Bank, Louisville Chamber of Commerce

68000
BROWSE for CP/M-68K

Look at any CP/M-68K file

Display ANY CP/M-68K disk file in ASCII or HEX. Scroll UP, DOWN, LEFT, or RIGHT any amount via PFKEYS or commands. Online HELP plus an extensive manual. Very powerful and flexible FIND command searches the file and highlights the found string. Repeat FIND, forward and reverse searches, hex, ascii, etc. etc. PRINT will print a line, group of lines, the screen image or the complete file. HELP, TABS, LOCATE, KEYS, CAPS, COLS etc. Support for PFKEYS, PRINTER, ANSI X3.64 and other terminals is provided through a menu driven configuration program. Written completely in 68000 assembler. Shipped on 8" SSSD (IBM 3740) diskette.

WB SOFTWARE DEVELOPMENT
112 OAKHAMPTON PL. S.W.
CALGARY, ALBERTA CANADA
T2V 4B2 (403) 238-3216

$49.95 US

MasterCard VISA

CP/M-68K trademark of DIGITAL RESEARCH.

C Source Code for the PC

Concurrent C	$45.00
LEX	$25.00
YACC	$25.00
Tools	$15.00

Austin Code Works
11100 Leafwood Lane
Austin, TX 78750
512-258-0785

A N N O U N C I N G

SOURCE-IT™

The Cross Reference / Documentation Aid for Microsoft Basic

KEY FEATURES

- Formatted Source Listing Complete With All Lines Referenced
- Alphabetized Listing of Variables, Files, Calls, Compiler-With Referencing Line Numbers
- Processes Up To 100 Programs At One Time
- Not Copy-Protected - May Be Used On Hard-disks
- 30 Day Money Back Guarantee

Special Introductory Offer $89.99
Call Or Write For A Free Brochure

SOURCE-IT Is Available For Most Micro Computers

Software Innovations, Inc.
16621 Dolores St., Huntington Beach, CA 92649
(714) 840-6411

The Public Domain Software Copying Company

est. 1983

brief Catalogue & Sample Disk (2 sided) $12

PC-DOS MS-DOS © SIG/M
also
Commodore, CPM/UG Apple II,
~ Osborne Dealer ~

33 Gold St. L3M N.Y.C. 10038
212-732-2565

Overseas include Post.-N.Y. incl. tax
Prepaid Only

©copyrights acknowledged

THE SOFTWARE DIRECTORY

When contacting software publishers please mention that you read about them in Micro/Systems Journal.

Program Name: Instant-C V1.0
Requirements: MS-DOS, CP/M-86 or MP/M-86
Minimum Memory: 320K
Language: Assembler
Description: Full C-language integrated package that includes a full-screen editor, a compiler/interpreter and source level debugger. Claims to be "the fastest interpreter and C compiler for MS-DOS." Includes float's and long's. Compatible with other C-compilers such as Lattic-C. Over 200 diagnostic messages.
Price: $500 Free 2.0 update included
Publisher: Rational Systems
Box 480
Natick MA 01760
(617)653-6194

Program Name: ELECTRA-FIND
Requirements: CP/M 2.2
Description: Does global search of disk and retrieves all instances of searched word or phrase, in context, tells which file each came from and puts results in new file. Menu driven with 20 search options. User can set automatic defaults.
Price: $50 + $4 shipping. 30-day money-back guarantee.
Publisher:
O'Neill Software
Box 26111
San Francisco CA 94126
(415)398-2255

Program Name: DSD80 Symbolic Debugger
Requirements: CP/M-80 or equiv.
Description: Full screen symbolic debugger for 8080,8085 and Z80 code with display of instruction, registers, stack and two memory windows. DDT compatible. Includes commands for port I/O, string searching and defining symbols. Single keystroke commands for single stepping and scrolling. In-circuit-emulator protects against program crashes with write protected memory, execute-only code and stack overflow protection. Z80 instruction set fully supported using either extended Intel or Zilog mnemonics. On-line help and 50-page user manual.
Price: $125 plus shipping
Publisher: Soft Advances
Box 49473
Austin TX 78765
(512)478-4763

Introducing the MIX Editor

(with Split Screen - both horizontal and vertical)

A Powerful Addition To Any Programmer's Tool Box

Full Screen Editing
WordStar Key Layout
Custom Key Layouts
Terminal Configuration
Help Files
Backup Files

Introductory Offer Only

29.95

30 Day Money Back Guarantee

Programmable
Macro Commands
Custom Setup Files
Mnemonic Command Mode
Multiple File Editing
Split Screen Editing

For PCDOS/MSDOS (2.0 and above/128K) • IBM PC/Compatibles, PC Jr., Tandy 1000/1200/2000, & others
For CPM80 2.2/3.0 (Z80 required/64K) • 8" SSSD, Kaypro 2/4, Osborne I SD/DD, Apple II, & others

Great For All Languages

A general purpose text processor, the MIX Editor is packed with features that make it useful with any language. It has auto indent for structured languages like Pascal or C. It has automatic line numbering for BASIC (255 character lines). It even has fill and justify for English.

Terminal Configuration

A utility for defining terminal features (smart features included) allows the editor to work with any terminal. Over 30 of the most popular terminals are built-in.

Custom Key Layouts

Commands are mapped to keys just like WordStar. If you don't like the WordStar layout, simply change it. Any key can be mapped to any command. You can also define a key to generate a string of characters, great for entering keywords.

Split Screen

You can split the screen horizontally or vertically and edit two files simultaneously.

Macro Commands

The MIX Editor allows a sequence of commands to be executed with a single keystroke. You can define a complete editing operation and perform it at the touch of a key.

MIX software
2116 E. Arapaho
Suite 363
Richardson, Tx 75081
(214) 783-6001

MSDOS is a trademark of Microsoft
PCDOS is a trademark of IBM
CPM80 is a trademark of Digital Research
WordStar is a trademark of MicroPro

Custom Setup Files

Custom keyboard layouts and macro commands can be saved in setup files. You can create a different setup file for each language you use. The editor automatically configures itself using a setup file.

Command Mode

Command mode allows any editor command to be executed by name. It is much easier to remember a command name versus a complicated key sequence. Command mode makes it easy to master the full capability of the editor. Frequently used commands can be mapped to keys. Infrequent commands can be executed by name.

Editor Commands

The editor contains more than 100 commands. With so many commands, you might think it would be difficult to use. Not so, it is actually extremely simple to use. With command mode, the power is there if you need it, but it doesn't get in your way if you don't. Following is a list of some of the commands.

Cursor Commands

Left/Right/Up/Down
Tab Right/Tab Left
Forward Word/Backward Word
Beginning of Line/End of Line
Scroll Up/Scroll Down
Window Up/Window Down
Scroll Left/Scroll Right
Top of File/Bottom of File
• • •

Block Commands

Copy/Move/Delete
Read/Write
Lower Case/Upper Case
Fill/Justify
Print

File Commands

Directory (with wild cards)
Show File/Help File
Input/Output File
Delete File/Save File

Other Commands

Split Screen/Other Window
Find String/Replace String
Replace Global/Query Replace
Delete Line/Undelete Line
Delete Word/Undelete Word
Insert Mode/Overwrite Mode
Open Line/Join Line
Duplicate Line/Center Line
Set Tab/Clear Tab
• • •

To Order: **Call Toll Free 1-800-523-9520,** (Texas only **1-800-622-4070**)
Mix Editor ___ $29.95 + shipping ($5 USA/$10 Foreign) Texas residents add 6% sales tax

Visa ___ MasterCard ___ Card # _____ Exp. Date _____
COD ___ Check ___ Money Order ___ Disk Format _____
Computer _____ Operating System: MSDOS ___ PCDOS ___ CPM80 ___
Name _____
Street _____
City/State/Zip _____
Country _____
Phone _____

MIX software
2116 E. Arapaho
Suite 363
Richardson, Tx 75081

Dealer Inquiries Welcome
Call (214) 783-6001

PT

The Slicer

By William Earnest

A High Performance, 80186-Based Single Board System For The Technical User, Available As Boards, Kits, or Assembled

Although the S-100 bus provides great flexibility and capability in the microcomputer world, there is also a place for more fixed designs. Single-board computers often represent a good combination of performance and cost that make them the preferred choice in many applications. One of these that I have been working with recently is the Slicer, from Slicer Computers & Controls.

THE HARDWARE

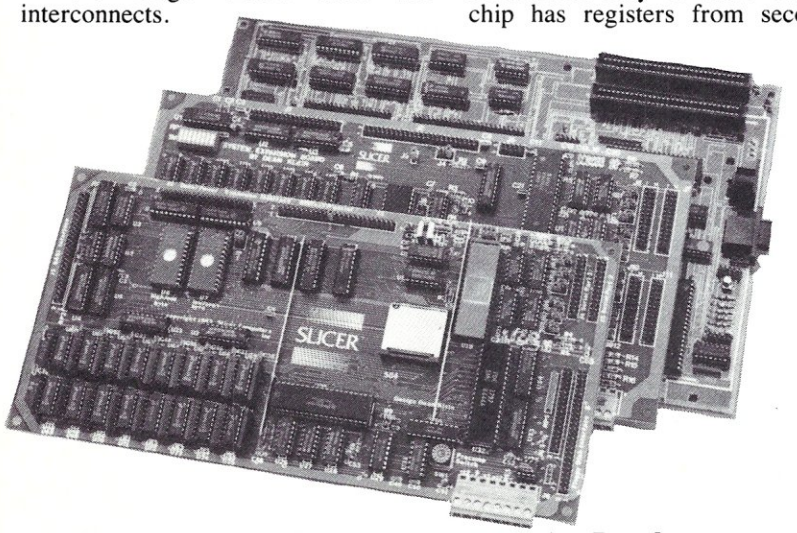
The basic system is on one printed circuit board a bit under 6'' by 12''. This contains a 80186 CPU, running at 8 Mhz, 256k bytes of memory, two serial ports capable of rates up to 38,400 baud, a floppy disk controller supporting both 8'' and 5.25'' drives, and a SASI compatible port that will interface to a hard disk controller such as the Xebec or Western Digital.

ROM circuitry supports up to 64k of code via 27256 chips, and a 8k byte monitor comes with the system (more on that later).

While this may seem to be quite a bit, the user needing more can add expansion boards. Along one edge of the main board are two headers carrying the data, address and control lines that might be needed. The boards are meant to be stacked with spacers, and the bus structure connects the boards via ribbon cable jumpers. While not as easy to reconfigure as a mother-board with edge connectors, this approach should be quite reliable, and allows future expansion without the initial investment in the large board used for interconnects.

The basic expansion board adds several capabilities to the system. It can add another 256k bytes of RAM, a parallel port for a printer, a real-time clock chip with battery backup, and 4 more serial ports, 2 of the same type as on the main board, and 2 that support both asynchronous and synchronous communication at rates to at least 1M bits per second. To enable these higher rates, interface is provided as TTL levels in addition to the standard RS-232 level conversion. For high speed burst transfer, a tie is provided between the Z8530 controller and the processor wait line so that string move instructions may be used. The clock chip has registers from seconds to years, with an internal 100 year calendar that handles leap years. It also has 50 extra byte size registers that are retained by the backup battery, where you can keep flags or data useful for system configuration or recovery from power failure.

Other expansion boards are now becoming available from slicer. One is a video display controller and keyboard interface, set up to match the structure of the IBM-PC. Also available is a prototype development board, a ''forest of holes'' with power bus and interface circuitry (address and data buffers, and high-order address decoding). This provides an ''out'' for those who need some non-standard interface or have an urge to tinker. I am presently awaiting these boards, with specific applications in mind for each.



The Slicer, System Expansion and PC Expansion Boards.

THE MONITOR ROM

As I mentioned earlier, a monitor is supplied in ROM as a part of the main board. This provides support for all the hardware on this board, and a flexible set of debugger commands. Included with the standard dump, modify memory, examine register and such, are a set which can access any disk on the system, determine its type, and read or write multiple sectors. With the command processsor having looping capability, it is easy to issue a command line to do a full disk to disk copy from the monitor. Also available are a set of commands to read and write all I/O ports for checking out hardware. One nice feature is a memory test section that needs no memory. By clever use of registers, the monitor can start up, communicate with the terminal, and execute the memory checkout with NO good RAM on the board. This makes it easy to isolate problems that might appear to be elsewhere, but are actually due to bad memory chips.

To support the technically inclined user for whom this system is intended, it is made available in several forms.

Starting with a bare board (which includes special parts), kits progress through ones having the hard to get parts (easy kits), to full kits, lacking only solder and some of your time. Assembled and tested systems are also available for those not interested in building.

The expansion board is sold similarly, with the added flexibility of getting

one with components for a memory board, an I/O and clock board, or both for a full house.

SOFTWARE

Software to use the board is also available. Starting with CPM-86 ($85) as the "standard" system, MS-DOS ($175) is now available, with Concurrent DOS hoped for soon if licensing arrangements can be straightened out. The system can boot up from any drive, with the default being the A drive, but a monitor boot command allowing selection of any other. The system will auto-configure itself, recognizing what type of drive is at each position, and the density and format of the disk inserted. A SETUP program will interactively patch a data area of the boot routine placed on track 0 of the system disk, to customize other details of your system. Such things as the CRT screen clear and cursor positioning commands are defined so that the monitor (and user programs that want to) can be terminal independent. Drive step rates are also set here, as are timeout delays for drive deselect and motor shutoff. The hard disk details such as number of heads, tracks, and capacity get defined, and the system partitions the available space into the needed number of logical drives. A memory disk may be defined for whatever amount of RAM you feel you can spare from program space. Two logical drives (E & F) are reserved to reference any two of the floppy drives, but with different format definitions, aiding file transfer between different type systems.

The way I got involved with the Slicer computer is a little tale with some interesting points itself. Last summer, a computerfest known as SOG III took place the last weekend of July in Bend, Oregon. Sponsored by Micro-Cornucopia, a hobbyist magazine primarily for single board systems such as the Kaypro, Xerox 820, and the like, it fit in nicely with our vacation plans and provided an excuse to visit a part of the country I had never seen. Among the people there were the principal hardware and software designers of the Slicer. I had been reading about the various 16 bit system coming out, but hadn't been happy with some of the technical compromises in the more popular "compatible" systems. Their presentation of the design choices made sense, and they made quite clear some of the hazier areas of this CPU type.

The next item was an invitation to build a system the next morning. The Slicer people had brought along several boards and bags of components, and ran a kit building session that Saturday morning on the front porch of the meeting hall.

Several brave souls turned up, and dug in. The experience ranged from several long time builders to some such as my wife, who could solder but didn't really have much technical experience. In a period of about 4 hours, thirteen kits were assembled, and then checked one at a time in a system cabinet with power supply, terminal and disk drive. The initial checkout without RAM showed all 13 working the first time, a very good yield. After correcting one or two reversed memory chips, all systems passed with no more problem than a defective socket on one board.

The builders had the option of buying the board they had built at a show discount, and many took advantage of the offer, including yours truly. It was refreshing to get sensible answers to your questions, and the hands-on experience was an excellent way to judge the quality of design and material. One thing that surprised several people there who had not seen the new chips was the size of the CPU. Packaged in a 68 lead chip carrier, the processor is less than 1" square, looking quite small in relation to the 40 pin DIP packages used for the peripheral devices. It seems that as the capability goes up, the size goes the other way.

OPINION DEPARTMENT

This has been my first experience in the 16-bit micro system world, and I have been having great fun. I can't comment on how the many commercial software packages work because I just haven't had occasion to try them. My interests being more along the line of system level software, the evenings have been spent with writing drivers, modem programs, poking around in the monitor and trying its many capabilities. Much of this was to develop a "feel" for the expanded instruction set of the CPU, and learn how best to manage the larger available memory. Some problems can be handled so much more easily when you aren't cramped into a 64k memory.

One software package that I can mention positively is the Turbo Pascal system by Borland. I am not particularly a fan of Pascal, with the strict checking of the compiler discouraging some of the shortcuts that assembly language programmers resort to. This version, however, is very easy to work with, and has enough enhanced system calls to let you do what you need. One of the intended uses for the Slicer involved moving an existing program from a Z-80 system. Written in an older dialect of Basic, an extensive rewrite was needed, and Turbo was chosen for the project. Once the data file formats were converted, the rest went surprisingly easily.

The system still runs in an experimental configuration most of the time. At present it has two 8" disks, two 5.25" drives (one each 48 tpi and 96 tpi), and a 34Mbyte formatted hard disk. It will definitely take a while to fill up all that space on the hard disk, but I am sure the old saw of "work expanding to fill the available space" will hold. The PC expansion board kit just arrived, so the program work will take a break while I warm up the soldering iron.

One area that still troubles me has nothing to do with the Slicer system, although working with it brought the problem into focus for me. This is the present "bandwagon" tendency shown by the numerous "compatibles" being pushed to the exclusion of possibly better systems. While standard systems make it easier for the software developer, I feel the neglect of more advanced systems can hurt the industry in the long run.

Both MS-DOS and CP/M have advantages and disadvantages with respect

	Prices A&T	Full kit	Easy kit	Bare Board
Single Board Computer	$995	$795	$450	$150
Expansion Board	$750	$575	$95	
PC Expansion Board	$600	$550	$400	$200
Math Coprocessor Board	$300			
Hard Disk Controller	$200			
Enclosure with Pwr Spply	$125			
Enclosure with P.S. 2-floppies	$695			
PC compatible keyboard	$150			
MS-DOS	$175			
CP/M-86	$85			

Bare board includes essential parts.
Easy kit includes hard to find parts.
Full kit includes all parts.
A&T is completly assembled and tested.

Slicer Computers Inc.
2543 Marshall Street N.E.
Minneapolis MN 55418
(612)788-9481

Slicer

Continued from page 57

to each other, and at times it is hard to decide which way to go. Perhaps Concurrent DOS will at least partially help this problem.

SUMMING UP

In summary, I feel the Slicer is a good example of an integrated system. Although nominally a "single board" approach, the bus structure maintains expandability, although not quite as flexible as the mother board approach used in the S-100 systems.

It is certainly not a system for the novice or one looking for a "plug in and run" machine, but it can offer much to the technically oriented experimenter. If you want to try the slicer call the Slicer Bulletin Board System at (612)788-5909.

Our latest software package

ModemMail

. . . the next generation in CP/M communications

*Now you can set up your own bulletin board system, transfer files, and operate a store and forward mail network that spans the globe—all with one CP/M program . . . . . **ModemMail.***

FEATURES

☐ The ModemMail Bulletin Board System ☐ The ModemMail Store and Forward Electronic Mail Network ☐ Automatic File Transfer . . . including XMODEM ☐ Data Encryption ☐ Full Programmable, Unattended Operation ☐ Terminal Communications ☐ Switchable Modem Ports ☐ and a Comprehensive User's Manual, featuring a Step-By-Step tutorial for new users . 249.00

ORDERS & INQUIRIES CALL 415 / 658-2881 ☐ MASTERCARD AND VISA ACCEPTED

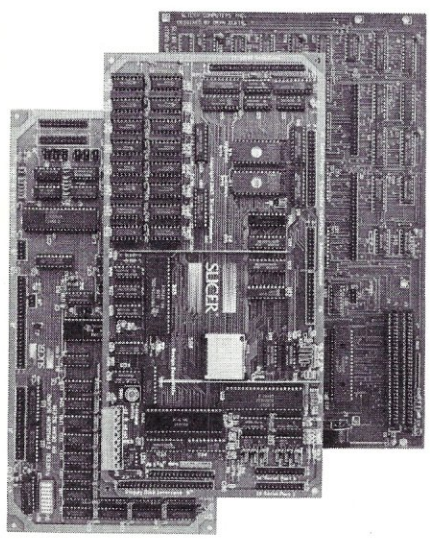
VISA AutoSoft MasterCard

INCORPORATED

VISION...INTEGRITY...SERVICE

166 SANTA CLARA AVENUE OAKLAND, CA 94610

SLICER—THE SYSTEM THAT GROWS TO FIT YOUR NEEDS



THE SLICER
Real 16 Bit Power on a Single Board—Featuring the Intel 80186

- Complete 8 MHz 16-bit microprocessor on a 6" × 12" board
- 256K RAM, plus up to 64K EPROM
- SASI port for hard disk controller
- Two full function RS232C serial ports with individually programmed transmission rates—50 to 38.4K baud
- Software compatibility with the 8086 and 8088.
- 8K of EPROM contains drivers for peripherals, commands for hardware checkout and software testing
- Software supports most types and sizes of disk drives
- Source for monitor included on disk
- Bios supports Xebec 1410 and Western Digital WD 1002 SHD controller for hard disks

Fully assembled and tested only $995
Also available in several kit forms

THE SLICER SYSTEM EXPANSION BOARD
For expanded memory, additional ports, and real time clock

- Up to 256K additional dynamic RAM
- 2 RS232C asychronous ports with baud rates to 38.4K for serial communication
- 2 additional serial ports for asynchronous RS232C or synchronous communication (Zilog 8530 SCC)
- Real Time Clock with battery backup for continuous timekeeping
- Centronics type parallel printer port

Fully assembled and tested only $750
Available in several kit forms also

THE SLICER PC EXPANSION BOARD
Gives your Slicer high performance video capability

- IBM compatible monochrome video
- Video memory provides 8 pages of text or special graphics capability
- 2 IBM type card slots for color video, I/O expansion, etc.
- IBM type keyboard port

Fully assembled and tested only $600
Available in several kit forms also

Also available: The μSLICER 188 $700; 8087 Math Co-Processor Bd. (call); 10 MB Hard Disk $700; W.D. 1002-SHD H.D.C. Bd. $200; Enclosures, Power Supply, and Support Hardware.

Operating systems are CP/M 86 by Digital Research, Inc. ($85), and MS DOS by Microsoft Corporation ($175).

MasterCard, Visa, Check, Money Order, or C.O.D. Allow four weeks for delivery. Prices subject to change without notice.

The SLICER Bulletin Board at 300/1200 Baud 612/788-5909

SLICER

Slicer Computers, Inc.
2543 Marshall St. N.E., Minneapolis, MN 55418
612/788-9481 • Telex 501357 SLICER UD

A C/nix Update

by R.A. Langevin

The May '84 issue of Micro systems contained my review of Version 1.0 of C/nix, an outstanding set of programs that add many UNIX and MSDOS 2.1-like features to CP/M 2.2. Since the review was published, C/nix has continued to evolve and Version 1.56, now in distribution, introduces a sufficient number of new features to warrant an update to the review.

If You Came In Late

Since some readers of Micro/Systems Journal may not have read or have access to the earlier review, the features that C/nix adds to CP/M include:

Hierarchical Directories
Input/Output Redirection and Pipes
UNIX-like Command Files
Online Documentation
UNIX-like Commands and Utilities
Limited Search Path Capabilities

A quick overview of C/nix commands is provided by Table 1, which is extracted from the Version 1.56 documentation.

What's New

The numerous enhancements incorporated in Version 1.56 are:

- Root directory on each drive can now contain 31 subdirectories (versus 26 in Version 1.0).

- Version 1.0 input/output redirection operated only to and from files or the console. Now redirection works with files and following devices:

lst:	listing device
con:	console
pun:	punch device
err:	"error" device, i.e. direct console output, bypassing BDOS
nul:	bit bucket (Output discarded)

The CP/M STAT utility can be used to associate physical devices with the foregoing logical devices as desired.

- In Version 1.0, simple redirection to a file resulted in the creation of the target file. As a consequence, the target could not already exist. Now the form, >!, will overwrite the target file if it exists.

- The cp and mv commands will now copy over an existing file of the same name. Inclusion of the -f switch will force the copy or move without confirmation.

Adding the verbose switch, -v, will cause the names of the files copied or moved to be echoed to the console, a useful feature when cp or mv are used with wildcards.

- The directory command, ls, now has three new switches:

ls -f results in listing of files, but not subdirectories, in current directory.
ls -d results in listing of subdirectories only.
ls -t results in listing of total disk space used in current directory.

- A new command, walk [-b] command, has been added. Walk will display the pathname of the current directory and the pathnames of all of its subdirectories, followed by the pathnames of their subdirectories, etc. If walk includes 'command', then 'command' will be executed in each directory.

Thus, for example, 'walk rm *.bak' will erase ALL files with the extension, bak, in the current directory, in all of its subdirectories, in their subdirectories, etc., asking for confirmation before they are erased. As another example, how about getting a nicely organized listing of all the files on your hard disk? This is easily done with 'walk ls -l'. This command will print the contents of the current directory, all of its subdirectories, their subdirectories, etc. If the -b switch is used following 'walk', it will cause walk to do a bottom-up rather than a top-down search.

Clearly, 'walk' is a VERY powerful command!

- Finally, the distribution disk now contains a new utility, set.com. Used with any or all of several switches, set has the following results:

set +p enbles paging of help files on the console. The next page of a help file can be displayed by striking any key.

set +v turns on the verbose mode for cp, mv, and rm as well as for command files.

set +b will cause the console bell to sound at the end of each page of a help file.

set +c forces confirmation before proceeding with cp, mv, or rm.

The use of - rather than + with a switch will turn the corresponding function off rather than on.

Worthy of Note

Although not new with Version 1.56, it is worth noting that the help facility works with any files that have the extension 'hlp'. As a result, you can use the help facility to access text of your own choosing simply by putting it in an ASCII file with that extension. You can then display your file by typing 'help filename'. For example, if you have limited disk space, instead of using the supplied help files (which are about 100K in size) you may want to create a new help file similar to Table 1 which contains just a command summary. Note that help files must reside on the disk from which C/nix is loaded. They can be invoked, however, from any directory on any disk. Finally, I must mention that the distribution disk also contains grep.pre. Although a slightly different syntax is used, grep is a full implementation of the marvelously useful UNIX generalized regular expression parser. Once you have used it, you will wonder how you managed without it. It is almost worth the price of the C/nix package by itself.

File Sizes

The addition of the enhancements described above has resulted in a modest increase in the size of the C/nix files as shown below:

FILE	Old Size	New Size
CNIX.COM	25.4K	28.5K
CNIXHIGH.SYS	1.5K	1.9K
CNIXUTIL.PRE	10.8K	11.0K
SET.COM	N/A	2.9K
GREP.PRE	10.4k	10.4k

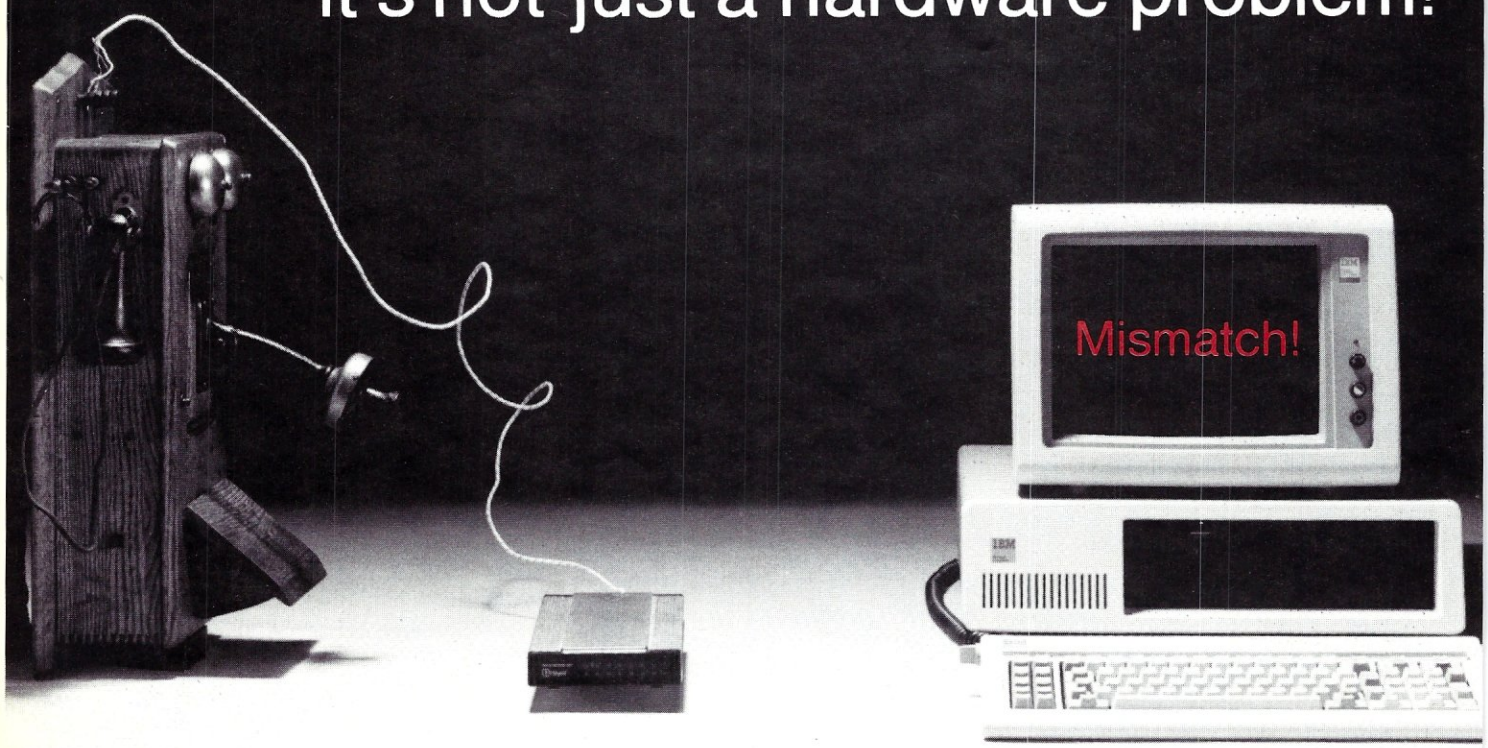
CNIXHIGH.SYS is the only one of these files that must be RAM resident, so Version 1.56 requires you to give up an additional 0.4K of TPA space, a modest sacrifice for the additional capability provided.

The Manual

The C/nix manual has been upgraded in several respects. It now contains substantially more explanatory material and has expanded to 58 pages from 41 in the earlier manual. In addi-

(continued on page 61)

Obsolete technology — it's not just a hardware problem!



You wouldn't dream of wiring your state-of-the-art modem to turn-of-the-century technology. Why strangle your computer with an antique communications program?

NightOwl Software believes your modem should open a window on the world — but without slamming a door on the power of your IBM-PC or compatible. That's why we designed our MEX-PC communications package to allow you complete access to your operating system, utilities and software while connected to a remote system.

Other communications programs limit your on-line options to a small set of built-in commands. Not MEX-PC. Our SHELL feature lets you run your spreadsheet, word processor, database management system, or any other program, from within MEX-PC while you're on-line — and without loss of text or data.

That's a claim no other communications software can make — and it's just one of many reasons MEX-PC is setting new standards for power, flexibility and performance in the world of telecomputing.

Consider the features:

- **A powerful command processor** allows fully automated dialing, log-ons, uploads, downloads and log-offs.
- **A built-in HELP program** explains all aspects of the software, on-line or off. A complete status screen immediately lists all current settings.
- **Command driven.** No need to wade through level after level of menus in a time-consuming search for the commands you want to enter or the features you need to change.
- **Fully documented.** Includes a typeset, ring-bound, 180-page user's manual and complete tutorial, fully indexed.
- **Wide range of protocols,** including Christensen XMODEM with both CRC and Checksum error correction.

$59.95*

Why spend more for a lot less power?

Write or call to request our free brochure.
Credit card orders welcome.

VISA master charge

Give us a toll-free call at 1-800-NITEOWL (In Wisconsin, call 1-414-563-4013)

NightOwl software, inc.

* Plus $5 shipping. Wisconsin residents add 5 percent sales tax.

NightOwl Software, Route 1, Box 7, Fort Atkinson, WI 53538

(continued from page 59)

tion, it now includes a useful discussion of C/nix troubleshooting techniques and hints. Finally, a very detailed index is a welcome addition.

The Hope Chest

As much as I like C/nix, there are three additional capabilities that I really want. First, is the ability to easily establish a user-defined search path. Second is a reasonable implementation of the UNIX link command, ln, to avoid unnecessary duplication of files. Third, and most important of all, I'd like to see the addition of a miniature command language to command files. At least, let's have 'if' and 'endif' and some kind of iteration. Perhaps we will see these in a subsequent version. I hope so.

In Summary

C/nix is better than ever! Collectively, the foregoing functions constitute significant enhancements to C/nix and give it even more of a UNIX flavor. The enhancements eliminate most of the shortcomings of Version 1.0 so that C/nix now provides a powerful set of utilities for the CP/M user, although it will still be of most interest to those who have lots of disk space. If you have a hard disk, I think C/nix is an essential addition to CP/M. I strongly recommend it.

C/nix Distribution

The distribution of C/nix is now handled by The Software Toolworks, a well respected software publisher. Their address is:

The Software Toolworks
15233 Ventura Boulevard,
Suite 1118
Sherman Oaks, California, USA
Phone (818) 986 4885

Price

The price of C/nix has been reduced to $59.95 (US dollars). Shipping costs $2.00 for 5 1/4 inch disks and $3.00 for 8 inch. Outside the United States, shipping costs $5.00 (US Dollars).

TABLE 1

```
commands (cmnd)          Summary of C/NIX Commands          commands (cmnd)

NAME
        commands — Summary of C/NIX Commands
SYNOPSIS

bye                                 — Leave the C/NIX shell (exit)
cat file1 file2 ... > outfile       — Concatenate text files
cat < ( ... )                       — Output text in parens
chdir                               — Change to top-level directory
chdir dir                           — Change to a new directory
chmod [+w|-w|+s|-s] file1 ...       — Change "mode" of files
cd                                  — Synonym for chdir
cp filefrom fileto                  — Copy a file
cp [-f|-c|-v|-q] file1 ... dir      — Copy files to new directory
csh [-v|-q] cmndfile param1 ...     — Invoke sub-shell on command file
dir [-lfdt] pattern1 pat2 ...       — List directories (ls)
dir [-lfdt]                         — List current directory (ls)
echo param1 param2 ...              — Echo parameters to console (csh)
era [-f] file1 file2 ...            — Erase files (rm)
exit                                — Exit the C/NIX shell
grep "pattern" file1 ...            — Search files for a pattern
grep "pattern"                      — Search console input for a pattern
help topic1 topic2 ...              — Display help information
help                                — Display list of help topics
ls [-lfdt] pattern1 pat2 ...        — List directories
ls [-lfdt]                          — List current directory
man topic1 topic2 ...               — Display pages from manual (help)
man                                 — Display list of manual pages (help)
mkdir dir1 dir2 ...                 — Make directories
mkrel file100 file200 file.pre      — Make page-relocatable program
mv [-f|-c] oldname newname          — Move/rename a file
mv [-f|-c|-v|-q] file1 ... dir      — Move files to new directory
pwd                                 — Print pathname of working directory (chdir)
ren oldname newname                 — Rename a file (mv)
ren newname=oldname                 — Rename a file (mv)
rm [-f] file1 file2 ...             — Remove files
rmdir dir1 dir2 ...                 — Remove directories (mkdir)
set [+|- [vcbp?]] ...               — Set certain user interface parameters
submit cmndfile param1 ...          — Submit command file (csh)
type file1 file2 ...                — Type text files (cat)
walk [-b]                           — Walk directory tree
walk [-b] command param1 ...        — Walk and execute command

NOTES
Optional flags are given in brackets, with alternatives separated with vertical
bars.  Ellipses (...) are used to represent a list of files, etc.

All of the above are recognized within the shell, except for grep and mkrel.
All but these two and chmod, mkdir, and rmdir, are also implemented entirely
within the shell.  The commands chmod, mkdir, and rmdir are implemented by
cnixutil.pre.
```

S–100
BARE BOARDS
8086/8087 CPU - plus 2764 or 27128, 8253, 8259
8088 Auxiliary Processor I/O mapped, 4K EPROM, 4K RAM, prototype area
$45.00
Each
Call or write for brochure.
Terms: Check or money order only. CA residents add sales tax. Prices include UPS shipping.
Applied Innovations
3000 Scott Blvd. Suite 106
Santa Clara, CA 95054
(408) 748-1875

PC–PRO
is
MS–DOS
for
CompuPro
for
information
write
Computer House, Inc.
P.O Box 709 Woodacre, CA 94973
(415)892-3591

FORTRAN
PROGRAMMERS
Discover why
you should be using
F77L
the complete implementation
of the ANSI FORTRAN 77
Standard for the IBM PC and
compatibles.
If you are serious about your
FORTRAN programming, you
should be using F77L.
$477
Lahey Computer Systems, Inc.
31244 Palos Verdes Drive West, Suite 243
Rancho Palos Verdes, California 90274
(213) 541-1200
Serving the FORTRAN community since 1969

Program Interfacing To MS-DOS

by William G. Wong

The second part in this series of articles explores some specific character device input and output functions. Character devices include the keyboard, display console, auxiliary devices (usually a serial port), and the printer. The character functions can deal with a single character or a string of characters.

The character devices can also be accessed as files using file names such as CON:. Use of these filenames and file input and output functions will be described in a later article which will also cover disk-file support, since the functions are the same.

The functions described in the rest of this section are essentially identical to the analogous CP/M functions . However, there are a number of nice improvements. The following table provides a quick overview of the functions available.

Character Input and Output Support Functions

Function	Parameters/results
1 Keyboard input	AL char
2 Console output	DL char
3 Auxiliary input	AL char
4 Auxiliary output	DL char
5 Printer output	DL char
6 Direct console I/O	DL char or FF hex AL char/status
7 Direct console input w/o echo	AL char
8 Console input w/o echo	AL char
9 Print string on console string ends with '$'	DS:DX string address
A Buffered keyboard input	DS:DX buffer address size, entered, string
B Check console status	AL (0 not ready, FF ready)
C Clear input buffer and do function	AL function code (1, 6, 7, 8, A)

The function number is passed in the AH register. Output functions use the DL or DS:DX registers for parameters and input functions return the results in the AL register.

Part II — Dealing with character input and output functions.

OUTPUT FUNCTIONS

The output functions have two forms for the console and one for the printer and auxiliary port. The following example shows the console output functions.

```
        MOV AH,2  ;AH :=console output
        MOV DL,'A';DL :=display charac
        INT 21H   ;call DOS & output char

        MOV AH,6  ;AH :=direct console I/O
        MOV DL,'A';DL :=char to display (not FF)
        INT 21H   ;call DOS and out charac

        MOV AH,9  ;AH :=console string out
        MOV DX,STR;DX :=string offset
        INT 21H   ;call DOS & out string
                  ;to display
STR: DB 'This is a sample text.$'
```

There are some restrictions on use of these functions. The first is that normal console output (function 2) recognizes the control-S and the control-BREAK (or control-C on some systems) input from the keyboard. The control-S causes the program to wait for a subsequent control-Q from the keyboard. This is useful because it allows a user to stop and restart the display of text, so that the information can be viewed at their leisure. The control-BREAK causes this function to be executed, while the default function aborts the program. Note, the control-BREAK function can be redefined as any function to be supplied by the program as described in my the previous article.

The control-S and control-BREAK options will not be checked if the direct console output function (function 6) is used. Instead, it is then up to the program to poll the console to see if any of the various control key options have been entered. This is the normal method of retaining program control during output without replacing the control-BREAK function. There is one restriction in using this method of output. That is the character FF hex cannot be sent to the console because this value indicates that a console input function is to be performed. This restriction is usually not a problem.

The print-string function is very useful since it allows a number of characters to be displayed at one time, without having to resort to single-character output functions. The function is actually a remnant of CP/M because the string terminator is the dollar sign ('$', 24 decimal). The unfortunate result is that you cannot print a string which contains a dollar sign. Also, this function has the same restrictions as console output function 2.

The following is an alternative to function 9 in that the string terminator is a FF hex and it uses function 6 to output each character. Note, FF hex was chosen as the string terminator because it cannot be output using function 6. It also allows a NUL character (0 hex) to be in the output string. This is useful because many function key-programming options on terminals often use the NUL character as a function-key string terminator.

```
; ----    DS:BX  = string index  ----

STROUT:MOV DL,[BX];DL :=string character
       CMP OFFH   ;zero :=end of string
       JZ  STREND ;skip if end of string
       PUSH BX    ;save string index
        MOV AH,6  ;AH :=output char func.
        INT 21H   ;output character in DL
       POP BX     ;BX :=string index
       INC BX     ; increment string index
       JMP STROUT ; loop until string end
```

A similar function can be created for auxiliary and printer output since a function-6-type operation does not exist for these devices.

The auxiliary and printer output functions (functions 4 and 5, respectively) perform just like function 2, except that keyboard input is not

examined. One useful function to create, is a printer-output function which checks the console after each character is printed, to see if the user wants to stop or pause a printout.

INPUT FUNCTIONS

Character input is available from the keyboard and the auxiliary device, but not from the printer. There are a number of keyboard-input functions, but only one auxiliary-input function. This really limits the latter.

The auxiliary-input function gets a single character from the serial port. It waits until a character is available and there is no associated-status function extant. For this reason, the function is not normally used for communication programs which access the serial port. The following is an example of the auxiliary input function:

```
MOV AH,3 ;AH :=aux input func.
INT 21H  ;AL :=aux input char.
```

Luckily, the keyboard-input functions are more varied and flexible. Keyboard-status functions are available, in addition to keyboard-input functions. The normal single character-input functions is number 1. When selected, the system waits until a single character is available from the keyboard, before returning to the calling program. The following is an example of this basic-keyboard input function.

```
MOV AH,1 ;AH :=keybd input func.
INT 21H  ;AL :=keybd character
```

The character entered is also echoed on the console. Entry of a control-BREAK key sequence sauses the control-BREAK function to be invoked. On some systems, such as the IBM-PC, a function key often generates characters at once. The first is a 0-hex, followed by a second character requiring two input-function invocations to produce two characters.

Unfortunately, this basic keyboard-input function is not sufficient for most sophisticated programs, because control-BREAK will normally be processed by the program. Also characters entered may not always be echoed on the console. Some of the following functions provide this support.

Function 8 provides the same type operation as function 1, except that the character is not echoed. It still recognizes the control-BREAK key. Function 7 is the same as function 8, except that it does not recognize the control-BREAK at the DOS level. Instead, the character is returned as a result of the function invocation in the AL register, thereby allowing the program to process the key sequence.

In the last three cases, DOS waits for a key to be pressed before returning to the program. A program can check the keyboard status before using any of these functions. This would allow the program to continue to perform useful work, such as copying a file while polling the keyboard. The following example shows how the keyboard status is obtained.

```
MOV AH,0BH ;AH :=keybd status func.
INT 21H    ;AL :=(0-no char. available,
           ;    OFF if no character)
```

The only drawback to using this function is that the control-BREAK will cause the control-BREAK function to be invoked.

The control-BREAK check can be bypassed by using the direct-console I/O function (function 6) with OFF-hex as a parameter in the DL register. The function always immediately returns the value of the key entered or a zero indicating that no character is available. The control-BREAK is returned as with any other key.

The last-single character input function is the clear-input buffer and read-next-character function. This is function 0C-hex requiring a sub-function code in the AL registers. This must be a 1, 6, 7, 8 or 0A hex, which are keyboard input or status functions. The input buffer must be cleared of any characters before the sub-function is invoked. This alleviates the need for an internally-coded function in a program which would read everything in the input buffer before invoking the subfunction.

Finally, there is the buffered keyboard input function. This function requires a buffer address as its parameter. It allows the user to enter a line of text, which is terminated by the return key. Each character will be echoed on the screen and basic edit capabilities can also be used.

The buffer contains two prefix bytes located before the actual buffer address. The first contains the size of the buffer in bytes and the second the number of characters entered, not including the return key. Filling up the buffer does not cause the function to complete as on some CP/M systems. The following is an example using the buffered input.

THE WORLD'S FASTEST MOST POWERFUL 8080 RELOCATING MACRO ASSEMBLER

SLRMAC™

ONLY $49.95

AT THIS PRICE, SHOULD YOU BE WASTING YOUR TIME USING SOMETHING ELSE?

This is what they said about Z80ASM, our Z80 assembler. Now the same features and performance are available in our Intel Mnemonic product. SLRMAC is compatible with M80 in .8080 mode, with many extensions. Too many features to list here.

To order or to find out more about our complete family of development tools, call or write:

SLR Systems

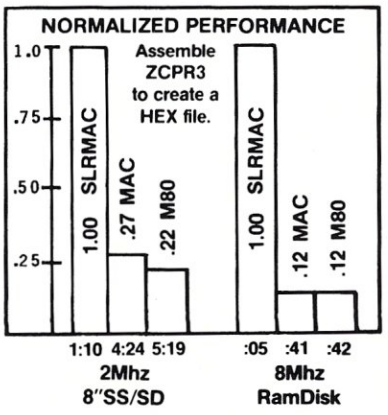
1622 N. Main St., Butler, PA 16001
(800) 833-3061, (412) 282-0864
Telex 559215 SLR SYS

MasterCard VISA C.O.D., Check or Money Order Accepted.

Add $3 shipping.
Z80 CP/M compatibility required.

". . a breath of fresh air . ."
Computer Language, Feb. 85

". . in two words, I'd say speed & flexibility",
Edward Joyce, Nov. 84 Microcomputing



```
    MOV DX,OFFSET BUF;DS:DX ::buffer addr.
    MOV AH,0AH        ;AH ::buffered read func.
    INT 21H           ;read data into buffer

BUF:DB  20            ;buffer 20 char. long
    DB   0            ;filled in by DOS
    DB  20            ;buffer area
```

Buffered input is an easy way to provide a standard input method to the user. The alternative is to write another procedure which does the same thing, plus giving the programmer more control over the characters entered and the function keys.

SUMMARY

MS-DOS provides a good set of character I/O functions for the keyboard and console, but is lacking, when with respect to the printer and auxiliary ports. Adding input and outputstatus calls would have made the system more flexible.

The next installment in this series will cover diskfile support, including namedfile support for the console and printer.

Bill Wong is the President of Logic Fusion, Inc., a systems software development firm located in Yardley, PA.

ENGINEERING SOFTWARE
CPM-80 • MSDOS • TRSDOS • PCDOS*

Free Catalog and Signal Processing Booklet

Professional — Affordable

PLOTPRO — Scientific Graph Printing Program	**$49.95**
ACNAP2 — AC Circuit Analysis Program	**$69.95**
DCNAP — DC Network Analysis Program	**$59.95**
SPP — Transient Signal Processing	**$59.95**
PCPLOT — Pixel Resolution Graphics Program	**$59.95**

*96 Computers and formats, 8087 Co-processor versions of ACNAP2, DCNAP, SPP available for $10.00 more.

BV Engineering
Professional Software
2200 Business Way, Suite 207 • Riverside, CA 92501
(714) 781-0252

PARDON OUR BLUNDERS

We regret the following errors in Micro/Systems Journal and wish to thank our readers for pointing them out to us:

Vol 1/No 2, Page 21: Due to a printer error one line of code in the Turbo Pascal program was obliterated. It should read:

```
if lonum = spc[indx2] then
```

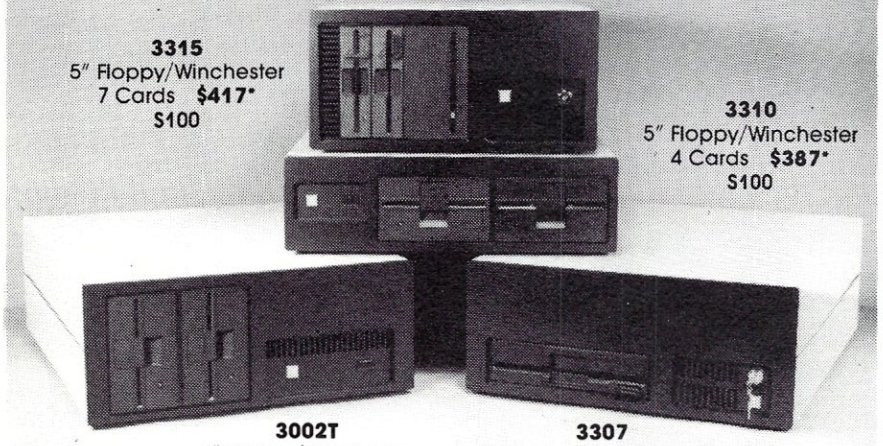
And, in the article titled "Interfacing To MS-DOS", pages 33-35, all references to moving a function number into the CL register before calling INT 21H are incorrect. One must use the A register.

Vol 1/No 1, page 49, column 1, line 18: The author neglected to indicate that address "0800" is a hex value. Thus the line should read:

```
LD BC,0800H
```

And, the code on pages 54 and 55 were shown in reversed order. The code on page 54 should have appeared on page 55 and vice versa.

Chassis Classics



3315
5" Floppy/Winchester
7 Cards **$417***
S100

3310
5" Floppy/Winchester
4 Cards **$387***
S100

3002T
5" Floppy/Winchester
10 Cards **$565***
S100

3307
8" Floppy/5" Winchester
7 Cards **$494***
S100

laser 3000 **MAIN/FRAMES & DISC ENCLOSURES** FROM **$100**

LASER 3000 DISC/COVERS (not shown)

3916F
5" Floppy **$100***

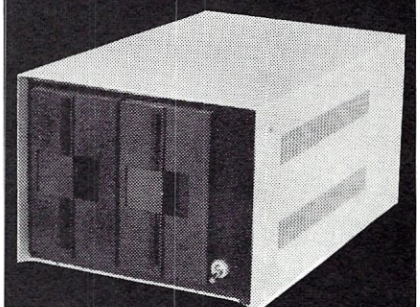
3915
2 ea. 5" Winchester **$199***

(Disk drives not included)

*1 piece; prices lower in quantity.

INTEGRAND
RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291 209/651-1203

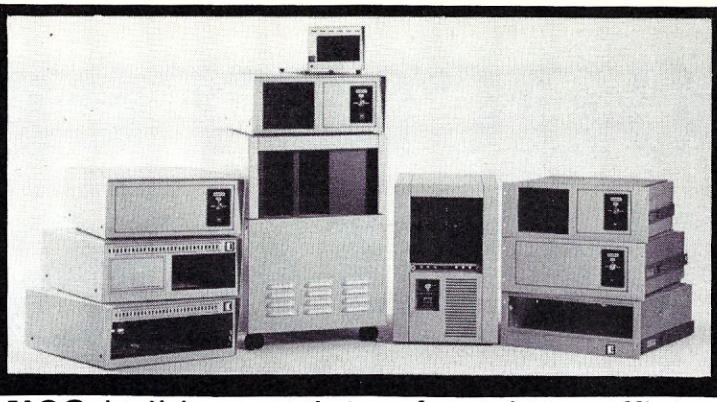
"Little Board"
MAIN/FRAMES
6 Models from **$125***



$150 (1 piece*)
MODEL 2800
Includes power supply & fan
(Disk Drives and Little Board not included)
AMPRO & Little Board are TM AMPRO computers

INTEGRAND
RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291
209/651-1203

S-100 BUS Mainframes with Patented cooling, and constant voltage power supplies.



Disk Drive Enclosures for Floppy disk, Hard disk, Tape Backup or Combinations.

PARA DYNAMICS builds a variety of modern, efficient, trouble-free expandable housing systems for most S-100 BUS configurations. Whether a rack mount, desk top, or stand alone, our patented super-efficient heat dissipation system can end your board-level failures due to high temperatures. Please call today for full details. **(602) 991-1600**

PARA DYNAMICS CORPORATION • 7895 EAST ACOMA • SCOTTSDALE, AZ 85260

Coming in future issues of Micro/Systems Journal

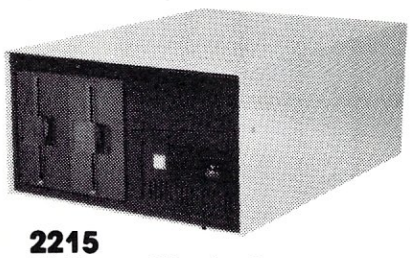
- Bringing up CP/M-68K
- Transfer Files Between CP/M & MS-DOS
- Modify CompuPro S-100 Boards for 16-Bits
- Overlays For Basic
- Interrupt Borrowing With Turbo Pascal

and lots more...

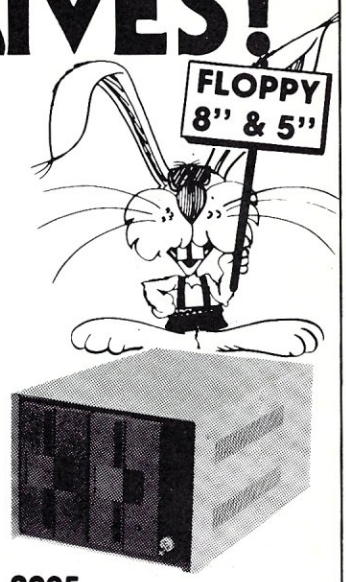
SATISFY YOUR DRIVES!

70 MAIN/FRAMES & DISK ENCLOSURES FROM $100

800D2F
5" Floppy Main/Frame
(10 cards) **$392**



2215
5" Floppy Winchester
Main/Frame (7 cards)
$380



2905
5" Disk Enclosure **$100**

Write or call for our brochure which includes our application note: "Making micros, better than any ol' box computer"

NEGRAND
8620 Roosevelt Ave./Visalia, CA 93291
209/651-1203
We accept BankAmericard/Visa and MasterCard

Concurrent PC-DOS

by Michael Guttman and Vincent Mills

Editor's Note: Concurrent DOS, is available for some 8086/186/286-based S-100, and other systems, supporting up to 16 users. It is essentially the same as Concurrent PC-DOS with the exception that it lacks MS-DOS compatibility.

A review of DRI's new 16-bit multi-processing multi-user operating system.

Concurrent PC-DOS (C/PC-DOS) is the latest in a series of multitasking, multiuser operating systems from Digital Research Incorporated (DRI) of Pacific Grove, California. Designed to run on a IBM-PC or compatible (either alone or with up to two terminals), C/PC-DOS features support for applications developed under either PC-DOS and C/PM-86.

Under C/PC-DOS, up to four applications can run simultaneously on a stand-alone PC, each appearing in its own 'window' on the screen. These windows can sized, placed and 'tiled' over one another as the user desires, allowing a full or partial view of every background and foreground activity.

As a windowing environment, C/PC-DOS is entering what promises to be a crowded field, a field that may soon be dominated by IBM's Topview and various Topview clones already nearing release. However, C/PC-DOS does offer an opportunity for those most familiar with CP/M and its variants to migrate to the IBM-PC environment with a minimum of hassle, allowing them to mix their applications with the more ubiquitous PC-DOS productivity software such as Lotus 1-2-3, Symphony, Framework, etc.

In addition, C/PC-DOS is the only windowing environment currently offering multiuser capabilities. Up to three users can work simultaneously, with one user multi-tasking from the IBM-PC screen and keyboard, while two other users are at work at standard terminals connected to serial RS-232 ports at up to 9600 baud.

For a mere $295, C/PC-DOS offers all this as well as a smorgasbord of system utilities and bundled software. It definitely seemed worth a gander.

GETTING STARTED

C/PC-DOS arrived in a brown binder box containing a thick manual and five distribution diskettes. According to the instructions, C/PC-DOS requires a minimum of 256K and two floppy disks. However, this would be a pretty sparse configuration, and DRI recommends at least 512K and a hard disk drive. We certainly can't imagine trying to run four applications concurrently from a floppy-only system.

As far as the hard disk is concerned, C/PC-DOS can be installed on either the root directory of DOS or on a separate CP/M-86 partition of its own. If a hard disk already has data and is partitioned to PC-DOS only, the DOS option is simplest. We tried both and rather liked the CP/M-86 option; it created two partitions, one in DOS format, the other in C/PM format.

In our case, these functioned respectively as drives C: and D:. Interestingly, media of either type would copy to one another, from or to hard disk or floppy, and programs from either operating system would run on either partition. We did notice that program loading seemed faster from the CP/M partition.

DRI has made installing C/PC-DOS a breeze. A batch file invoked upon boot-up automatically copies all the distribution to the hard drive, prompting the user appropriately at every step. The installation also creates an appropriate 'autoexec.bat' file for boot-up loading of C/PC-DOS.

GETTING FAMILIAR

Once loaded, C/PC-DOS doesn't appear very different from PC-DOS. A familiar C> prompt appears, and most PC-DOS commands work as expected. At this point, the only obvious difference is the appearance of a C/PC-DOS status line on the 25th video row. Multitasking is already in progress, however, and some performance degradation versus PC-DOS will already be apparent (we figured about 10-15%).

C/PC-DOS really starts cooking when the user enters the control key sequences that activate windowing. Using a CTRL-1, CTRL-2, CTRL-3, and CTRL-4, the user can round-robin among the four available tasks, starting and stopping programs in each at will. Of course, as tasks mount up, degradation rises, particularly if these tasks are simultaneously updating the screen or writing to disk.

Invoking the WMENU window manager from any window adds some spice to C/PC-DOS. After loading WMENU, typing a CTRL-+ activates this memory-resident program, allowing the user to easily change the window positions, sizes, colors, etc. even while other programs are executing. WMENU generates another window with easy-to-understand prompts on the 25th video line, temporarily replacing the C/PC-DOS status line. It is possible to create some very artistic collections of windows, manipulating the screen colors in each window to heighten the effect. When the user is done manipulating his windows, WMENU politely disappears into the background and the main status line reappears.

However, there are some severe limits to what can be effectively placed on the screen. C/PC-DOS itself takes up some 150K, and each active process will require at least 64K and usually 128K or more. Before you know it, even a fully loaded 640K system will

soon be out of memory. Combined with the speed degradation experienced with several tasks running simultaneously, these memory restrictions make trying to exercise C/PC-DOS to its theoretical potential a rather frustrating experience.

In addition, not all programs are well-behaved enough to co-exist in the windowing environment. Our version of Wordstar 3.3, for example, wrote all over the screen without any respect for window boundaries. NewWord, the Wordstar clone, worked better, keeping within its window boundary, but its cursor and text would disappear behind other windows, despite a 'tracking' feature that is supposed to keep the current cursor line in the display.

Other programs, such as Sidekick or Prokey, simply collide with C/PC-DOS, particularly if they are designed to remain memory resident. Finally, programs that use copy-protection schemes not compatible with C/PC-DOS will also not run properly.

We find it hard to look at these as problems with C/PC-DOS itself. The real problem lies with the fact that not all software is 'well-behaved' enough to exist in any off-the-shelf windowing environment. This is because many programs need to bypass standard DOS calls to achieve acceptable performance, particularly in the area of video I/O (see sidebar). In addition, it seems painfully clear that the IBM-PC's hardware is just not powerful enough to effectively support the kind of multitasking theoretically possible under environments like C/PC-DOS.

C/PC-DOS does have some compatibility problems with PC-DOS for which there seems to be no good excuse. Some of the problems have to do with the fact that C/PC-DOS was originally designed to match DOS 1.1, leaving out the 2.x level system calls that many programs and particularly many hardware drivers have come to depend on. A new version, C/PC-DOS 4.1, due shortly, is supposed to solve these system call problems. (*Editor's Note: we hope, shortly, to publish a review of C/PC-DOS 4.1 running on an S-100 system.*)

DRI did make C/PC-DOS work with 2.x directory structures, supporting hard disk, subdirectories and the 360K floppy format. However, DRI fudged in other areas, such as leaving out the very useful PATH command. The function of PATH has, however, been partially replaced by a SYSDISK command that sets a single global system directory and supplemented by a scheme of 'floating' logical disk drives N: and O: that can be reassigned to other drives and sub-directories.

DRI also left out the popular advanced batch file commands, such as IF, FOR, and GOTO, crippling us inveterate users of sophisticated batch files. Another annoying change is the removal of the editing functions normally assigned by DOS to function keys F1 through F5, particularly key F3, which is used to repeat the last DOS command given. DRI has replaced these with a function key assignment system that is user-definable but not compatible with PC-DOS.

GOING MULTIUSER

Perhaps the most interesting aspect of C/PC-DOS is its multiuser capabilities. For the price of two terminals and connecting cables, C/PC-DOS users can add two workstations to their PC. A few other products (Software Link's Multilink, for example) offer similar capabilities, but not with all the other functions offered by C/PC-DOS.

Activating the extra users first involves using the SETPORT command to set the appropriate serial ports to the hardware characteristics of the attached terminals. Then, using the SETUP program, the user enables the ports as separate consoles. The configuration thus created can be saved using the Save System Parameters function in the SETUP.

Getting the terminals operational

THE BACKREST UTILITY

C/PC-DOS supports so many utilities and programs, it is impossible to review all of them in depth. However, here is a detailed look at one very good example of a DRI utility, the BACKREST program.

BACKREST consists of the files BACK.CMD, REST.CMD and COMMAND.BR. BACK and REST make backing up and restoring hard disk partitions a pleasure. When backing up, these programs copy or restore only files changed or created since the last backup. The user is prompted to insert the diskettes in the correct order. Each disk used for backup is labelled with a unique volume identifier that makes restoration a breeze.

BACK keeps track of where things can be found in a special file called DIR.BR, which contains a record of all files by date and storage volume on the hard disk. This file is saved at each BACK session automatically.

True hackers will appreciate the customized command file, COMMAND.BR. Using this file, you can specify what defaults the utilities should use in execution, such as whether to reuse disks in your backup set and what colors you prefer for your color monitor. Unfortunately, only hackers will use it. Reading the 88K of directions for customizing the command file took one night by itself (and they are on the distribution disk, and not in the User's Guide where they belong.) Some of the commands, like ERASE and REUSE, we think are especially prone to be misunderstood and therefore be used destructively.

CP/M file password structures (for CP/M media only) are maintained automatically, provided the command file contains exception records for this purpose. However, the files (and the backed-up copy of COMMAND.BR) are NOT protected on the backup disks. This implies anyone with access to a backup set could conceivably breach the password feature.

More than one drive can be specified in the source set; CP/M and DOS partitions are handled separately and require respectively formatted media.

Each backup disk generated by BACKREST contains a volume number as a directory entry with a zero length. The files are stored literally as they were on the hard disk, making emergency restoring easy. This is quite a change from PC-DOS which added a header to each file, making them impossible to restore without RESTORE. Prior to DOS 3.00, RESTORE couldn't restore to floppies, which meant that if your hard disk should quit you wouldn't be able to restore your files to temporary floppies.

Unfortunately, any advantages of BACKREST over PC-DOS BACKUP and RESTORE are overridden by a very drastic reduction in throughput. Here are some representative test results running BackRest.
Saving 338 files.

	Time for whole backup	Number of disks used
With BackRest:*	2 hours, 9 minutes	16 disks
With PC-DOS:	36 minutes	14 disks

* Used the default VERIFY type of backup. Non-VERIFY is not recommended.

In our opinion, there is no excuse for the slow execution time of BACKREST. Most users will probably lose track of what they're doing before the backup is finished.

was a piece of cake. A null modem to an Apple IIc running terminal emulator worked just fine. If problems do occur, they can be alleviated by turning off some of the handshaking options using SETPORT.

The slave consoles are subject to some restrictions not applicable to the PC console. No multitasking or windowing is allowed, no C/PC-DOS status line appears, and IBM-PC function keys are not operational. Unfortunately, this effectively disables any software for which function key input is mandatory.

In addition, the restrictions on video are even more severe; only ANSI video display calls will work. Since the terminals are also restricted to a 9600 baud serial transfer rate, serious speed degradation occurs with programs doing extensive video updating.

Obviously, these terminals would only be useful running very vanilla-flavored programs (Multiplan, Supercalc, DbaseII). Furthermore, these programs would have to run within the restrictions of available memory, and would therefore further limit the multitasking activities of the main PC console.

These restrictions severely limit the real usefulness of th multiuser features of C/PC-DOS. Onl small tasks with limited console I/O would really be appropriate. Once again, the problem seems to be not so much with C/PC-DOS, but with the underpowered hardware of the IBM-PC.

Another weakness that affects C/PC-DOS multiuser operation is the absence of file password protection. Once again, this is more attributable to PC-DOS than C/PC-DOS, since the DOS file and directory structure does not provide space for password flags. Using C/PM-86 media, however, password protection is preserved.

THE OPERATING SYSTEM AND ITS UTILITIES

The C/PC-DOS operating system is invoked with LOADCCPM.COM, which loads CCPM.SYS, the resident module that handles user requests, task scheduling and system input and output. It can emulate much of the PC-DOS commands as well as certain C/PM-86 commands. Upon loading, CCPM.SYS scans the file CONFIG.SYS, looking for FIXED-DEVICE commands that designate installable device drivers. CCPM.SYS recognizes only CONFIG.SYS commands in the form FIXED-DEVICE = filename.ext.

As mentioned earlier, the operating system is supported by two configuration programs, SETPORT and SETUP. SETPORT configures the serial ports for console and communications uses. You can set the RS-232 parameters and handshaking to match any device from 110 to 9600 baud.

SETUP lets you change a variety of start-up operating system values. These include:

MDISK - creates a memory disk drive (M:) at the top end of memory. Since the operating system requires

DRI's C/PCDOS Programmer's Pak

These rich and wonderful volumes include the following:

RASM-86 Relocatable Assembler - a full-featured assembler, with support for the 8087 coprocessor built-in. Mnemonics are Intel standard with a few variations. This assembler seems much easier to use than IBM's own assembler.

LIB-86 Object Librarian - keeps track of object, map and cross-reference files in a library format accessible to the linker. In any large programming task, it is useful to have a way to organize object and other compiler-generated files in a single consolidated scheme. It also cuts down on the number of files the linker has to keep open when running. This tool is not included with IBM's assembler.

LINK-86 Object Linker - makes .CMD files for C/PC-DOS. It can also handle overlays, which require a little trickier programming but result in smaller memory requirements. Of course, memory consumption is a crucial point with multi-tasking systems. .CMD files are the combination of one or more .OBJ files. .CMD files can be no greater than 64K (just like .COM files.)

XREF-86 Cross Reference Generator

SID-86 Symbolic debugger - just pass it the names of the .CMD and .SYM (symbol table) files and away you go. Everything is here. Symbolic table(s) aid greatly in the debugging process. It's almost like watching the source code running. Could be improved greatly if issued as a screen-formatted system.

Sample Programs - DRI includes some useful sample programs such as:

Window source code - WMENU, written in C and assembler, is a great example of how to do windows. High-level routines in C call several small assembler routines that do all the muscle-work in altering the characteristics of the four windows. Since WMENU uses a keyboard interrupt (Ctrl-Grey-+) to become visible, it shows how to write memory-resident desk accessory software. The Digital Research C(TM) compiler is required to compile the programs with extension .C (not included.)

Queue communications examples - DRI provides two communications examples in assembler: PROTOCOL.A86 and SAMPLE.A86. Queue Driven Serial Communications refer to the buffering of communications data between an outside source and your computer. In a multitasking environment, communications data (say, from your modem) can pile up faster than your computer can handle. TERM.A86 shows how to use the operating system to do this buffering for you.

PROTOCOL.A86 - program to set interrupt protocol for serial ports

SAMPLE.A86 - program shows how to use the XIOS backdoor entry points that allow you to write directly to the virtual console buffer and still work in C/PC-DOS's windows. Although it is a bit complex, the fast speed makes it worthwhile if you're programming for windows.

TERM.A86 - dumb terminal program which uses the Queue Driven Serial Communications capabilities of C/PC-DOS.

SYSTAT - dynamically shows the status of system queues, memory and process status. You can select the rate of update, and run this in the background of any program.

GSX-86 User's Guide and Programmer's Guide - GSX is the Graphics Extension to C/PC-DOS, the precursor to GEM, their Graphics Environment Manager. It allows you to address a graphics input or output device in a nonspecific way, a way compatible with many other device drivers. Unfortunately, no device drivers are included with either C/PC-DOS or in the Programmers Pak. Without these device drivers, you can't use the GSX software.

Programmer's Reference Guide for Concurrent PC-DOS - goes in-depth into the inner workings of C/PC-DOS and lists the operating system calls available to the programmer. Though these calls are not compatible with PC-DOS itself, they provide access to the guts of the operating system and the communication queues. Programs that operate as separate, concurrent processes can pass data to each other using queues. Any process can create queues or spawn other processes.

at least 256K of memory, only memory in excess of 256K can be used as a memory disk. MDISK provides up to 256K of RAM disk. The user can change both the size and the starting address of MDISK.

Diskette head step time — You can improve your diskette performance sometimes by adjusting how fast the head moves across the diskette surface. The default value is fine for starters.

Verify after diskette write— Enabling write verification results in greater data security at the expense of processing time. After each write to a floppy diskette, a read checks to be sure the write worked. This feature was put in by IBM as "insurance", in case their original CDC floppy drives were to act erratically once the machine was in the field. Luckily, the drives turned out to work fine without verification. (Interestingly, the two DOS Interrupt 21H calls that change the verify flag in PC-DOS are not supported by C/PC-DOS!)

Maximum memory per process— You may set a limit on the amount of memory available to each .CMD program, normally the full available amount. This can become a problem if you start a very large CP/M program and want to change windows in-process.

Color monitor scroll mode—Use this menu selection to select the display update rate, from wide open (flickering) to slow and sure.

Updating CCPM.SYS and rebooting — After any changes have been made, SETUP updates the copy of CCPM.SYS on the system disk. Note that none of the changes made appear until the system is rebooted.

HDMAINT is a menu-driven command that performs hard disk maintenance tasks. Although the main purpose of HDMAINT is to format your hard disk before initial usage, you can also use it to change the partitioning on a blank disk or the volume name.

DSKMAINT is C/PC-DOS's version of DISKCOPY, FORMAT and DISKCOMP, combined into one menu-driven program. DSKMAINT can copy, format or verify either DOS or CP/M floppies.

BACKREST is a disk backup and restore utility written by Stok Software, Inc. and licensed for distribution by DR. It uses a customized control file to direct the backup process in an automatic and painless fashion (see sidebar).

WMENU and WINDOW - as discussed earlier, WMENU is a memory resident program that allows you to easily change the window formats from the keyboard while perserving the operation of all concurrent tasks. WINDOW is a batch version of WMENU that allows the user to set up window attributes in advance via a system batch file.

BATCH Commands - like PC-DOS, C/PC-DOS can invoke a series of commands from a batch file. Unlike PC-DOS, Concurrent has a BATCH.CMD file that controls batch processing when needed, line by line. Having an external batch processor (necessary because of the hybrid nature of C/PC-DOS) makes command files rather slow. They seem to virtually crawl across the screen, with 4 seconds or more between prompts.

CHDIR and MKDIR - familiar subdirectory commands cloned from PC-DOS. For some reason, these work with multi-level subdirectory commands such as CHDIR \ralph\ws\txt, while other commands such as DIR will support only one level of subdirectory referencing.

TIME and DATE - These CMD files allow you to change the on-screen time and date, two functions normally handled internally by COMMAND.COM. I've found that the ASTCLOCK command that comes with AST boards for the PC sets the time and date for the system satisfactorily.

TYPE - You can display the contents of a file on the screen, as well as on the printer. In C/PC-DOS, ^P toggles the printer. Halt and resume listing with ^S and ^Q.

COPY - COPY can copy files between consoles, printers and drives. One annoying limitation - you can't copy directly between subdirectories without using a "floating" drive.

RUNMENU - Menus are a powerful tool provided by C/PC-DOS. Using EDITMENU (see below), you can create custom menu screens or modify system menus. The RUNMENU command invokes a menu from a menu file. This is the mechanism used in the default STARTUP1.BAT file to pull up the main menu. Using this command, you can execute predefined or your own menus.

EDITMENU - the menu design tool. Using this program, you can design a full-screen menu using character graphics and text. You can specify character attributes (color, brightness and flashing) as well as limit the user's input. On a valid response, your menu can execute another menu, return back one menu level or execute a program or batch file.

COPYMENU - copies menu descriptions between menu library files.

DELQ and ERAQ - allow the user to delete files interactively. The DELQ and ERAQ commands query before erasing files to ensure the erasure is truly wanted. Using these commands

PERFORMANCE TESTING

We ran some performance tests to find out just how much degradation occurs when we take advantage of C/PC-DOS's concurrency. As a typical disk- and computation-intensive task, we compiled two separate programs in PC-DOS and one and two process C/PC-DOS jobs.

The programs were chosen for their generality. They contain no macros or INCLUDE calls. The PC-DOS command used to start the test was: MASM progname;<cr> For the one job test of C/PC-DOS, the same command as the PC-DOS test was entered from the window 1 with no other windows or consoles active - the best possible conditions.

For the two job test of C/PC-DOS, one compile was run from the console in the first window with no other windows active, and the second compile was submitted from an Apple //c running in terminal mode at 9600 baud.

RAM DISK TESTS

Since both PC-DOS and C/PC-DOS contain some form of RAM disk, each test was repeated for RAM disk. The RAM disk was loaded with the executible program MASM and the source code. Then, the default directory was switched to the RAM disk and the "MASM program;<cr>" was typed.

The results of these tests are tabulated below. The results seem to indicate that:

1) the overhead of concurrency slows down a single job in the order of 15-20%.

2) the addition of a second job did not seem to affect total throughput on the system very much; running two identical jobs concurrently took just about twice as long as one alone.

Figure 1 : Hard-disk test of Compile MASM instruction (min:sec):

Hard disk - PC-DOS		1 job C/PC-DOS	2 jobs C/PC-DOS
Program 1	1:01	1:10	2:30
Program 2	1:15	1:24	2:45

RAM disk - PC-DOS		1 job C/PC-DOS	2 jobs C/PC-DOS
Program 1	:57	1:02	2:17
Program 2	1:12	1:17	2:32

(available for some time for PC-DOS in the public domain,) selective housecleaning tasks are made rather simple.

DIR and SDIR - DIR lists the files in the current directory in a CP/M-style format, with 5 files per line. Using the /L option displays the more standard directory format, including file sizes and creation dates. SDIR is a more exhaustive directory function that lists all files in a CP/M or PC-DOS directory, whether CP/M system files or normal files. Files are listed in alphabetical order by filename and extension. Each file listing also shows the amount of space used by the file and the file's attributes. SDIR is somewhat slower than DIR, due to the sorting process required.

SET - The SET command changes CP/M file attributes, such as system (read-only) status and password protection. Passwords apply only to C/PM files and media.

FUNCTION - helps the user change the commands assigned to function keys and numeric keypad keys. Like C/CPM, C/PC-DOS cannot emulate the normal PC-DOS editing keys that make re-editing a bad command easy. Instead, you can assign up to 20 characters per function key, and up to four characters per shifted, alted or ctrled function key. After assigning new keys, you can save the entries as a .PFK file that can be used in batch mode with FUNCTION to set the default keys the next time you boot.

HELP - displays information about C/PC-DOS commands and their proper usage. HELP also allows you to add new HELP information and topics interactively, a useful feature for dealers wanting to install turnkey systems in the C/PC-DOS environment.

ORDER - displays or changes the file extension search order used by C/PC-DOS to find a program when the file extension is not specified in the command. The default order is CMD, COM, EXE then BAT. You can reduce the search time by omitting or rearranging the extension list to your liking.

PIP (Peripheral Interchange Program) - familiar to C/PM users, can be used to copy files between drives, directories or devices. PIP can also copy files between CP/M and PC-DOS media. Using PIP transfer options, the user can filter or customize the copying process more effectively than the COPY command. PRINTER - displays or sets the printer assigned to the current window. C/PC-DOS can accomodate up to four printers as either parallel (Centronics) or serial (RS-232) devices. The LST:, LPT1: and PRN: device names used by PIP and COPY always refer to the current printer.

PRINTMGR - C/PC-DOS's version of PRINT in PC-DOS. PRINTMGR sets up a background print queue, manages the performance of the print queue and allows you to change the default page length, margins, tab value, etc.

STOP - displays the programs currently running in all windows and allows you to terminate them. STOP also shows the total memory available in the system for more windows or programs.

SUSPEND - since many PC-DOS programs write directly to the screen buffer, PC-DOS provides a mechanism to subdue other processes running in background windows. Set SUSPEND to ON if your application is well-behaved, or uses the Interrupt 10h screen printing functions. This will allow your .EXE and .COM program to continue running in the background while you do other tasks in other windows.

BUNDLED SOFTWARE

C/PC-DOS includes some software not ordinarily distributed with PC-DOS, including a full-screen editor, cardfile, communications program and interactive file manager.

DREDIX - is a full-screen text-editing program used to create or modify batch, program or

WHAT WORKS, WHAT DOESN'T

A PC-DOS compatible program list

We tested a grabbag variety of programs we had lying around, and also included here some of the programs listed in the User's Guide Supplement. Programs marked "works" we tested; "certified" programs are DRI tested.

Programmer tools (all work)

PROLOG TED, a programmer's editor

MAKE Microsmiths tools (downloader systems)

2500 A.D. cross-assemblers

ASMGEN disassembler

Modula-2 from Modula Research Institute

TURBO Pascal from Borland International

Communications programs

PC-DIAL works

Qmodem works - written in TURBO Pascal

Graphics Processors

DR Draw certified

DR Graph certified

PFS:Graph certified

Accounting programs

General Ledger certified

Home Accountant certified

Word processing programs

Word processing is a tricky subject. Although you can only process a word so many ways, many programs on the market use clever ways to update the screen faster. Programs don't follow the rules for displaying a character because the IBM-supplied routines are just too slow.

PC-WRITE doesn't works - crashes system

Microsoft WORD freezes up just before starting up

Wordproof from IBM works

WordStar works - certified (but 3.3 ignores windows)

NewWord (WS clone) works

MultiMate certified

PFS:Write certified

ThinkTank doesn't work - bombs out on copy protection

Database programs

PFS:File works - certified

PFS:Report certified

dBASE II works - certified

Friday works

Spreadsheet programs

Spreadsheet programs are basically character-oriented and have very simple I/O tasks; therefore all tested programs designed for DOS 1.00 or 2.x0 seem to work correctly. Large spreadsheets usually indicate large memory requirements, which might be a problem to a prospective business user.

Lotus 1-2-3 works - certified

MultiPlan works - certified

SuperCalc 2 certified

SuperCalc 3 certified

VisiCalc certified

Utility programs

These utility programs were kicked out for differing reasons

PC Mouse didn't work because Terminate and Stay Resident programs usually change some interrupt vectors to make calling them automatic. C/PC-DOS replaces all the vectors after a job finishes executing incorrectly.

PC Mouse doesn't work (uses interrupt handler)

SideKick doesn't work (uses interrupt handler)

Norton Utilities didn't work

The Norton Utilities main program NU.EXE came up with an Error 1 when reading the disk information using relative sector addressing with Absolute Disk Read. Peter Norton answered our letter, telling us that C/PC-DOS generates this error when trying to read the FAT on the hard disk.

word-processing files. DREDIX can work on up to four files at the same time. Each file is assigned a buffer, and you can switch between these buffers easily. Although four files can be referenced, only a maximum of two can be windowed on the screen at the same time. Most of the commands that do things in DREDIX are combinations of the Alt or Ctrl key with a letter key. For example, a comprehensive help screen can be called up with Alt-H.

Feature-wise, DREDIX is a great replacement for the shallow EDLIN program supplied with PC-DOS. Unfortunately, using an IBM color card, the screen flickers annoyingly whenever a character is typed; after each typed character, there must be at least 20 flickers. It appears that any action on the keyboard triggers the flicker.

CARDFILE - helps you store and retrieve name and address information on electronic card file. You can enter the first and last name, business, address, phone number, and any comment for each associate you want on file. Once entered, commands in CARDFILE allows you to search for matching cards by name, business or phone number and display them one at a time. DRTALK- C/PC-DOS's version of PC-TALK. Using this comprehensive communication package, you can converse with other systems, upload or download files using binary (XMODEM) or line-oriented protocols. Communications parameters can be set while inside DRTALK which override any set using SETPORT. The dialing directory can be customized to include your most-used phone numbers.

FM (File Manager) - takes the most common computer commands and lists them in a menu. You can select commands from the menu with a few keystrokes. In a default C/PC-DOS system, the file manager can be selected from the startup menu. FM will be useful for those who might be unfamiliar with the PC-DOS command structure.

DOCUMENTATION

All end-user documentation is contained in a single ring binder. It consists of Installation Notes, a User's Guide, and a Getting Started section. All the sections are dense and lengthy with small print and only a few illustrations, but the writing is clear and the manual is certainly exhaustive.

Within the User's Guide is the bulk of information. The File Manager, DR Talk, DR EDIX Editor, the Cardfile application, the Printer Manager and the Menu creation and modification programs are each described in a separate chapter. In another chapter, the commands that operate under C/PC-DOS are listed alphabetically. This takes a full 116 pages, with about 1/2 page to 4 pages per command. Typical of DR manuals, everything about C/PC-DOS is included SOMEWHERE, but it is not always organized or indexed usefully.

An appendix to the user guide provides a helpful list of Required Program Files for the start-up and file manager menus, an ASCII table in decimal and hex, and 53 pages of error messages. An additional User's Guide Supplement lists the software that has been checked for proper operation under C/PC-DOS, including special notes on needed modifications. See later for compatible software listing.

Getting Started with Concurrent PC-DOS coaches the beginning user in how to install the system and back up the source disks, use of the File Manager utility, how to use and change the windows, what programs are bundled with the software, and how to get interactive help.

The end-user volume outlines the concepts of C/PC-DOS and describes the commands, utilities and programs included; it does NOT explain how the operating system actually works. These tidbits are left to a separate set of Programmers Guides. You can order this Programmer's Pak directly from Digital Research for $75.00. It consists of two fat volumes: Concurrent CP/M Programmer's Guide and DR Assembler Plus Tools (see sidebar).

SOME FINAL THOUGHTS

Digital Research has been struggling for some time to regain the momentum it lost when C/PM was overshadowed by PC-DOS. With the introduction of C/PC-DOS, we think it's safe to say that DRI will contain some of this slide, although it certainly will not regain its leadership role.

Overseas, however, another story emerges. CP/M is still a major force, with many vendors wanting both DOS and CP/M compatibility. The result is that DRI has already issued some 100 licenses for C/PC-DOS worldwide. Of these, 30 to 35 major European manfacturers have brought C/PC-DOS to market. According Mini-Micro Systems (March 1985, pg. 68), the following manufacturers offer C/PC-DOS as well as MS-DOS for their computers:

- Ericsson Information Systems division of L.M. Ericsson AB
- Ferranti Computer Systems Ltd.
- Olivetti SpA
- Seimens AG
- Applied Computer Techniques Plc. (ACT)
- ICL Plc.
- British Telecom Merlin Ltd.

Therefore, we can expect DRI and C/PC-DOS to continue to prosper, regardless of its impact in the American market.

However, in America C/PC-DOS does face stiff competition from other windowing programs, particularly IBM's Topview. The weak points for C/PC-DOS are its slow performance and absence of any feature to transfer data between applications. As these other products mature, the window of opportunity for C/PC-DOS will start to close unless substantial improvements are made.

In the multi-user area, C/PC-DOS is too weak a product to last before the onslaught of networks and UNIX-based systems that are emerging to run on PC, AT and S-100 systems. Once again, the market window will be very brief.

One particular problem that DRI will face with C/PC-DOS is wooing software developers to the product. Developers familiar with PCs will likely stick to familiar confines of MS-DOS, waiting for new power from upwardly compatible improvements, rather than switch back to C/PM.

MODEL 100 C COMPILER

Now you can write efficient programs for your TRS-80 model 100 with ease. Or, learn the essentials of C programming while traveling!

C/100 - THE "PORTABLE" C COMPILER

Cassette version $49.00
Disk/Video interface version $59.00
Model II version (run on mod II, then download object code to model 100) .. $79.00
Model III version (as above for Mod III) . $79.00

Write or call for information on other TRS-80 software.

MODELS II, 12, 16
MODELS III, 4

TRS/C C COMPILER

Full K&R with source to the function library. UNIX compatible $85.00

ZSPF EDITOR

SPF, the choice of most mainframe programmers, is now available for Z80 machines. And it's panel driven so you can customize it! $75.00

business utility software
109 minna ste 423 san francisco ca 94105
(415) 397-2000

Turbo Pascal 3.0

by David W. Carroll

A review of the new upgraded version for MS/DOS & CP/M

It has been about a year and a half since my review of Turbo Pascal version 1.0 was published in *Microsystems* Magazine ("The Versatile Turbo Pascal", February 1984). In that time, over a quarter million copies of Turbo Pascal have been sold.

International Data Corp., a major market research firm, recently predicted that Pascal, or a Pascal-like language such as Modula-2, will begin to replace BASIC as the primary programming language for users of PCs this year. He credited Borland's Turbo Pascal with being "the prime motivating factor in Pascal's imminent rise in popularity."

In the meantime, Borland International has been busy creating other high quality, low cost products for microcomputer users. Borland has released SideKick and SuperKey as well as upgrading Turbo Pascal to version 2.0 and releasing Turbo Toolbox and Turbo Tutor.

Proving that we can expect the unexpected from them, Borland suprised many Turbo users by releasing an even more powerful Turbo Pascal version 3.0 in March. Turbo 3.0 claims to offer even faster compilation and execution than its forerunner, version 2.0. In addition, a number of new features have been added to the 16-bit version of the product, including:

* DOS 2.0 path support and I/O handling
* Turtle Graphics (IBM PC Version)
* Enhanced graphics support
* Optional 8087 and BCD arithmetic versions (16-bit)
* Improved file handling functions, including APPEND
* All known Turbo 2.0 bugs and problems have been fixed

One unwelcome change in Turbo 3.0 is its price - increased $15 from the $54.95 (including shipping) for version 2.0 to $69.95, but it is still a teriffic deal. Unfortunately, version 2.0 is no longer being sold, so new users will have no choice but to buy the new version.

For many of the estimated 300,000 Turbo Pascal owners, the question is "Should I upgrade?" Borland offers an attractive upgrade incentive - just return your distributon disk for version 2.0 and Borland will credit you with $39.95 toward Turbo version 3.0 (or if you have the 8087 version, you will be credited $59.95 toward an enhanced version - 8087, BCD, or both). This offer expired on June 1, 1985.

Although the many benefits of upgrading are obvious for PC-DOS Turbo users, CP/M Turbo 2.0 users may wish to review the new features section below before deciding on an upgrade.

NEW MS-DOS VERSION FEATURES

In addition to its increased speed (discussed below), Turbo 3.0 for the PC and 100% compatibles has many powerful additions. Borland certainly didn't "leave well enough alone."

It now includes standard procedures for directory access and manipulation. In addition, file names may now contain drive/subdirectory information, thus a file name like 'C:\mydirec\myfile.dat' is now legal.

Full I/O redirection on standard files is provided. The run-time package also uses standard MS-DOS file handles. Of course this means that Turbo 3.0 will only run with MS-DOS 2.0 and later. There are a number of system-interface procedures for getting command line arguments, defining a path for TURBO overlays, etc. The installation program allows defining a path for the Turbo message overlay file as well.

There are a number of advanced graphics functions - for generating circles, fill patterns, etc. - as well a very nice implementation of Turtle graphics.

In addition, you can change file buffer sizes, open a file for append, and text I/O is considerably faster - especially with larger buffer sizes - and a number of new procedures are available for manipulating text files.

The distribution disk still includes the CALC.PAS example spreadsheet program, one of the most interesting programming examples I have seen to date.

8087 SUPPORT VERSION

Turbo 3.0 is also available with 8087 math co-processor support, as was Turbo 2.0. This version is designed primarily for number crunching applications (scientific, etc.) and provides dramatically increased speed and 16-digit accuracy over a range of 4.19E-307 to 1.67E+308. It uses the long-real 64-bit data type of the 8087 chip.

NEW BCD REALS VERSION

A new optional version of Turbo 3.0 supports Binary Coded Decimal (BCD) real numbers instead of the standard binary reals in the normal Turbo package. Using BCD reals provides a significant increase in accuracy and virtually eliminates rounding error - making this version attractive for developing business accounting software packages. The range of the BCD reals is 1.0E-63 to 1.0E+63 with 18 significant digits.

Several numeric and string formatting procedures are included to simplify screen and printout design, much like the PRINT USING feature of BASIC. BCD reals occupy 10 bytes compared to 6 bytes in standard MS-DOS Turbo 3.0.

NEW CP/M-80 VERSION FEATURES

Here's a summary of what CP/M-80 users get when they update from Turbo 2.0 to 3.0, as described by a file on the CompuServe Borland SIG:

-You get an entirely new, rewritten

reference manual.
-ParamCount and ParamStr functions parse the command line for you.
-Overlays now work according to the specifications in the manual.
-Mark and Release now work correctly.
-Functions SeekEoln and SeekEof which act like EOLN and EOF except that they skip blanks and Tabs before testing for end of line or file.
-Simplified error handling - a new standard variable, ErrorPtr can be assigned the value of ADDR of the users error handler. Can trap and identify user break (Ctrl-C), I/O errors, and run-time errors.
-EXIT procedure for immediately terminating execution of the current procedure or function.

As can be seen, the CP/M-80 upgrade provides several new functions, some fixes, and possibly increased speed (not tested in this review).

ALL BUGS FIXED?

According to Borland, all known Turbo 2.0 bugs and problems have beem fixed in 3.0. However, there is at least one bug and an irritating problem that are new with Turbo 3.0.

The new bug has to do with an error in the *FileSize* function and a work-around fix has been published.

The new problem is that the Turbo 3.0 compiler always leaves the CGA in BRIGHT YELLOW when it exits to DOS. No fix is available. Compiled programs will leave the CGA in the last mode and color specified upon return to DOS.

NOT 100% COMPATIBLE

Some programs written in Turbo 2.0 will not function the same (or in some cases not function at all) when compiled with version 3.0. First, PC-DOS and MS-DOS versions no longer work with DOS 1.X.

On the simple side, Turbo 3.0 no longer automatically clears the screen at the beginning of a program, and the program's termination no longer places the cursor at the beginning of line 25.

Turbo 3.0 uses a different EOLN function that handles keyboard input differently, which means that some INKEY$ like routines using EOLN(Kbd) will no longer work.

Users of Turbo Toolbox will find that an upgraded ACCESS3.BOX file is included with Turbo 3.0 to allow the new compiler to be used with the Toolbox - an indication that upward incompatibilities can be a real problem.

DOCUMENTATION

The 376 page typeset reference manual is a comprehensive guide to Turbo 3.0. However, there are a number of errors in the manual, most of which are noted in a READ-ME file on the disk. One major topic I found lacking was a discussion of the differences between Turbo 2.0 and Turbo 3.0 to aid in upgrading existing programs. It looks like users must use trial and error to try and debug any problems running Turbo 2.0 programs under the new version.

SPEED

One of the major reasons for buying Turbo Pascal has always been its speed. Turbo 2.0 was at least 10 to 20 times faster than other Pascal compilers. Execution speed was also 3 to 10 times better than other compilers. Turbo 3.0 goes even farther to break the Pascal speed barrier.

The Turbo Editor screen display code has been rewritten to make screen updates virtually instantaneous. The "flash and flicker" that sometimes occurs when using the IBM CGA card is notably absent with the new Editor. With the improved editor and even faster compilation speed, the Turbo 3.0 programming environment really does "work like magic"

Compiling with Turbo 3.0 really is twice as fast as with version 2.0! Several large test programs were compiled in both versions and the results were consistant. On the IBM PC, the 731 line TPL0.PAS compiler from Nikalus Wirth's book **Algorithms + Data Structures = Programs** compiled in 23 seconds with Turbo 2.0 and 12 seconds in version 3.0. The 1257 line CALC.PAS example spreadsheet program provided by Borland on the distribution disk compiled in 29 seconds with version 2.0 and in 15 seconds with version 3.0. I noted that loading program files from disk into the compiler was also somewhat faster, especially when using floppies.

Compiled code was between 416 and 512 bytes smaller in version 3.0, with the code size decrease varying between 4.6% and 2.7%. This non-linear decrease seems to indicate that the standard library functions which Turbo 3.0 includes in each program, are somewhat smaller, rather than a decrease in the size of compiled program code itself.

Several tests with text file input/output indicate that Turbo 3.0 is consistantly 15.6% faster than version 2.0 on the IBM PC with a hard disk and standard buffer sizes, probably due to its use of DOS 2.0 file handles. This improvement increases with increased buffer size, now allowed in Turbo 3.0. A simple disk I/O intensive text filter program which reads text files character-by-character and writes each character to a new file showed a 6 second improvement from 2.0 to 3.0 on a 25,229 byte file (38 seconds with 2.0 and 32 seconds with 3.0). When a 126,336 byte file was processed by the same program, an improvement of 28 seconds was noted, from 179 seconds in 2.0 to 151 seconds in 3.0.

On programs that are not real number arithmetic intensive, no speed improvement in Turbo 3.0 execution was found, other than that accounted for by the improvement in file handling noted above.

I entered the 179 line "Guass-Seidel" program for the simultaneous solution of linear equations from Alan Miller's book **Pascal Programs for Scientists and Engineers** that Borland refers to as an execution speed benchmark in their 3.0 advertisments, and ran Miller's specified example, but was unable to duplicate Borland's published results. After modifying the program to remove intermediate screen displays and to include a DOS timer function, I found that Miller's example of a three equation electrical current problem using a relaxation factor of 1.0 (requiring 20 iterations) took 1.21 seconds to solve with Turbo 2.0 and 0.66 seconds to solve with Turbo 3.0 and the same problem using a relaxation factor of 1.8 (requiring 44 iterations) took 3.08 seconds to solve with Turbo 2.0 and 1.49 seconds to solve with Turbo 3.0. Since no file I/O was involved, this represents a 2 to 1 improvement in execution speed - an improvement that appears to remain linear as program iterations increase.

(continued on page 84)

Product: Turbo Pascal 3.0 (PC-DOS)
Company: Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066
408/438-8400
Price:

Standard Turbo Pascal	$ 69.95
Standard plus BCD Reals version	$109.90 *
Standard plus 8087 version	$109.90 *
Standard plus BCD and 8087 version	$124.95 *

Turbo 2.0 upgrade, subtract $39.95 and return disk
Turbo 2.0 8087 upgrade, subtract $59.95 and return disk
(upgrade valid for the same or higher level productthrough June 1, 1985 only)
Manual: 376 page typeset softbound reference manual
Available for: PC-DOS, MS-DOS, CP/M-86, CP/M-80 (Z-80 only)
Note: Not copy protected.
* Available for 16 bit versions only

The Coherent System

by Les Hancock

A Low-cost Multiuser UNIX Look-a-like for the XT & XT compatibles

COHERENT is an operating system sold by the Mark Williams Company, a small software house started in 1976. Like most such shops they've concentrated on OEM sales, where they seem to have done well -- for example, COHERENT will be bundled with a new Z8000 workstation just announced by Commodore for sale in Europe. In 1983 they broached the end-user market with an IBM PC version of COHERENT.

I've used COHERENT often over the last two years. It's a UNIX lookalike: it looks and acts like the Seventh Edition of UNIX released in 1979, but it's homegrown, written without reference to the code from AT&T Bell Labs.

The system tested here is COHERENT 2.3.39. The testbed was an IBM PC XT with 512K RAM and a ten-megabyte Seagate disk. (According to Mark Williams this version will also run on a COMPAQ+ or on PC's with hard disks by Corvus, Davong, Micronetworks, NCR and Tecmar.) For a multiuser, multitasking system, COHERENT is quite small. The root files take up about two megabytes of disk space, plus half a meg for swapping, and the kernel needs only 78K of RAM.

Version 2.4, due out later this year, will include some enhancements and utilities now in test. The update policy is reasonable: to get the latest version, send back the old diskettes with a check for 10% of the new version's cost. COHERENT is priced well below competitors like VenturCom's VENIX and Microsoft's XENIX.

BRINGING UP COHERENT

COHERENT comes in a packet of seven diskettes, six for the root files and one to build the system. Instructions in the release notes are straightforward; I read them over once, then went through them step by step and had a working system built in half an hour. The directions are so clear that any novice should be able to follow them. New users may not always understand what it is they're doing, but following the examples will give them a sensible system.

DOCUMENTATION

COHERENT 2.3.39 arrived with eleven pounds of documentation, packaged in two oversize three-ring binders. The documents are bulky but nicely laid out and well organized. A typeset version in IBM PC size is due shortly, but the difference is only cosmetic. Here are the titles:

bc Calculator Language Tutorial (27pp + index)

COHERENT Administrator's Guide (57pp + index)

COHERENT Assembler Reference Manual (35pp + index)

COHERENT Command Manual, Revision 3 (228pp + index)

COHERENT System Manual, Revision 3 (175pp + index)

COHERENT Version 2.3.39 Release Notes (77pp + index)

ed Interactive Editor Tutorial (66pp + index)

An Introduction to ELLE (36pp + index)

Introduction to the COHERENT System (91pp + index)

learn User's Guide (26pp + index)

lex -- Lexical Generator: Tutorial (28pp + index)

m4 Macro Processor User's Manual (15pp + index)

nroff Text Processor Tutorial (64pp + index)

Shell Command Language Tutorial (40pp + index)

The TROUT Full-Screen Editor, Version 1.3 (29pp)

yacc Parser Generator Tutorial (37pp + index)

Until very recently most UNIX lore was transmitted from druid to druid by oral tradition. The Mark Williams people seem to have understood that COHERENT users may not have druids handy. They've written an excellent suite of manuals aimed at intelligent readers who need to know how to install and administer a multiuser operating system. The emphasis is on examples and tutorials, which is all to the good. The reference sections are generally on a par with those in the standard UNIX user's manual, but the tutorials outshine any UNIX tutorials not written by Brian Kernighan. I especially appreciated the easygoing, take-my-hand introductions to **lex, awk** and **yacc**. Users learning those tools, like student skydivers, need all the hand-holding they can get.

Apart from the reference guides the most important documents are certainly the Release Notes, the Introduction to the COHERENT System, and the Administrator's Guide. With the slight exception noted below under **Communications**, these three manuals tell new users exactly how to install COHERENT, get it up and running, and maintain it in a multiuser environment. They can and should be read through from beginning to end, like a novel.

Here's a sample of the 2.3.39 release notes:

Adding new users

To add new users john and mary to your COHERENT system, use the **newusr command:**

```
/etc/newusr john mary
```

newusr will create home directories for **john** and **mary** in the directory **/u**.If you have a partition for users' directories (**/dev/hd1** in the example given above [and not shown in this review]), before you use **newusr** for the first time, you should be sure that **/u** is mounted with the command

```
/etc/mount /dev/hd1 /u
```

This will put the user files into the partition **hd1**. To make sure that **/u** is mounted whenever you boot the system and go to multiuser mode, put the

command to mount it into the file **/etc/rc**. **newusr** also creates a **.profile** file for each new user. When **john**'s **.profile** is created, all it will contain is the line

export PATH=:/u/john/bin$PATH

See the documents Introduction to the COHERENT System and **sh** Shell Command Language Tutorial for further information.

THE MARK WILLIAMS HOTLINE

The COHERENT documentation sets a high standard, with plenty of examples and good indexes, but the time may come when what you need to know can't be found in any manual. At that point registered owners of COHERENT can call a support hotline for advice. The idea's excellent and I wish I could tell you how well it works, but I've long since blown my cover as a reviewer of COHERENT, so my technical questions went directly to names I knew. When I first tried the product nearly two years ago, the staff politely and patiently answered my questions before they could know I was a reviewer. There's no reason to think that attitude has changed.

THE PROGRAMMING TOOLS

UNIX is a development system at heart, and COHERENT provides nearly all the usual development tools. **lex** and **yacc** are there, as well as **awk** and **m4**, and the obligatory C compiler. The only omissions worth mentioning are the Source Code Control System (too big to fit), the syntax checker **lint**, and the pretty-printer **cb**.

In a sense COHERENT's most important development tools are its system calls, which are identical to those of UNIX 7, and its C library, which is complete unless you need exotica like the double-precision log-gamma function. According to Mark Williams, C programs that run on UNIX 7 will run on COHERENT after recompiling, and I think that's true if you take it with a grain of debugging. After all, it's seldom possible to move code from UNIX on computer A to UNIX on computer B without fixing a few glitches, especially if the program's a large one and the coders took a few shortcuts. (They always do.)

Even shell scripts can often be moved as is from UNIX to COHERENT, since COHERENT's **sh** is a clone of the Bourne shell. Unfortunately, comments are limited to lines beginning with a colon; the comment symbol # would have made a nice enhancement.

Other tools don't always follow UNIX usage one hundred percent. Suppose you want to sort **filein** and send the sorted output to **fileout**. The command line

sort filein -o fileout

works fine with UNIX but not with COHERENT, which insists on the counter-intuitive

sort -o fileout filein

It's hard to fault Mark Williams here, since their version follows the command syntax given in the UNIX user's manual, while UNIX doesn't. Yet it's a fact that a UNIX shell script with the line

sort filein -o fileout

won't run under COHERENT. This points up a simple lesson: if you need UNIX, get UNIX. For instance, if your office or lab runs UNIX on minis, and you'd like to add PC workstations, and you expect to move the code you're developing back and forth between minis and PC's, you probably want a UNIX derivative that's based on the straight UNIX code. You can still expect some portability trouble -- that's inevitable -- but you'll suffer less from conflicts like the one shown above for **sort**.

Otherwise COHERENT's compliance with UNIX should be close enough to satisfy individual users. I've tested various tools by moving programs from COHERENT to UNIX and vice-versa, and by running the examples published in several standard UNIX guides. Results were good, though most code needed a little tweaking. The problems usually had to do with command syntax or the name of a library, and I was never really stymied.

Mark Williams' C compiler is excellent. It supports the standard features described in Appendix A of Kernighan and Ritchie's "C Programming Language", and adds a few popular extensions -- **void** and **enum** types, structure assignment, **readonly** statics, names with up to 39 significant characters. MWC86, the MS-DOS version of this compiler, sells for the same price as the whole COHERENT package. COHERENT's C is essentially the same as MWC86, though it doesn't support the 8086's large memory model, which means you're restricted to programs with 64K bytes of data and another 64K of executable code. MWC86 can use the 8086's whole memory space, a feature promised for a future release of the COHERENT compiler. Also promised is a COHERENT version of the symbolic debugger that's shipped with MWC86. I've used that debugger and found it very helpful. With it you can debug a C program in terms of the source code, which means cutting big chunks off a project's development time. For now COHERENT users have to make do with **db**, an assembly-level debugger like UNIX's **adb**. They'll also have to give up **lint** -- no great hardship, since the C compiler's error-checking is unusually good, with **lint**-ish warnings triggered by nonportable code.

Incidentally, the C library provided for MWC86 offers most of the functions in COHERENT's C library, even including **setjmp** and **longjmp**. That makes COHERENT a good environment for MS-DOS software design. Programs can be developed in COHERENT with UNIX-lookalike tools, then moved painlessly to the MS-DOS compiler.

TEXT PROCESSING

Text processing was UNIX's first application, and in many offices UNIX is still used mostly for editing, formatting, typesetting and vetting text. COHERENT supports editing and formatting with the line-oriented editor **ed**, two screen editors, and **nroff**. Mark Williams does have a spelling checker, but the dictionary takes up so much room that it's omitted in the XT distribution. Shops that do a lot of word processing may want to order it as an option.

ed is a common bond in the UNIX community. Most installations offer a screen editor of some sort, usually **vi** and sometimes **emacs**, but you can never be sure what you'll find on a stranger's system, or whether it will work with your screen. **ed** is an uncomplicated, reliable editor that works at any terminal. Personally I prefer **ed** to screen editors like **vi**, which gobble memory, run slow, and rely on keys that wander from place to place on different keyboards.

COHERENT's **ed** surprised me at first by refusing blanks in command strings. That is,

1,23 w thisfile

won't work -- you have to type

1,23w thisfile

The explanation is that COHERENT's version allows constructions like

1,23wq thisfile

where commands are piled together. I prefer consistent syntax to multiple commands. Otherwise **ed** is everything it should be, and unlike many editors it will handle files larger than 64K.

COHERENT's screen editors are **trout** and **elle** (rhymes with belly), both variants of **emacs**. They work quite well and should please those who like screen editors. **elle** offers windows, which will please those who like windows. I tried to stick with these editors long enough to overcome my prejudices, but had to give it up.

INTRODUCING THE LATEST IN
HIGH QUALITY PRODUCTIVITY TOOLS
FOR MICROCOMPUTER SOFTWARE
DEVELOPERS AND PROGRAMMERS

{SET} Tools –

- Operate on most popular MS-DOS and CP/M systems.
- Can be used with any source language.
- Improve development productivity.
- Provide assistance for the tedious task of maintenance.

{SET:DIFS}™ Source File Comparator $139.00
- Fast, smart and accurate
- Use for regression testing, too
- Difference display highlighting

A/P/L Options available as ***add-ons*** *to {SET:DIFS} to provide for minimized communications with the ADR or PanValet mainframe librarians or with {SET}'s Batch Line Editor, {SET:LIKE}. $20.00 per option*
{SET:LIKE} ***add-on*** *to {SET:DIFS}. $40.00*

{SET:GXREF}™ Cross Reference Utility $79.00
- Supplied parameter files allow use of any source language
- Cross references multiple files at once

{SET:PATCH}™ Object File Editor $79.00
- Quickly apply changes to any file type
- Hexadecimal and ASCII display and change entry
- Easy to use with cursor and function keys

{SET:SCIL}™ Source Code Interactive Librarian $349.00
- Maintains history of changes
- Reduces storage by identifying differences from level to level
- Provides control over concurrent development efforts by detecting overlapping changes

PC-Demo Disk and Manual available for $35.00

{SET} Tools are available individually or, better yet, select a combination of tools to meet your specific needs.

Multiple copy discount available.

{SET} Get {SET} for Success
System Engineering Tools, Inc.
645 Arroyo Drive • San Diego, CA 92103

COD, Check with order, Master Card or VISA accepted.

To order {SET} tools or for more information, call System Engineering Tools, Inc. (619) 692-9464.

Moving the cursor back and forth means typing two-finger chords like ctrl-F, ctrl-B and ctrl-A. I defy any touch typist to live with ctrl-A.

COHERENT's **nroff** comes with the standard **ms** macro package. I've used **nroff** heavily without problems. I do have a quibble, though.

nroff's output contains control codes for underlining, boldface, and so forth, and of course these codes differ from printer to printer -- a perennial problem with word processors. Mark Williams' **nroff** produces the sort of plain-vanilla codes that drive a Diablo or Qume printer. Since many PC's were sold with the Epson MX80 dot-matrix printer, a filter called **epson** is provided to reshape the output. Just type

nroff -ms myfile ¦ epson

and formatted output goes to the printer.

This works perfectly for the MX80, but my XT came with a slightly different Epson, IBM's "Personal Computer Graphics Printer," which doesn't support the MX80's italics. I noticed this problem in the earlier edition of COHERENT and wrote a filter to get around it. It's still a problem in 2.3.39.

COMMUNICATIONS

Over the years communications and networking have become very important to UNIX. From the beginning dialup ports were provided for remote access via phone lines and modems. Then utilities were added to ship data from one system to another, and an informal network of UNIX sites began to take shape. There are well over a thousand nodes in the UNIX net now, and the traffic in electronic mail and news articles is heavy enough to cause concern -- a busy site can run up four-digit phone bills. The mail system's ease of use is seductive, and most of us still get a ham-radio thrill out of sending notes to Europe or Australia. The utility programs that make all this happen are known as the **uucp** group, and they're invisible to most users. UNIX calls them out secretly, often late at night, and keeps the details to itself -- which is just as well, since those details are ugly. Communicating via UNIX is a pleasure, but the communications utilities are hard to write, debug and maintain, which may explain why no version of COHERENT has offered **uucp** so far. However, 2.3.39 includes an effective system-to-system communications package, and provision is made for dialup lines.

Installing a COHERENT dialup is easy. The XT comes with an asynchronous port and an RS-232 connector. Attach a Hayes-compatible modem, amend the **/etc/ttys** file to identify the port and set its baud rate, and you're ready to go. I should say that I had some trouble with this step because COHERENT 2.3.39 names its devices differently from the version I was used to, and for once the documentation didn't help. I worked this out with Mark Williams' technical staff, and they rewrote the release notes to make installation easier.

For inter-site data transfer there's **kermit**, COHERENT's version of the widely used communications package. It's a well-developed piece of software and worked perfectly when I tried it. With it you can dial in to any on-line system, using your XT as a dumb terminal. If that system also has **kermit**, you can transfer files. **kermit**'s error-checking protocols allow you to send and receive both ASCII and binary files with good confidence, even over normally noisy phone lines at 1200 baud. Unfortunately these file transfers can be done only between **kermit** systems -- since I have **kermit** you can use *your* **kermit** to send me files or to get files from my system, but you can't load up or down to a system that doesn't support **kermit**. By the way, despite its

name **kermit** is anything but cuddly; novices won't find it easy to use. One of Mark Williams' good tutorials is needed here.

Now for the good news. A future release of COHERENT will include the **uucp** utilities. The software is just going into test, but it does exist and it does work. Mark Williams sent me a prerelease copy, which I've used to communicate with other COHERENT and UNIX sites. This **uucp**, written by Lauren Weinstein, is functionally similar to the UNIX standard and should greatly enhance COHERENT's appeal. The eventual release will also supply cu, a dialup utility that lets you share ASCII files with non-**kermit** systems.

PERFORMANCE

The PC XT has never been a speed demon and wasn't meant for multitasking. It's certainly the least workhorse you'd want to saddle with timesharing, and UNIX sits heavy on it. True, the first versions of UNIX ran on even smaller hardware, but that was ages ago by the digital clock. Nowadays a middling UNIX shop may put fifteen or twenty users on a PDP-11/70 or a VAX, and no software can make a VAX out of a PC.

COHERENT certainly squeezes a lot of work out of a little computer, but here as in the old saw, two's company and three's a crowd. A 512K XT will reasonably serve three COHERENT users, one at the console and two at terminals hung onto the async ports; when memory falls short tasks are swapped out to disk, so no job will fail for lack of space unless the disk is full. But if all three users do heavy work --**nroff** jobs, for example, or compiles, or number crunching -- then response time will exceed the threshold of boredom. A two-user setup seems more reasonable, and that's how I tested COHERENT, with the second user dialing in through a U.S. Robotics modem.

In the two-user configuration I found response no worse than what I'm used to on a twenty-user 11/70, though disk-heavy jobs can always slow you down. Here are some timings that compare COHERENT on the XT with VenturCom's VENIX, and with UNIX or UNIX clones running on two new micros and a VAX.

The first benchmark does ten thousand system calls and barely touches the disk:

```
main(){
    int i;

    for (i = 0; i < 10000; ++i)
        getpid();
}
```

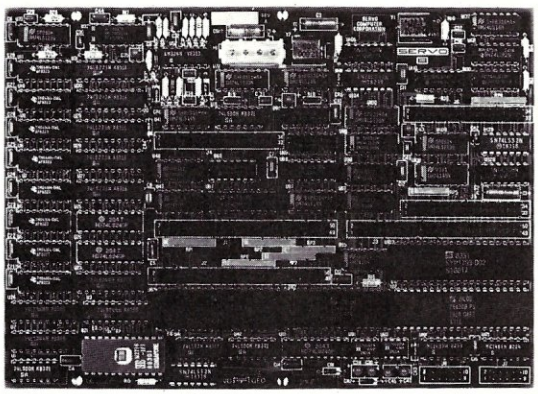
Here are the timings:

DEC VAX 11/780, UNIX 4.2 BSD	2.4 min
IBM PC AT, XENIX	3.0 min
AT&T UNIX PC, UNIX V	3.2 min
IBM PC XT, VENIX/86 2.0	8.4 min
IBM PC XT, COHERENT 2.3.39	1 min

That's a little artificial, of course -- real jobs move data. So I tried another benchmark, one that leans heavily on **nroff** and exercises programming tools like **awk, sed, sort** and the shell. It formats a 24K text file, searches it for five keywords, and prints an index with page numbers. Here's the shell script:

```
: Script to index keywords in an nroff text.
: Usage: index nroff-file-name keyword-file-name
:
: Use sed to turn keyword file into an awk prog file
sed  '/^.x$/s///&/{printf".IX &\n"}/' $2 > /tmp/_xw$$__
: Add null prog to print all lines
echo "/^/{print}" > /tmp/_xt$$__
```


4 MHZ ON YOUR Z80 IS LIKE DRIVING 40 ON THE FREEWAY, *GO 60 INSTEAD!*



SERVO 8 HIGH PERFORMANCE 6 MHZ SINGLE BOARD COMPUTER

- 6 MHZ Z80B CPU — RUNS AT FULL SPEED WITH NO WAIT STATES
- FOUR LAYER BOARD (5.75" x 8") CAN MOUNT DIRECTLY TO MINIFLOPPY
- POWER REQUIRED 5 VOLTS AT 1.4 AMPS. NO OTHER VOLTAGES NEEDED
- UNIQUE FLOPPY CONTROLLER WITH AUTOMATIC SELF-ADJUSTMENT (NO POTS) FOR; 3.5" DRIVES, 5.25" DRIVES, 8" DRIVES, 1.6 MB 5.25" DRIVES
- CONFIGURATION MANAGEMENT UTILITY INCLUDED ALLOWS EASY MENU-DRIVEN SELECTION OF OVER 50 DIFFERENT FLOPPY AND WINCHESTER FORMATS AS WELL AS BAUD RATES, PRINTER PORT SELECTION AND TURN-KEY AUTOLOAD
- S.A.S.I. (SCSI) BUS FOR WINCHESTER CONTROLLER (XEBEC 1410)
- TWO RS232 PORTS WITH SOFTWARE SELECTABLE RATES 300 TO 153.6K BAUD
- STANDARD CENTRONICS TYPE PARALLEL PRINTER PORT
- 2K EPROM WITH AUTO SELECTION FOR BOOTSTRAP (FLOPPY OR WINCHESTER)
- 64K 150NS DYNAMIC RAM WITH 128K EXPANSION AVAILABLE
- 50 PIN SYSTEM EXPANSION BUS WITH Z80 TERMS PLUS ADDITIONAL TERMS
- REAL TIME CLOCK, TENTHS OF SECONDS, SECONDS, MINUTES, DAYS, WEEKS
- NOT A TOY, SERVO USES MIL-SPEC OR INDUSTRIAL GRADE PREMIUM PARTS

- A & T SERVO 8 COMPUTER — **$389** FOR CP/M ADD **$70** VISA M/C COD
- CP/M V2.2 CBIOS SOURCES — **$50**; INCLUDES WINCHESTER FORMATTER, EPROM, CBIOS (Z80 CODE), CONFIGURATION UTILITY (TURBO PASCAL CODE)

- SERVO EXPANSION BOARD WITH 128K ADDITIONAL RAM, CLOCK/CALENDAR WITH BATTERY BACKUP, TWO ADDITIONAL SERIAL PORTS, — **$384**
- SERVO CONTROL INTERFACE WITH 24 ANALOG INPUTS AND 8 ANALOG OUTPUTS (12 BIT ADC, DAC) PLUS 16 DIGITAL INPUTS, 64 DIGITAL OUTPUTS — **$495**

SERVO

SERVO COMPUTER CORPORATION
360B N. ELLENSBURG ST. BOX 566
GOLD BEACH, OREGON 97444
(503) 247-2021

```
cat /tmp/_xt$$__ /tmp/_xw$$__ > /tmp/_xs$$__
: Use awk to insert .IX lines in nroff text
awk -f /tmp/_xs$$__ $1  > /tmp/_xu$$__
: Insert macro definition
echo ".de IX" > /tmp/_xx$$__
echo '.tm \$1" \n%' >> /tmp/_xx$$__
echo ".." >> /tmp/_xx$$__
: nroff the file, saving .tm output
cat /tmp/_xx$$__ /tmp/_xu$$__  |
   nroff -ms >/dev/null 2> /tmp/_xy$$__
: C program ndex formats index into 24-char col,
sort -u /tmp/_xy$$__ | ndex 24
: remove temp files
rm /tmp/_x*__
```

And here are the timings:

DEC VAX 11/780, UNIX 4.2 BSD	62.9
AT&T UNIX PC, UNIX V	65.3
IBM PC AT, XENIX	88.9
IBM PC XT, COHERENT 2.3.39	200.3
IBM PC XT, VENIX/86 2.0	210.5

(All timings are in cpu seconds, user + sys. The UNIX PC was tested with the 10-megabyte disk.) I think those tests justify two general conclusions. First of all, COHERENT is a bargain for XT owners, who can convey the feel of a minicomputer UNIX to two users for $500. (In fact, if a mini's users are numerous and busy the real turnaround time for a job might be better on the XT.) But -- conclusion number two -- UNIX fans looking for a new computer will find a still better bargain in the likes of the PC AT and the AT&T UNIX PC, which cost less now than I paid for my XT two years ago.

I understand that COHERENT runs on the PC AT in 8086 mode, and I'm sorry I wasn't able to time it. Unfortunately Mark Williams has no immediate plans for a version that would exploit the extended features of the AT's 80286 chip, as XENIX does. Let's hope such a version is forthcoming; COHERENT on the AT could make an attractive package.

All of which brings up a point about performance that can't be measured in cpu seconds. In 1970-71 developers at Bell Labs set up the first commercial UNIX installation, a word processing system for the Labs' Patent department. It was installed on a PDP-11 with 24K bytes of core, a half-megabyte hard disk, and no memory protection, at a cost of about $65,000. Today, for much less money, you can put COHERENT or some other UNIX offshoot on an XT with 640K of RAM, twenty megabytes of disk -- *and no memory protection*. Chips meant for multitasking, like those in the AT and the AT&T UNIX PC, have ways of protecting memory so no program can accidentally destroy another. The XT's chip does not.

Writing of the first PDP-11 installation, Bell Labs' Dennis Ritchie recalls: "On a machine with no memory protection . . . every test of a new program required care and boldness, because it could easily crash the system . . . ."* This still holds true for the XT. If you run COHERENT with reliable applications software you'll probably have no trouble. But if you use your XT as a development system, whether with COHERENT or any other OS, sooner or later you'll write a program with a bug that causes a crash. "Sooner or later" is a figure of speech: most programmers will have the problem sooner, not later, and more than once on bad days.

If you're alone on the system such crashes are merely annoying. COHERENT is robust, and I've never had a crash I couldn't fix with a reboot and the check -s command. But if there's a second user, who's probably doing your monthly billing -- well . . .

COHERENT AND MS-DOS

COHERENT and MS-DOS can coexist in a strictly physical sense. Installing the system involves partitioning the XT's disk into sections -- one for the root files, one for users' files, and so forth --and the release notes explain how to set up an MS-DOS partition so you can boot the MS-DOS operating system. However, you can't jump back and forth between MS-DOS and COHERENT; you'll have to shut one down completely before you can start up the other. The only communication between the two systems is by diskette. COHERENT comes with utilities that read and write diskettes in all the XT formats.

THE BOTTOM LINE

There's a feeling of rightness about COHERENT that's hard to quantify, something to do with appropriate technology. That's a 70's catchphrase, and perhaps I can sum up my feelings by saying that COHERENT reminds me of UNIX in the 70's, before it took on so many responsibilities and put on so much weight. COHERENT works as advertised, it's well documented, and it doesn't overload the PC's little-league chip. If you own a PC with hard disk and want the functionality of UNIX, COHERENT is a bargain. It will be an even bigger bargain when Mark Williams adds **uucp**.

---

* D. M. Ritchie, "The Evolution of the UNIX Time-sharingSystem," AT&T Bell Laboratories Technical Journal, October 1984. All articles in this number of BLTJ deal with UNIX and are highly recommended.

VITAL INFORMATION

COHERENT 2.3.39
Mark Williams Company
1430 West Wrightwood Avenue
Chicago, Illinois 60614
(312) 472-6659
Price: **$500**
Hardware required:
IBM PC XT, or compatible with 256K RAM

But the PC's chip limits the system. If I were in the market for a brand new UNIX-like workstation, I'd look at micros built around a chip designed for multiuser service: the PC AT, for example, or the AT&T UNIX PC --or the C900 with COHERENT, if Commodore sells it in the United States.

Thanks to Dave Fiedler of InfoPro Systems, to Ron Underwood of Bellcore, and to Randolph Fritz of Western Union for help with uucp and comparative timings.

Les Hancock claims to have reached his peak in the 70's, when nobody was looking and has been very tired since 1968. He also happens to be a co-author of "The C Primer", a software consultant to Bell Communications Research and lives in Suffern NY.

CLASSIFIEDS

Micro/Systems Journal will accept Classified Ads. The charge is $6/line (3 lines minimum); 50 characters maximum per line. Three times Frequency $5/line; six times $4/line; non-profit clubs $2/line. Logos, special type, etc., are an extra charge. Check must accompany ad copy. Send to **Micro/Systems Journal**, POB 1192, Mountainside NJ 07092.

WANTED: Issue March/April 1982 of Microsystems. Will pay $10 for a good copy. Don Pannell, 6428 Edgemoor Wy, San Jose CA 95129; (408)-253-5108.

PUBLIC DOMAIN SOFTWARE: SIG/M, CP/MUG (5" & 8"), PC, Commodore, Apple II. Brief catalog and sample disk- $12. The Public Domain Software Copying Co., 33 Gold St., L3M, NYC NY 10038, (212)732-2565.

The PC Bus

by Dave Hardy

Editor's Note: The "PC-Bus" is used today by over 200 system and plug-in board manufacturers. It is today a de-facto standard for most single-user systems. I feel that many of our readers are interested in developing special purpose hardware interfaces to systems using this bus and hence I asked Dave Hardy to write this regular column.

Many IBM-PC and PC-compatible computer users are surprised to find out how easy it is to add peripheral devices to their machines. In spite of the fact that the PC is a rather complex microcomputer, it is designed to allow all kinds of additions to its bus, from simple I/O like game ports to complex DMA circuits like hard disk controllers and video graphics boards.

Even more surprising to most PC users is the fact that add-on boards are very simple to design and build. Many companies (including IBM) make cards for "home-brew" interfaces (called the "IBM Prototype Card") that contains most of the interface circuits that are needed to build a simple project.

This month, to show just how easy it is to use the PC bus, I'd like to discuss some of the basic concepts of the PC's expansion bus, including some of its pin assignments and definitions, and a simple I/O design.

The I/O Channel

Although the IBM-PC is basically a single-board computer, provision has been made to add additional devices via a series of 62-pin connectors called the "I/O Channel." The I/O channel is simply an extension of the data, address, and control buses of the PC's 8088 microprocessor. Along with the 8088's 8-bitbidirectional data bus, its 20 bit address bus, and associated control lines, the I/O channel also contains 6 of the PC's 8interrupt lines, 3 of its 4 DMA lines, a separate high-speed clock (along with the system clock, of course), a Power-On Reset line, a Wait line, and several others.

The I/O channel also provides enough power (+-5V and +-12V) to run most boards. The standard PC has five expansion connectors and the PC-XT has eight, although two or three of these slots are used for standard boards like the disk controller, video output, etc.

Figure one shows a diagram of the PC's I/O channel (a.k.a. expansion slot).

For a full explanation of the PC's I/O expansion lines (and a dandy do-it-yourself reference), you might want to pick up a copy of the IBM Technical Reference Manual. It probably contains everything that you'd ever want to know about the hardware inside the PC, along with all of the information that you'd ever need to build plug-in boards for it. There are very few books written about designing and building boards for the PC, and the reason why may be that the Technical Reference manual is complete enough that additional books are not necessary.

Some Pin Definitions

Fortunately, only the most complex expansion boards would ever need to use all of the pins in the PC bus. (This is where it starts to get easy.) In addition to the address and data lines, most simple I/O interfaces to the PC (like the simple parallel I/O port shown in figure two) require only 3 control lines: IOR*, IOW*, and AEN.

IOR* - I/O Read command line. This line goes active (low) when the processor (or DMA device) controlling the system requests that data from the simple interface be placed onto the system data bus.

IOW* - I/O Write command line. This line goes active (low) when the processor (or DMA device) controlling the system requests that the simple interface read from the data bus.

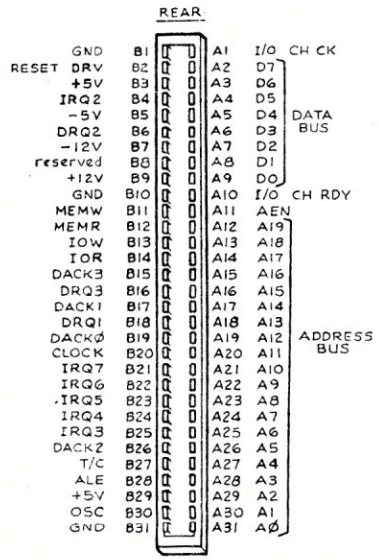


FIGURE 1 — BUS PIN FUNCTIONS

AEN - Address ENable line. This line goes active (high) whenever a DMA controller has control of the address, data, and control buses.

Beside the three lines above, our simple parallel I/O board (figure 2, again) needs to see the first ten address lines (A0-A9), all eight data lines (D0-D7), the +5V power line, and a ground line (GND). The reason that we only need to see the first ten address lines is that we are doing ported input and output, not memory reads and writes. Although the PC can address much more memory, it can only handle 1024 I/O ports (0-3FF, ten bits).

Figure Two Explained

Figure two is actually remarkably simple, and can be broken down into three parts to help understand its operation: The address decoder, the I/O decoder, and the bus interface.

The address decoder (the 11 Exclusive-OR gates in figure two) constantly watches the ten address lines used for ported I/O and sends the signal SELECT (active high) to the I/O decoder whenever it sees an address that matches the one set in the 10 switches. IBM set aside port addresses 300H to 31FH for use with its prototype board, so it might be a good idea to use one of these addresses for our simple parallel interface. Address 300H, for example, could be selected by closing switches A9 and A8.

The address decoder also watches the AEN line to make sure that the address it is seeing is a valid processor address, and not something pumped out by a DMA device that has taken temporary control of the PC's address bus.

The I/O decoder watches the IOR* and IOW* lines to see if an I/O read or write is taking place. If it sees the SELECT signal at the same time as it sees an I/O read or write, then it assumes that the processor wants to read from it or write to it, and asserts the proper line (either GATE* or LATCH).

During an I/O read, the 74LS244 Octal Buffer IC is activated by the GATE* signal, and places data from the DATA INPUTS onto the PC's data bus so that the PC can read it.

During an I/O write, the 74LS374 Octal Flip-Flop IC is activated by the LATCH signal, and latches the value on the PC's data bus into its DATA OUTPUTS.

The circuit shown in figure 2 could be even simpler if the address were hard-coded, instead of being switch-selectable, and the whole thing could also be done with fewer IC's, too, but I have left it in its unreduced form to make it easier to understand. It is easy to see, in its present form, how this circuit could be expanded into something more useful, such as a parallel printer port, or serial communications port, or game port.

In future "PC Bus" columns, I hope to discuss many of these applications, along with some of the more complex features of the PC's I/O channel, including DMA circuits, memory expansion (how about 200MB with bank select?), parity circuits, and even a front panel/hardware debugger for your PC.

As always, I welcome reader questions, comments, and suggestions, and will attempt to answer as much mail as possible each issue. If you have any input at all, please feel free to send it to me at any of the following addresses:

CIS: 70150,102
SOURCE: TCH054
Technical BBS: (313) 846-6127

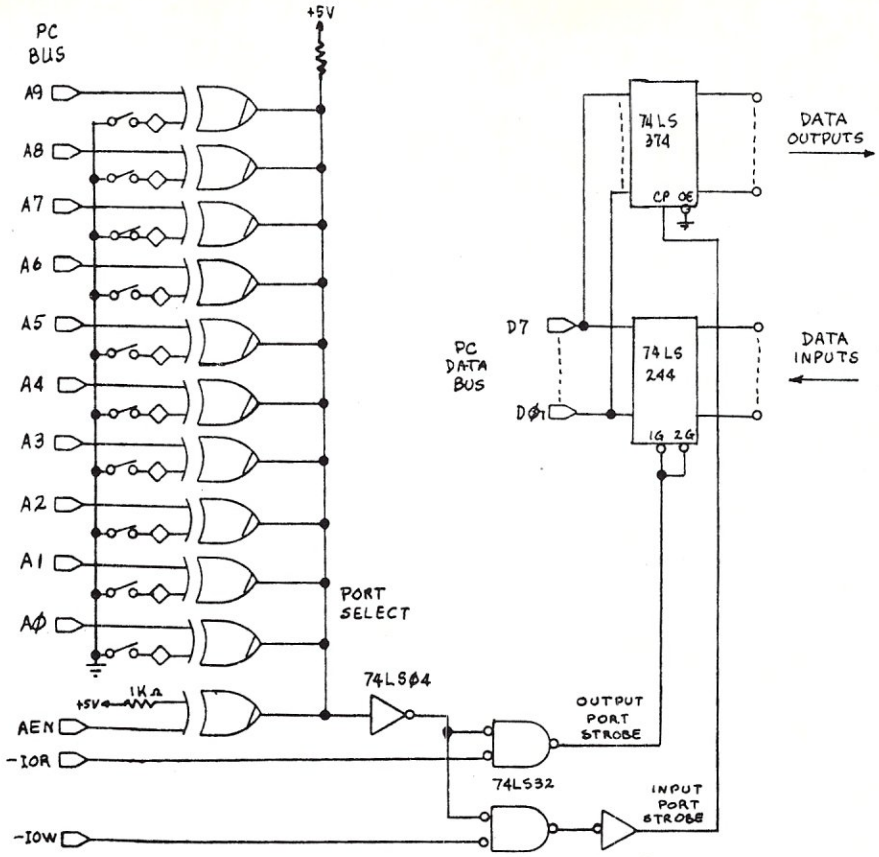


Figure 2 — A simple parallel I/O port for PC-bus systems

THE WORLD'S FASTEST

S-100 Z-80 SLAVE PROCESSOR

TurboSlave I

- 8 Mhz Z-80H
- Data transfers to 1 mbyte/second
- S-100 IEEE-696 compatible
- 4k Monitor rom
- Low parts count
- No paddle boards
- 128k Ram with parity
- 2 RS-232 Ports. 50-38.k baud
- F.I.F.O. communications
- On board diagnostics
- Low power consumption
- TurboDOS compatible
- **GUARANTEED COMPATIBLE WITH ALL S-100 SYSTEMS RUNNING TURBODOS**

INTRODUCTORY PRICE $495

Includes TurboDOS drivers (a $100 value) and TurboSlave I with 128k ram.

P.O. Box 8067, Fountain Valley, CA 92728
TELEX: 910 997 6120 EARTH FV

FOR MORE INFORMATION AND QUANTITY DISCOUNTS
CALL: (714) 964-5784

Registered trademarks: Z-80H, Zilog Inc.; TurboDOS Software 2000, Inc.

*** IBM PC VERSION COMING SOON ***

Poor Person Software
Introduces

Write-Hand-Man
Desk accessories for CP/M

Write-Hand-Man lets you take notes, check phone numbers, make appointments, and countless other tasks without leaving Wordstar, dBase, Multiplan, or any other application. Enter **Write-Hand-Man** with a single keystroke and choose the program you want. When you leave **Write-Hand-Man,** your application continues normally.

$49.95 plus tax delivers **Write-Hand-Man** and 4 companion programs; **Notepad, Phonebook, Calendar,** and **Termcomm.** User written programs are easily added. All you need is M80 or some other LINK-80 compatible assembler.

Other CP/M products available from **Poor Person Software: Poor Person's Spooler** ($49.95), **Poor Person's Spelling Checker** ($29.95), **Poor Person's Spread Sheet** ($29.95), **Keyed Sequential Files** ($39.95), **Poor Person's Menus** ($29.95), **aMAZEing Game** ($29.95), **Window System** ($29.95), **Crossword Game** ($39.95), **Mailing Label Processor** ($29.95). Shipping included.

All products available on IBM 8 inch and Northstar 5 inch disks. Other 5 inch formats add $5 handling charge. No credit cards.

Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
tel 415-493-3735

CP/M is a registered trademark of Digital Research

The UNIX File

by Ian F. Darwin

The UNIX File looks at many aspects of the UNIX operating system. If you have comments or questions about UNIX or this column, feel free to write to Ian Darwin at Box 603, Station F, Toronto, Ontario, Canada M4Y 2L8. If you have UNIX mail access to the UUCP network, you can contact me at "ihnp4!darwin!ian". I can't always answer immediately, but I will get back to you (electronic mail gets answered first!). And I'm always glad to hear from readers with comments either on the column itself or on their reactions to particular UNIX systems or products.

This issue's column discusses a name and address *database* written using only the UNIX shell, a UNIX word processing book, some comments on C programming style, and a pointer to some other UNIX articles.

MAILING LIST SYSTEM

I talk a lot about shell programming as a means to portability and ease of maintenance. Here's a mailing list system of moderate complexity, implemented without writing any C programs. If you have a commercial mailing list program (such as Leverage, reviewed in Microsystems, August 1984, page 122), the editing will be faster. My file format is intentionally compatible with Leverage and other UNIX-conscious programs using ASCII files to store the data.

All the files for the system are kept in a subdirectory called 'nad' for name and address data. The main file is called 'database' (for Leverage compatibility; otherwise any reasonable name would do). File DATABASE has one line per person or company, with a dozen or so fields separated by the vertical bar character. The inestimable advantage of this format, as opposed to any binary organization, is that most any UNIX program can handle it with ease. You can SORT, GREP, AWK or EDit the file anytime. Here's what my own record might look like:

```
Darwin|Ian|p|y|Box 603, Station F|
   Toronto|ON|M4Y 2L8|416-555-1212|
   416-555-1212|416-555-1212||
   unixfile|ihnp4!darwin!ian
```

That's all one line, but it's been broken up to make it fit in a narrow column. What are the fields, and what order to they appear in? I'm glad you asked, for I have in a file called LAYOUT an AWK-language description of the fields:

```
SNAME=0     +1 # surname
GNAME=SNAME+1 # given name
LIST =GNAME+1 # which mailing list?
XMAS =LIST +1 # does this slug
              merit a xmas card?
ADDR =XMAS +1 # number and street
CITY =ADDR +1
PROV =CITY +1 # state if U.S.A.
ZIP  =PROV +1
HFONE=ZIP  +1
BFONE=HFONE+1
DFONE=BFONE+1 # data phone (rare
              this year)
FFONE=DFONE+1 # family phone
              number if known
ENICN=FFONE+1 # e-mail nicname
EMAIL=ENICN+1 # e-mail address
```

That gives an AWK name to each column; SNAME for surname - column one, GNAME for given names - column two, and so on. The slightly-roundabout way of assigning the numbers is so that I can add or subtract a field without having to tediously renumber all the rest, an example of letting the computer do the walking.

How do I get the data into the computer? Naturally, there's another shell script. It's called 'bp', and here's how it looks in action:

```
% cd $HOME/nad
% bp
SNAME? Darwin
GNAME? Ian
LIST? p
XMAS? y
ADDR? Box 603, Station F
CITY? Toronto
PROV? ON
ZIP? M4Y 2L8
HFONE? 416-555-1212
BFONE? 416-555-1212
DFONE? 416-555-1212
FFONE?
ENICN? unixfile
EMAIL? ihnp4!darwin!ian
SNAME? ^q
%
```

This prompting, while not as sophisticated as a full-screen product like Leverage or dBASE, is adequate for low-volume work. Note that BP's main loop is built entirely of shell-programming primitives, and isn't very long:

```
for f in GNAME LIST XMAS ADDR ...
do
        echo -n "$f? "
        read $f
done
echo
```

You will recognize that the names here are the same ones in the file LAYOUT; in fact I just read the layout file into the editor and deleted everything except the names, then joined them together into a single (long) line (shown here as '...' due to column width). The keywords FOR, DO and DONE delimit a simple programming loop; the variable f is given the value GNAME, LIST, and so on. The variable is echoed (sans newline; if you have a System III/V ECHO command use "\c" instead of '-n'), then a response from the terminal is READ into the variable named by the variable f. So after the whole loop has been through, there are shell variables SNAME, GNAME, LIST and so on, each with some reasonable value. To add them to the file, we just ECHO them all onto the end of the file:

```
echo $SNAME"|"$GNAME"|"$LIST"|
      "$XMAS"|"$ADDR... >>$OUTFILE
```

The first part of BP contains some checking and some special handling for the first prompt. This allows me to flip into the editor just after finishing entering a record if I see I have made a mistake. Alternatively, I can use the shell escape mechanism to perform any other task that might occur to me while my fingers are tied up and my mind is free. Listing one shows what the entire BP program looks like.

I don't make any special provision for editing the data. I use a text editor to update the file; this is serviceable but not as flashy as LEVERAGE. (A challenge for the technical reader is to write a program that lets you update fields on some full-screen basis. It must read a description of what the fields are, rather than knowing that the E-mail nickname is field 11, and it must use termcap and curses to update the screen. This is probably a term-length programming project; do not undertake it lightly.)

Now what do I do with the data? I have so far developed programs that consult it for an online telephone directory, keeps my .MAILRC file (the place where several mail systems look for mail aliases) updated, and prints my Christmas Card list. Here's the phone program:

```
: phone - find name in the phone book
CASEOPT=
: -y in V7, broken in III/V, `fixed'
                    to -i in Vr2?
```

```
for person
do
    grep $CASEOPT $person ${PHONE
          BOOK-$HOME/nad/database}
done | awk -F'|' '
BEGIN   {
  SNAME = 0       +1      # surname
  #rest of file LAYOUT included here
  }
  {
  print $GNAME, $SNAME, "- Home:"
        ,$HFONE,"Work:", $BFONE
  }'
```

What's this thing with brace brackets in the GREP command? The Bourne shell (sh) allows you to provide defaults for shell variables; the variant shown here allows me to have a shell variable PHONEBOOK (presumably set in my .PROFILE file) that would override the value given here. The GREP command will look in my PHONEBOOK file if I have the shell variable PHONEBOOK set; by default it looks in file nad/database from my home directory. A group of users could share a common file such as /USR/ADM/PHONEBOOK by setting that as the default here. I used the AWK program to break up the records for printing. Use of the CUT utility on System V would be faster; if you have System V feel free to use that program. Not all my readers have System V, however, so I've chosen portability over efficiency. As Kernighan and Plauger say in their book "The Elements of Programming Style",

```
Make it right before you make
        it faster.
```

I think the version that I've shown here is right for the vast majority of UNIXes out there, although there surely are some for which it's not. For pure efficiency, try building a version using the shell built-in command SET. It's not as easy as you might think! Or write a C program, perhaps patterned after the System V CUT program; don't go to the trouble of writing a C program that only knows how to read one file format!

Now about this .MAILRC file. It is the place where the Berkeley MAIL system looks to see if you have any personal mail aliases. Since I keep in touch with many people via electronic mail, I use this facility, and my homegrown mailer uses it too. The format of the common entries is:

```
alias nickname fullpath
```

where nickname is the name you can remember, and fullpath is the mail path to get to them. For me, you might have:

```
alias unixfile ihnp4!darwin!ian
```

to get mail to me by saying:

```
Mail unixfile
```

```
: bp - database prompter
OUTFILE=btmp

case $# in
        0) ;;
        *) echo "Usage: bp" >&2; exit 1;;
esac

while true
do
echo -n "SNAME? "; read SNAME
case $SNAME in
        ~e)     ed $OUTFILE
                continue        # and go around main loop again
                ;;
        ~!*)    sh -c "`expr \"$SNAME\" : '..\(.*\)'`"
                continue;;
        ~q|"")  exit;;          # beam me up, Scotty!
esac

for f in GNAME LIST XMAS ADDR CITY PROV ZIP HFONE BFONE DFONE
     FFONE ENICN EMAIL
 do
        echo -n "$f? "
        read $f
done
echo
echo
$SNAME"|"$GNAME"|"$LIST"|"$XMAS"|"$ADDR"|"$CITY"|"$PROV"|"$ZIP"|"
  $HFONE"|"$BFONE"|"$DFONE"|"$FFONE"|"$ENICN"|"$EMAIL >>$OUTFILE
done
```

FIGURE 1- BP shell file

Looking at the file layout, you'll see that I have the nickname and full path fields stored in the mailing list data-base. Then I just have a simple AWK script - left as an exercise for the reader - to extract them, and print "alias" with the nickname and the path. This rewrites my .MAILRC file whenever I run it.

I've built a reasonably comprehensive mailing list system including provision for both hardcopy and electronic mail addressing. It's all done in the shell, although you might want to write a C program here and there. It should work on any UNIX system later than V6 or PWB.

WORD PROCESSING BOOK

Morris Krieger's book *Word Processing on the UNIX System* (1985, McGraw/BYTE, ISBN 0-07-035498-7) arrived on my desk recently. And it is good. The book has a brief introduction to UNIX, then A detailed discussion of ED, NROFF/TROFF, formatter macros, and TBL (but not EQN). The editor discussed is ED, not VI; this might seem odd at first but there's a plausible aside on why ED is the one true editor for word processing and why none of this VI or EMACS stuff need apply. The set of macros discussed is MS, not MM. While that might appear to be a liability for System V users, the book describes the principles behind macros, so that the user of any version of UNIX can not only understand formatter macros, but even write new ones, based on this material.

I can't give this book the review I'd like to, partly because it's not the main focus of this column and partly out of modesty - I was consulting with Mr. Krieger while he wrote it.

But I do recommend the book. Its 380 pages are substantially more detailed than all the AT&T formatting documents. Krieger has spent considerable time trying some of the less-well-documented formatter features to be able to explain what they do. The book contains many examples, and illuminates not a few obscure points. The book is just what you wanted when introducing people to the topic, or learning it yourself. It's here now. If you're doing text processing work on UNIX, get a copy.

FORMATTING C PROGRAMS

There is room for considerable variety of formatting style in most block-structured languages. In my humble opinion, however, there are some programs being published throughout the C/UNIX community that aren't attractively formatted.

Newcomers to the language often wish for guidance in this area, and a formatting program or "pretty printer" can help. There are several C formatting programs out there. To my mind, the "best" of the lot I've seen is CB with the -s and -j options. These options - unfortunately not in all versions of UNIX yet - format your program according to the style used by Kernighan and Ritchie in "The C Programming Language". Since this program works as a filter, all you have to say is:

```
cb -sj prog.c >prog.new
```

and look at prog.new. If you're happy with it, just:

```
mv prog.new prog.c
```

to replace the old version with the neatly formatted one. Look at it before replacing the original since any automated formatter can occasionally be driven berserk by certain constructs that its authors didn't anticipate.

If you are developing C code on UNIX, please use cb -sj on your programs before you publish them or post them to the net. A uniform coding style goes a long way towards portability of both programs and programmers.

While I'm on my soap box, I'll say my piece about block comments. Since most of us are no longer using punched cards, we no longer need such fancy block comments as

```
/*********************************/
/********* DO SOMETHING *********/
/*********************************/
```

When I see one of these keypunch wonders, I wonder how much programmer time was wasted making it. It's much easier on the eyes (and the printer) to do something like this:

```
/*
 * do something
 */
```

They both take the same amount of paper, but the latter looks better - to my eyes at least.

MISCELLANY

Back issues; Micro/Systems Journal has inherited some back numbers of the old Microsystems). The November issue, which Ziff-Davis did not send to most newstand distributors, contains several UNIX articles, including an interview with Andy Hall (in charge of System V development); a good article on C portability; a review of UNIX on the Codata 3300; and a UNIX File column discussing the make utility and a shell efficiency tip and file system reliability. Plus there's a nine-page article on the history of UNIX that I co-wrote with Geoff Collyer. If you're interested in UNIX, I (modestly) suggest you get a copy of that issue if they're still available.

That's all for now. Next month: another installment on joining the UNIX network, and how to get the source for a UNIX-like operating system subset. See you then.

REVOLUTIONIZE

YOUR S-100 SYSTEM TO 68000 POWER WITH OUR MULTIUSER HARDWARE/SOFTWARE PACKAGE THAT SUPPORTS 8 OR MORE USERS

*includes our 68000 Processor and Mirage Multi-User operating system.

For as little as $1295*

- Time Sharing
- Multi-tasking
- Multi-Directory Filing System
- Password Protected Login
- RAM Disk
- Device Independence
- Printer Spooling
- Basic, Fortran, Pascal, APL, and FORTH compilers available

CALL OR WRITE FOR DETAILS AND FOR OUR FREE S-100 CATALOG

Inner Access Corporation

Box 888 · Belmont, CA 94002 · 415/591-8295

DSD 80

FULL SCREEN SYMBOLIC DEBUGGER

"THE SINGLE BEST DEBUGGER FOR CP/M-80. A TRULY AMAZING PRODUCT."

LEOR ZOLMAN
AUTHOR OF BDS C

- ☐ Complete upward compatibility with DDT
- ☐ Simultaneous instruction, register, stack & memory displays
- ☐ Software In-Circuit-Emulator provides write protected memory, execute only code and stack protection.
- ☐ Full Z80 support with Intel or Zilog Mnemonics
- ☐ Thirty day money back guarantee
- ☐ On-line help & 50 page user manual

NOW ONLY $125.

SOFTADVANCES

P.O. BOX 49473 AUSTIN, TEXAS 78765 (512) 478-4763

MasterCard VISA

UNIX POWER

100,000 software developers can't be wrong.*

UNIX is the chosen operating system for more than 100,000 software developers because it has the power they need. But developers aren't the only people who need computing power. Any business that wants multi-users to access the same files at the same time or wants to simultaneously run multi-task operations . . . needs UNIX. At Dynacomp, we offer UniPlus + ® System V by UniSoft Corp. For $1495. U.S. dollars you can run UNIX on the CompuPro® 816/E™ . . . a powerful 68K S-100 bus computer system that maximizes its memory for multi-user/multi-task operations.

UniPlus + includes all the standard UNIX System V features PLUS performance enhancements found only in UniPlus + . These features increase the portability, flexibility, and performance of UNIX, allowing an affordable operating system for program development, text preparation, and general office use.

If it's time for you to upgrade to UNIX, call your local Full Service CompuPro System Center in the United States or call Dynacomp in Canada for complete details.

FROM

210 W. Broadway
Vancouver, B.C.
V5Y 3W2
(604) 872-7737

DYNACOMP COMPUTER SYSTEMS LTD.

46-6535 Mill Creek Dr.
Mississauga, Ont.
L5N 2M2
(416) 826-8002

*AT&T estimates that there are more than 100,000 people currently developing software under UNIX. Dynacomp serves all of Canada and parts of Asia and the Pacific Rim. Call us for details and information on our full product line including Plexus. • UNIX is a trademark of Bell Laboratories, Inc. CompuPro is a registered trademark and System 816/E is a trademark of Viasyn Corp. UniPlus + is a registered trademark of Unisoft Corp. AT&T is a registered trademark of AT&T Information Systems.

TURBO PASCAL 3.0

continued from page 73

CONCLUSIONS

All in all, Turbo 3.0 is certainly still a good deal. As an applications development language it is without equal for ease-of-use, price, speed, and power. Turbo 3.0's added speed and features make it well worth the $30.00 upgrade cost for existing Turbo 2.0 owners (especially for IBM PC-DOS users). It is unfortunate that the long integer type is still not supported, but the BCD package can make up for this limitation. The new manual still has a few errors, but most are noted in a READ.ME file on the program disk.

The Turbo 3.0 compiler from Borland International still represents a value without equal in high quality, low cost microcomputer software.

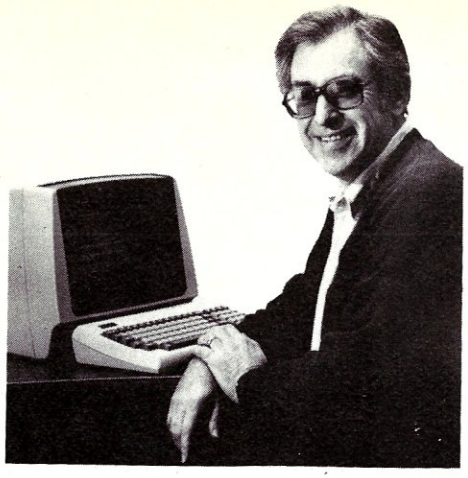
David W. Carroll is a freelance writer and computer consultant living in the Sierra Nevada foothills near Sacramento, California. He is the author of "Telecommunications with the IBM PCjr" co-published by Microtext/Prentice Hall and "Programming with Turbo Pascal" to be co-published by Microtext and McGraw Hill this summer.

Products for the Expanding Universe

NEW GENERATION SYSTEMS

1800 Michael Faraday Drive, Suite 206, Reston, VA 22090
(703) 471-5598 Order Line (800) 368-3359
Telex: 750417 NEW GEN SYS UD.

		Price
MicroShell	UNIX-like Commands for CP/M Fast command (submit) files, I/O redirection, command line editing, auto disk and user searches. (Usable under CCP/M). [CP/M-80]	125.00
Tool Pax	UNIX-like Utilities Sorting, searching, print formatting, file differences, vertical cut/paste, etc. [CP/M-80, 83; MP/M-816; CCP/M-816]	150.00
MicroTree	Named Directory Areas Hierarchical file directories and utilities. Permits named directory areas under CP/M, MP/M 816 with: change/make/list/remove directories, rename, copy and show directory tree. [CP/M-80, 83; CP/M-86; MP/M-816; CCP/M-816; CP/M-68K]	99.00
CPRO-UTIL	Multiuser Utilities Schedule and time commands, printer setup, disk utility (DU-like: unerase, view, change, etc.), write to terminals, electronic mail and others for CompuPro Concurrent CP/M systems.	99.00
TexSys	Text Control System Multiple versions of text files—source code, letters, spread sheets, contracts, any text. Keeps record of changes allowing any version to be recovered. Twenty revisions normally only require three times the disk storage space. [CP/M-80, 83; MP/M 816; CCP/M-816; PCDOS/MSDOS, Macintosh, Lisa]	195.00
Bibliofile	Bibliographic Data Base Maintain, search and format bibliographies. Compatible with most word processors. [CP/M-80, 83; MP/M & CCP/M-816; PCDOS/MSDOS]	250.00
DSD	Dynamic Screen Debugger Symbolic, full screen program debugger for assembly languge, BDS C, etc. programs. Dynamically displays instructions, registers, memory, and stack areas. 8080, 8085 and Z80 compatible. [CP/M-80, 83; MP/M-816; CCP/M-816]	125.00
BRX-80/86	Fast Hard Disk Backup (16 bit version 30-100% faster than PIP), handles random files, files larger than a floppy disk, backup only changed files (86 version only), selective restore. [CP/M-80, 83; MP/M-816; CCP/M-816]	195.00
Intro. C	Learn C on your PC Complete, comprehensive introductory course in C with a C interpreter to run your programs as you write and change them. [PCDOS 2.0]	95.00
CI C86	Full 16-bit C Compiler Computer Innovations' fast, optimized, 16-bit full C compiler. With source code for all library functions, 8087, small/big model support, comprehensive manual. We use it for all 16-bit programs; it's great!	349.00
BDS C	The Original—Still Fastest! Ultrafast C compiler for CP/M-80 with symbolic debugger, full library and run time source code. Large public domain library of programs available.	140.00
AMCALL	Multiuser Modem Program Both AMCALL and MCALL-II specially configured for CompuPro 816, CompuPro 10, or Gifford MP/M, CCP/M or MC-DOS Multiuser systems. Supports 7 file transfer protocols with the Hayes 300/1200 Smart-Modem. Baud rates to 19,200. Single-user AMCALL 150.00 / MCALL-II 125.00	300.00
Disk Maker	A Disk Maker System for Your Needs Moves programs, data, any files among over 200 different floppy disk formats. Can format, read, write and duplicate 3½", 48/96 tpi 5¼", IBM PCAT, and 8" diskettes (with appropriate drive options). Available options include 10, 15, 20Mb hard disks, 8086 Coprocessor, Word Processing/Typesetting	



Sol Libes says ...

"Call me crazy ... I'm doing it again!"

I'm back into magazine publishing ... something I swore I would never do again. When Microsystems magazine died I was in a depressed state for weeks. Countless letters and phone calls from devoted subscribers made it even worse. Everyone kept urging me to do it again.

I kept remembering what my wife Lennie and I went through when we started Microsystems, and I said no ... not again ... we want to live a normal life again. But there has been something missing from my life the last several months. The passing of Microsystems left a void. Let's face it, there really is no other magazine that caters to the advanced micro user the way Microsystems did.

*So I am starting a new magazine. It is in the tradition of the old Microsystems. Lots of practical info ... strictly technical ... no fluff ... stuff to keep every hacker up-to-date on the ever-changing micro technology ... software and hardware tutorials and reviews, public-domain software info and reviews (SIG/M, PC/Blue, PC-SIG, C-User Group and more) ... MS/DOS, CP/M, Turbo DOS, C, Pascal, Forth, Lisp, and of course Assembler ... S-100, IBM-PC, single board computers, multi-user systems ... a real micro systems-oriented journal ... in fact, that is its name—**Micro/Systems Journal!***

For the Advanced Computer User

Micro/Systems Journal™

Subscription Form

☐ **Yes! Start my subscription!**

Name:_____

Address:_____

_____

City:_____ State:_____ Zip:_____

☐ 1 Year	$18.00
☐ 2 Years	32.00
☐ 1 Year (Canada & Mexico)	24.00
☐ 2 Years (Canada & Mexico)	44.00
☐ 1 Year (other foreign)	32.00
☐ 2 Years (other foreign)	58.00

Please make check payable in U.S. funds by a U.S. bank.
Thank you.

published six (6) times a year

Make check payable to: Micro/Systems Journal, P.O. Box 1192, Mountainside, NJ 07092

SUBSCRIPTION EXPIRING?

The code in upper right corner of your mailing label indicates your last issue. To renew check box below, attach your mailing label below, enclose your check (deduct 10%) and mail it to address shown below.

			Canada & Mexico	Foreign
☐ Renewal	☐ 1 year	$18	$24	$32
☐ New Subscription	☐ 2 years	$32	$44	$58

MOVING?

Remove mailing label from magazine and attach it here, then print your name and *new* address in section below. Send it 8 weeks before you move.

Name: _____

Company: _____

Address: _____

City: _____ State: _____ Zip: _____

Micro/Systems Journal *is published six (6) times a year.*

PC/BLUE SOFTWARE DIRECTORY

A listing of the contents of the PC/BLUE public domain software library. Includes a cross-referenced index by subject; 30 pages. Price $4 U.S., Canada and Mexico; $6 elsewhere.

Micro/Systems Journal™ Back Issues

The following back issues are available. They are $4 per copy ($6 foreign, cash, Postal Order or U.S. bank check) and include shipping. If ordering 3-9 copies deduct 10%, 10 or more copies deduct 15%. Orders can be combined with back issues or old Microsystems for discount. Send to: M/SJ, Box 1192, Mountainside NJ 07092.

MARCH/APRIL 1985 (Vol 1., No. 1): Bringing Up CP/M Plus, Assembly Language Extensions for MS-Basic, New Tricks for CP/M2.2, Building an IBM/PC or XT Clone, Extended Single Density Storage, Variable Size Arrays in C;REVIEWS: dBase-III and 16-Bit Lisp & ProLog-Part I.

MAY/JUNE 1985 (Vol 1., No. 2): Build an S-100 to PC-Bus Converter, Interfacing to MS-DOS Part-1, Loadable Drivers for CP/M2.2, Roll Your Own PC-Clone, Bringing Up ZCPR-3, C & Godbout Disk-1 Controller, Writing Translation Programs in C and Turbo Pascal; REVIEWS: 16-Bit Lisp & ProLog-Part II.

Microsystems Back Issues

The following back issues of the old Microsystems magazine are still available. Quantities of many of the issues are, however, limited. They are $3.50 per copy ($5.00 foreign, cash or U.S. bank check) including shipping. If ordering 3-9 copies deduct 10%, 10 or more copies deduct 15%. Make check out to "Micro/Systems Journal", Box 1192, Mountainside NJ 07092.

1984

AUGUST: Intro to Local Area Networking; Graphics Subroutines in C For NAPLPS; Using YACC, MAKE and Prolog under Unix; Multiprocessing on S-100; Using Unix Sort, ciphers and enhancements; REVIEWS: TurboDOS, NCR-PC, MindSet-PC, Adding TurboDOS to NorthStar System, Leverage DBMS for Unix.

APRIL: Unix Software Directory; Upgrade NorthStar ZPB; MS-DOS 2.0 Overview - Part 2; S-100 Phantom & Bank Selecting; Upgrading FIG Forth; REVIEWS: UniPlus+, Informix, DRI-C.

FEBRUARY: Using WordStar to Create Mailmerge/DBase-II files; Moving data files between CP/M software packages; Datestamp DBase-II; CP/M 2.2 Deblocking; Building S-100 diagnostic hardware; Enhance CP/M+ with RSX; REVIEWS: DBase-II, S-100 Mainframes, DRI Display Manager, AutoDex, Turbo Pascal.

JANUARY: Enhancing MP/M - Part 1; Installing MP/M; Add Concurrency to MP/M; Two Users on CP/M; Relocating Assemblers & Linkage Editors - Part 3; S-100 Wait States; REVIEWS: MP/M-8/16, ProComp-8, Paragraphics Game Board, ProLog.

1983

DECEMBER: CP/M Software Directory; A Debug Subroutine; Implement IOBYTE on North Star; Floppy Disk Problems; Improve Trig Functions in CBasic-80; Build Cheap S-100 Memory; Extended Memory Management; CP/M-86 BDOS Calls; REVIEWS: XLISP, LISP/80, TLC LISP, APC Basic, Microdynamics S-100 EProm Programmer, Ackerman S-100 Digital Synthetalker, Digital Research 16K & 32K S-100 Memory cards.

NOVEMBER: Intro to 80286, 68000, and 16032 Microprocessors; Intro to Local Area Networks - Part 2; Extended Memory Management for older S-100 Systems; Notes on Microsoft Fortran-80; Building S-100 Parallel Ports; REVIEWS: CompuPro CPU-68K, System 8/16, Xenith Z-100, Nevada & Ellis Computing Fortran.

OCTOBER: Intro to Local Area Networks, Part-1; Build Low-Cost LAN; Build S-100 Bubble Memory Card; Use Radio Shack Model 100 portable with a CP/M system; Write Menu-Driven Utility for Setting Printer Options; North Star Improvement; True Z-80 Random Number Function; Hide Code in Basic REM statements; Machine Code loader for MBasic; Increase Single-Density Disk Formatting; Relocating Assembler & Linkage Editors, Part-2; Run MX-80 with North Star; User Group Directory; CP/M-86 Versus CP/M-80; REVIEWS: CP/NET, QBAX, S-Basic.

SEPTEMBER: Using RatFor; Relocating Assemblers & Linkage Editors, Part-1; Sleuth WordStar Files with Pascal; CrossCheck Program; CP/M<->NorthStar File Transfers; NorthStar DOS as a CP/M COM File; Add Rescue Key to System; S-100 TMA Interfacing; REVIEWS: Altos 586, CompuPro 8/16C, Ithaca InterSystems Encore, Dual Systems 83/20 Unix System, Supersoft C, Software Tools, Morrow Designs Micronix, Upgrade Older S-100 Systems to CompuPro Dual Processor.

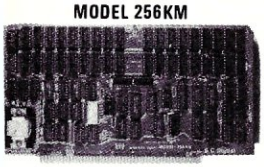
AUGUST: XERA Program; Logging-On CP/M; WordStar Date/Time Patch; Find Location of Variable in NorthStar Basic; Prevent System Crashes During Warm Boot; Enhance Spreadsheet Print Files; Plotting Package-Part 3; Run WordStar under TP/M; 50-line Text Formatter; Using the LU Utility; User Areas under CP/M; REVIEWS: Stiff Upper Lisp, MuLisp-80, Supersoft Lisp, Cromenco C-10, Access Manager, Fancy Font, Computime SBC-880 S-100 card.

JULY: Using RCPMs; RCPM Directory; PIP Data Between Computers; Toward Smarter Modem Programs; Interface MX-80 via Parallel Interface; Digital Audio On CP/M System; Customize CP/M CBIOS; Plotting Package Part-2; REVIEWS: DRI PL/I-86 and PL/I-80, S-100 PMMI MM-VT1.

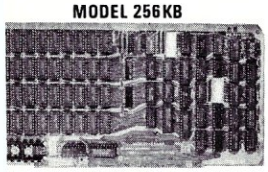
JUNE: Plotting Package Part 1; Drive HP Plotter; Laboratory Graphics Applications; Console Keypressed interrupts; Cutomize Wordprocessor Keyboard; WordStar Patch hor H-19/Z-19 Terminal; Relocatable Code; REVIEWS: Graftalk, JES S-100 Graphics Controller, ZCPR2.

For S100 bus by S. C. Digital, Inc.

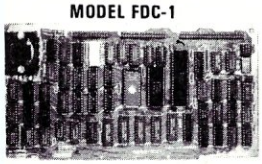
MODEL 256KM



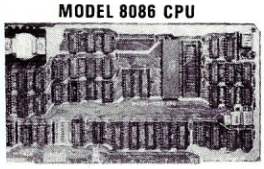
MODEL 256KB



MODEL FDC-1



MODEL 8086 CPU



256 K/1M DYNAMIC RAM Board Model 256KM $795
● 256K/1M bytes using 64K or 256K DRAMs ● 8/16b data ● 24b Address ● Parity per byte ● 175 nsec access time ● will run Z80/Z8000 to 6 mhz, 8086, 80186, 68000 to 8 mhz without wait states ● transparent refresh, unlimitted DMA ● with 1MB using 256K drams.

256K/1M DYNAMIC RAM BOARD Model 256KB-256 $459
● 256K/1M bytes using 64K or 256K DRAMs ● 8b data ● 16 or 24b address ● parity per byte ● Memory mapping in 16K blocks ● 175 nsec access time ● Addressable in 128k, 192k, or 256k boundaries, compatible with Z-100* systems ● with 256KB using 256K drams, expandable to 1 mega byte, less memory mapping ● add $20 for memory mapping.

FLOPPY DISK CONTROLLER Board Model FDC-1 $375
● Single or double sensity, sides, in any combination of up to four 8" or 5¼" drives ● Digital phase-locked loop ● DMA data transfer with cross 64K bountaries, 24b address, DMA arbitration ● built in monitor/boot EPROM that accomodates two different processors ● serial port to 19.2Kbaud ● uses 765A/8272 ● with CPM bios programs

8086 CPU Board Model 8086 CPU $375
● 8/4 mhz SW selectable ● 8087 interface ● provision to fun two processors on a bus such as our Z80 CPU ● convertable to 10, 12 mhz clock ● optimized for DRAM boards.

80286 CPU Board Model 80286 from $350
● 8/4 mhz switchable ● 80827 interface ● provision to run two processors on a bus ● convertable to 8 mhz ● separate built in colck for 80287 ● optimized for DRAM boards.

SUPPORT Board Model Support-1 $395
● 4 serial, full handshakes, two with software programmable baud rates ● Centronics ● SASI interface ● Real/interval times ● Calendar-clock with battery backup ● expandable interrupt controllers for 8086 or 8080/Z80 ● CPU switching to run 2 processors on a bus such as our 8086 or 80286 and Z80 CPU boards.

I/O Interface board Model 3SPC-N $229
● 3 serial RS-232C with switch selectable baud rates, 110 to 19.2kbaud.
● 1 parallel.

Board Sets: 8086, 256KM (with 512KB), FDC-1, 3SPC-N $1388
Z80, 256KB-256, FDC-1, 3spc-N $1150

S-100 COMPUTER 'System 16' $3200
8086 based at 8 mhz, with 512 kb, 5 serial ports, 1 centronics, 1 SASI, battery backed calendar clock, real time clock, interrupt driven, 10 slot card cage, two 5.25" floppy drives with 500 kb transfer rates and 1.2 mbyte storage each, with CPM 86 operating system (Concurrent Dos available soon). Cabinet has room for full size 5¼" hard disk.

Operating Systems available: CPM 2.2, CPM 3.0, CPM 86, MSDOS.

*CPM is registered trade mark of Digital Research Inc. Z-100 is registered trade mark of Zeith Corporation.

Please call for latest prices.
Prices subject to change without notice.

S.C. DIGITAL, INC.
1240 N. Highland Ave., Suite 4 ● P.O. Box 906, Aurora, Illinois 60507
Phone: (312) 897-7749

Microsystems Back Issues *continued*

JANUARY: Unix Vs CP/M; Intro to Xenix; Unix on Micros; Build S-100 DMA Adaptor; Interfacing to BSR X-10 Home Control System; S-100 Troublehooting; REVIEWS: InterSystems DPS-8000 and Coherent; MicroShell, UNICA, Small-VOS, Small-Tools, Five S-100 RAM Cards, SemiDisk.

1982

NOVEMBER/DECEMBER: CP/M Vs MS/DOS; CP/M-86 Vs MS-DOS; Intro to ADA Part 2; Virtual Disk for NorthStar; CP/M Program Auto-execute; Macros & Macro-Assemblers; REVIEWS: Janus, Aztec-C, C/80, Morrow S-100 M26 Hard Disk System, Teleram S-100 Bubble Memory Card, Jade S-100 Bus Probe.

JULY/AUGUST: Hardware Random Byte Generator; Error Detection & Correction Codes; Getfile CP/M Utility Program; CP/M Patches; CP/M Application Notes; Run old NorthStar programs under new DOS; Cloning Disk Drives; Low Cost Floppy Disk Power Supply; Intro to Computer Graphics; Using Supersub Utility; REVIEWS: D80, RAID-8080, Three Macro-Assemblers, PDS, Cer-Tek S-100 UniProm Board; GrafPak.

MAY/JUNE: Intro to DBMS; Three ways to implement a mail list; Cursor Addressing; Structured Programming in Basic; Replacement for CP/M Submit; CP/M Disk Directory & Table Secrets; Mods for SDS VDB-8024; Run NorthStar Basic with CP/M; REVIEWS: DataStar, MDBS, TIM, Mince, ZDM.

1981

NOVEMBER/DECEMBER: Introduction to the C Language, Virtual Segment Procedures, Little-Ada Part II, A Disk Alignment Routine, Northstar DOS/BIOS Directory & File Conversion using UCSD Pascal; REVIEWS: BDS-C, Small-C, Tiny-C and Whitesmiths-C; Tarbell Double-Density Disk Controller

JULY/AUGUST: 16-Bit Disk Operating Systems; Input Queuing For NorthStar; Variable Speed Automatic Slow Step; Build S-100 Clock/Calendar Card; REVIEWS: TEC-86 System, Seattle Computer 8086 System, AlphaMicro, Godbout Dual Processor, CP/M-86, Televideo 920-C Terminal.

1980

MAR/APRIL: Linear Programming Techniques in Pascal; Intro To CP/M part 2, Addressing The Cursor; S-100 Bus - New Vs Old; Tarbell Disk Controller Mods; REVIEWS: CGS-808 S-100 Color Graphics Controller.

THE BOOK MART

The following selected books can be ordered thru Micro/Systems Journal. We furnish these titles only because we feel they are very worthwhile publications.

Interfacing to S-100/IEEE-696 Microcomputers by Sol Libes and Mark Garetz. The definitive book on the subject. Also has great application to hardware interfacing in general.
$17.95 (US. Can. Mex)
$27.95 (other foreign)

The 8086 Book by Russell Rector & George Alexy. Probably the most comprehensive reference book on both software and hardware for the 8086 and 8088. Packed with lots of program and circuit examples.
$19.95 (US. Can. Mex)
$29.95 (other foreign)

68000 Assembly Language Programming by Gerry Kane, Doug Hawkins & Lance Leventhal. A reference and how-to book with a wealth of fully debugged programming examples. Includes assembler conventions, I/O device programming and interfacing.
$19.95 (US. Can. Mex)
$29.95 (other foreign)

The Programmer's CP/M Handbook by Andy Johnson-Laird. A comprehensive reference on CP/M's internals, file system, CCP, BDOS BIOS, etc. Examples of CP/M customizing and utilities written in C are given.
$22.95 (US. Can. Mex)
$32.95 (other foreign)

The following books, by Dave Cortesi, are among the best books on CP/M and contain many programming examples. They contain a great deal of info on installation, use of command sequences, file handling, directories, etc. And comprehensive reference sections are also included.

Inside CP/M" $27.50 (US. Can. Mex) $37.50 (other foreign)

Inside CP/M-Plus $19.95 (US. Can. Mex) $29.95 (other foreign)

Inside CP/M-86" $18.95 (US. Can. Mex) $28.95 (other foreign)

Inside Concurrent CP/M $18.95 (US. Can. Mex) $28.95 (other foreign)

Send check for full amount (NJ residents include sales tax). Shipping is included in price. Professional books may be tax deductible. All checks must be payable in U.S. funds by a U.S. bank. If billed to a firm or institution add $5.00 and include a Purchase Order Number. All books will be shipped within 2 weeks of receipt of order.

Mail to:
Micro/System Journal
Box 1192
Mountainside NJ 07092

Advertiser Index

2500AD Software ..... 44 & 45
Amanuensis 25
AndraTech 50
Applied Innovations 61
Austin Code Works 54
Autosoft 58

BDSoftware 11
BG Computer Applications 53
Business Utility Software 71
BV Engineering 64

Competitive Edge 16
CompuPro (Viasyn) 1
Computer House 61
Computime 37
Computing! 47
C Users' Group 51

Data Access 6
Dynacomp Computer Systems 84

Earth Computer 80
Echelon 9
Electronic Control Technology 17

First Capitol Computer 15
Fulcrum Computer Products 5

GSR Computers 20

Inner Access 83
Integral Systems 47
Integrand ..... 64 & 65
Intercontinental Micro Systems C3
I/O Technology 7

John D. Owens Associates 47

KC Systems 32

Lahey Computer Systems 61
Lattice 29
Lomas Data 52

Macrotech International C2
Martian Technologies 53
Mendocino Software 50
Microcomputer Systems Consultants 37
Micro Methods 53
MicroSystems Journal 86
Micro/Systems Books 88
Micro/Systems Back Issues ..... 86 & 87
Mix Software 55
Morgan Computing 53
Motel Computers Ltd. 21

New Generation Systems 84
Night Owl ..... 23 & 60

Para Dynamics 65

Peak Electronics 50
Performics 24
Poor Person Software 80
Programmer's Shop, The 25
Public Domain Software 54

Rational Systems 30
R & L MicroServices 51

S-100 39
S.C. Digital 87
Semidisk Systems C4
ServoComputer 77
Shaw American Technologies 54
Simpliway Products 36
Slicer Computers 58
SLR Systems 63
Soft Advances 83
Softaid 50
Softfocus 37
Software Innovations 54
Software Wizardry 15
System Engineering Tools 76
Systems Guild 13

Teletek 2

WB Software Development 54
Western Wares 37

Viasyn (CompuPro) 1

They Called It Impossible.

We Call It TurboLAN.™

Imagine IBM PCs, intelligent workstations and S-100 BUS multiuser systems all on the same local area network.

Impossible?

Not with TurboLAN, Intercontinental Micro's networking solution.

The key is a unique hardware and software approach to local area networks.

You get the flexibility of 8-bit and 16-bit processors on the same network, the cost savings of shared peripherals, and expandability to over 4000 users.

You simply organize the network the way you want, mixing IBM PCs, S-100 BUS multi-user systems, S-100 BUS file servers, and workstations.

Intercontinental's sophisticated single board computers, 8-bit and 16-bit slaves, memory and interface boards make your job even easier. TurboLAN will work with almost any S-100 BUS system including NORTHSTAR, IMS, Teletek, and any other S-100 master featuring phantom deselection or extended addressing capability.

The ability to run PC-DOS, MS-DOS, CP/M, MP/M, and other applications programs means whatever the future holds, TurboLAN is ready.

And so is Intercontinental Micro.

You see, our S-100 BUS products revolutionized an industry. Intercontinental's slaves, masters and memory boards introduced advanced features like Direct Memory Access and Memory Management to micro environments. TurboLAN.™ The new networking solution from the company that's been building networking solutions for years.

Still think TurboLAN's impossible?

Then call, write or circle the bingo number below—we'll send you complete information on TurboLAN and descriptions of all our products, including the most complete S-100 BUS product line in the business.

OUR COMPLETE TURBOLAN AND S-100 BUS PRODUCT LINE

CPS4800X-SINGLE BOARD COMPUTER.
4 or 6 MHz processor (Z80A/B), onboard floppy disk controller, 64K RAM, 4 channel DMA controller, 24 line parallel I/O port, two serial I/O channels, real time clock. Memory mapped or I/O mapped capability.

CPS-16X—16-BIT 8086 SLAVE PROCESSOR.
256K, 512K or 1MB RAM, bank selectable, memory mapped, two serial ports, 24 line parallel I/O port, 8 MHz.

CPS-BMX/MX—Z80 SLAVE PROCESSORS
4 to 6 MHz processors, 64K RAM or 128K bank selectable memory, two serial ports, 24 line parallel I/O port.

256KMB—MEMORY BOARD.
Hard disk cache, linear addressable to two megabytes, bank selectable in 16K increments, configures for phantom deselection, parity error detection.

LANS-100—TURBOLAN FOR S-100 BUS SYSTEMS
ARCNET controller meets 696.2/D2S-100 spec, coax cable interface, 255 nodes per network segment, 2.5 megabit/sec. data rate.

LANPC TURBOLAN FOR THE IBM PC
Plug-in expansion ArcNet board. Custom software drivers integrates IBM PC into TurboLAN networks. 64K or 256K RAM options available.

WS80X-DISKLESS WORKSTATION
Converts almost any dumb terminal into intelligent workstation with networking capability. Floppy and hard disk options available.

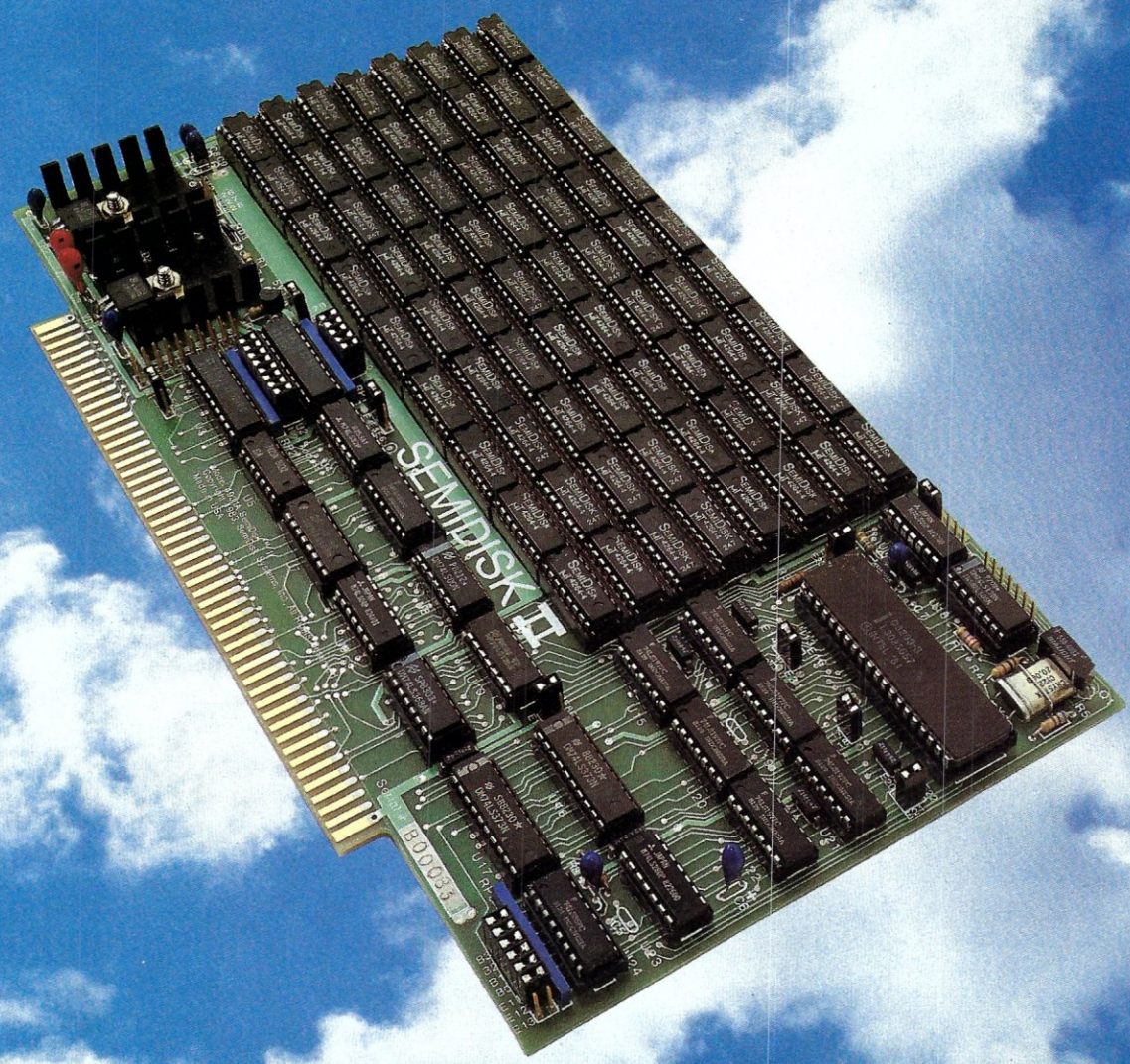
PERSONALITY BOARDS—
SASI, Centronix, PRIAM, Clock/Calendar, RS232, Modem, RS422, long distance serial communications (up to 4000 Ft.)

Intercontinental Micro Systems

4015 Leaverton Ct., Anaheim, CA 92807, (714) 630-0964, TELEX: 821375 SUPPORT UD

TurboDOS is a Trademark of Software 2,000, Inc. ARCnet is a Trademark of Datapoint. CP/M, CP/M 86, MP/M, MP/M 86, CP/M PLUS, Concurrent CP/M 86 are Trademarks of Digital Research. PC-DOS, IBM-PC are Trademarks of International Business Machines. MS-DOS is a Trademark of Microsoft. TurboLAN is a Trademark of Intercontinental Micro Systems.

2 Megabytes



THE LARGEST CAPACITY DISK EMULATOR YOU HAVE EVER SEEN.

You know about disk emulators. They're fast semiconductor disk drives. Very fast. But until now, the most disk storage you could get on a single board was 1Mbyte. (That was from us, too.) Now we have news that'll really blow your socks off... 2 Megabytes on a single board. Available NOW. That's not a pie-in-the-sky promise.

That's enough storage for dozens of large programs and hundreds of kilobytes of data files. Enough for almost anything you want to do with a disk drive. But that's not all. With SemiSpool, our CP/M print spooler, you can implement a print buffer hundreds of kilobytes long in seconds. All in software. At no extra cost.

Another thing about disk emulators. Unless they're from SemiDisk Systems, they're probably afraid of the dark: Lose power or turn the computer off, and your valuable data goes to that big backup disk in the sky. But our Battery Backup Units keep SemiDisk data flying high while your computer is off, and up to 10 hours during a complete blackout.

So remember this: SemiDisk Systems has been building dedicated microcomputer disk emulators longer than anyone. And larger. And faster. And at a much lower cost. And that's not a lot of hot air.

AT A PRICE YOU NEVER THOUGHT YOU'D SEE

	512K	1Mbyte	2Mbyte
SemiDisk I, S-100	$995	$1795	
SemiDisk II, S-100	$1295	$2095	$2549
IBM PC, XT, AT	$945	$1795	$2499
QX-10,QX-16	$799		$2499
TRS-80 II,12,16	$995	$1795	$2499
Battery Backup Unit	$150		

SEMIDISK

SemiDisk Systems, Inc.
P.O. Box GG, Beaverton, Oregon 97075
503-642-3100

Call 503-646-5510 for CBBS/NW, 503-775-4838 for CBBS/PCS, and 503 649-8327 for CBBS/Aloha, all SemiDisk-equipped computer bulletin boards (300/1200 baud). SemiDisk, SemiSpool trademarks of SemiDisk Systems. CP/M trademark of Digital Research.