THE CP/M USER'S JOURNAL

$2.95

Microsystems

Volume 4 / Number 6

June 1983

Graphics for More Microcomputers

David Freese provides a substantial plotting package of subroutines to drive a graphics printer. Steve Leibson reviews the Hewlett-Packard Graphic Language as a means of communicating with a wide range of plotters. Kalle Gehring and John Moore, of Eastern Michigan University, show how graphics capability can speed data collection and analysis in a chemical laboratory. Joseph Long reviews GrafTalk, a business-oriented package for the production of pie, bar, and line plot graphics.

Plotting isn't always subversive! Hitch your micro to a plotter or graphics printer

Hardware Review

Tom Ceska reviews the J.E.S. Super Compuprism graphics board.

System Enhancements

Richard Conn describes ZCPR2, his powerful and flexible replacement for the CP/M command processor. John Potochnak describes a type-ahead buffer that prevents loss of characters during disk I/O. Bradford Thompson shows how to customize your keyboard to generate the codes needed by your word processor.

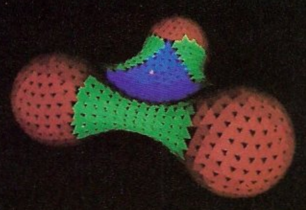
WordStar and the Heath/Zenith H-19 Terminal

Norman Dresner and Bill Machrone show how to modify WordStar to use all the power of your H-19 keyboard.

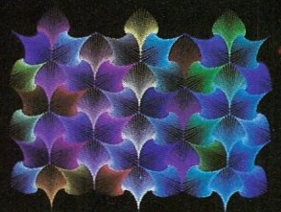
06

0 14024 14044

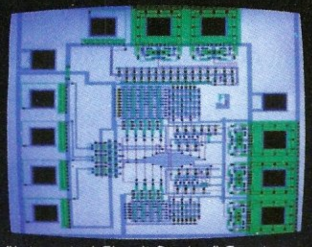
SUPERIOR GRAPHICS HAVE COME DOWN TO EARTH.



"Three Atoms" Courtesy of Greg Abram, University of North Carolina at Chapel Hill



"Aurora" By Richard Katz, Vectrix Corporation



"Integrated Circuit Design" Courtesy of Floyd J. James, University of North Carolina at Chapel Hill

"In The Beginning" By Richard Katz, Vectrix Corporation

$1995 AND THE FIRST AFFORDABLE HIGH RESOLUTION COLOR GRAPHICS MACHINE IS YOURS

VX128

- **VERY HIGH RESOLUTION** 672 by 480 pixels individually addressable
- **EIGHT COLORS PER PIXEL** 3 bit planes of memory totalling 128K graphics RAM
- **ON-BOARD 16 BIT MICRO-COMPUTER** Intel 8088 microprocessor with additional PROM and RAM and built-in expansion capability
- **3D GRAPHICS SOFTWARE PACKAGE** built-in command set includes: rotation, scaling, translation, perspective, clipping, viewport, polygon, and filled polygon
- **HARDWARE LINE AND ARC GENERATION** on-board VLSI graphics display controller, 1600 nanoseconds pixel drawing time
- **USER DEFINABLE CHARACTER GENERATION** built-in character set includes zoom, slant, and variable spacing, or upload your own character definitions

VECTRIX

- **SERIAL AND PARALLEL INTERFACE** 300-19.2K baud and 8 bit parallel port
- **USER FRIENDLY COMMAND FORMAT** supports high level language and hexadecimal transmissions

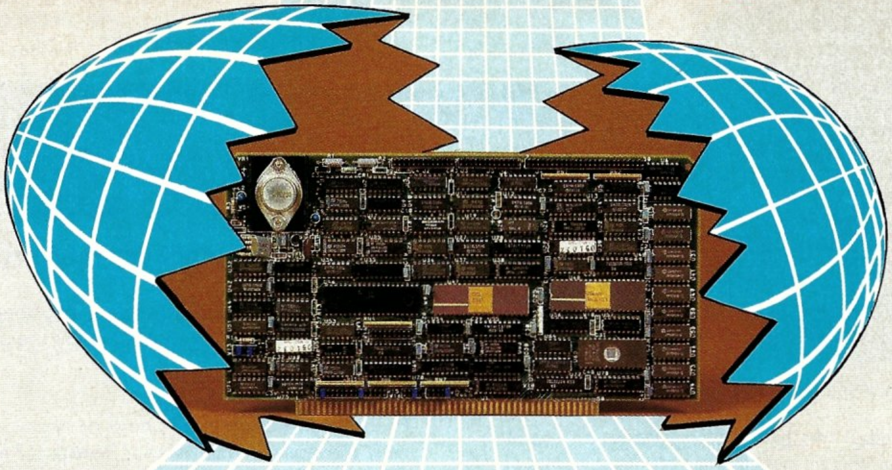
VX384

- **512 COLORS PER PIXEL** 9 bit planes of memory with 384K graphics RAM
- **COLOR LOOKUP TABLE** 8 bit digital-to-analog converters provide a 16 million color palate
- **INCLUDES ALL FEATURES** of VX128 for total of $3995
- **VXM HIGH RESOLUTION COLOR MONITOR** RGB analog input with 24 kiloHertz scan rate, long persistence phosphor $1295
- **COLOR GRAPHIC PRINTER** with interface cable $1295

For additional information on VX128, VX384, VXM Monitor or VXP Printer call Toll Free 1-800-334-8181, or 919-272-3479, or write Vectrix Corporation, 700 Battleground Avenue, Greensboro, NC 27401

CIRCLE 160 ON READER SERVICE CARD

Incubation Complete



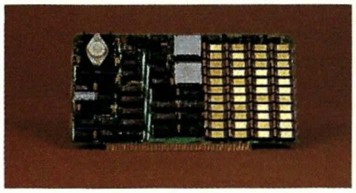
A Third Generation is Born

SBC 300

A Z80 based microcomputer board with memory and I/O functions

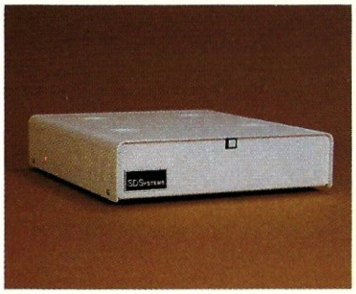
- ***Fully complies with IEEE 696 Standard***
- 4/6/8 MHz Z80 A/B/H*
- Supports CP/M® Plus
- Operates as bus master/slave for multi-user, multi-processor architecture.
- 64K on board memory, dual ported, parity checked
- 2 serial ports, 1 full SASI port
- All I/O drivers on board
- Memory management
- Full 24 bit address capability
- 3-16 bit CTC's

(Pictured above)



ExpandoRAM IV—Random access memory board utilizing 64K or 256K NMOS RAM chips

- ***Fully complies with IEEE 696 Standard***
- 256K capacity with 64K chips
- 1024K capacity with 256K chips
- Error checking and correction (2 bit detection, 1 bit correction)
- On board refresh
- Supports both 8 and 16 bit data transfers
- 24 bit addressing



SD300—A new series of compact yet expandable S-100 microcomputers.

- Compact size approximately 4" × 14" × 17"
- 6 Slot motherboard
- Rugged metal enclosure
- Supports up to 5 users

OEM Version: Designed for ease of integration and maximum flexibility

- Z80 CPU
- 256K RAM
- Versafloppy II with free CP/M Plus™

Discless Version: An ideal high performance system for disk intensive applications. Eliminates disk wait states for spread sheets, spelling checkers, and network operation. Utilizes SDSystems RAMDisc and ROMDisc modules.

VFW-3: A single board controller for floppy and Winchester disk drives:

• ***Fully complies with IEEE 696 Standard*** • Free copy of CP/M Plus™ included • Up to 4 floppies and three Winchester drives may be controlled by VFW-3 • Data transfers to and from board under DMA or programmed I/O control • Supports 24 bit address space.

CP/M Plus™: high performance single user operating system.

• CP/M® 2.2. compatible—no modification! • When used with SDSystems 256K memory board speeds are up to 7 times faster than CP/M® 2.2. • High performance file system • MP/M® II file password protection • Time and date stamps on files • Support for 1 to 16 banks of RAM • Support for 1 to 16 drives of up to 512 MB each • Easy to use system utilities with HELP facility • Powerful batch facility • Sophisticated programmer utilities.

RAMDisc 256: A solid state disk emulator that greatly increases system performance by eliminating disk waits in disk intensive applications. Excellent for spreadsheets, spelling checkers and software development.

• 256K capacity • 1 mb total bus capacity • CP/M® 2.2, PLUS™ compatible • I/O port addresses user selectable • Storage locations addressed by on board 20 bit counter • On board refresh.

ROMDisc 128: An EPROM board that replaces a floppy disk drive for the purposes of booting CP/M® and loading application programs.

• Provides non volatile, permanent storage of programs and data • Utilizes 2732 or 2764 EPROMS, (16 max) • 128K capacity per board • 512K system capacity • Use with SDSystems RAMDisc to configure a stand alone or network discless system • CP/M Plus™ available in eproms • Serial port provided.

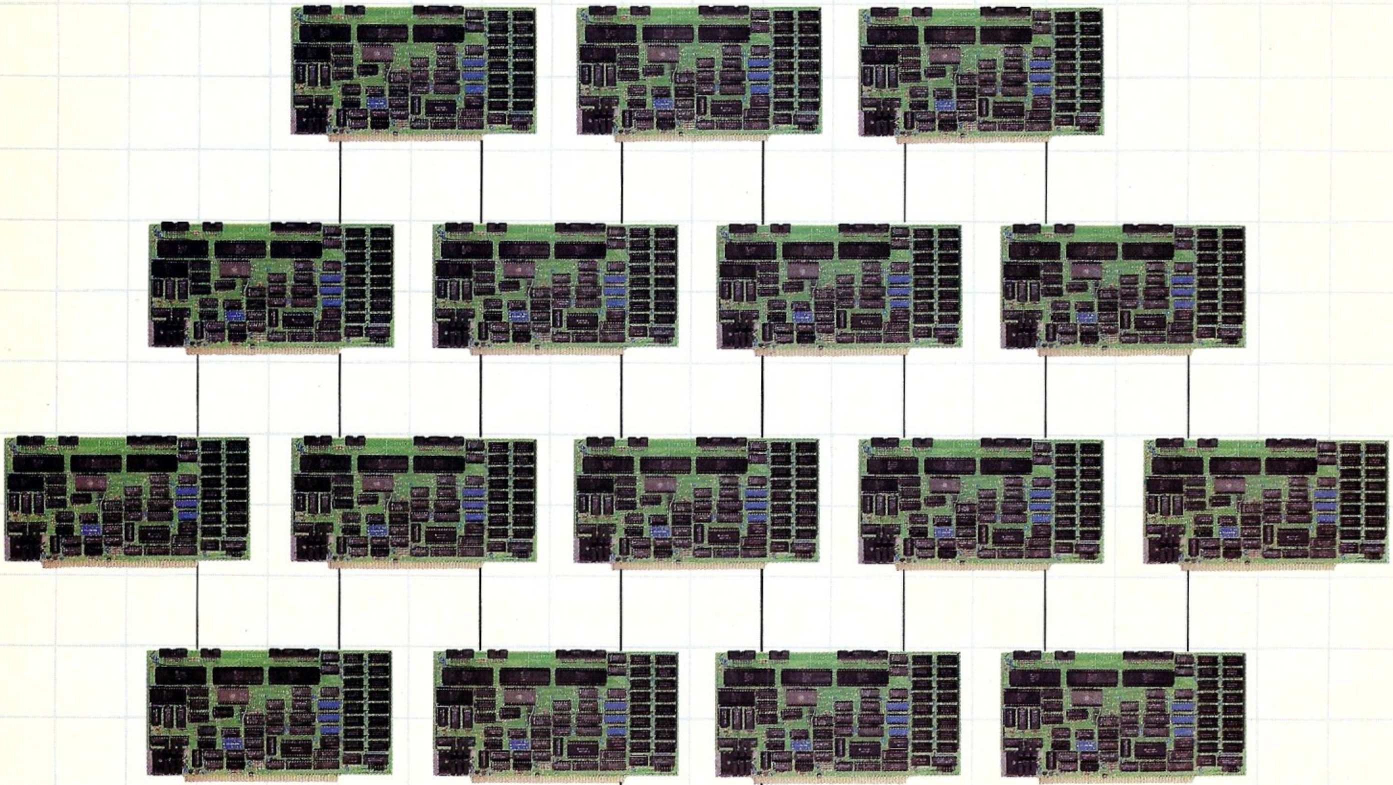
CP/M® 2.2 and CP/M Plus™ are registered trademarkes of Digital Research, Inc. *Z80 product of Zilog Corp.

SDSYSTEMS
A Syntech Company

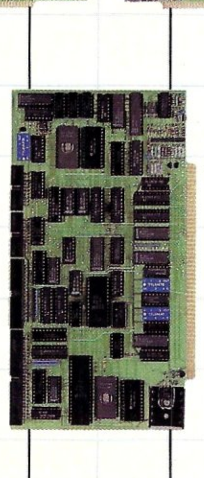
10111 Miller Road • Dallas, Texas • (214) 340-0303 • TLX-682 9016

CIRCLE 3 ON READER SERVICE CARD

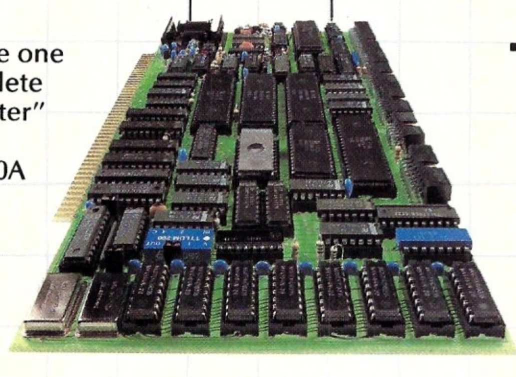
Now Our Family Tree Is Complete



SBC-1 (Above) A multiprocessing slave board computer with Z-80 CPU (4 or 6 MHz), 2 serial ports, 2 parallel ports, and up to 128K RAM. Provides unique 2K FIFO buffering for system block data transfers. When used with TurboDOS or MDZ/OS the results are phenomenal!



HD/CTC (Left) A hard disk and cartridge tape controller combined together on one board! A Z-80 CPU (4 or 6 MHz); 16K ROM, and up to 8K RAM provide intelligence required to relieve disk I/O burden from host system CPU. Round out your multiprocessing system with an integrated mass storage/backup controller.



Systemaster® (Right) The ultimate one board computer; use it as a complete single-user system or as the "master" in a multi-processing network environment. Complete with Z- 80A CPU, 2 serial and 2 parallel ports, floppy controller, DMA, real time clock, and Teletek's advanced CP/M BIOS. Also supports MP/M-II, MDZ/OS, and TurboDOS.

TELETEK

9767F Business Park Drive
Sacramento, CA 95827
(916) 361-1777
Telex #4991834
Answer back-Teletek

CIRCLE 174 ON READER SERVICE CARD

Your Single Source Family of S-100 Products.

Contents

Microsystems

Volume 4/Number 6
June 1983

DEPARTMENTS

Microsystems

STAFF

Sol Libes	editor
Chris Terry	technical editor
Ian Darwin/Dave Fiedler/Dave Hardy/Bill Machrone/ Ernest E. Mau/Bruce Ratoff/Anthony Skjellum	contributing editors
Andrew Bender/David Gewirtz/Fred Gohlke/ Steve Leibson/Don Libes/Randy Reitz	assisting editors
Ann Ovodow	editorial coordinator
Nancy Metz	editorial secretary
Mariano Nicieza	art editor
Jim Beloff	advertising manager

ADVERTISING SALES OFFICES

New England, Midatlantic
Jim Beloff, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-3452

Southeast
Mark Browning, *Browning Publications*
P.O. Box 81306
Atlanta, GA 30366
(404) 455-3430

Advertising Coordinator
Rosemarie Caruso, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-5386

Midwest
Jeff Edman, *The Pattis Group*
4761 W. Touhy Avenue
Lincolnwood, IL 60646
(312) 679-1100

Southern California, Southwest
Barbara Farkas,
Ziff-Davis Publishing
3460 Wilshire Blvd.
Los Angeles, CA 90010
(213) 387-2100

Northern California, Northwest
Jeff Cohen,
Ziff-Davis Publishing
3030 Bridgeway
Sausalito, CA 94965
(415) 331-7133

Canada
Frank Lederer, *The Pattis Group*
1623 Yonge St.
Toronto, Ontario M4T241
(416) 482-6288

Direct Retail Sales
Lynn Kujawa, *Ziff-Davis Publishing*
One Park Ave.
New York, NY 10016
(212) 725-7679

CONSUMER COMPUTER & ELECTRONICS DIVISION

President	Larry Sporn
Vice President Marketing	J. Scott Briggs
Vice President Circulation	Carole Mandel
Vice President General Manager	Eileen G. Markowitz
Creative Director	Peter J. Blank

ZIFF-DAVIS PUBLISHING

Richard P. Friese	**President**
Albert S. Traina	**President, Consumer Magazine Division**
Furman Hebb	**Executive Vice President**
Paul H. Chook	**Executive Vice President, Marketing and Circulation**
Phillip T. Heffernan	**Senior Vice President**
Sidney Holtz	**Senior Vice President**
Edward D. Muhlfeld	**Senior Vice President**
Philip Sine	**Senior Vice President**
Baird Davis	**Vice President**
George Morrissey	**Vice President**
Selwyn Taubman	**Treasurer**
Bertram A. Abrams	**Secretary**

Second Class postage paid at New York, NY 10016 and at additional mailing offices. POSTMASTER: Send address changes to MICROSYSTEMS, PO Box 1987, Morristown, NJ 07960. Subscriptions are $24.97 for 12 issues. Canadian prices are $5.00 per year additional; other foreign $8.00 per year additional (U.S. currency only). For information or questions about subscriptions phone: (800) 631-8112.

Editorial correspondence is welcomed and should be sent to: MICROSYSTEMS, One Park Avenue, New York, NY 10016. Phone: (212) 725-6856.

For information on commercial advertising, write to: MICROSYSTEMS Advertising Dep't., One Park Avenue, New York, NY 10016, or call Jim Beloff at (212) 725-3452.

☐ VISISCHEDULE ☐ SUPERCALC ☐ VISICALC ☐ WORDSTAR ☐ D.B. MASTER ☐ MULTI PLAN ☐ VISIFILE ☐ dBASE II ☐

☐ AZTEC ☐ SUBMARINE COMMANDER ☐ WIZARDRY—SCENARIO I ☐ PREPPIE ☐ CANYON CLIMBER ☐ DEADLINE ☐ STAR RAIDERS ☐ CENTIPEDE ☐ ZORK II

VOLKSWRITER ☐ DATA PERFECT ☐ FILE MANAGER 80+ ☐ SCREENWRITER PROFESSIONAL ☐ PFS: GRAPH ☐ THE SENSIBLE SPELLER ☐ VISITREND/PLOT

RENT SOFTWARE BEFORE YOU BUY!

from our

SOFTWARE
RENTAL LIBRARY

You can now RENT the most popular software available for just 20-25%* of Manufacturers' Retail Price

- Eliminate the risk—rent first!
- 100% of rental fee applies toward purchase
- All purchases are 20% Off of Manufacturer's Suggested List
- Rentals are for 7-days (plus 3 days grace for return shipping)
- No Membership Fees

Now currently available for:

Apple	IBM, PC	Standard CP/M
Eagle	TRS-80 II	Xerox 820
Northstar	Osborne	Heath/Zenith 89
	Franklin	

REMEMBER, THESE ARE NOT DEMOS, BUT ORIGINAL UNRESTRICTED SOFTWARE PROGRAMS

(complete with manuals in original manufacturers' packages)

To Immediately Order, or for more information:

UNITED COMPUTER CORP.
Software Rental Library
Culver City, California

Toll Free CALL 1-800 992-7777
In California CALL 1-800 992-8888
In L.A. County CALL 1-213 823-4400

MasterCard | AMERICAN EXPRESS | VISA

*Plus postage and handling.

☐ FROGGER ☐ CHOPLIFTER ☐ GORF ☐ DAVID'S MIDNIGHT MAGIC ☐ EASTERN FRONT (1941) ☐ ZORK I ☐

CIRCLE 4 ON READER SERVICE CARD

Editor's Page

by Sol Libes

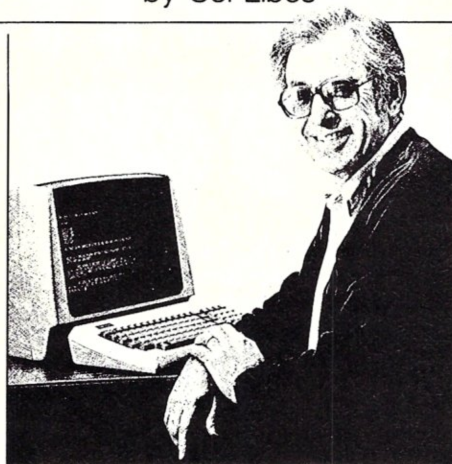
Graphics is becoming of increasing importance in the micro field. The Apple Lisa is the most current example of the trends in the improved graphics interface of micros. The hardware has improved tremendously in recent years, and software to take advantage of this improved hardware is now starting to appear.

Tektronix was the early leader in CRT graphics technology, introducing the storage tube with vectorgraphics type displays in the early 1970s and making computer graphics widely accessible to engineers doing CAD/CAM work. In the mid '70s, Ramtek introduced graphics terminals that used raster scan technology (essentially the same as that used in TVs), significantly increasing the capabilities of CRT graphics while reducing the cost. The raster scan technology was based on using a large memory system to store a memory image of the display in the form of dots, and displaying the dots on the raster scan. Some of the high-resolution graphics required as much as a megabyte of memory. Further, the same color technology used in color TV could be used for CRT graphics.

Color for micros

The next most significant event occurred in the late '70s with the introduction of the Apple computer, which included raster color graphics capability in which the user had access to the screen bit-map in memory. This made it possible to create excellent real-time graphics that extended far beyond the alphanumeric limitations of standard CRT displays for under $2000. This led to the development of some excellent games and business-oriented graphics.

In the very late '70s, IBM introduced the 3279 color graphics terminal using raster scanning. With the blessing of IBM, business color graphics began to take hold. In 1980, NEC introduced the first personal computer with



a separate microprocessor dedicated to supporting high-density color graphics. This is the current state of the art in CRT graphics developed for personal computers.

Graphics software in the '60s and even up into the mid '70s was a very custom thing, as there was no such thing as a standard interface. Textronix realized that, to sell their graphics displays, they needed some software to that was user friendly and easy to get at. They created a subroutine library linkable to Fortran called the Plot-10 Software, which in effect became the first standard graphics package. Then a company called ISSCO (Integrated Systems and Software Co., San Diego CA) introduced the first variable bus subroutine library and subsequently introduced the first interactive graphing package, called "Telegraph." These were mainframe products in the $20-30,000 range.

Soon afterwards, Precision Visuals brought out a package at roughly one-third the price. There are now similar packages for micros (VisiPlot was the first micro-level product in the $300-400 range. GSS4 from Graphics el product) in the $300-400 range. GSS4 from Graphics Software Systems is the first subroutine library in the micro world that has minicomputer capabilities. The functionality of the 16-bit micro packages now being introduced is bringing capabilities previously available only on mainframes to the micro level.

Graphics standards development have moved very slowly. The Calcomp plotter represented, in effect, the first standard graphics device developed in the early '70s. The Textronix Plot-10 software became the standard for a graphics subroutine library. In the early '70s, the ACM put together a Special Interest Group for graphics, which published a very comprehensive report in 1979. This served as the point of departure for the ANSI committee, which was formed in 1979. Today the X3H3 ANSI committee is diligently working to finish up a graphics standard that was started nearly 10 years ago.

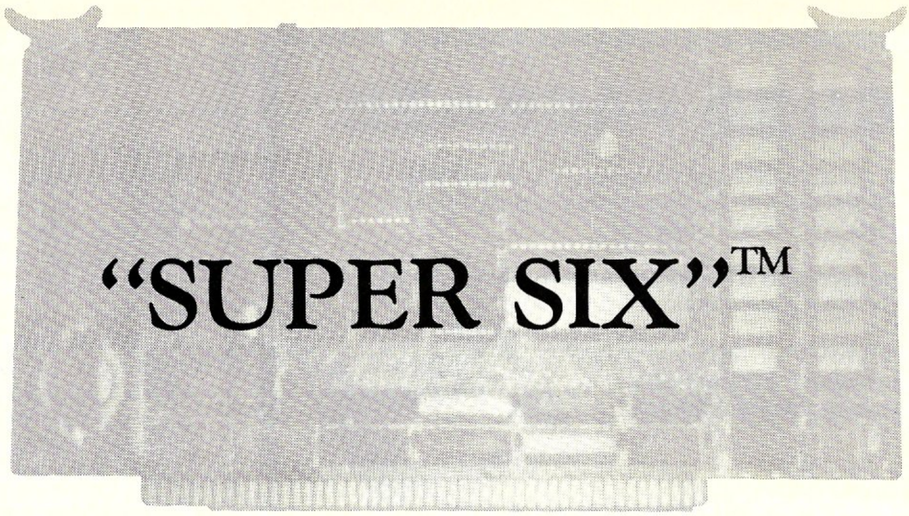
Standardization

There are three significant areas of graphics standardization. First is American National Standards Institute work, which is addressing the source and object code level. Along with the ANSI work is an ISO activity which is moving with a European graphics standard called Graphics Kernel System (GKS), which is moving toward adoption by countries internationally. The third area of standardization is called "Videotext." In the U.S., ANSI is working with NAPLPS (North American Presentation Level Protocol Syntax), which is really the Telidon work from Canada with AT&T's endorsement.

Today, when writing an application program using graphics that requires the drawing of lines, solid areas, etc., one can do this through a software interface such as GKS, which provides the function primitives to draw lines, fill areas, create polygons, windows, viewport transformations, etc. These functions are contained in subroutines that form a linkable library.

A universal interface?

The software interface that will be coming to the market soon is termed the "virtual device interface" and contains hardware



“SUPER SIX”™

ITS POWER IS AWESOME!

- 6MHz Z-80B CPU
- S-100 IEEE-696 Compatible
- 6MHz Z-80 SIO, PIO
- 128KBytes Bank Selectable RAM
- Floppy Disk Controller, Simultaneous Operation of 5-1/4 and 8-inch Drives
- DMA Controller
- CTC Interrupt Clock
- CP/M* 3.0 (Plus), TurboDos* and MP/M II* Compatible

We ASTONISHED the industry when we introduced SUPER QUAD*, the first single board S-100 computer. Now you're going to be AMAZED when you discover “SUPER SIX”™. It's power, it's speed, it's AWESOME.

The SUPER SIX runs 1-1/2 times faster than any other S-100 computer available today. 128KBytes of RAM with DMA is standard. So is simultaneous operation of 5-1/4 and 8-inch floppies. No more worries about non-standard 5-1/4 inch formats.

Combine the power and speed of the SUPER SIX as the Master and the proven performance of SUPER SLAVE™ processor boards, and you've got a multi-processor, multi-user system that is nothing short of AWESOME.

SPEED UP WITH SUPER SIX . . . see the SUPER SIX at your quality local dealer or contact:

ADVANCED DIGITAL CORPORATION

12700-B Knott Avenue
Garden Grove, California 92641
TEL: 714-891-4004 TELEX: 678401 TABIRIN

*CP/M and MP/M are trademarks of Digital Research Corp

TurboDos is a trademark of Software 2000 Inc.

SUPER SIX, SUPER QUAD, and SUPER SLAVE are trademarks of Advanced Digital Corp.

CIRCLE 148 ON READER SERVICE CARD

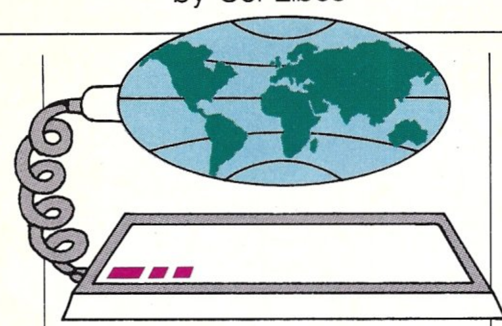
Editor's Page
continued . . .

protocols that allow a graphics software package to talk to a specific piece of graphics hardwre. This standard is currently being worked on by the ANSI committee. When this standard is finished and adopted, it is expected that manufacturers of graphics devices and IC graphics controllers will conform to the standard simplifying interfacing. Digital Research has already introduced a graphics virtual device interface extension to CP/M, called GSX, that conforms to the proposed ANSI standard. There is no doubt that we will see similar products from other companies. This should create a broader market for applications software packages using graphics. Companies that do not conform to the standard will find that they alone are supporting their graphics interfaces, thereby reducing the amount of software available for their systems.

Microsystems plans to include reviews of these packages in future issues.

News & Views

by Sol Libes



Random rumors

Steve Wozniak, creator of the Apple II, is rumored to be working independently on a project using a new video graphics display technology which will eliminate most of the video support circuitry. . . . Rumors are getting more rampant that American Bell will soon market a microcomputer system using the 32-bit BellMac 32 microprocessor. . . . It looks like the American National Standards Institute (ANSI) standard for microdisks is floundering and there will be no standard adopted this year, leaving the battle to be fought in the marketplace. . . . Data General is promising, at last, to enter the personal computer market later this year with a 32-bit system. . . . And Prime Computer is also expected to enter the personal computer market with a desktop workstation using either the 8086 or 80186. . . . Osborne Computer is expected to introduce two new systems this year (one of which will most likely be out by the time this appears in print). One is expected to be a lower-cost, more compact, and lighter version of the current Osborne One with a larger screen. The second is expected to be a dual processor (Z80/8088) unit with color display, IBM compatibility, and hard disk option. . . . Coleco is also expected to introduce a Z80-based microsystem this summer.

Low-cost CP/M systems hit market

Spectra Video Inc., NY NY, a manufacturer of electronic game units, has announced a Z80-based system with "CP/M compatibility" that carries a list price of $299.95. Called the SV-318, it is backed by 14 hardware peripheral products and 100 entertainment and software programs. It has 32K of ROM (expandable to 96K, contains Microsoft Basic and word processor), 32K of RAM (expandable), TV modulator, 71-key ASCII keyboard (with 52 graphics characters), user-definable keys and color capability. There is also a built-in joystick, plug-in cartridge slot, 256 x 192 pixel graphics, and lots more.

And Personal Micro Computers Inc., 475 Ellis St., Mt. View CA, showed their MicroMate computer at the recent West Coast Computer Faire with a special show price of $695. The unit included CP/M Plus (version 3.0) with a bank-switched 128K RAM, a 400K SDDD disk, and Basic, spreadsheet, spelling checker, and utilities programs. Just add a standard terminal and printer for a complete system.

Supermicroprocessors: A status report

National Semiconductor demonstrated their 16032 system recently at a UNIX conference, running a port of UNIX done by Human Comput Resources Corp., of Toronto. Further, they have been accepting orders for the chip set, promising production quantities this summer. They have been shipping sample quantities for several months. Thus we can expect to see initial shipments of 16032-based systems before year-end.

NS is expected to start shipping their "Mesa" system this fall. The Mesa, an 8-user system, is expected to contain the 16082 memory management unit with a 32-bit-wide virtual memory. NS is promising to start sampling its floating point co-processor chip this month.

The 16032 is reported to really be a 32-bit processor with 16-bit I/O. The 32032, which NC says it will begin sampling in the fourth quarter of this year, is reported to be an upgrade of the 16032 with true 32-bit I/O. Further, NS claims that it will be possible to build a fault-tolerant transaction system by running two 32032s in parallel and comparing results on alternate memory cycles to detect soft errors. NS says the 32032 will be 1.8 times faster than the 16032.

The Motorola 68000 although doing many operations 32-bits at a time has an internal 16-bit-wide data path, and hence the redesign task to move up to a 32-bit version (68020) is a more difficult undertaking for Motorola vis-a-vis National. The 68020 is being promised for April '84 (samples) and August '84 (production). This will be Motorola's true 32-bit micro operating at 16 MHz. It will do 32-bit multiply/divide, use 150,000 transistors, consume 1.5 watts, and have approximately 100 pins.

The Motorola 68881 floating point co-processor production schedule has slipped: Motorola is now promising samples in July '84 and production quantities in December '84. It will contain eight 80-bit registers, have all the addressing modes of 68020, be compatible with the IEEE floating point standard plus some additional functions.

The Motorola 68010, their 16-bit micro with virtual memory facilities, is now being sampled and production is expected this summer. The maximum clock speed will be 16 MHz.

A crimp was thrown into Intel's 286 project and 32-bit upgrade when a large group of engineers up and left their Portland OR operation to form a UNIX systems house based on Intel's technology. However, Intel is already shipping limited production quantities of the 80286.

The NS 16032 will compete

with the Motorola 68010 while the 32032 will compete with the Motorola 68020. Samples of both the 16032 and 68010 became available at about the same time, with NS promising earlier production quantities. It appears that NS may be sampling and producing the 32032 well ahead of Motorola's 68020. Thus, although National was very late entering the supermicro marketplace, it appears that they now have a good opportunity of garnering enough of it for it to be worthwhile for them.

Tandy finally goes CP/M

The last major CP/M holdout has finally knuckled under . . . Tandy has finally decided to sell CP/M for their new Model 12 system. They have decided to offer the new CP/M Plus (version 3) as an option, as they will include their own TRSDOS with the unit. However, be forewarned that the CP/M Plus implementation will limit the CP/M TPA (Transient Program Area) to only 48K because of the bank-switching hardware used. Thus, one of the primary advantages of CP/M Plus is immediately lost! On most other CP/M Plus systems, TPAs as large as 62K can be attained.

Competition heats up in the DOS world

Digital Research and SoftTech Microsystems have both dropped the price of the Disk Operating Systems for the IBM-PC. DR's CP/M-86 has been available from IBM for $250 and SM's p-System was $600, while PC-DOS (Microsoft's MS-DOS with minor changes) was offered for only $40. Reportedly, only about 3% of the PCs have been sold with CP/M-86 and less than 1% with the p-System. IBM also held up supplying these packages so that PC-DOS had the field to itself for several months, giving it another advantage over its competition.

DR therefore decided to take marketing matters into its own hands and market an improved version of CP/M-86 for the PC themselves; they are making it available for only $60. SM and IBM announced that they will make the p-System available for $50 for the runtime system.

Microsoft has also released version 2 of MS-DOS, which IBM is now furnishing with the new upgraded version of the PC called the PC-XT. Microsoft claims that version 2 has features to make it an upgrade path to their XENIX multiuser system. Further, they have already disclosed that the next version of MS-DOS will provide an even closer link to XENIX. Considering that IBM is continuing to give strong backing to PC-DOS, does this

mean that they are also planning to introduce a XENIX system? Of course IBM could drop always introduce their own implementation of PC-DOS and sever their connection with Microsoft.

Where is the iAPX432?

It is over two years now since Intel introduced their iAPX432 32-bit microprocessor chip set; as yet no commercial product has been introduced using it.

Many of the pioneering users who started product development projects using the 432 have since cancelled the projects or put them on hold due to performance limitations that have shown up with the 432. First of all, Intel was a year late in delivering samples, and then some users claimed that the samples delivered operated at only 20% of their rated speed.

Further, the unique architecture of the 432 meant that companies were starting from ground zero in software development, and software development tools and support chips have been slow in becoming available. The complex architecture also means that software development is more complex and expensive, and few companies have been willing to make the investment.

Intel is currently starting production on the third revision of the 432 chip set, which they hope will finally meet their original performance claims. Also, the bus interface and memory control ICs are expected to finally become available, and it is expected that we will soon see 432-based systems being introduced.

UNIX update

The implementation of UNIX on 68000-based micros appears to be split evenly between two different implementations: Xenix from Microsoft, Bellevue WA, and UniPlus+ from Unisoft, Berkeley CA. There are a few vendors who have done their own 68000 ports, but it is likely that they will switch to either Xenix or UniPlus+ to achieve some level of compatibility and a wider market base, since there already is some software available for these systems. UniPlus+ was the first port to reach the market and implements the Berkeley version of UNIX, which accounts for its popularity. Xenix, on the other hand, is an implementation of the Bell Labs version of UNIX and comes from one of the largest micro software houses in operation today (namely Microsoft).

The 8086 and Z8000-based micros appear to be going with Xenix. The only other alternative here is Coherent from Mark Williams Co., Chicago, and they have not been aggressive in getting their product out.

National Semiconductor has already demonstrated their UNIX implementation and one done by Human Computing Resources, Toronto Canada.

The initial results do not indicate that this implementation is any faster than the 68000, 8086 and Z8000 implementations, despite the claim by National Semiconductor that this micro has been specifically designed to support UNIX. The initial demonstrations of the product, however, are still considered prototypes, and tuning should improve its performance.

Intel introduces text co-processor chip

Intel has announced an IC (82730) to operate in conjunction with its 8086 16-bit micro that provides word-processing display functions such as proportional character spacing, smooth scroll, display of superscripts and subscripts, variable fonts, and allows users to define their own character sets and provide complete correspondence between display and printed copy. The unit can also be used in conjunction with Intel's new 82720 graphics coprocessor, allowing mixing of text and graphics on the screen. IC samples are now being supplied to OEMs, and production is expected this fall.

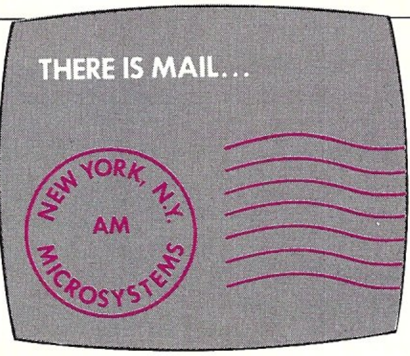
Random news bits

The IEEE 796 Bus (Multibus) Standard has been adopted by the IEEE and is now an official standard. Also, the IEEE expects to shortly adopt the Binary Floating Point Arithmetic (IEEE 754) and Assembly Language Mnemonics (IEEE 694) standards . . . . For those into country and western music, check out the record entitled "Basic Ain't the Language of Love." It tells the story of a woman involved with a personal computer addict . . . . . . Toshiba has announced a floppy disk system in development that packs 3MB on a 3.5″ floppy . . . . Zilog has announced that they will shortly introduce the Z80H, an 8MHz version of the Z80 . . . . Hidisk, San Diego CA, has announced that they now have available CP/M for the Digital Equipment LSI-11. They call it the Z-11 . . . . Do you know which is the largest number of CP/M-based systems in current use? I bet you'll never guess. Look below to see if you are right.

News bit answer

The largest number of systems running CP/M are the Apple II computers. Microsoft alone has equipped one third of them with their Z80 softcard running CP/M. Add to that Z80 cards from two other suppliers, and I estimate that about 40% of the Apple IIs in current use are running CP/M. Since Apple has already shipped over 800,000 Apple IIs, that means that there are about 400,000 Apple II CP/M-based systems!

Letters to the Editor



Dear Editor,

I strongly disagree with the experience F.B. Laughlin reported on his interactions with the people at Lexisoft, Inc. who produce Spellbinder. I have had my S-100 based system since 1979 and Spellbinder word processing software since then (it was first known as Autospell). In that time, I have written several journal articles, collaborated on two plays, two chapters in books, and a test manual. All of this was done using my word processing software.

Of course, there have been times when I wished the word processing software could do something special. Every time I've contracted Lexisoft, they have been extremely cooperative with my problem. For instance, I have an NEC spinwriter and I wanted to use the proportional space thimbles. The letters are palced in different locations than on non-proportional spaced thimbles. Lexisoft told me of a modification with a lookup table to allow me to use the special thimbles.

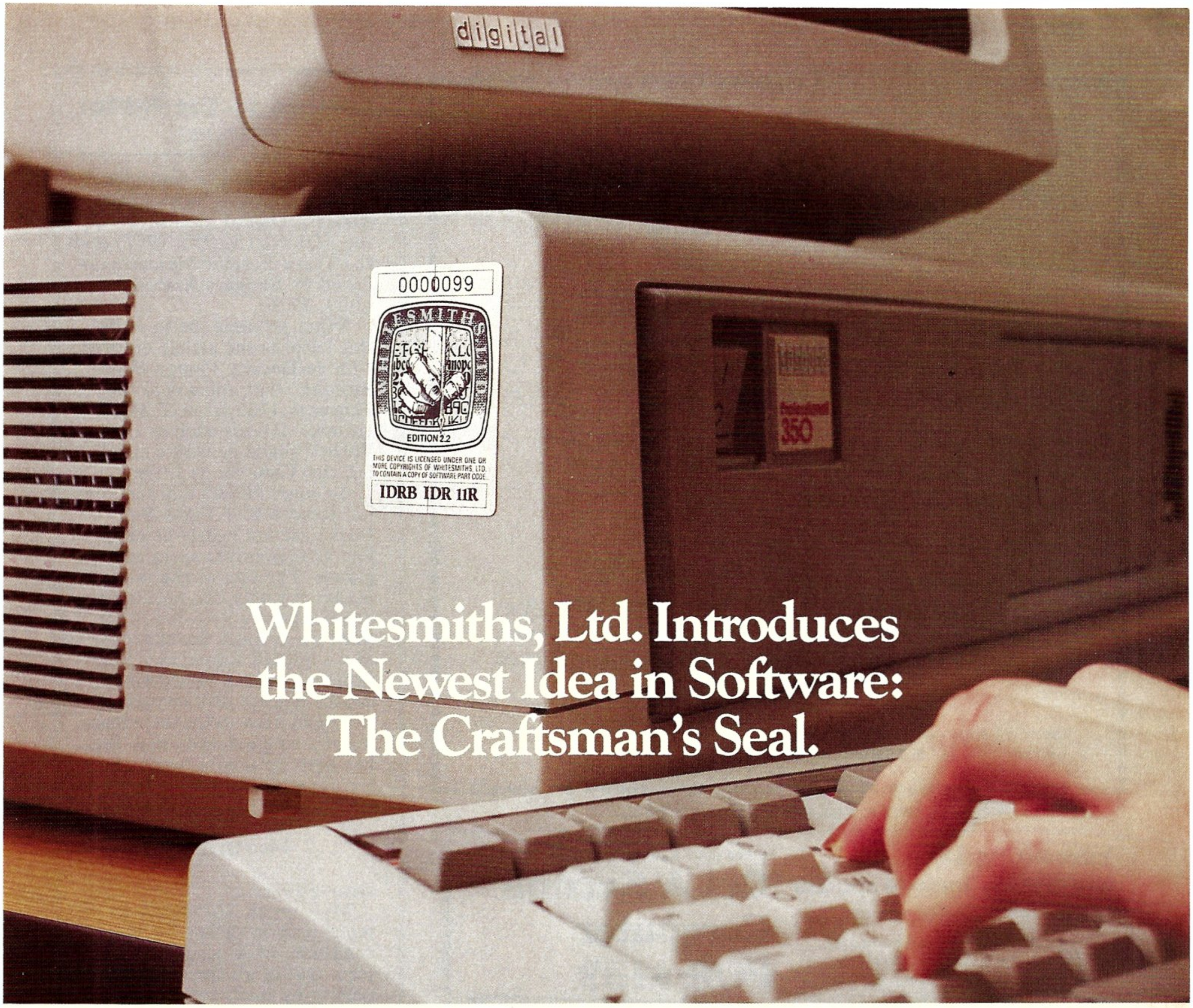
Ronald E. Olson, Ph.D.
Univ. of Illinois at Chicago
Office of the Dean
808 South Wood Street
Post Office Box 6998
Chicago, Illinois 60680

Dear Editor,

Computerists who are still using disk controllers based on the Western Digital FD1771 chip can increase the CP/M file capacity of their 8-inch single-density single-sided disks from 241K to 354K by formatting each of the non-system tracks (track 2 through track 76) into two 2432-byte sectors. The deblocking algorithm provided by Digital Research must be changed in order to handle these sectors, whose length is equivalent to 19 128-byte logical sectors; as listed in Appendix G of the "CP/M System Alteration Guide," DEBLOCK.ASM is predicated on the use of sectors whose byte size is 128 times an integer power of two.

BIOS space can be increased from 0.875K to 1.625K by formating tracks 0 and 1 into 29, as opposed to the usual 26, 128-byte sectors. Compatibility is still maintained with any existing ROM-based routine for loading the coldstart boot program on sector 1, track 0.

For complete formating in-



Whitesmiths, Ltd. Introduces the Newest Idea in Software: The Craftsman's Seal.

Like the craftsman's seal on a fine work of art, Whitesmiths' Authorization Seal is more than just a means of identification. It represents the pride in workmanship, sense of tradition, and commitment to value on which we have always prided ourselves.

At Whitesmiths, we've established a tradition of producing some of the industry's most innovative and dependable software. Our C and Pascal compilers and cross compilers run in over 30 environments on DEC, Intel, Motorola and Zilog computers. Idris, our highly portable operating system, is compatible with UNIX and runs on all those vendors' machines too. And our full line of Software A La Carte items – from assemblers to special purpose tools – permits our customers to pick and choose what they need to tailor our products for unusual requirements.

The Authorization Seal is one more way we can take pride in our products. Once affixed to the machinery on which the software is to run, the Authorization Seal verifies, at a glance, ownership, product code, and the software copyright – all of which reduce order turnaround time and upfront expenses for both OEMs and end-users. No contracts to sign, no documents to read. Just good software, plain and simple.

The Whitesmiths, Ltd. Authorization Seal. It's one more way that we're committed to becoming a software tradition in your time.

Contact Whitesmiths, Ltd.,
97 Lowell Road, Concord,
MA 01742, (617) 369-8499
TLX 951708 SOFTWARE CNCM.

Whitesmiths, Ltd.
Software Craftsmen

UNIX is a trademark of Bell Laboratories; IDRIS is a trademark of Whitesmiths, Ltd.
Distributors: **Australia,** Fawnray Pty. Ltd., P.O.B. 224, Hurstville, NSW 2220 (612) 570-6100; **Japan,** Advanced Data Controls Corp., Chiyoda-ku, Tokyo (03) 263-0383; **United Kingdom,** Real Time Systems, Newcastle upon Tyne 0632 733131

CIRCLE 7 ON READER SERVICE CARD

Letters to the Editor
continued . . .

formation and disk-controller command details, refer to the Western Digital data sheets on the FD1771 (March, 1977) and the Track Format Manual published by Shugart Associates (July, 1976).

What prompts me to write this letter is the article entitled "Triple-Density Floppy Disk Storage" that appears in the February 1983 issue of *Microsystems.* Boards such as the Tarbell single-density controller, the Versafloppy I, the Cromemco 4FDC, and the Digital Research of Texas Big Board are not nearly as obsolete as they might seem at first glance. The fact of the matter is that not only can no existing double-density disk controller yield three times the capacity of a standard formated single-density disk, but no existing double-density disk controller can yield even twice the capacity of a single-density disk that has been optimally formatted for use with the FD1771.

Robert Lurie
8 Tingley Road
Morristown, NJ 07960

Dear Editor,

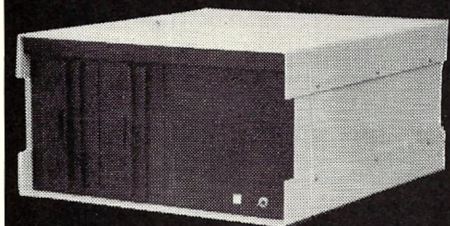
My copy of *Microsystems* often arrives in rather shabby condition, with pages bent or torn. Out of all the magazines I receive by mail, I must say that yours comes in the worst condition of the lot. I don't mean to imply that I've *never* received others in the same condition, but that this was/is rare (rather remarkable considering the Post Office!). I must also say that the magazines are not beyond use, by any means, but I haven't been called a perfectionist for nothing! At any rate, I thought it something you should know about.

Thomas C. Smith
506 19th Lane
Vero Beach, FL 32960

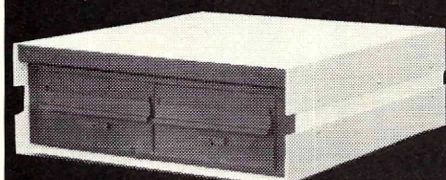
If your magazine arrives in bad condition from the postal service, call the Microsystems *subscription department and request another copy (800) 631-8112 or (201) 540-0445.*
—Editor

Main/Frames
Main/Frames
from
$200

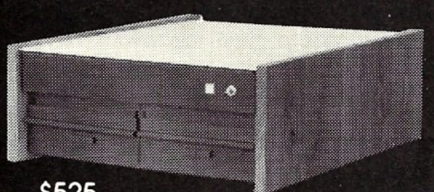
- 30 Models of Enclosures
- Assembled and tested
- Quasi-Coax Motherboards
- Power Supply
- Card cage and guides
- Fan, line, cord, fuse, power & reset switches



8" Floppy Main/Frame
$482



8" Disc Enclosure
$250



$525
Phase/80 8" Floppy Mainframe

$900
Phase/80 Desk + Mainframe

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

INTEGRAND

8620 Roosevelt Ave.•Visalia,CA 93291
209/651-1203

We accept BankAmericard/Visa and MasterCharge

CIRCLE 49 ON READER SERVICE CARD

GREAT FEELING
#PR 5500

Word processing bi-directional daisy wheel printer. 185 words per minute. Micro justification. 48 positions per horizontal inch. 120 positions per vertical inch.

$995.

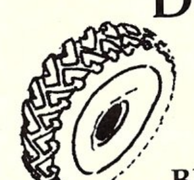
YOUR CUSTOMER GETS $2,000 OF FREE SOFTWARE WITH EVERY SANYO COMPUTER YOU SELL.

SANYO DESK-TOP COMPUTERS
"It's a great feeling"

SANYO
DESK-TOP BUSINESS COMPUTERS BY
SANYO BUSINESS SYSTEMS CORP.
51 Joseph St., Moonachie, N.J. 07074 (201) 440-9300 (800) 526-7043

CIRCLE 23 ON READER SERVICE CARD

Don't Re-Invent the Wheel - Use Ours!



Blaze/lib™

A solid, time-tested core of linkable modules providing buffer manipulation, string/number conversions, directory search, character I/O, and screen formatting functions in Pascal/MT+™. The Terminal Dependent Library provides full screen control for Lear-Siegler, Soroc, Xerox 820, TRS-80 II, Televideo, and Heath terminals.

.ERL $75 Source $200

Blaze/IO™

All Pascal/MT+ file I/O and UTILMOD functions, completely rewritten in optimized 8080 assembly code. Typically 40% faster and 50% smaller than the original MT+™ routines written in Pascal. Blaze/IO™ can lop 5-7K off a 25K .COM file! Also includes APPEND(F) function for text files, and TAB(X) and COLUMN(X) functions for formatting text output.

.ERL $75 Source $2500

Phonedex I™ $49.95

Personal phone/mailing list data base. Prints mail labels 1-4 up, address book pages. Data can be queried on any field and extract files created. Phone numbers can be dialed through D.C. Hayes Smartmodem™. Dumb terminal function allows communication with CBBS/timesharing systems.

ANOVA/Plus™ $69.95

Step up from two-level "T" tests to full 5-factor ANOVA, with an option for comparison of individual mean levels through Scheffé's Contrasts. Data files may be created and edited from within the program. In machine code Pascal for lightning speed beside versions in BASIC.

VISA MasterCard

STARSIDE ENGINEERING
PO Box 18306 • Rochester NY 14618 • (716) 461-1027

Please add $3 shipping/handling for all orders. NYS add 7% tax.

CIRCLE 266 ON READER SERVICE CARD

The CP/M Bus

Relocating modules and their use in overlaying programs, segmented programs, overlay management, and system extensions

by Anthony Skjellum

This month's CP/M Bus column is a discussion of program overlays. Included in the discussion is the use of relocatable modules to create such overlays. Also discussed are programs which are segmented because of their size. We have previously mentioned another type of relocatable program. This is the type of program which relocates itself under CP/M's BDOS (Basic Disk Operating System) for the purpose of extending the environment for other software. These system extensions will be discussed further in this installment.

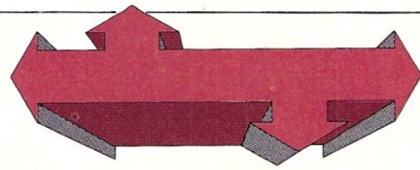
The initial discussion on relocatable code was given in the February 1983 column. Readers are encouraged to refer to this previous column before proceeding. System extensions were also introduced there.

Page relocatable code

For the purposes of the following discussion, all relocatable code will be of the page relocatable variety. As mentioned previously, such code is capable of relocation at page boundaries (XX00 hexadecimal) via a simple relocating subroutine. For 8086/88 code, no such relocation step is needed, but the concepts which follow will still be of interest.

Programs and overlays

A program overlay is a segment of code stored on disk that is conditionally brought in for execution, depending on the needs of the main program. Overlays can be used to permit large programs to run in segments or prevent programs with many diverse functions from requiring unduly large amounts of memory. We will distinguish between programs which execute segments in a sequential fashion from those which call up overlays as needed. Each will be discussed in turn.



Segmented programs

As mentioned above, a segmented program is one that is divided into several parts. The parts are executed serially. At the completion of a segment, the next one is loaded and replaces its predecessor in memory. Shared variables are provided through memory locations selected by the programmer, and some segments may share larger amounts of information via temporary files.

The mechanism of segmentation is quite straightforward. The initial .COM file includes both the first segment and a segment loader module. When a segment completes its task, it relinquishes control to the segment loader. The loader then loads the next segment into the area previously occupied by the former segment. Finally, the loader gives control to the new segment. The final segment does not call the loader, but instead returns control to CP/M via a warm boot request.

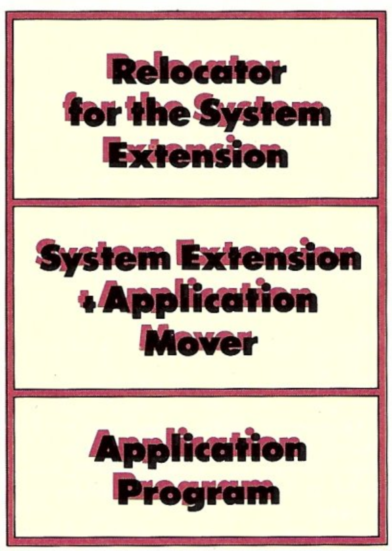


Figure 1. System extensions can be made to load transparently with the application program by arranging them in a special binary module.

A familiar program that operates in a segmented fashion is the BDS C compiler. The two phases of compilation (parsing and code generation) are divided between the CC1 and CC2 programs respectively. If CC1 is executed in the default mode, it loads CC2 after it has completed. However, CC2 is not loaded by CC1 if CC1 discovers a fatal error in the source file under compilation. Furthermore, CC1 passes all information to CC2 via memory when the compiler runs in the segmented mode. Only when the programs are executed separately is a temporary file created by CC1. (This turns out to be undesirable, since optimization ('-o') cannot be performed when the compile steps are executed separately.)

Programs with overlays

A program that uses overlays will have to be modular so that its functions may be conveniently divided into overlays. Conceivably, programs could be constructed which use more than one overlay simultaneously. Then such overlays would have to be page relocated after loading for execution at the address chosen by the program. Consequently, such overlays would have to be relocatable object modules.

It is also conceivable that overlays would refer to other overlays. This could open the way for primary, secondary, and possibly tertiary overlays. For example, the main program itself could request primary overlays. These primary overlays would in turn request secondary overlays. (A typical primary overlay would be a CRT

IS YOUR NEW CP/M™ MICRO STILL INTIMIDATING YOU?

By now, you may have discovered that CP/M is a wonderful operating system. But not an easy one to work with.

Here at last is **POWER!**, a super-power-packed, user-friendly program that takes the frustration and intimidation out of using CP/M.

POWER! offers 55 prompted, easy-to-use CP/M utilities in one 15k package. It automatically numbers disk files. From a screen menu, you pick the number to Copy, Erase, Reclaim, Rename, Type, etc. Your computer feeds the file names automatically. No more typing errors. And you can even run programs from the numbered menu.

No more BDOS errors. And you don't need a system disk in any disk drive!

POWER! lets you Test and Fix bad disks. Reclaim accidentally erased files or programs. Single step through memory up or down. Search, View, Change memory or disk in a snap. See Status and File Size instantly. Verify Checksums for programs. Load or Save programs at any address. Convert Hex numbers to Decimal or Binary instantly.

Now, **POWER!** also includes a special program that lets you lock sensitive files, so that only you can access them. Without the secret PASSWORD which you can create and change at will, no prying eyes will ever know your secret file even exists.

POWER! will operate in any standard CP/M or MP/M system, including CP/M-86, IBM PC, Apple (Z80 card), Osborne, Kaypro, HP, TeleVideo, TRS-80 conversions, S100's including NorthStar, Vector, Morrow, CompuPro, etc. Available in all disk formats.

It comes with a 120-page easy-read guide. And at only $3.00 per utility command, **POWER!** is your best buy in software. It's 55 friendly programs rolled into one to simply put you in control of CP/M.

Don't be intimidated by CP/M. Call or send in your order today and **Take The POWER! Trip.**

InfoWorld
Software Report Card

Power Version 2.55

	Poor	Fair	Good	Excellent
Performance	☐	☐	☐	☑
Documentation	☐	☐	☑	☐
Ease of Use	☐	☐	☐	☑
Error Handling	☐	☐	☐	☑

© 1982 by Popular Computing, Inc. A subsidiary of CW Communications Inc., Framingham MA. Reprinted from InfoWorld.

Only $169. Money Back Guarantee

Charge & COD Orders Welcome VISA MasterCard

TOLL FREE (800)428-7825 Ext. 96N
IN CA: (800)428-7824 Ext. 96N
DEALERS AND OEM'S (415)567-1634 Ext.N

Take The 'POWER!' Trip.

COMPUTING!
2519 Greenwich San Francisco, CA 94123
TOLL FREE (800)428-7825 Ext. 96N
IN CA: (800)428-7824 Ext. 96N
DEALERS AND OEM'S (415)567-1634 Ext.N
ONLY $169. Calif. add 6½% sales tax
☐ **CP/M $169.** ☐ **CP/M-86 $169.** ☐ **MP/M $249.**

Card No. ____________
Exp. Date ____________
Computer ____________
Your Name ____________
Company Name ____________
Address ____________
City/State/Zip ____________

CIRCLE 14 ON READER SERVICE CARD

management package, while a typical secondary overlay routine could be the low-level CRT routines for a specific terminal.) Each overlay would contain a list of the overlays which it in turn references. These overlays would be loaded at once so that the overlay which requested them may function immediately.

In order to house the overlays used by a program, an overlay file would be needed. An overlay file would consist of a list of overlays and their sector addresses (and lengths) within the file. Such a directory would be referenced for the purpose of finding overlays. A copy could be optionally kept resident in memory to minimize the access time for acquiring a given overlay.

Overlays will consist not only of executable subprograms. Many programs will find it useful to provide on-line information via text overlays. Such overlays would be stored in their own text overlay file with a directory of sector addresses and lengths (the same way as for code overlays). Reference to a given overlay would be through that directory.

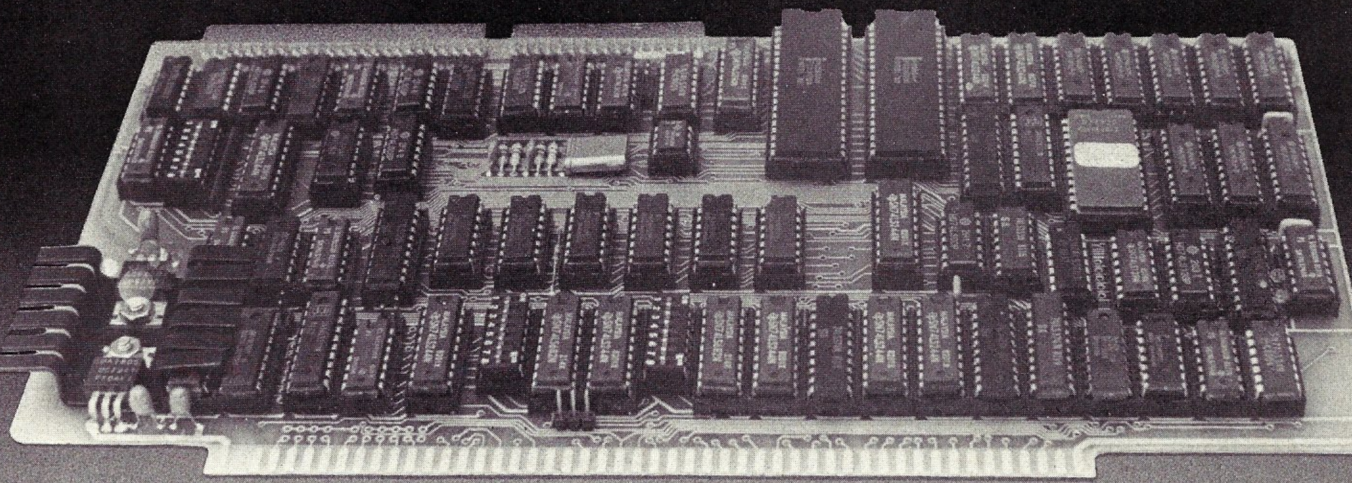
Once overlays are loaded and used in a program's execution, a method for efficient deletion of these overlays must be available. A least-recently-used (LRU) technique could be applied to such overlays. Such a technique would need to be carefully constructed and would require that the calling subroutines work through a counting procedure so that a running usage count on each overlay could be maintained. Such a system is likely to be complicated and not necessary in most applications. However, it will be universally advisable to delegate all overlay handling to a central overlay manager. Centralization of the overlaying process would increase efficiency and permit portability of the overlay manager to other programs.

The overlay manager would provide a command structure for handling overlays. Specific requests for overlay deletion would also be supported. This would allow explicit removal of overlays once they are no longer needed. This could almost completely remove the need for an LRU system for overlay deletion if used properly by the programmer.

A well-known software system that uses overlays is the WordStar text processor. The menu-driven commands used by WordStar are particularly suited to overlays. Furthermore, Word Star uses text overlays to provide users with on-line help and other information about the editor's operation.

In the above paragraphs, we have outlined the possibilities for using overlays in certain types of programs. Now we

Does Yours Compare with OMNIDISK?



Introducing tomorrow's disk controller . . . OMNIDISK offers S-100 users a unique combination of compatibility and technological innovation that together produce features not found in any conventional disk controllers. See for yourself what tomorrow looks like:

- ✓ **Simultaneous support of both 5¼" and 8" floppy disks allows software transfer between disks.**
- ✓ **24 bit DMA allows CPU by-pass.**
- ✓ **Power-on boot PROM gets you up and running in a hurry.**
- ✓ **On-board de-blocking conserves valuable RAM space above bios.**
- ✓ **Interfaces with the WD 1001® hard disk controller. No need to buy a host adapter.**
- ✓ **Full 16 bit port addressing.**
- ✓ **Full track buffer allows controller to recall entire track, not just sectors. Results in a speed increase 3-to-7-times greater than conventional controllers for both read and write operations.**

OMNIDISK'S features reflect our committment to designing S-100 products with an eye on the future. OMNIDISK'S price reflects our committment to offer products with an eye on the needs of today's user. You can begin using tomorrow's disk controller today for only. . . **$399***

So why wait, order now.

***CP/M configured for OMNIDISK, only $25 with purchase.**

FULCRUM™
COMPUTER PRODUCTS

distributed by:
W.W. COMPONENT SUPPLY INC.
1771 Junction Avenue
San Jose, CA 95112
(408) 295-7171

FREE U.P.S. ground shipping on pre-paid orders. Shipping will be added to C.O.D., VISA and M/C orders. CA residents please add sales tax.

Tomorrow's 8 MHZ Z80 CPU coming soon from FULCRUM

CIRCLE 10 ON READER SERVICE CARD

turn to a discussion of system extensions.

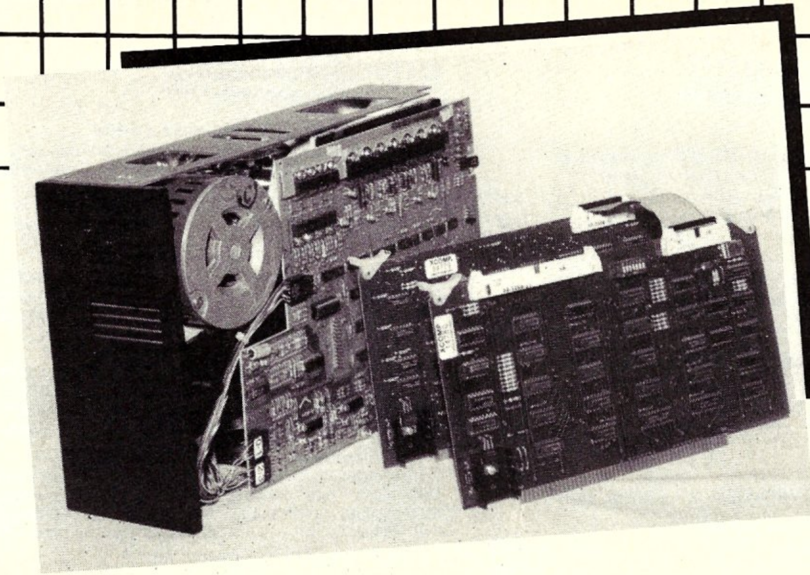
System extensions

A system extension is a program that relocates itself beneath BDOS, intercepts BIOS and BDOS CALLs and extends the capabilities of the operating system by interpreting these calls in new or more general ways. The method for implanting system extensions was mentioned previously. System extensions could also be made accessible to application software through a RST calling mechanism.

System extensions can simplify and enhance the CP/M environment for specific tasks. For example, a graphics manager could be brought in below CP/M for the benefit of a Basic interpreter in which graphics are to be created via PRINT commands. Instead of modifying the BIOS to handle the specific application (which is undesirable), a system extension is provided to do so. Many such extensions are possible, and several could co-exist at once. Furthermore, system extensions need not be loaded explicitly by the operator, but can be made to load transparently with the application program. This is done by arranging a special binary module (see Figure 1 on page 24).

The relocator gets control from CP/M, installs the system extension and then gives control to the application mover. The application mover (really part of the system extension) moves the application program down to the start of the transient program area (TPA) and then relinquishes control to the application program. The application program executes as if run as a normal transient program.

The above concept can be generalized to relocate several system extensions for a given application program. The overhead for several relocations would be at most a few seconds. This time is not significant compared to the length of time a typical application program is used during a single execution. ■

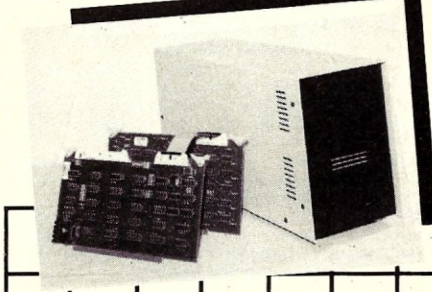


10 MEGABYTE S-100 PACKAGE $1295.00

"THE WINCHESTER DEAL OF THE YEAR"

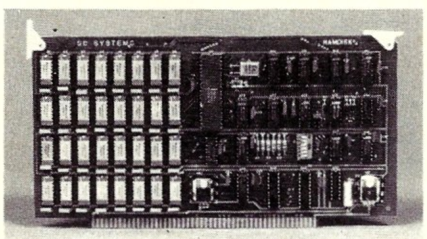
If your data storage needs have outgrown your floppy - based S-100 system, Jade now gives you the opportunity to add a fast, reliable hard disk package to your system *at a bargain price.*
If purchased seperately, the components of this package would cost you well over $3000.00; our special purchase now allows us to give you a special price of only $1295.00 CALL NOW... These units will not be on the shelf for long at this price!

- *Shugart SA1004 8-inch 10MB Winchester*
- *X Comp S-100 Controller For SA1004*
- *Your Choice of X Comp CP/M Auto-Attach Software (for most S-100 Systems) or "WD BOOT" PROM (for SD Systems board sets)*
- *Cables and Manuals Included*
- *Limited Quantity*



For Cabinet & Power Supply Please Add $200.00

MSH-991000 *w/SD Boot Prom* **$1295.00**
MSH-991002 *w/Auto Attach Sftwr* **$1295.00**



RAM DISC

The RAM Disc from SD Systems is a "solid-state disk drive" for IEEE696/S-100 systems. It allows programs to execute from high speed RAM, circumventing the mechanical problems and speed limitations of floppy disk drives. RAM Disc increases system performance substantially in disk intensive applications. 256K storage per board. "install" programs available for CP/M & MP/M

MEM-66256A *RAM DISK* **$799.95**

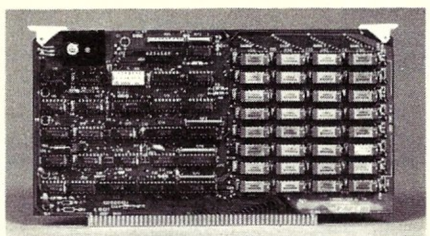
ROM DISC

SD Systems ROM Disk is a 128K EPROM/ROM board which is similar in concept to the RAM Disc. Like the RAM Disc it allows execution of software at high speed directly from memory, without the use of disk drives. The ROM Disc is non volatile memory ideally suited for use with the new CP/M Plus, now available in EPROM

MEM-12850A *ROM DISC Board* **$295.00**

CP/M Plus On EPROM:

SFC-55009059E *Unbanked RS232* **$400.00**
SFC-55009057E *Banked RS232* **$400.00**



EXPANDORAM III

SD Systems new ExpandoRAM III is a high density S-100 memory board utilizing the new 64K x 1 dynamic RAM chips. It allows memory sizes of 64K, 128K or 256K all on a single S-100 board.
The ExpandoRAM III marks a new generation of highly reliable and versatile dynamic RAM boards suitable for expanding current S-100 systems, or for use in new multi-user systems.

MEM-65064A *64K* **$495.00**
MEM-65128A *128K* **$595.00**
MEM-65192A *192K* **$675.00**
MEM-65256A *256K* **$755.00**

VERSAFLOPPY II

SD Systems' Versafloppy II is a reliable field-proven S-100 double-density floppy disk controller. For full software compatibility it can also read and write standard IBM 3740 single density format. The Versafloppy II controls single or double sided drives in single or double density format, 5¼" or 8" in any combination and up to 4 drives simultaneously. The Versafloppy II is faster, more stable and more tolerant of bit shift and "jitter" than most other controllers. All control and diagnostic firmware is included

IOD-1160A *Versafloppy II* **$359.95**

SBC 200

SD Systems' SBC-200 is an S-100 bus compatible single board computer based on the 4MHz Z80A CPU. It contains a sychronous/asynchronous serial port with software programmable baud rates, a parallel I/O port, a four channel counter timer and 1K on-board RAM. Up to 8K of firmware may be added to the board's 4 EPROM sockets. The SBC 200 is the ideal heart of a system composed of the other boards in the SD product line, or can stand alone in process control applications.

CPC-30200A *SBC-200* **$399.95**

CP/M 3.0 PLUS

SD Systems' implementation of Digital Research's CP/M 3.0 Plus is a unique combination of hardware and software that optimizes the powerful new features of CP/M Plus. CP/M Plus requires 192K of memory for optimum performance, making it an ideal match for the 256K ExpandoRAM III

SFC-55009059F *Unbanked, RS232* **$250.00**
SFC-66009058F *Unbanked, VDB8024* **$250.00**
SFC-55009057F *Banked, RS232* **$250.00**
SFC-55009056F *Banked, VDB8024* **$250.00**
SFC-55009057D *Manual* **$50.00**

FREE!* CP/M 3.0

S-100 board set with MHz Z-80A, 64K of RAM expandable to 256K, serial and parallel I/O ports, double density disk controller for 5¼" and 8" disk drives, new and improved CP/M 3.0 manual set, system monitor, control and diagnostic software. Includes SD Systems SBC-200, 64K ExpandoRAM III, Versafloppy II, and FREE CP/M 3.0 — all boards are assembled and tested.

***64K Board Set with FREE CP/M 3.0** **$1195.00**
***256K Board Set with FREE CP/M 3.0** **$1395.00**

Place Orders Toll Free

Continental U.S.
800-421-5500
Inside California
800-262-1710

JADE
Computer Products

4901 West Rosecrans, Hawthorne, California 90250

CIRCLE 16 ON READER SERVICE CARD

5 1/4 DISK DRIVES

Tandon TM100-1 *single-sided double-density 48 TPI*
MSM-551001 ________ $219.95 ea 2 for **$199.95 ea**

Shugart SA400L *single-sided double-density 40 track*
MSM-104000 ________ $234.95 ea 2 for **$224.95 ea**

Tandon TM100-2 *double-sided double-density 48 TPI*
MSM-551002 ________ $294.95 ea 2 for **$269.95 ea**

TEAC FD55A *single-sided double-density 40 track*
MSM-660551 ________ *$299.95 ea 2 for* **$239.95**

5¼" Cabinets with Power Supply
END-000216 *Single cab w/power supply* ________ **$69.95**
END-000226 *Dual cab w/power supply* ________ **$94.95**

S-100 MOTHERBOARDS

ISO-BUS - Jade
Silent, simple and on sale - a better motherboard.

6 Slot (5¼" x 8⅝")
MBS-061B *Bare board* ________ **$22.95**
MBS-061K *Kit* ________ **$39.95**
MBS-061A *A & T* ________ **$69.95**

12 Slot (9¾" x 8⅝")
MBS-121B *Bare board* ________ **$34.95**
MBS-121K *Kit* ________ **$69.95**
MBS-121A *A & T* ________ **$109.95**

18 Slot (14½" x 8⅝")
MBS-181B *Bare board* ________ **$54.95**
MBS-181K *Kit* ________ **$99.95**
MBS-181A *A & T* ________ **$149.95**

8" DISK DRIVES

Siemens FDD 100-8 Single-sided double-density
MSF-201120 ________ **$274.95 ea 2 for $249.95 ea**

Shugart SA801R Single-sided double-density
MSF-10801R ________ **$394.95 ea 2 for $389.95 ea**

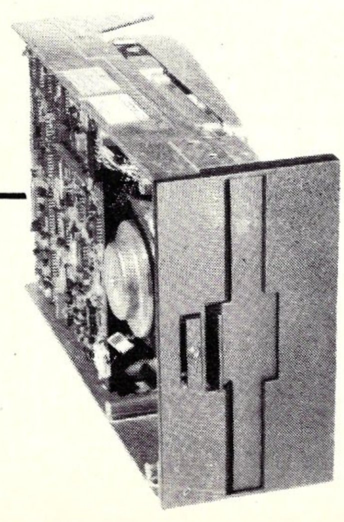
Shugart SA851R Double-sided double-density
MSF-10851R ________ **$554.95 ea 2 for $529.95 ea**

Qume DT-8 Double-sided double-density
MSF-750080 ________ **$554.95 ea 2 for $529.95**

Tandon TM848-1 Single-sided double-den thin-line
MSF-558481 ________ **$379.95 ea 2 for $369.95 ea**

Tandon TM848-2 Double-sided double-den thin-line
MSF-558482 ________ **$494.95 ea 2 for $484.95**

HALF-HEIGHT 8 INCH - NEC
The FD1165 is a double sided, double-density 8-inch floppy disk drive which uses a high speed steel-band head positioning mechanism and a direct drive D.C. spindle motor. One of the most reliable 8-inch drives available.
MSF-851164 *FD1164 single sided* ________ **$389.95**
MSF-851165 *FD1165 double sided* ________ **$475.00**



MODEMS

SMART BUY in MODEMS - Signalman
1200 and/or 300 baud, direct connect, automatic answer or originate selection, auto-answer/auto-dial on deluxe models. 9v battery allows total portability, full one year warranty.
IOM-5600A *300 baud direct connect* ________ **$89.95**
IOM-5610A *300 baud Deluxe* ________ **$149.95**
IOM-5620A *1200/300 baud Deluxe* ________ **$369.95**
IOM-5650A *300 baud for Osborne* ________ **$119.95**
IOM-5630A *300 baud card for IBM* ________ **$269.95**

SMARTMODEM - Hayes
Sophisticated direct-connect auto-answer/auto-dial modem, touch-tone or pulse dialing. RS-232C interface, programmable.
IOM-5500A *Smartmodem 1200* ________ **$574.95**
IOM-5400A *Smartmodem 300* ________ **$224.95**
IOK-1500A *Hayes Chronograph* ________ **$218.95**
IOM-1100A *Micromodem 100* ________ **$368.95**
IOM-2010A *Micromodem II w/term prgm* ________ **$329.95**
IOM-2012A *Terminal program for MMII* ________ **$89.95**

1200 BAUD SMART CAT - Novation
103/212 Smart Cat & 103 Smart Cat, 1200 & 300 baud, built-in dialer, auto re-dial if busy, auto answer/disconnect, direct connect, LED readout displays mode, analog/digital loop-back self tests, usable with multi-line phones.
IOM-5241A *300 baud 103 Smart Cat* ________ **$229.95**
IOM-5251A *1200 baud 212/103 Smart Cat* ________ **$549.95**

J-CAT™ MODEM - Novation
1/5 the size of ordinary modems, Bell 103, manual or auto-answer, automatic answer/orginate, direct connect, built-in self-test, two LEDs and audio beeps provide status information.
IOM-5261A *Novation* ________ **$149.95**

NEW! CP/M PLUS 3.0

CP/M 3.0 is Digital Research's latest version of the industry standard disk operating system. It features many performance improvements, such as intelligent record buffering, improved directory handling, "HELP" facility, time date stamping of files and many more improvements. AND A TREMENDOUS INCREASE IN SPEED!!! it is fully CP/M 2.2 compatible and requires no changes to your existing application software. Available only to Versafloppy II owners with SBC-200 CPU's
SFC-55009059F *Unbanked, RS232* ________ **$250.00**
SFC-55009058F *Unbanked, VDB8024* ________ **$250.00**
SFC-55009057F *Banked, RS232* ________ **$250.00**
SFC-55009056F *Banked, VDB8024* ________ **$250.00**
SFC-55009057D *Manual* ________ **$50.00**

CPU 8086/87 - CompuPro
CPU 8086/87, based around Intel's highest performance 8086 sixteen bit processor, also includes sockets for their 8087 math co-processor, and 80130 Operating System Firmware compoent.
CPU-70520A *CPU 8086/87 A & T* ________ **$699.00**
CPU-70520C *CPU 8086/87* ________ **$764.00**

S-100 MAIN FRAMES AND DISK CABINETS ON SALE

S-100 MEMORY BOARDS

64K STATIC RAM - Jade
Uses new 2K x 8 static RAMs, fully supports IEEE 696 24 bit extended addressing, 200ns RAMs, lower 32K or entire board phantomable, 2716 EPROMs may be subbed for RAMs, any 2K segment of upper 8K may be disabled, low power typically less than 500ma.
MEM-99152B *Bare board* ________ **$49.95**
MEM-99152K *Kit less RAM* ________ **$99.95**
MEM-32152K *32K kit* ________ **$199.95**
MEM-56152K *56K kit* ________ **$289.95**
MEM-64152K *64K kit* ________ **$299.95**
Assembled & Tested ________ **add $50.00**

RAM 17 - CompuPro
Meets or exceeds all IEEE 696/S-100 specifications including timing - works up to and including 10 MHz with 8088/86 CPUs. Guaranteed to perform flawlessly with any IEEE 696/S-100 extended addressing specification.
MEM-64180A *3RAM 17, 64K A & T* ________ **$459.95**
MEM-64180C *RAM 17, 64K CSC* ________ **$549.95**

RAM 16 - CompuPro
There is an alternative to quirk-prone dynamic memories: thanks to low power consumption, high speed operation, fully static technology, and suitability for 8 or 16 bit systems, RAM 16 reprsents a new high in performance for sophisticated S-100 systems.
MEM-32180A *RAM 16, A & T* ________ **$495.00**
MEM-32180C *RAM 16, CSC* ________ **$595.00**

RAM 21 - CompuPro
Operates at 12 MHz with 8088, 8086, 68000, 80286, and 16032 type CPUs, extremely low power consumption, meets or exceeds all IEEE 696/S-100 specifications, including timing.
MEM-12810A *RAM 21 A & T* ________ **$995.00**
MEM-12810C *RAM 21, CSC* ________ **$1125.00**

DISK TUBS

Diskette Storage Boxes
Clear Plexiglas storage boxes for up to 75 Diskettes
MMA-505 *Holds 75 5¼" Diskettes* ________ **$19.95**
MMA-508 *Holds 50 8" Diskettes* ________ **$24.95**

EPROM ERASERS

ULTRA-VIOLET EPROM ERASERS
Inexpensive erasers for industry or home.
XME-3100A *Spectronics w/o timer* ________ **$69.50**
XME-3101A *Spectronics with timer* ________ **$94.95**
XME-3200A *Logical Devices* ________ **$49.95**

Visit Our Stores In
Los Angeles
Orange County
Dallas
San Diego
San Jose

JADE
Computer Products
4901 West Rosecrans, Hawthorne, California 90250

CIRCLE 16 ON READER SERVICE CARD

A General-Purpose Graphic Plotting Package

A plotting interface—written for Basic, Pascal/Z, and JRT Pascal to drive a BASE2/IDS-560 printer—that can be easily modified for other printers and other languages

by David H. Freese, Jr.

There are many excellent dot matrix graphic printers that are underused simply because there is inadequate software available to extend high-level languages with plot functions. Those of us who first learned programming on a mainframe computer (with Fortran and directly callable PLOT funtions) may yearn for those good old days when we need graphic output now. Well, yearn no longer—contained herein is all you need to provide yourself with a substantial plotting package.

Functions provided

The subroutines contained in this group of assembly-level programs provide the interface between a Basic, Pascal/Z, or JRT Pascal program and the BASE2/IDS-560 graphics printers. These subroutines are designed to be called from Basic, using function calls similiar to those found in Fortran, and from Pascal using external procedures. The functions provided include:

a. pixel plotting
b. data point plotting
c. line plotting
d. circle plotting
e. axis plotting
f. initialization of plot map, and
g. transfer of plot map to the printer

The assembly code is designed to execute under CP/M. Conditional assembly is provided to allow these subroutines to operate under various conditions. These include Pascal/Basic linkage, BASE2/IDS560 printer drivers, and relocation for the CDR Systems, Heath/Zenith, and Lifeboat Associates versions of CP/M. The routines can be easily modified for different operating systems by redefining the equates associated with the memory allocation. These routines have been successfully transferred to a TRS-80 Model I.

Linkage to other languages

While these subroutines are shown with linkage to Basic and Pascal, it is quite easy to extend them to other high-level languages. The entrance code for each subroutine and the passing of parameters from the high-level language is specified in the listings. It is only necessary to rewrite these to provide the recognition of multiple parameters from the language in question. For example, the subroutines could have been written to use the CALL function in Microsoft's MBasic. This would also have been compatible with Microsoft's Fortran compiler. However, those who still use OBasic would then be unable to use the subroutines without considerable rewriting. (Microsoft sometimes refers to their earlier version (4.5) of Basic as OBasic, as opposed to the current version (5.4) which they refer to as MBasic.)

It is easy to provide the interface to printers other than the BASE2 or the IDS560. The subroutine that transfers the plot image to the printer will have to be modified to reflect the specific requirements of your printer.

Background

The BASE2 and IDS-560 graphic printers can be operated in an alphanumeric mode or a graphic mode. In either mode the printer can be configured to print in several horizontal and vertical densities. Horizontal densities are specified in characters/line (or dots/line), and vertical densities are specified in half-dots/inch. For a complete description of the BASE2 or IDS-560, refer to the respective manuals.

The selection of 14 half-dots/inch vertical density for the BASE2 printer will guarantee a uniform distribution of dots in the vertical direction without overlap or interline spaces. The selection of 96 characters/line (or 576 dots/line) guarantees the same dot density in the horizontal direction. The graphic routines are based on this selection of horizontal and vertical dot densities.

The vertical and horizontal density is fixed at 84 dots/inch in the IDS560, and no special setup codes are needed to insure proper graphics output.

When operating in the graphics mode, the printer accepts data in the form of ASCII characters, each of which represent seven vertical dots. Each successive character contains the next horizontal position of seven vertical dots. Each dot is printed or skipped, depending on the value of its corresponding bit in the ASCII character. The uppermost dot is specified by bit 0, and the lowermost dot by bit 6. Bit 7 is not used by the printer and can be either 1 or 0. The BASE2 prints the line of graphics when all 576 ASCII characters have been received. Printing commences when the ETX DC4 sequence is received by the IDS-560. The printer remains busy until that line has been printed, and will then accept another line of characters. For example, the following sequence would print the equivalent of the letter K:

David H. Freese, Jr., Clermont Computer Consultants, R.D. 1—Box 316, Cape May Court House, NJ 08210

```
1 0 0 1
1 0 1 0
1 1 0 0
1 0 1 0
1 0 0 1
0 0 0 0
0 0 0 0
| | | |
| | | --- char(n+3): &H11, or &H91
| | ----- char(n+2): &H0A, or &H8A
| ------- char(n+1): &H04, or &H84
--------- char(n+0): &H1F, or &H9F
```

The BASE2 graphics printer is not an *x-y* plotter, since a reverse line feed cannot be implemented on this machine. The IDS-560 does have the capability to reverse line feed and/or to move the print head to an absolute vertical and horizontal position on the paper. Graphics output would be slow using absolute print head movement.

Two methods of creating the graphics image are available:

a. compute the status of each bit as it is printed, or

b. create an image of the entire plot area and transfer it to the printer when it is complete.

Method (a) minimizes the memory requirements, but requires that any equation being plotted be evaluated for each and every dot in the plot space. For a plot 300 x 300 pixels in area, this means 90,000 evaluations of an equation, possibly using floating point arithmetic and transcendental functions. The result is an extremely slow plotting routine.

Method (b) minimizes the plotting time, but requires a significant amount of memory to store the image. A plot area 300 x 300 will require 11,250 bytes, an area 400 x 400 will require 20,000 bytes. Reducing the area results in both a smaller printed plot and reduction in the resolution of the plot. It is necessary to find a compromise that has sufficient resolution/size and a moderate memory requirement.

The second method has been selected for this application. The subroutines have been written in a manner that will allow a change in the desired plot area by a change in one variable—the number of lines/plot. A selection of 43 lines/plot results in a plot area 301 x 301 and occupies 11,438 bytes of memory, while a selection of 59 lines/plot results in a plot area 413 x 413 and occupies 21,476 bytes of memory. The choice of a plot area of 413 x 413 will be approximately 5″ per side on the IDS-560.

Passing parameters from the high-level language

With the exception of the initialization and transfer functions, the plot functions all require multiple integer parameters to be passed by the high-level language. A language such as Pascal or Fortran maintains multiple parameter lists on a stack. Whether the stack contains the actual value or a pointer to the storage location of the parameter is a function both of the function Declaration and the specific compiler/interpreter implementation. In order to maintain compatibility with as many versions of Basic as possible, the USR function will be used to gain access to the plot functions. The older versions of Basic allow only a single parameter to be passed via the USR function call. It will be necessary to subvert this limitation by using additional Basic functions normally used during random disk input/output.

Linkage to Pascal/Z

Parameters are passed by Pascal/Z on the stack. The stack is organized in the following way for integer/byte values:

```
p1 (2 bytes): integer,
p2 (2 bytes): integer,
.
.
pi (1 byte):  byte/boolean,
.
.
pn (1/2 bytes)
return address                  ; stack pointer
```

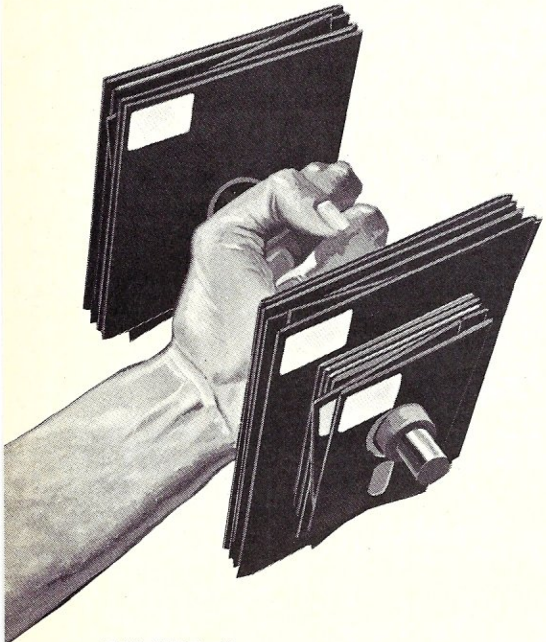
The Pascal program must declare the plotting subroutines to be external and designate the proper number of passed parameters. Each subroutine removes its allocated parameters and restores the return address to the stack. No check is made for improper precedure calls.

Linkage to JRT Pascal

JRT Pascal uses two stacks, one to pass parameters and one to contain global variables. Both functions are needed to support the graphics plotting package. JRT Pascal treats the disk as an extension of memory. When an external procedure or function is declared, it is handled as follows: During the compile process, a list of external procedures is made a part of the program intermediate file. During execution, the interpreter loads the required external procedure the first time it is invoked. That procedure may then either remain in memory or be purged, depending on the dynamic memory requirements of the program. In this manner the disk is treated as an extension of memory. Each procedure in the plotting package could have been separately assembled and linked, but

To create the graphics image, we may either compute the status of each bit as it is printed (slow), or create an image of the entire plot area by using a method that minimizes plotting time but requires a significant amount of memory.

WHY YOU NEED MICROSHELL™: NEW POWER FOR CP/M® SYSTEMS



PHASE 1: "LITTLE" FEATURES THAT ADD UP TO BIG TIME-SAVERS

You appreciate MicroShell from the moment you run it. Suddenly your CP/M computer seems much smarter, more powerful, more tolerant of your errors. That's because MicroShell adds features to CP/M normally found only on more powerful operating systems.

You spend a lot of time with your computer. MicroShell makes your time more productive. Take the warm-start after you run a program or when you change disks. MicroShell eliminates it — you can run another program immediately. Change a disk — even in drive A — and a quick command logs it in. No more time lost booting up — no more juggling "system" disks with CP/M on them!

Or how about finding files? Under CP/M you have to know what disk the file is on and then change disks or preface the file name with the disk drive. And if you have a hard disk and are using CP/M user numbers, you also need the user number the file is in. MicroShell automatically searches for files on all disks, or selected disks — your option. And it works even when a program calls for a file. So you can run WordStar, for instance, from any disk or user number and MicroShell finds the necessary WordStar files.

Running several programs in a row? MicroShell lets you put a string of CP/M commands on one line. You end the distraction of entering commands, waiting for execution, then entering the next command.

Make a mistake? With CP/M you have to start over. Add MicroShell and you can edit the last command line (with simple WordStar commands). MicroShell saves the last command string, so you can run through a series several times — or change commands slightly and run again — without retyping.

MicroShell doesn't affect how your CP/M runs. It helps CP/M, but it doesn't get in the way. And your system probably never misses the 8K bytes MicroShell uses because most all popular programs run with 48K. So in a 64K CP/M system, there's plenty of room. For the rare, big program, it's easy to exit MicroShell, run the program and re-enter MicroShell.

You appreciate MicroShell from the start. It has powerful features so easy-to-use that the MicroShell manual tells you "How to Run MicroShell Without Reading the Manual".

PHASE II: NEVER RE-ENTER DATA AGAIN

But please — read all 75 pages of the manual and get to know MicroShell in depth. You appreciate MicroShell more and more when you do the big projects you bought your micro for.

Take the time you turn a spreadsheet analysis into a table for a report. Without MicroShell you have to re-enter data in a file for your word processing software. Why the duplication of effort? MicroShell lets you *redirect* the output of any program to a file — no duplicated keystrokes.

MicroShell captures what appears on your CRT and sends it anywhere. Send data to your printer for hard-copy or save it on disk. Or capture that report your accounting program sends to the printer. MicroShell tames your data-handling jobs — it even "slows down" to feed programs that process input slower than MicroShell sends it.

When you review your work, MicroShell has a "TYP" command that displays 23 lines (an average CRT screen) then stops until you hit a key. If you've ever scanned a long document whipping by at 10 or 20 lines per second, you know you need this feature. Or have you tried to hit control-S in time to stop the display and missed?

For more information about MicroShell, see the following reviews:
Christopher Kern, BYTE, December, 1982
Alan R. Miller, Interface Age, July, 1982
David Fiedler, Microsystems, January, 1983

PHASE III: THE SYSTEM EVOLVES WITH YOU

If you're in computers for the long haul, MicroShell will grow on you. MicroShell helps you teach new users. And it lets you develop new capabilities in a clear and logical way.

MicroShell lets you set up "menus" to make your system "friendly" for non-technical users. And fast — no slow CP/M "submit" files used. As you develop your own software, MicroShell's "pipe" feature lets you grow into ever more sophisticated applications. It links simple programs together by directing the output of one program to the input of another.

Just as MicroShell lets you combine commands on one line, it lets you save any command series in a file. MicroShell reads and executes commands directly from the file — no slow, intermediate steps. For example, you can compile, link and run a program with one command, "compile filename". MicroShell executes all the commands from a file you've built.

And you can call other command files from within a command file to enlarge on what you've built. You can set conditional commands with an "IF" statement. Or you can set command files to call for input from a user and act on it. Before you know it, you have a system much more powerful and friendly than bare CP/M.

SO DON'T WASTE ANY MORE TIME WITH YOUR CP/M SYSTEM — ADD MICROSHELL TO IT TODAY.

ORDER LINE 800-368-3359

Available for all popular 8080, 8085 or Z80, CP/M 2.2 systems on 8" or 5¼" diskette.

Price: $150.00 ($25.00 manual only). Foreign orders add $20.00 for Air Mail shipment.

NEW GENERATION SYSTEMS, inc.

2153 Golf Course Drive
Reston, VA 22091
(703) 476-9143

CIRCLE 19 ON READER SERVICE CARD

Graphic Plotting Package continued . . .

I chose to keep the entire package in a single external file. This decision forces the external plotting procedures to be entered with a fixed parameter list.

JRT Pascal places data in the parameter stack in the following manner:

```
value parameters:  present on the stack,

      integer:  2 bytes
      boolean:  2 bytes
      char:     1 byte
      set:      16 bytes
reference parameters: address of the variable
      address:  2 bytes
```

The external plotting procedure is declared in the following manner:

```
procedure JPLOT(cmd: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;

    The data stack after the procedure call JPLOT will be:
```

The data stack after the precedure call JPLOT will be

```
xx          (DE) -> current stack address
0F00        16 bit integer = length of stack = 15
p7L,p7H     16 bit integer = p7
p6L,p6H     16 bit integer = p6
p5L,p5H     16 bit integer = p5
p4L,p4H     16 bit integer = p4
p3L,p4H     16 bit integer = p3
p2L,p2H     16 bit integer = p2
p1L,p1H     16 bit integer = p1
'X'         8 bit ASCII character
            a,A,p,P,l,L,d,D,c,C,i,I,t,T
.
.
.
image       (HL + 6) -> address of first global variable
```

Linkage to a Basic program

Parameters are passed in an ordered list, which is contained in a string variable passed by Basic. Each of the parameters in the string is created by the use of MKI$(a) and concatenated to the next parameter. The following Basic calling procedure is recommended:

```
100   DEF USR1=&HXXXX
110   DEF FNPLOT(X1,Y1,X2,Y2)=
                  LEN(USR1(MKI$(X1)+MKI$(Y1)+MKI$(X2)+MKI$(Y2)))
.
.
.
1000 ZZ=FNPLOT(A,B,C,D)      ' execute the funtion
```

(**Note:** *The use of the LEN() function prevents Basic from assigning an additional string variable to string space. This reduces string clutter and subsequent time loss when Basic collects garbage and reallocates string space.)*

Description of subroutines

Axis is a subroutine that plots the *x,y* axis pair and tick marks at required locations. A grid consisting of single points spaced at the major tick marks is optional. The calling routine is set up as follows:
When called by Basic:

```
DEF USRk = &Hkkkk
DEF FNAXIS(X0,Y0,XA,YA,XB,YB,GRID) =
              LEN(USRk(MKI$(X0) + MKI$(Y0) + MKI$(XA)
              + MKI$(YA) + MKI$(XB) + MKI$(YB) + MKI$(GRID)))
ZZ=FNAXIS(0,0,5,5,10,10,1) :' execute Axis
```

```
;**********************************************************
;
;  GENERAL PURPOSE GRAPHICS PRINTER PLOTTING PACKAGE
;
;  Plotting Subroutines/Procedures
;     compatible with the following high level languages:
;
;     . BASIC,
;     . PASCAL/Z,
;     . and JRT PASCAL
;
;  which support the following printers
;
;     . BASE-2
;     . IDS-560
;
;  10 February 1983
;
;  Ver: 2.2
;
;  Courtesty:    Clermont Computer Consultants
;                RD 1 Box 316
;                Cape May Court House, NJ 08210
;                  (609) 263 7511
;
;**********************************************************
;
        .Z80
;
;       EQUATES FOR ALL SUBROUTINES
;
TRUE    EQU     1
FALSE   EQU     0
;
LIFEBT  EQU     FALSE
ZENITH  EQU     FALSE
CDR     EQU     TRUE            ; assemble for CDR SYSTEMS BIOS
;
BASE2   EQU     FALSE
IDS560  EQU     TRUE            ; assemble for IDS-560
;
BASIC   EQU     FALSE
PAS.Z   EQU     FALSE
PAS.J   EQU     TRUE            ; assemble for JRT PASCAL
;
        IF      LIFEBT
BASE    EQU     4200H
        ENDIF
;
        IF      NOT LIFEBT
BASE    EQU     0000H
        ENDIF
;
BDOS    EQU     BASE+5
;
;==========================================================
;
;       Change LINES equate to produce a plot area of a given size
;
;       LINES = int(#dots/7) + 1
;
LINES   EQU     59              ; produces 413 x 413 image
;
;==========================================================
```

Graphic Plotting Package continued . . .

Seven parameters are passed:

```
X0, Y0: intersection of axis
XA, YA: minor tick interval
XB, YB: major tick interval
GRID:    0 suppress grid marks
         1 plot grid marks at major tick intervals
```

When called by Pascal/Z:
Declaration:

```
procedure axis(x0,y0,xa,ya,xb,yb: integer  grid: boolean); external;
```

Invoking:

```
axis(0,0,2,2,10,10,true);
```

When called by JRT Pascal:
Declaration:

```
procedure jplot(cmd:char;p1,p2,p3,p4,p5,p6,p7:integer); extern;
```

Invoking:

```
jplot('a',0,0,2,2,10,10,1);
```

Line is a subroutine that plots a line between the pair of ordered points ($x1,y1$) to ($x2,y2$). The calling routine is set up as follows:
When called by Basic:

```
DEF USRi = &Hiiii
DEF FNLN(x1,y1,x2,y2) = LEN(USRi(MKI$(x1) + MKI$(y1) +
                        MKI$(x2) + MKI$(y2)))
ZZ=FNLN(5,10,100,110) :' execute Line
```

When called by Pascal/Z:
Declaration:

```
procedure line(x1,y1,x2,y2: integer); external;
```

Invoking:

```
line(5,10,100,110);
```

When called by JRT Pascal:
Declaration:

```
procedure jplot(cmd:char; p1,p2,p3,p4,p5,p6,p7: integer); extern;
```

Invoking:

```
jplot('L',5,10,100,110,0,0,0);
```

Point is a subroutine that plots a square centered at the point x,y. The calling routine is set up as follows:
When called by Basic:

```
DEF USRn=&Hnnnn
DEF FNPOINT(X,Y)=LEN(USRn(MKI$(X)+MKI$(Y))
ZZ=FNPOINT(50,100) :' execute Point
```

When called by Pascal/Z:
Declaration:

```
procedure point(x,y: integer); external;
```

Invoking:

```
point(50,100);
```

When called by JRT Pascal:
Declaration:

```
procedure jplot(cmd:char; p1,p2,p3,p4,p5,p6,p7: integer); extern;
```

Invoking:

```
jplot('d',50,100,0,0,0,0,0);
```

```
;
DOTS    EQU     LINES*7         ; number of vertical dots in plot
NLINES  EQU     DOTS/8 + 1      ; number of lines in image area
LENGTH  EQU     DOTS            ; number of horizontal dots in plot
;
        IF      BASE2
;
FILL1   EQU     (576-LENGTH)/2  ; fill for left margin
FILL2   EQU     576-LENGTH-FILL1; 576 dots per line in 96 char/line mode
        ENDIF
;
        IF      IDS560
;
FILL1   EQU     126             ; 1.5 inch left margin
        ENDIF
;
AREA    EQU     NLINES*LENGTH   ; number of bytes for plot area
;
        IF      LIFEBT
TMEM    EQU     0DF00H          ; lower bound of CP/M (48K ORG 4200H)
        ENDIF
;
        IF      NOT LIFEBT
TMEM    EQU     0DC00H          ; lower bound of ORG 0 versions
        ENDIF
;
        IF      BASIC
;
BOTTOM  EQU     TMEM-AREA-06EAH ; leave room for prog + plot area
                                ; and start at next lowest page boundary
ORIGIN  EQU     (BOTTOM AND 0FF00H)
        ENDIF
;
        IF2
        IF      PAS.Z
;
        .PRINTX/        PASCAL-Z version/
        ENDIF
;
        IF      PAS.J
;
        .PRINTX/        JRT PASCAL version/
        ENDIF
;
        IF      BASIC
;
        .PRINTX/        BASIC version/
        ENDIF
;
        IF      BASE2
;
        .PRINTX/        For BASE-2 printer/
        ENDIF
;
        IF      IDS560
;
        .PRINTX/        For IDS-560 printer/
        ENDIF
;
        ENDIF
;
        IF      BASIC
;*****************************************************************
;
;       Subroutine to link the parameters passed from BASIC
;
DTLNK:
;
```

Plot is a subroutine that sets the pixel specified by *x,y*. The calling routine is set up as follows:

When called by Basic:

```
DEF FNUSRi=&Hiiii
DEF FNPLOT(X,Y)=LEN(USRi(MKI$(X) + MKI$(Y)))
ZZ=FNPLOT(2,5) :' execute Plot
```

When called by Pascal/Z:

Declaration:

```
procedure plot(x,y: integer); external;
```

Invoking:

```
plot(2,5);
```

When called by JRT Pascal:

Declaration:

```
procedure jplot(cmd: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;
```

Invoking:

```
jplot('p',2,5,0,0,0,0,0);
```

Init is a subroutine which clears the plot memory. Callable from Basic by:

```
DEF USRq=&Hqqqq
DEF FNCLRPLT = USRi
ZZ=FNCLRPLT ' execute clear plot
```

When called by Pascal/Z:

Declaration:

```
procedure init; external;
```

Invoking:

```
init;
```

When called by JRT Pascal:

Declaration:

```
procedure jplot(cmd: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;
```

Invoking:

```
jplot('i',0,0,0,0,0,0,0);
```

XFR is a subroutine that transfers the plot memory to the line printer.

Callable from Basic by:

```
DEF USRm=&Hmmmm
DEF FNXFR = USRi
ZZ=FNXFR  'execute transfer
```

When called by Pascal/Z:

Declaration:

```
procedure xfrplt; external;
```

Invoking:

```
xfrplt;
```

When called by JRT Pascal:

Declaration:

```
procedure jplot(cmd: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;
```

Invoking:

```
jplot('t',0,0,0,0,0,0,0);
```

Circle is a subroutine that plots a circle of radius *r*, centered at location *x,y*. The calling routine is: When called by Basic:

```
DEF USRk = &Hkkkk
DEF FNCIRCLE(X,Y,R) = LEN(USRk(MKI$(X) + MKI$(Y) + MKI$(R)))
.
.
ZZ=FNCIRCLE(25,25,10)
```

```
        LD      A,(DE)          ; fetch # bytes
        SLA     B               ; 2*(# parameters)
        CP      B               ; are they =?
        JR      NZ,STERR        ; no, an error in BASIC
        INC     DE              ; point to string address
        EX      DE,HL
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        EX      DE,HL           ; HL points to first byte in string
        LD      DE,P1
DTLNK1: LD      A,(HL)          ; transfer 2*(# param's) bytes of data
        LD      (DE),A
        INC     HL
        INC     DE
        DJNZ    DTLNK1
        RET
;
        ENDIF
        IF      PAS.J
;**********************************************************************
;
;       PARAMETER LINKAGE ROUTINE FOR JRT PASCAL
;
;       All calls to plot functions must be via this parameter linkage
;       routine.  JRT PASCAL expects each external .INT file to contain
;       a single unique function or procedure.  All of the plot procedures
;       must be treated as a single entity with one entry parameter acting
;       as a pointer to the desired sub-procedure.
;
JPLOT:
;       entry code
;
;       DEFB    95,06,00        ; int vmcode
        DEFB    92              ; lpn vmcode
        DEFB    00              ; mode vmcode
;
;       hl = base
;       de = current
;       bc = tos
;
        LD      BC,6
        ADD     HL,BC           ; HL = IMAGE address
        LD      (IMGPTR),HL     ; store for later use
;
        EX      DE,HL
        DEC     HL
        DEC     HL              ; point to length
        LD      A,(HL)          ; get length
        CP      0FH             ; see if correct
        JP      NZ,JERR         ; bad procedure call
        DEC     HL              ; do each parameter in reverse order
        LD      D,(HL)
        DEC     HL
        LD      E,(HL)
        LD      (P7),DE         ; DE = parameter 7
        DEC     HL
LD      D,(HL)
DEC     HL
LD      E,(HL)
LD      (P6),DE         ; DE = parameter 6
DEC     HL
LD      D,(HL)
DEC     HL
LD      E,(HL)
LD      (P5),DE         ; DE = parameter 5
DEC     HL
LD      D,(HL)
DEC     HL
```

When called by Pascal/Z:
Declaration:

```
procedure circle(x,y,r: integer); external;
```

Invoking:

```
circle(25,25,10);
```

When called by JRT Pascal:
Declaration:

```
procedure jplot(cmd: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;
```

Invoking:

```
jplot('c',25,25,10,0,0,0,0);
```

Next month, in Part 2, David Freese will discuss how to install the program, as well as how to link it to Basic, Pascal/Z and JRT Pascal, and will give some typical examples. The source code is available from the author in 8″ SSSD, 5¼″ Osborne, 5¼″ CDR, and 5¼″ Heath/Zenith formats. The cost is $15 prepaid.

```
        LD      E,(HL)
        LD      (P4),DE         ; DE = parameter 4
        DEC     HL
        LD      D,(HL)
        DEC     HL
        LD      E,(HL)
        LD      (P3),DE         ; DE = parameter 3
        DEC     HL
        LD      D,(HL)
        DEC     HL
        LD      E,(HL)
        LD      (P2),DE         ; DE = parameter 2
        DEC     HL
        LD      D,(HL)
        DEC     HL
        LD      E,(HL)
        LD      (P1),DE         ; DE = parameter 1
        DEC     HL
        LD      A,(HL)          ; A = type
        CP      'A'
        JP      Z,AXIS
        CP      'a'
        JP      Z,AXIS          ; vector AXIS procedure
        CP      'L'
        JP      Z,LINE
        CP      'l'
        JP      Z,LINE          ; vector LINE procedure
        CP      'P'
        JP      Z,PLOT
        CP      'p'
        JP      Z,PLOT          ; vector PLOT procedure
        CP      'D'
        JP      Z,POINT
        CP      'd'
        JP      Z,POINT         ; vector POINT procedure
        CP      'I'
        JP      Z,INIT
        CP      'i'
        JP      Z,INIT          ; vector INIT procedure
        CP      'C'
        JP      Z,CIRCLE
        CP      'c'
        JP      Z,CIRCLE        ; vector CIRCLE procedure
        CP      'T'
        JP      Z,XFRPLT
        CP      't'
        JP      Z,XFRPLT        ; vector XFRPLT procedure
JERR:
        JP      STERR + 1       ; bad procedure call
;                                 print error message & return

        ENDIF
;*********************************************************************
;
;       Test & failure message subroutines
;
YTEST:
        LD      DE,DOTS-1       ; y <-- DOTS-1-y
        OR      A
        EX      DE,HL
        SBC     HL,DE
;
TEST:                           ; limits the bounds on x and y to
                                ; be  0 <= HL <= DOTS-1
        XOR     A               ; clear test flag
        LD      B,A
        LD      A,H
        AND     80H             ; is HL negative?
        JR      Z,NNEG          ; no
        LD      HL,0            ; yes, set lower limit of 0
        DEC     B               ; set test flag
        RET
NNEG:   LD      DE,DOTS-1       ; test HL>=DOTS
        OR      A
        EX      DE,HL
        SBC     HL,DE
        JR      NC,UPOK         ; no carry, DE within bounds
        LD      HL,DOTS-1
        DEC     B               ; set test flag
        RET
UPOK:   EX      DE,HL
        RET
;
;
STERR:  POP     HL              ; modify stack to eliminate the return
        LD      DE,STERR$       ; from DTLINK
MSG:    LD      C,9             ; print the buffer under CP/M
        JP      BDOS            ; and return to interpreter when done
;
STERR$: DEFB    13,10,'Improper Number of Parameters',13,10,10,'$'
;
;*********************************************************************************
;
;       INIT: image initialization subroutine
;
INIT:                           ; CALL & USR entry point, no parameters
        IF      PAS.Z
;
        ENTRY   INIT
        ENDIF
;
        LD      HL,(IMGPTR)     ; HL = first address of IMAGE
        PUSH    HL
```

2500AD SOFTWARE INC.

offers you a variety of Cross Assemblers for your Z-80 CP/M* System.

Z-8000 Cross Development Package $179.50

Instant Z-8000 Software! This package allows development and conversion of software for the Z8001, 8002, 8003 and 8004 based machines on a 64K Z-80 CP/M machine. This powerful package includes:

- a Z-80 to Z-8000 Assembly Language Source Code Translator
- an 8080 to Z-8000 Source Code Translator
- Z-8000 Macro Cross Assembler
- Linker and Loader
- COM to Hex File Converter
- a 100 page User Manual
- a Zilog Z-8000 Technical Reference Manual

The Translators provide Z-8000 source code from Intel 8080 or Zilog Z-80 source code. This source code expansion is from 2% to 11%. The Translator outputs a worksheet and a Z-8000 source file. The worksheets show each line of 8080/Z-80 code, with notes to help the programmer to optimize performance, and further lower code expansion. It even comments lines it adds! 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.

Z-80 Macroassembler $49.50

Power for larger programs! This 2500AD macro-assembler includes:

- Zilog Z-80 Macroassembler (with the same powerful features as all our assemblers)
- powerful linker that will link up to 400 files
- 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

6502 Macro Cross Assembler $79.50

6502 software on your Z-80 machine! This 2500AD software allows you to develop large, powerful assembly language 6502 programs on a Z-80 CP/M System. The program uses standard mnemonics with a straightforward Zilog Z-80 type syntax. This package includes:

- 6502 Macro Cross Assembler
- the powerful 2500AD Linker
- a 50 page User Manual

Other Macro Cross Assemblers available:

- Zilog Z-8 $79.50
- Intel 8748, 8749, etc. $79.50

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

Relocatable Code—the packages include a versatile Linker that will link up to 400 files together, or just be used for external reference resolution. 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 parenthesis.

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.

Plan English Error Messages—

System requirements for all programs

Z-80 CP/M 2.2 System with 64K RAM and at least a 96 column printer is recommended.

Z-8000 based versions of all programs are available. **Call 2500ADSOFTWARE, 303-752-4382.**

Coming soon! Call for details.

- 8086 Macro Cross Assembler
- 8080 and Z-80 to 8086 Translator
- 8086 Relocating Macro-assembler for CP/M 86
- all 2500AD products running under CP/M 86

I would like to order:

- ☐ Z-8000 Cross Development Package **$179.50**
- ☐ Z-80 Macroassembler **$ 49.50**
- ☐ 6502 Macro Cross Assembler **$ 79.50**
- ☐ Zilog Z-8 **$ 79.50**
- ☐ Intel 8748, 8749, etc. **$ 79.50**

Check one: total **$**______

- ☐ 8" Single Density
- ☐ 5¼" Osborne

shipping/handling ($6.50 per unit) **$**______

total order **$**______

Name________________

Company________________

Address________________

City________ State______ Zip______

Phone________ Ext.______

Make and model of computer system____________

☐ C.O.D. (2500AD pays C.O.D. charges)
☐ VISA or MasterCard #, Exp. Date (mo. yr.)

Signature________________

2500AD SOFTWARE INC
P.O. Box 441410 Aurora, CO 80014 303-752-4382

CIRCLE 17 ON READER SERVICE CARD

```
        POP     DE
        INC     DE              ; DE = second address of IMAGE
        LD      BC,AREA-1
        LD      (HL),0          ; set all bytes to 00
        LDIR
;
        IF      PAS.Z
;
        XOR     A               ; flag valid subroutine call
        ENDIF
;
        RET
;
;**********************************************************************
;
;       Plot Subroutine:
;
PLOT:                           ; USR entry point
        IF      PAS.Z
;
        ENTRY   PLOT
        POP     BC              ; fetch return address
        POP     HL              ; fetch y: integer
        LD      (P2),HL
        POP     HL              ; fetch x: integer
        LD      (P1),HL
        PUSH    BC              ; restore return address
        ENDIF
;
        IF      BASIC
;
        LD      B,2             ; transfer data from string buffer
        CALL    DTLNK           ; and test/limit to boundaries
        ENDIF
;
        LD      HL,(P1)
        CALL    TEST
        BIT     7,B             ; if out-of-bounds
        JP      NZ,SETRET       ; do not plot point
        LD      (STARTX+2),HL
        LD      HL,(P2)
        CALL    YTEST
        BIT     7,B
        JP      NZ,SETRET
        LD      (STARTY+2),HL
        JP      SET1
;
SET0:   LD      HL,(STARTX+2)   ; entry point for subr call
        CALL    TEST            ; when point may be outside boundaries
        BIT     7,B             ; if out-of-bounds
        RET     NZ              ; do not plot point
        LD      (STARTX+2),HL
        LD      HL,(STARTY+2)
        CALL    TEST            ; if out-of-bounds
        BIT     7,B             ; do not plot point
        RET     NZ
        LD      (STARTY+2),HL
;
SET1:   LD      B,0             ; entry point for subr call
        LD      HL,(STARTY+2)   ; when point is within boundaries
;
;       Determine address and bit number of pixel
;               ADDR = LENGTH*(INT(Y/8) + X
;               BIT  = 2^(8 - (Y MOD 8))
;
        LD      DE,8
        OR      A
DIV8:   SBC     HL,DE
        JR      C,DONDIV
        INC     B
        JR      DIV8
DONDIV: ADD     HL,DE
        LD      C,L
        INC     B
        LD      DE,LENGTH
        LD      HL,(IMGPTR)
        OR      A               ; clear carry
        SBC     HL,DE           ; HL = IMAGE - LENGTH
SMULT:  ADD     HL,DE
        DJNZ    SMULT
        LD      DE,(STARTX+2)
        ADD     HL,DE           ; HL = ADDR
        LD      B,C
        INC     B
        LD      DE,TABLE-1
FINDT:  INC     DE
        DJNZ    FINDT
        LD      A,(DE)          ; A has proper bit set
        OR      (HL)
        LD      (HL),A
;
SETRET:
        IF      PAS.Z
;
        XOR     A               ; flag valid subroutine call
        ENDIF
;
        RET
;
TABLE:  DEFB    80H             ; bit position table
        DEFB    40H
        DEFB    20H
        DEFB    10H
        DEFB    08H
        DEFB    04H
        DEFB    02H
        DEFB    01H
;
;**************************************************************
;
;       Point Subroutine:
;
POINT:                          ; CALL entry point
        IF      PAS.Z
;
        ENTRY   POINT
        POP     BC              ; fetch return address
        POP     HL              ; fetch y: integer
        LD      (P2),HL
        POP     HL              ; fetch x: integer
        LD      (P1),HL
        PUSH    BC              ; restore return address
        ENDIF
;
        IF      BASIC
;
        LD      B,2
        CALL    DTLNK
        ENDIF
;
        LD      HL,(P1)
        CALL    TEST
        LD      (P1),HL
        LD      HL,(P2)
        CALL    YTEST
        LD      (P2),HL
        LD      HL,(P1)
```

```
        DEC     HL
        LD      (STARTX+2),HL
        LD      HL,(P2)
        DEC     HL
        LD      (STARTY+2),HL
        CALL    SET0
        LD      HL,(P1)
        LD      (STARTX+2),HL
        CALL    SET0
        LD      HL,(P1)
        INC     HL
        LD      (STARTX+2),HL
        CALL    SET0
        LD      HL,(P2)
        LD      (STARTY+2),HL
        CALL    SET0
        LD      HL,(P2)
        INC     HL
        LD      (STARTY+2),HL
        CALL    SET0
        LD      HL,(P1)
        LD      (STARTX+2),HL
        CALL    SET0
        LD      HL,(P1)
        DEC     HL
        LD      (STARTX+2),HL
        CALL    SET0
        LD      HL,(P2)
        LD      (STARTY+2),HL
        CALL    SET0
;
        IF      PAS.Z
;
        XOR     A                       ; flag valid subroutine call
        ENDIF
;
        RET
;
;***************************************************************************
;
;       Line Subroutine:
;
LINE:                                   ; CALL entry point
        IF      PAS.Z
;
        ENTRY   LINE
        POP     BC                      ; fetch return address
        POP     HL                      ; fetch y2: integer
        LD      (P4),HL
        POP     HL                      ; fetch x2: integer
        LD      (P3),HL
        POP     HL                      ; fetch y1: integer
        LD      (P2),HL
        POP     HL                      ; fetch x1: integer
        LD      (P1),HL
        PUSH    BC                      ; restore return address
        ENDIF
;
        IF      BASIC
;
        LD      B,4                     ; 4 parameters in the CALL list
        CALL    DTLNK                   ; fetch the values P1, P2, P3, P4
        ENDIF
;
        LD      HL,(P1)                 ; test for boundary conditions & transfer
        CALL    TEST                    ; to working storage
        LD      (STARTX+2),HL           ; X1 <-- P1
        LD      HL,(P2)
```

```
        CALL    YTEST
        LD      (STARTY+2),HL   ; Y1 <-- P2
        LD      HL,(P3)
        CALL    TEST
        LD      (ENDX+2),HL     ; X2 <-- P3
        LD      HL,(P4)
        CALL    YTEST
        LD      (ENDY+2),HL     ; Y2 <-- P4
        LD      HL,(ENDX+2)     ; test for ENDX = STARTX
        LD      BC,(STARTX+2)
        OR      A               ; 0-->carry
        SBC     HL,BC
        JR      NZ,LINE1        ; ENDS<>STARTX
        LD      HL,(ENDY+2)     ; same for ENDY, STARTY
        OR      A
        LD      BC,(STARTY+2)
        SBC     HL,BC
        JP      Z,SET1          ; co-resident points, just plot single point
;
LINE1:                          ; entry point for other subroutine calls
        LD      HL,(ENDX+2)     ; evaluate 32 bit fixed point
        LD      DE,(STARTX+2)   ; numbers: DIRX & DIRY
        OR      A
        SBC     HL,DE
        LD      (DIRX),HL       ; DIRX <- ENDX - STARTX; fractional part
        BIT     7,H             ; extend sign of .DIRX to integer part
        JR      Z,X0
        LD      HL,-1
        JR      X1
X0:     LD      HL,0
X1:     LD      (DIRX+2),HL     ; do same for DIRY
        LD      HL,(ENDY+2)
        LD      DE,(STARTY+2)
        OR      A
        SBC     HL,DE
        LD      (DIRY),HL
        BIT     7,H
        JR      Z,Y0
        LD      HL,-1
        JR      Y1
Y0:     LD      HL,0
Y1:     LD      (DIRY+2),HL
        LD      HL,(DIRX)       ; HL <- .DIRX
        LD      DE,(DIRY)       ; DE <- .DIRY
        LD      A,H
        AND     80H
        LD      B,A             ; B <- sgn(.DIRX)
        LD      A,D
        AND     80H
        LD      C,A             ; C <- sgn(.DIRY)
SHLFT:  SLA     L               ; multiply HL, DE by 2
        RL      H               ; until sign change occurs
        SLA     E               ; on one or the other
        RL      D
        LD      A,H
        AND     80H
        CP      B
        JR      NZ,DSHLFT
        LD      A,D
        AND     80H
        CP      C
        JR      NZ,DSHLFT
        JR      SHLFT
DSHLFT: LD      (DIRX),HL       ; restore .DIRX & .DIRY with one equal
        LD      (DIRY),DE       ; to 1/2 & the other less (magnitudes)
        LD      HL,8000H        ; HL <- 1/2
        LD      (STARTX),HL     ; round up STARTX, STARTY by 1/2
        LD      (STARTY),HL
NXTBL:  CALL    SET1
        OR      A
```

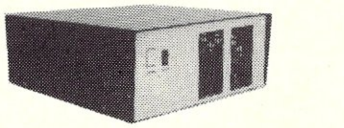
```
        LD      HL,(ENDX+2)
        LD      DE,(STARTX+2)
        SBC     HL,DE
        JR      Z,NXTBL2        ;STARTX = ENDX
NXTBL1: LD      HL,(STARTX)
        LD      DE,(DIRX)
        ADD     HL,DE           ;.STARTX = .STARTX + .DIRX
        LD      (STARTX),HL
        LD      HL,(STARTX+2)
        LD      DE,(DIRX+2)
        ADC     HL,DE           ;STARTX. = STARTX. + DIRX. + .CY
        LD      (STARTX+2),HL
        LD      HL,(STARTY)
        LD      DE,(DIRY)
        ADD     HL,DE           ;.STARTY = .STARTY + .DIRY
        LD      (STARTY),HL
        LD      HL,(STARTY+2)
        LD      DE,(DIRY+2)
        ADC     HL,DE           ;STARTY. = STARTY. + DIRY. + .CY
        LD      (STARTY+2),HL
        JR      NXTBL
NXTBL2: OR      A
        LD      HL,(ENDY+2)
        LD      DE,(STARTY+2)
        SBC     HL,DE
        JR      NZ,NXTBL1       ;STARTY. <> ENDY.
;
        IF      PAS.Z
;
        XOR     A               ; flag valid subroutine call
        ENDIF
;
        RET
;
; ***************************************************************************
;
;       CIRCLE Subroutine:
;
CIRCLE:
        IF      PAS.Z
;
        ENTRY   CIRCLE
        POP     BC
        POP     HL                      ; R VALUE
        LD      (P3),HL
        POP     HL                      ; Y VALUE
        LD      (P2),HL
        POP     HL                      ; X VALUE
        LD      (P1),HL
        PUSH    BC
        PUSH    IX
        PUSH    IY
        ENDIF
;
        IF      BASIC
;
        LD      B,3
        CALL    DTLNK
        ENDIF
;
        LD      HL,(P1)
        CALL    TEST
        LD      (P1),HL
        LD      HL,(P2)
        CALL    YTEST
        LD      (P2),HL
        LD      IX,SINTBL
        LD      IY,COSTBL
        LD      B,45
CIRC0:
        PUSH    BC              ; SAVE COUNTER
        LD      D,(IX+1)        ; SIN->(DE)
        LD      E,(IX)
        LD      BC,(P3)         ; R->(BC)
        CALL    MULT
        LD      (RSIN),DE       ; PROD->RSIN
        LD      D,(IY+1)
        LD      E,(IY)
        LD      BC,(P3)
        CALL    MULT
        LD      (RCOS),DE       ; PROD->RCOS
        LD      HL,(P1)         ;X->HL
        ADD     HL,DE
        CALL    CIRC1           ; X+RCOS ,Y+RSIN
        LD      DE,(RSIN)
        ADD     HL,DE
        CALL    CIRC2
; X+RSIN, Y+RCOS
        LD      DE,(RSIN)
        ADD     HL,DE
        CALL    CIRC1
        LD      DE,(RCOS)
        ADD     HL,DE
        CALL    CIRC2
; X-RSIN, Y+RCOS
        LD      DE,(RSIN)
        OR      A
        SBC     HL,DE
        CALL    CIRC1
        LD      DE,(RCOS)
        ADD     HL,DE
        CALL    CIRC2
; X-RCOS, Y+RSIN
        LD      DE,(RCOS)
        OR      A
        SBC     HL,DE
        CALL    CIRC1
        LD      DE,(RSIN)
        ADD     HL,DE
        CALL    CIRC2
; X-RCOS, Y-RSIN
        LD      DE,(RCOS)
        OR      A
        SBC     HL,DE
        CALL    CIRC1
        LD      DE,(RSIN)
        OR      A
        SBC     HL,DE
        CALL    CIRC2
; X-RSIN, Y-RCOS
        LD      DE,(RSIN)
        OR      A
        SBC     HL,DE
        CALL    CIRC1
        LD      DE,(RCOS)
        OR      A
        SBC     HL,DE
        CALL    CIRC2
; X+RCOS, Y-RSIN
        LD      DE,(RCOS)
        ADD     HL,DE
        CALL    CIRC1
        LD      DE,(RSIN)
        OR      A
        SBC     HL,DE
```

Retail Sales
123 East 200 South
Salt Lake City, Utah 84102

The Great Salt Lake Computer Company, Inc.

Mail Orders
P.O. Box 26816
Salt Lake City, Utah 84126

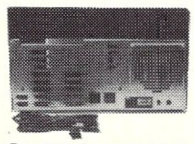
QT 5¼" MAINFRAME

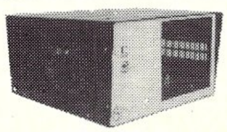


- Provisions for any 2 - 5¼" drives • 15 ea DB 25 cutout • 2 ea 50 pin • 2 ea 34 pin • 1 ea Centronic • EMI filter (fused) • 2 AC outlets • Avbl. with 6 - 8 or 12 slot motherbd • Power supply (+8V16A/±16V3A/+12V5A)

QTC-MF+MD8 (8 slot MB) **$720.00**
QTC-MF+MD12 (12 slot MB) **$765.00**

QT 8" MAINFRAME



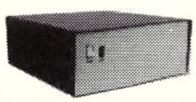


- Provision for any 2 - 8" drives (hard or floppy) • 15 ea DB 25 cutout • 2 ea 50 pin • 2 ea 34 pin • 1 ea Centronic • EMI filter (fused) • 2 AC outlets • Avbl with 6 - 8 or 12 slot motherbd
- Power supply (+8V16A/±16V3A/+5V6A/−5V1A/+24V6A)

Desk Top Version
QTC-MF+DD8 .... **$800.00**
QTC-MF+DD12 .... **$850.00**

Rack Mount Version
QTC-RMF-DD8 .... **$900.00**
QTC-RMF-DD12 .... **$950.00**

QT STANDARD MAINFRAME

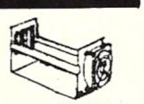


- Provisions for any 2 - 5¼" drives
- 15 ea DB 25 cutout • 2 ea 50 pin
- 2 ea 34 pin • 1 ea Centronic
- EMI filter (fused) • 2 AC outlets
- Avbl with 6-8-12-18 or 22 slot MB
- Power supply (+8V16A±16V3A)

Desk Top Version
QTC-MF+12 .... **$675.00**
QTC-MF+18 .... **$720.00**
QTC-MF+22 .... **$800.00**

Rack Mount Version
QTC-RMF+12 .... **$775.00**
QTC-RMF+18 .... **$820.00**
QTC-RMF+22 .... **$900.00**

QT S-100 CARD CAGES



- Made of anodized steel • Card guides for ea. MB • 1 - Indicates w/MB
- 2 - Indicates w/MB + 1 ea fan
- 3 - Indicates w/MB + 2 ea fans

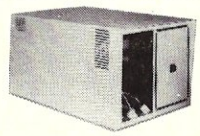
QTC-CC4	**$20.00**	CC12	**$60.00**
CC4-1	**$75.00**	CC12-1	**$165.00**
CC6	**$25.00**	CC12-2	**$200.00**
CC6-1	**$90.00**	CC12-3	**$225.00**
CC6-2	**$115.00**	CC18	**$75.00**
CC8	**$35.00**	CC18-1	**$220.00**
CC8-1	**$100.00**	CC18-2	**$250.00**
CC8-1	**$135.00**	CC18-3	**$280.00**

QT S-100 MOTHERBOARDS

- Silence Plus • Built in Termination • IEEE696 • Terminal strip for easy power connection

4 Slot Motherboards		**6 Slot Motherboards**	
QTC-MB4BB	**$20.00**	QTC-MB6BB	**$30.00**
QTC-MB4K	**$35.00**	QTC-MB6K	**$50.00**
QTC-MB4A	**$45.00**	QTC-MB6A	**$65.00**
8 Slot Motherboards		**12 Slot Motherboards**	
QTC-MB8BB	**$35.00**	QTC-MB12BB	**$40.00**
QTC-MB8K	**$65.00**	QTC-MB12K	**$95.00**
QTC-MB8A	**$95.00**	QTC-MB12A	**$135.00**
18 Slot Motherboards		**22 Slot Motherboards**	
QTC-MB18BB	**$65.00**	QTC-MB22BB	**$75.00**
QTC-MB18K	**$135.00**	QTC-MB22K	**$150.00**
QTC-MB18A	**$185.00**	QTC-MB22A	**$200.00**

QT 8" THINLINE MAINFRAME



- Provisions for 2 ea 8" thinline drives • 15 ea DB 25 cutout
- 2 ea 50 pin • 2 ea 34 pin • 1 ea Centronic
- EMI filter (fused) • 2 AC outlets
- Power supply (+8V16A/−5V/+24V6A+5V6A)

Cadillac Version
QTC-IMF+DD6C (6 slot MB) **$500.00**

Ford Version
QTC-IMF+DD6F (6 slot MB) **$389.00**

DISK DRIVES / DRIVE SUB ASSEMBLY

8" Disk Drives
801R Shugart SS/DD **$355.00**
851R Shugart DS/DD **$455.00**
DT8 (842) Oume DS/DD **$465.00**
BEST BUY M-2894-63 MIT DS/DD **$409.00**
(call for qty price)
M-2896-63 MIT 8" thinline DS/DD **$475.00**

5¼" Disk Drives
TM100-1 or B-51 for IBM **$200.00**
TM100-2 or B-52 for IBM (In stock) **$270.00**

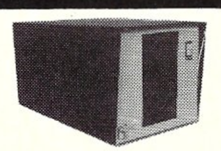
- Vertical or horizontal disk cabinet - 11"h × 11"w × 20"d 5"h × 17"w × 20"d

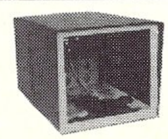
QTC-DDS+0 *w/2 ea SS/DD Siemens drives* **$595.00**
QTC-DDS+1 *w/1 ea DS/DD MIT 2894-63* **$695.00**
QTC-DDS+2 *w/2 ea DS/DD MIT 2894-63* **$1150.00**
QTC-DDS+3 *w/2 ea SS/DD 801R Shugart* **$975.00**
QTC-DDS+4 *w/2 ea DS/DD 8" thinline drives* .. **$1150.00**

OUR SALTIEST DEALS

Franklin Ace Computer **$919.00**
"Thinline" Apple Compatible **$235.00**
Disk Drive - ½ size/163 K/40 track
Dumpling CX/Grappler compatible **$99.00**
C. ITOH 8600 (80 cps ltr quality/
18 pin printhead/180 cps reg. mode .... **CALL**
FX80FT Epson **$569.00**
MX100FX Epson .... **CALL**
Cabinet for 2 ea. 5¼" disk drive w/p.s. **$89.00**
Toshiba P-1350
100 cps letter quality 200 cps reg. mode
Par or serial - full graphics - 24 pin printhead ... **$1575.00**
Bi-directional tractor **$275.00**
Sheet feeder **$1095.00**

QT DISK DRIVE CABINETS





"All in One" Vertical Disk Drive Cabinet
For 1) 2 ea or 4 ea 8" thinline drive
2) 2 ea standard 8" drives
3) 1 ea hard disk + 1 ea standard 8"
- Power supply (+5V6A/−5V1A/+24V6A)
- Positive pressure fan w/filter • EMI filter
- Power interface cable for any 8" drive

QTC-DDC88V *For 2 standard size 8" drives* **$325.00**
QTC-DDC88T *For 2 thinline 8" drives* **$325.00**
QTC-DD8V *For 1 ea 8" drive* **$275.00**

Horizontal Disk Drive Cabinet
For 2 ea 8" standard size drives
- Power supply (+5V6A/−5V1A/+24V6A)
- Interface cable for any 8" drive

QTC-DDC88H **$300.00**

TERMINALS / MONITORS

Televideo 925C **$700.00**
Televideo 970 .... **$1095.00**
BMC12A (15 MHZ) **$80.00**
BMC Color #9191 .... **$299.00**

CIRCLE 39 ON READER SERVICE CARD

PARTS

TR1602B	**$2.50 ea.**	TMS2716(5+12V)	**$3.95ea.**
2114L2	**$1.50 ea.**	TMS2732	**$4.95 ea.**
TMS2532	**$5.95 ea.**	TMS4164(200NS)	**5.95 ea.**
2716 (5V)	**$3.95 ea.**	TMS6116-4(200NS)	**$4.75 ea.**

MANY OTHERS IN STOCK

SIERRA DATA S-100 BOARDS

SDS-SBC-100-Z80(4mhz)
master 2 serial 2 par/floppy controller/64k ram . **$655.00**
SDS-SBC-100S-4mhz slave/2 serial
2 par/64k ram **$565.00**
SDS-ZSIO/4 - 4 serial port I/O bd **$235.00**
SDS-MUX - RS232 multiplexer bd **$235.00**
SDS-HDI-M - Hard disk bd for micropolis **$129.00**
SDS-CPM/B105 - cp/m for SBC100 w/BIOS **$150.00**
SDS-Turbodos - Multi-user for master & slaves ... **$645.00**

PRINTERS

Star-Gemini 10 .... **$335.00**
OK1-82A (120 cps) Serial + par **$389.00**
OK1-83A (120 cps) Serial + par **$585.00**
OK1-84A (200 cps) Serial **$1079.00**
OK1-92A (160 cps) **$519.00**
OK1-93A (160 cps) **$899.00**
NEC-8023A **$475.00**

BOARD SETS

Best Bare Board Set Available
QTC-SBC 2/4 CPU
QTC-EXP+III 256K Memory bd.
QTC-FDC 5/8 Floppy disk controller
Bare Board Set **$150.00**
1) Includes manuals & assembly instructions
2) Parts available 3) Monitor & B10S available

QT CLOCK/CALENDAR

S-100 or Apple • Time in hrs, min, sec. • AM/PM or Military Format • Date in Mo., Day, Yr, Day of Week & Leap Year recognition • 4 hard interrupts (1024 Hz, 1Hz, 1 min, 1 hr) • On board battery (will last 14 mos. w/no power on)

QTC-CCS-BB (S-100) **$60.00**
QTC-CCS-K (kit) for S-100 .... **$100.00**
QTC-CCS-A (A+T) for S-100 .... **$125.00**
QTC-CCA-BB (for Apple) **$40.00**
QTC-CCA-A (for Apple) **$125.00**

CPU BOARDS / MEMORY BOARDS / I/O

QTC-SBC 2/4 BB **$50.00**
QTC-SBC 2/4 A A+T **$265.00**
QTC-Z+80 BB **$28.00**
Teletek FDC-I A+T **$595.00**
Teletek Systemaster A+T **$695.00**

DYNAMIC (64K/256K or 1 MEG)
QTC-EXP+III Bare Bd. **$65.00**
QTC-EXP+III 64K A+T **$450.00**

STATIC
S-RAM 64K BB **$60.00**
S-RAM 64 A+T **$349.00**
S-RAM 128K A+T *(Fully IEEE696)* **$895.00**

QTC-I/O+BB *2 SER 3 PAR* A+T **$60.00**
QTC-I/O + A+T **$340.00**
QTC-ADA *ADA Converter* A+T **$375.00**
QTC-Dual GPIB-488 *IEEE 488 Interface bd.* A+T . **$475.00**

APPLE CORNER / DISKETTES

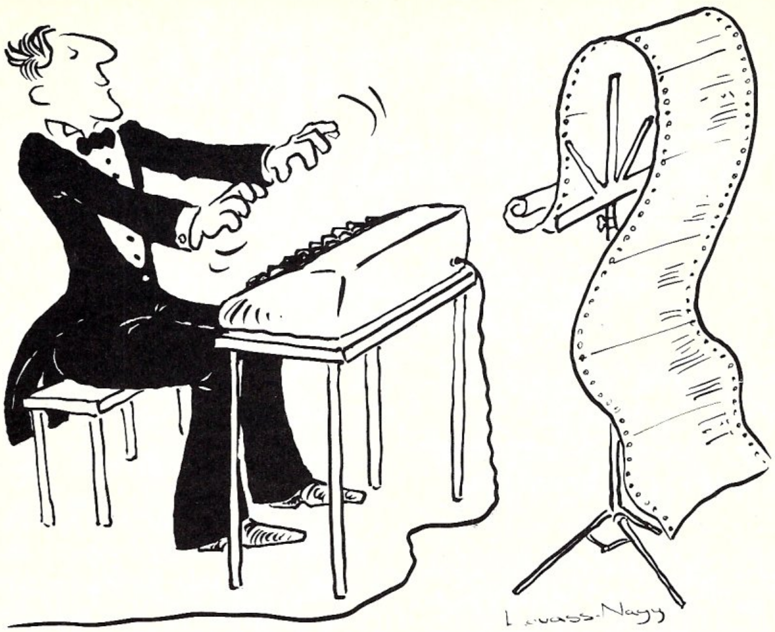
Micro-SCI Drive **$259.00**
w/controller .... **add $79.00**
Rana Elite I **$299.00**
Rana Elite II **$439.00**
Rana Elite III **$569.00**
Rana Controller **$99.00**
QT Apple Compatible Drive **$235.00**
5¼" soft sector (for Apple) .... **$19.00 for 10**
1 yr warranty - with hub ring
5¼" soft sector DS/DD .... **$28.00 for 10**
5¼" plastic library case .... **$2.50 ea, $20.00 for 10**
Flip file storage case *(stores 70 diskettes)* ...... **$19.00 ea**

In Continental U.S.
(800) 545-2633

Inside Utah
(801) 363-3314

We accept cash, checks, credit cards, or Purchase Orders from qualified firms and institutions. **Minimum prepaid order $15.00** Export cusotmers outside the U.S. or Canada please add 10% to all prices. **Prices and availability subject to change without notice.** Shipping and handling charges via UPS Ground 60¢/lb. UPS Air $1.00/lb. **minimum charge $3.00**

```
        CALL    CIRC2
; X+RSIN, Y-RCOS
        LD      DE,(RSIN)
        ADD     HL,DE
        CALL    CIRC1
        LD      DE,(RCOS)
        OR      A
        SBC     HL,DE
        CALL    CIRC2
;
        INC     IX
        INC     IX
        INC     IY
        INC     IY
        POP     BC
        DEC     B
        JP      NZ,CIRC0
;
        LD      HL,(P1)         ; X+R, Y
        LD      DE,(P3)
        ADD     HL,DE
        CALL    CIRC1
        CALL    CIRC2
; X-R, Y
        LD      DE,(P3)
        OR      A
        SBC     HL,DE
        CALL    CIRC1
        CALL    CIRC2
; X, Y+R
        CALL    CIRC1
        LD      DE,(P3)
        ADD     HL,DE
        CALL    CIRC2
;X, Y-R
        CALL    CIRC1
        LD      DE,(P3)
        OR      A
        SBC     HL,DE
        CALL    CIRC2
;
        IF      PAS.Z
;
        POP     IY
        POP     IX
        XOR     A
        ENDIF
;
        RET
;
CIRC1:  LD      (STARTX+2),HL
        LD      HL,(P2)
        RET
;
CIRC2:  LD      (STARTY+2),HL
        CALL    SET0
        LD      HL,(P1)
        RET
;
RSIN:   DEFW    0
RCOS:   DEFW    0
;
;
;       MULT: from ELECTRONICS/Feb 24, 1982 Designer's Casebook
;             article by Jerry L. Goodrich
;             performs a 2-byte by 2-byte integer multiply
;             (BC)*(DE)-->(DE),(HL)
;
MULT:
        LD      A,E             ;load lowest-order byte of multiplier
        PUSH    DE              ;save highes-order byte multiplier
        CALL    BMULT           ;do 1-byte multiply
        EX      (SP),HL         ;save lowest-order bytes product,get multiplier
        PUSH    AF              ;store highes-order byte of first product
        LD      A,H             ;load highest-order byte of multiplier
        CALL    BMULT           ;do second 1-byte multiply
        LD      D,A             ;position highest-order byte of product
        POP     AF              ;get highes-order byte of first product
        ADD     A,H             ;update third byte of product
        LD      E,A             ;and put in E
        JP      NC,NC1          ;don't incr D if no carry
        INC     D               ;incr D if carry
NC1:
        LD      H,L             ;relocate lowest-order bytes of sec. prod.
        LD      L,0
        POP     BC              ;get lowest-order bytes of sec. prod.
        ADD     HL,BC           ;get final product lowest-order 2 bytes
        JR      NC,NC2          ;done if no carry
        INC     DE              ;otherwise update highest-order 2 bytes
NC2:
        BIT     7,H             ;round up if frac part => .5
        RET     Z
        INC     DE
        RET
;
;       BMULT performs a 1-byte by 2-byte multiply
;       (A)*(BC)-->(A),(BC)
;
BMULT:
        LD      HL,0            ;zero partial product
        LD      DE,7            ;D=0,E=bit counter
        ADD     A,A             ;get first mulitplier bit
LOOP1:
        JP      NC,ZERO         ;zero-skip
        ADD     HL,BC           ;one-add multiplicand
        ADC     A,D             ;add carry to third byte of product
ZERO:
        ADD     HL,HL           ;shift product left
        ADC     A,A
        DEC     E               ;decrement bit counter
        JR      NZ,LOOP1        ;loop until done
        RET     NC              ;done if no carry
        ADD     HL,BC           ;otherwise do last add
        ADC     A,D
        RET                     ;and return
;
;       COSTBL and SINTBL are tables of cosine and sine values
;       specified as 16-bit fractions.  Each table is 45 units
;       (degrees) long.
;
COSTBL:
        DEFW    65526           ;1 DEG
        DEFW    65496           ;2
        DEFW    65446           ;3
        DEFW    65376           ;4
        DEFW    65287           ;5
        DEFW    65177           ;6
        DEFW    65048           ;7
        DEFW    64898           ;8
        DEFW    64729           ;9
        DEFW    64540           ;10
        DEFW    64332           ;11
        DEFW    64104           ;12
        DEFW    63856           ;13
        DEFW    63589           ;14
        DEFW    63303           ;15
        DEFW    62997           ;16
        DEFW    62672           ;17
        DEFW    62328           ;18
```



The Well-Tempered Cross-Assembler

Before Johann Sebastian Bach developed a new method of tuning, you had to change instruments practically every time you wanted to change keys. Very difficult.

Before Avocet introduced its family of cross-assemblers, developing micro-processor software was much the same. You needed a separate development system for practically every type of processor. Very difficult and very expensive.

But with Avocet's cross-assemblers, a single computer can develop software for virtually any microprocessor! Does that put us in a league with Bach? You decide

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 3 years of actual use. Ask NASA, IBM, XEROX or the hundreds of other organizations that use them. Every time you see a new microprocessor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on any computer with CP/M* and process assembly language for the most popular microprocessor families.

XASM05	6805	$200 each
XASM09	6809	
XASM18	1802	
XASM48	8048/8041	
XASM51	8051	
XASM65	6502	
XASM68	6800/01	
XASMF8	F8/3870	
XASMZ8	Z8	
XASM400....	COP400	
XASM75	NEC 7500	$500

(Coming soon: XASM68K .... 68000)

Turn Your Computer Into A Complete Development System

Of course, there's more. Avocet has the tools you need from start to finish to enter, assemble and test your software and finally cast it in EPROM:

Text Editor VEDIT -- full-screen text editor by CompuView. Makes source code entry a snap. Full-screen text editing, plus TECO-like macro facility for repetitive tasks. Pre-configured for over 40 terminals and personal computers as well as in user-configurable form.

CP/M-80 version	$150
CP/M-86 or MDOS version	$195

(when ordered with any Avocet product)

ROM Simulator -- ROMSIM by Inner Access eliminates need to erase and reprogram EPROM. Installed in an S-100 host, ROMSIM substitutes RAM for EPROM in external target system. 16K memory can be configured to simulate the 2708, 2758, 2716, 2516, 2732, 2532, 2764, 2564 in either byte or word organization. Avocet's configurable driver makes loading of HEX or COM files fast and easy.

From $495 depending on cabling and RAM installed.

EPROM Programmer -- Model 7128 EPROM Programmer by GTek programs most EPROMS without the need for personality modules. Self-contained power supply ... accepts ASCII commands and data from any computer through RS 232 serial interface. Cross-assembler hex object files can be down-loaded directly. Commands include verify and read, as well as partial programming.

PROM types supported: 2508, 2758, 2516, 2716, 2532, 2732, 2732A, 27C32, MCM8766, 2564, 2764, 27C64, 27128, 8748, 8741, 8749, 8742, 8751, 8755, plus Seeq and Xicor EEPROMS.

(Upgrade kits will be available for new PROM types as they are introduced.)

Programmer	$389
Options include:	
Software Driver Package	$ 30
RS 232 Cable	$ 30
8748 family socket adaptor	$ 98
8751 family socket adaptor	$174

Call Us

If you're thinking about development systems, call us for some straight talk. If we don't have what you need, we'll help you find out who does. If you like, we'll even talk about Bach.

VISA and Mastercard accepted. All popular disc formats now available -- please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

*Trademark of Digital Research.

AVOCET SYSTEMS INC.
DEPT. 683-M
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TLX 467210

CIRCLE 201 ON READER SERVICE CARD

```
        DEFW    61966           ;19
        DEFW    61584           ;20
        DEFW    61183           ;21
        DEFW    60764           ;22
        DEFW    60326           ;23
        DEFW    59870           ;24
        DEFW    59396           ;25
        DEFW    58903           ;26
        DEFW    58393           ;27
        DEFW    57865           ;28
        DEFW    57319           ;29
        DEFW    56756           ;30
        DEFW    56175           ;31
        DEFW    55578           ;32
        DEFW    54963           ;33
        DEFW    54332           ;34
        DEFW    53684           ;35
        DEFW    53020           ;36
        DEFW    52339           ;37
        DEFW    51643           ;38
        DEFW    50931           ;39
        DEFW    50203           ;40
        DEFW    49461           ;41
        DEFW    48703           ;42
        DEFW    47930           ;43
        DEFW    47143           ;44
        DEFW    46341           ;45
;
SINTBL:
        DEFW    1144            ;1 DEG
        DEFW    2287            ;2
        DEFW    3430            ;3
        DEFW    4572            ;4
        DEFW    5712            ;5
        DEFW    6850            ;6
        DEFW    7987            ;7
        DEFW    9121            ;8
        DEFW    10252           ;9
        DEFW    11380           ;10
        DEFW    12505           ;11
        DEFW    13626           ;12
        DEFW    14742           ;13
        DEFW    15855           ;14
        DEFW    16962           ;15
        DEFW    18064           ;16
        DEFW    19161           ;17
        DEFW    20252           ;18
        DEFW    21336           ;19
        DEFW    22415           ;20
        DEFW    23486           ;21
        DEFW    24550           ;22
        DEFW    25607           ;23
        DEFW    26656           ;24
        DEFW    27697           ;25
        DEFW    28729           ;26
        DEFW    29753           ;27
        DEFW    30767           ;28
        DEFW    31772           ;29
        DEFW    32768           ;30
        DEFW    33754           ;31
        DEFW    34729           ;32
        DEFW    35693           ;33
        DEFW    36647           ;34
        DEFW    37590           ;35
        DEFW    38521           ;36
        DEFW    39441           ;37
        DEFW    40348           ;38
        DEFW    41243           ;39
        DEFW    42126           ;40
        DEFW    42995           ;41
        DEFW    43852           ;42
        DEFW    44695           ;43
        DEFW    45525           ;44
        DEFW    46341           ;45
;
;**********************************************************************
;
;       Axis Subroutine
;
AXIS:
        IF      PAS.Z
;
        ENTRY   AXIS
        POP     BC              ;fetch return address
        LD      HL,0
        ADD     HL,SP
        LD      L,(HL)          ;fetch GRID: boolean
        LD      (P7),HL
        INC     SP
        POP     HL              ;fetch YB: integer
        LD      (P6),HL
        POP     HL              ;fetch XB: integer
        LD      (P5),HL
        POP     HL              ;fetch YA: integer
        LD      (P4),HL
        POP     HL              ;fetch XA: integer
        LD      (P3),HL
        POP     HL              ;fetch Y0: integer
        LD      (P2),HL
        POP     HL              ;fetch X0: integer
        LD      (P1),HL
        PUSH    BC
        ENDIF
;
        IF      BASIC
;
        LD      B,7             ; pass seven parameters
        CALL    DTLNK
        ENDIF
;
        LD      HL,(P1)
        CALL    TEST            ; keep within x,y bounds
        LD      (P1),HL
        LD      (STARTX+2),HL
        LD      (ENDX+2),HL
        LD      HL,0
        LD      (STARTY+2),HL
        LD      HL,DOTS-1
        LD      (ENDY+2),HL
        CALL    LINE1           ; draw Y axis
        LD      HL,(P2)
        CALL    YTEST
        LD      (P2),HL
        LD      (STARTY+2),HL
        LD      (ENDY+2),HL
        LD      HL,0
        LD      (STARTX+2),HL
        LD      HL,DOTS-1
        LD      (ENDX+2),HL
        CALL    LINE1           ; draw X axis
;
        IF      PAS.Z
;
        PUSH    IX              ; save IX, IY for PAS.Z/Z
        PUSH    IY
        ENDIF
;
        LD      IX,STARTX+2     ; point to Xt, Yt positions
```

```
        LD      IY,STARTY+2
        LD      HL,(P3)         ; fetch X tick minor
        LD      (TM),HL
        LD      A,L
        OR      H
        JR      Z,XTMAJ         ; skip if Xt minor=0
        LD      HL,(P1)
        LD      (TAXIS),HL
        LD      HL,(P2)
        LD      (CAXIS),HL
        CALL    TICK            ; fill in tick marks
        CALL    MINOR
XTMAJ:
        LD      HL,(P5)         ; fetch X tick major
        LD      (TM),HL
        LD      A,L
        OR      H
        JR      Z,YTMIN         ; skip if Xt major =0
        LD      HL,(P1)
        LD      (TAXIS),HL
        LD      HL,(P2)
        LD      (CAXIS),HL
        CALL    TICK            ; fill in tick marks
        LD      (XGRID),HL
        CALL    MAJOR
YTMIN:
        LD      IX,STARTY+2     ; same as above, but rotate axis
        LD      IY,STARTX+2
        LD      HL,(P4)         ; fetch Y tick minor
        LD      (TM),HL
        LD      A,L
        OR      H
        JR      Z,YTMAJ         ; skip if Y tick minor=0
        LD      HL,(P2)
        LD      (TAXIS),HL
        LD      HL,(P1)
        LD      (CAXIS),HL
        CALL    TICK            ; fill in tick marks
        CALL    MINOR
YTMAJ:
        LD      HL,(P6)         ; fetch Y tick major
        LD      (TM),HL
        LD      A,L
        OR      H
        RET     Z               ; all done no Y tick major
        LD      HL,(P2)
        LD      (TAXIS),HL
        LD      HL,(P1)
        LD      (CAXIS),HL
        CALL    TICK            ; fill in tick marks
        LD      (YGRID),HL
        CALL    MAJOR
        CALL    GRID
;
        IF      PAS.Z
;
        POP     IY              ; restore IX, IY
        POP     IX
        XOR     A               ; indicate valid external procedure call
        ENDIF
;
        RET
;
TICK:   LD      HL,DOTS-1       ; find largest value for tick mark
        LD      DE,(TAXIS)
        OR      A
        SBC     HL,DE
        LD      DE,(TM)
TICK0:  SBC     HL,DE
        JR      NC,TICK0
        ADD     HL,DE
        LD      DE,DOTS-1
        EX      DE,HL
        OR      A
        SBC     HL,DE           ; HL = highest value for tick mark
        RET
;
MINOR:  LD      (IX+0),L        ; plot minor tick marks on indicated axis
        LD      (IX+1),H
        LD      HL,(CAXIS)
        DEC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      HL,(CAXIS)
        INC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      L,(IX+0)
        LD      H,(IX+1)
        LD      DE,(TM)
        OR      A
        SBC     HL,DE
        JR      NC,MINOR
        RET
;
MAJOR:  LD      (IX+0),L        ; plot major tick marks on indicated axis
        LD      (IX+1),H
        LD      HL,(CAXIS)
        DEC     HL
        DEC     HL
        DEC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      HL,(CAXIS)
        DEC     HL
        DEC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      HL,(CAXIS)
        DEC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      HL,(CAXIS)
        INC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      HL,(CAXIS)
        INC     HL
        INC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      HL,(CAXIS)
        INC     HL
        INC     HL
        INC     HL
        LD      (IY+0),L
        LD      (IY+1),H
        CALL    SETO
        LD      L,(IX+0)
        LD      H,(IX+1)
        LD      DE,(TM)
        OR      A
        SBC     HL,DE
        JR      NC,MAJOR
        RET
```

NEW
Retail Showroom
Come in for
FREE
Demonstration

WE WILL BEAT OR MEET ALL PRICES ON ALL PRODUCTS IN THIS MAGAZINE

TO PLACE ORDER CALL (213) 219-0808 NOW!

48K APPLE COMPATIBLE COMPUTER

1 yr. Warranty

Introductory Price	$595
Call for Availability	

ALL COMPUTERS

Apple II E	$ Call
Commodore 64	469
Kaypro	1649
Osborne	Call
Sanyo MBC 1000	1599
Apple Compatible Computer	595
S-100 System	Call for lowest price
IBM PC	Call

IBM PC PRODUCTS

Ast Research

Combo Plus 64K w/clock, Par. Port, Serial Port, Software	$359
Mega Plus 64K Expandable to 512	430

Tandon

TM100-2 Dbl Side Dbl Density	$ 235

5¼ & 8″ DISK DRIVES

Shugart

SA400, 40 Track	$ 145
SA801R, Sgl Side/Dbl Den	369
SA851R, Dbl Side/Dbl Den	479

Tandon

TM 100-1, Sgl Side/Dbl Den	$ 179
TM 100-2, Dbl Side/Dbl Den	235
TM 848-1, Sgl Side/Dbl Den	369
TM 848-2, Dbl Side/Dbl Den	449

Qume

DT-8 Dbl Side/Dbl Den	$ 465

Siemens 8″

FDD 100-8, Sgl Side/Dbl Den	$ 209

Mitsubishi

8″ Dbl Side/Dbl Den	$ 409

FOR FRANKLIN & APPLE

All Apple Products Not Seen in This Ad Are Available-Call for Guaranteed Lowest Price

Grappler +	$118
Par. Interface Card & Cable	49
TG Joystick	39
16K Memory Card	44
80 Column Card (View Max 80)	169
RH Fan	69
RF Modulator	25

UNBEATABLE PRINTER PRICING

Star Micronics

Gemini 10 (100 cps)	$ 355
Gemini 15 (15″ carriage)	465

C. Ioth

Prowriter 8510AP (120 cps)	$ 365
Prowriter 8510ACD (Serial)	549
Starwriter F-10 40pu	1249

Okidata

Microline 82A	$ 379
Microline 83A	649
Microline 84P	940
Microline 84S	1050
Microline 92	509

Nec

Nec 8023 (price reduction)	$ 449
Spinwriters	Call

Toshiba

P1350	$1699

SPECIALS OF THE MONTH

5¼″ Disk Drive
SA400L Shugart
90 Day Warranty
ONLY $145.00

MODEM
Smart Modem 300
Hayes Top of the Line
ONLY $195.00

DRIVE FOR APPLE OR FRANKLIN
Micro Sci A-2
Fully Compatible
$249.00

PRINTER
Gemini 10
Same as Epson MX80FT
$355.00

MONITOR
Sanyo 2112 Green Screen
• 15MHz • Composite
ONLY $79.00

USI Amber 12″
• 1000 Lines
• Horizontal Resolution
$149.00

All Products Carry at least a 90 Day Warranty

MODEMS

Hayes

Smart Modem 300	$ 199
Smart Modem 1200	498

Novation

D-Cat	Call
Apple Cat	Call

TERMINALS

Televideo

925C Green Screen	$ 699
950 Detachable Keyboard	899

Adds

Viewpoint 1A, 2A, 3A	Call

DISKETTES

All packages of diskettes come with free library case and reinforced hub. Full 1 year warranty. Prices are for Packs of 10.

5¼″ Disks

Sgl Side/Sgl Density	$18.50/10
Sgl Side/Dbl Density	20.50/10
Dbl Side/Dbl Density	26.50/10

8″ Disks

Sgl Side/Sgl Density	$25.00/10
Sgl Side/Dbl Den	29.00/10

CABINETS & MAINFRAMES

QT Computer Systems

DDC-8 Cabinet w/pwr. supply & fan
ONLY $209.00

DDC-88-H Dual Cabinet w/pwr. supply & fan
ONLY $259.00

Mainframe
6 slot w/2, 8″ cutouts, pwr. supply, fan, filter, Connector Cutouts
ONLY $549.00

5¼″ Cabinets

Single Cabinet w/pwr. supply	$ 59
Dual Cabinet w/pwr. supply	89

DISK DRIVES FOR FRANKLIN & APPLE

Rana Systems

Rana Elite I	$ 285
Rana Elite II	439
Rana Elite III	569
Controller	89

Micro Sci

A-2 (35 Track)	$ 249
Controller	75
Apple III Drives Available	Call

Quentin Research

Apple Mate	$ 249
Controller	60

MONITORS

Sanyo

12″ Green Screen, 2112 Composite	$ 79

BMC

12A (15MHz) Composite	84
12EUN (20MHz) Composite	129
9191 Color Composite	309

Taxan

12″ Green (18MHz) Composite	$ 129
12″ Amber (18MHz) Composite	129
RGB 1 Color	319

Zenith

ZVM 121 12″ Green (15MHz)	$ 94

USI

9″ Green (20MHz)	$ 139
12″ Green (20MHz)	139
12″ Amber (20MHz)	149

CABLES

IBM to Printer	$ 32
Osborne to Printer	32
Kaypro to Printer	32

Call for all cable configurations

DISKETTE STORAGE

Mini Files—Holds 70 Disks	$16.50
Maxi Files—Holds 70 Disks	24.00
5¼″ Library Cases	3.00
8″ Library Cases	4.00

Computer Components Unlimited

CIRCLE 208 ON READER SERVICE CARD

NEW RETAIL STORE:
11976 Aviation Blvd.
Inglewood, CA 90304

MAIL ORDER:
P.O. Box 1936
Hawthorne, CA 90250

ORDER DESK:
(213) 219-0808

No Surcharge for Credit Cards

VISA MasterCard

All merchandise new. We accept MC, Visa, Wire Transfer, COD Call, Certified Check, P.O.'s from qualified firms, APO accepted. Shipping: Minimum $3.50 first 5 pounds. Tax: California Res. Only add 6½% sales tax.

Mon.-Fri. 8 a.m. to 7 p.m.
Sat. & Sun. 11 a.m. to 4 p.m.

The computers of tomorrow will have multiple windows. Yours can have them today. The Electric Blackboard™ multi-window text editor allows you to divide the screen of *your* CRT into any combination of horizontal and vertical windows.

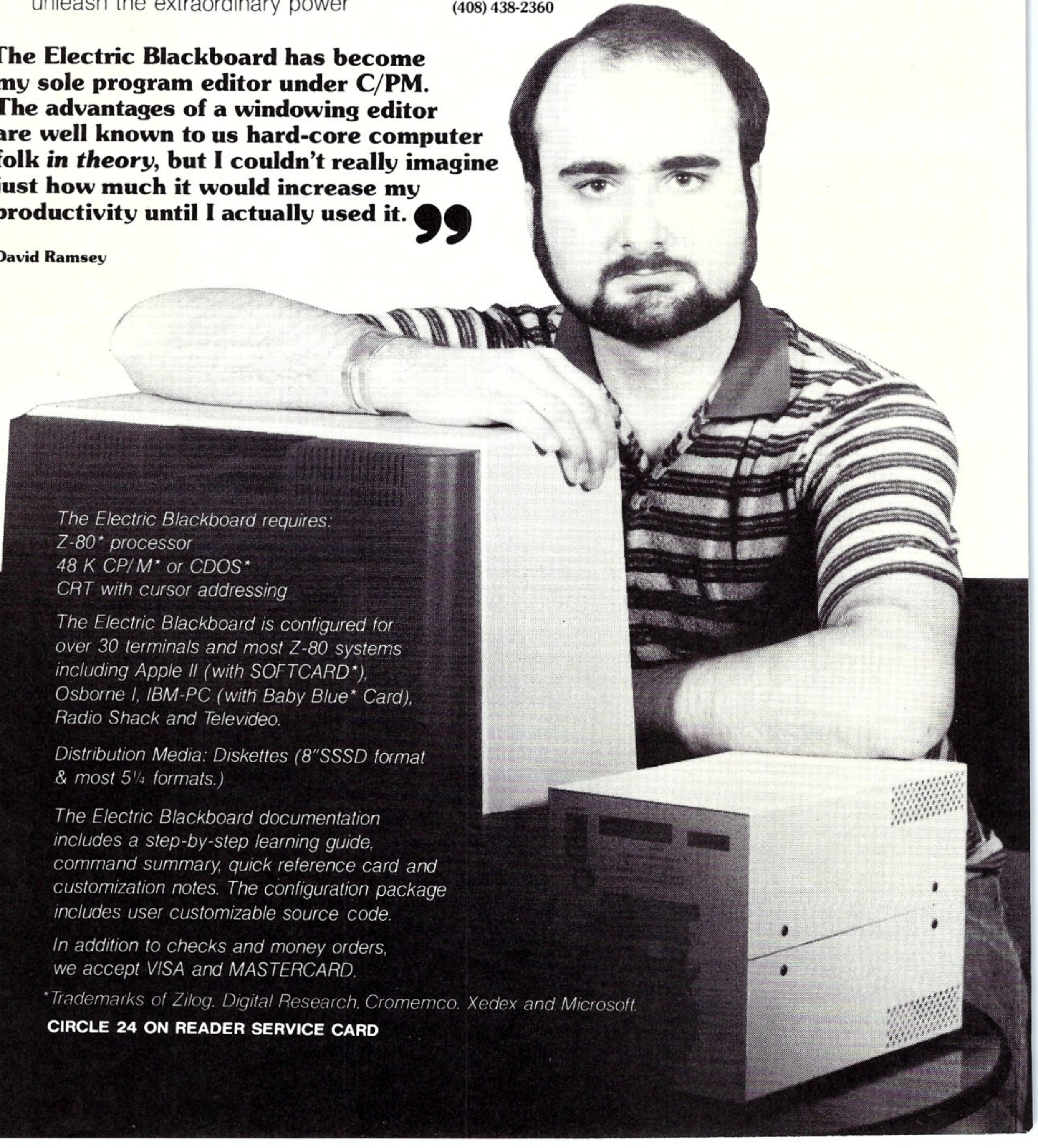
Cut, paste, compare, draw charts, review, scroll and edit your favorite 4, 5, or even 10 files simultaneously. Create your own commands for your own applications. Increase your productivity. Whether scientist or novice, it takes only minutes to get started. Let *our* multiple windows unleash the extraordinary power and flexibility of *your* micro for only $198.00 (manual alone: $30.00)

SANTA CRUZ SOFTWARE SERVICES

1711 Quail Hollow Road
Ben Lomond, CA 95005
(408) 438-2360

"The Electric Blackboard has become my sole program editor under C/PM. The advantages of a windowing editor are well known to us hard-core computer folk *in theory*, but I couldn't really imagine just how much it would increase my productivity until I actually used it."

David Ramsey



The Electric Blackboard requires:
Z-80 processor*
48 K CP/M or CDOS**
CRT with cursor addressing

The Electric Blackboard is configured for over 30 terminals and most Z-80 systems including Apple II (with SOFTCARD), Osborne I, IBM-PC (with Baby Blue* Card), Radio Shack and Televideo.*

Distribution Media: Diskettes (8"SSSD format & most 5¼ formats.)

The Electric Blackboard documentation includes a step-by-step learning guide, command summary, quick reference card and customization notes. The configuration package includes user customizable source code.

In addition to checks and money orders, we accept VISA and MASTERCARD.

**Trademarks of Zilog, Digital Research, Cromemco, Xedex and Microsoft.*

CIRCLE 24 ON READER SERVICE CARD

```
;
GRID:
        LD      HL,(P7)         ; plot x, y grid marks if required
        LD      A,H
        OR      L
        RET     Z               ; no grid required
        LD      HL,(P5)
        LD      A,H
        OR      L
        RET     Z               ; impossible x grid
        LD      HL,(P6)
        LD      A,H
        OR      L
        RET     Z               ; impossible y grid
        LD      HL,(XGRID)      ; put grid marks at intersections of
GRID0:  LD      (STARTX+2),HL   ; axis major tick marks
        LD      HL,(YGRID)
GRID1:  LD      (STARTY+2),HL
        CALL    SET1
        LD      HL,(STARTY+2)
        LD      DE,(P6)
        OR      A
        SBC     HL,DE
        JR      NC,GRID1
        LD      HL,(STARTX+2)
        LD      DE,(P5)
        OR      A
        SBC     HL,DE
        JR      NC,GRID0
        RET
;
;************************************************************************
;
;       LINE PRINTER DRIVER
;
PRNTR:
        PUSH    AF
        PUSH    BC
        PUSH    DE
        LD      E,A
        LD      C,5
        CALL    BDOS
;
        IF      BASE2
;
        LD      B,32
DELAY:  DJNZ    DELAY           ; give BASE2 some more time
        ENDIF
;
        POP     DE
        POP     BC
        POP     AF
        RET
;
;************************************************************************
;
;       XFRPLT: transfer image-to-printer subroutine
;
XFRPLT:                         ; CALL & USR entry point, no parameters
        IF      PAS.Z
;
        ENTRY   XFRPLT
        PUSH    IX              ; save IX, IY for PAS.Z/Z
        PUSH    IY
        ENDIF
;
        IF      BASE2
;
        LD      A,27            ; set up PRNTR for:
        CALL    PRNTR           ;       96 characters/inch
        LD      A,50            ;       14 vertical half-dots/inch
        CALL    PRNTR
        LD      A,27
        CALL    PRNTR
        LD      A,98
        CALL    PRNTR
        LD      A,14
        CALL    PRNTR
        ENDIF
;
        IF      IDS560
;
        LD      A,03            ; send ETX character
        CALL    PRNTR
        ENDIF
;
;       row:    continous sequence of memory locations of length LENGTH
;       line:   continous sequence of graphic characters of length LENGTH
;       shift:  number of 16 bit left shifts required to recover graphic
;               character from 16 bit word formed from location
;                       plot+i (IX)->H, and plot+i+LENGTH (IY)->L
;
        LD      IX,(IMGPTR)     ; IX = IMAGE
        LD      IY,(IMGPTR)
        LD      DE,LENGTH
        ADD     IY,DE           ; IY = IMAGE + LENGTH
        LD      C,LINES         ; transfer this number of print lines
XFR0:   LD      D,7             ; # shifts in first group
;
XFR1:
        IF      BASE2
;
        LD      A,27            ; set up for graphics
        CALL    PRNTR
        LD      A,99
        CALL    PRNTR
        ENDIF
;
        PUSH    BC              ; save line counter
        LD      B,FILL1         ; & fill left margin with blanks
XFR10:  LD      A,128
        CALL    PRNTR
        DJNZ    XFR10
        LD      BC,LENGTH       ; transfer the graphics characters
XFR2:   LD      H,(IX+0)
        LD      L,(IY+0)
        CALL    ROTL
;
        IF      BASE2
;
        CALL    PRINTER
        ENDIF
;
        IF      IDS560
;
        PUSH    AF
        CALL    PRNTR
        POP     AF
        CP      03              ; was ETX sent?
        CALL    Z,PRNTR         ; yes, must be sent twice
        ENDIF
;
        INC     IX              ; increment image pointers
        INC     IY
        DEC     BC              ; decrement character counter
        LD      A,C
        OR      B
        JR      NZ,XFR2         ; do entire line of graphic characters
;
        IF      BASE2
```

```
;
        LD      B,FILL2
XFR20:  LD      A,128           ; fill right margin with blanks
        CALL    PRNTR
        DJNZ    XFR20
        LD      A,10            ; terminate with line feed
        CALL    PRNTR
        ENDIF

        IF      IDS560
;
        LD      A,03            ; send line terminator sequence
        CALL    PRNTR
        LD      A,14
        CALL    PRNTR
        ENDIF
;
        DEC     D               ; do one less shift on next line
        LD      A,D             ; test for D=6, special case
        CP      6
        JR      NZ,XFR21
        LD      BC,-LENGTH      ; if D=6 then repeat row for next line
        ADD     IX,BC
        ADD     IY,BC
XFR21:  POP     BC
        DEC     C               ; decrement line counter
        JR      Z,XFRDN         ; all lines done, exit from loop
        LD      A,D             ; shifts 7 --> 0 done?
        CP      -1
        JR      NZ,XFR1         ; no, do next shift
        JR      XFR0            ; yes, next row, line, shifts 7-->0
XFRDN:
        IF      BASE2
;
        LD      A,27            ; reset printer to normal
        CALL    PRNTR
        LD      A,98
        CALL    PRNTR
        LD      A,24
        CALL    PRNTR
        ENDIF
;
        IF      IDS560
;
        LD      A,03
        CALL    PRNTR
        LD      A,14
        CALL    PRNTR
        LD      A,03
        CALL    PRNTR
        LD      A,02            ; return to NORMAL mode
        CALL    PRNTR
        ENDIF
;
        IF      PAS.Z
;
        POP     IY              ; restore IX, IY for PAS.Z
        POP     IX
        ENDIF
;
        RET
;
ROTL:   PUSH    DE              ; save DE, BC
        PUSH    BC
        XOR     A               ; test for case 0
        CP      D
        JR      Z,ROTL1         ; no shifts for case 0
        LD      A,D             ; test for case 7
        CP      7
        JR      NZ,ROTL0
        SRL     H               ; if D=7, then one shift right
        JR      ROTL1
ROTL0:  SLA     L
        RL      H               ; 16 bit rotate D bits to left
        DEC     D
        JR      NZ,ROTL0
ROTL1:  XOR     A               ; reverse bit order for PRNTR
        LD      B,7
ROTL2:  SRL     H
        RLA
        DJNZ    ROTL2
        POP     BC              ; restore BC, DE
        POP     DE
        RET                     ; & return graphic char in Accum
;
;*************************************************************************
;
;       working storage locations
;
STARTX: DEFW    0               ; X1, 16 bit integer, 16 bit fraction
        DEFW    0
STARTY: DEFW    0               ; Y1, same
        DEFW    0
ENDX:   DEFW    0               ; X2, same
        DEFW    0
ENDY:   DEFW    0               ; Y2, same
        DEFW    0
DIRX:   DEFW    0               ; (X2 - X1)/256, same formate as X1
        DEFW    0
DIRY:   DEFW    0               ; (Y2 - Y1)/256, same
        DEFW    0
;
;       passed parameters storage
;
P1:     DEFW    0               ; memory reserved for passing up to
P2:     DEFW    0               ; eight parameters from BASIC via
P3:     DEFW    0               ; CALL subr(P1,P2,P3,P4,P5,P6,P7,P8)
P4:     DEFW    0
P5:     DEFW    0
P6:     DEFW    0
P7:     DEFW    0
P8:     DEFW    0
;
TM:     DEFW    0               ; reserved for AXIS routine temp storage
TAXIS:  DEFW    0
CAXIS:  DEFW    0
;
XGRID:  DEFW    0
YGRID:  DEFW    0
;
IMGPTR: DEFW    IMAGE
IMAGE   EQU     $
;
        IF      PAS.Z
;
        DEFS    AREA
        ENDIF
;
        END
```

Z80 Software

DEBUG
QED
LINKER
MODEM
MACRO
TPM
BASIC I
ZDDT
ZTEL
QSAL
ZEDIT
TOP I
6502X
TPM II

SOFTWARE DESCRIPTIONS

TPM (TPM I) - $80 A Z80 only operating system which is capable of running CP/M programs. Includes many features not found in CP/M such as independent disk directory partitioning for up to 255 user partitions, space, time and version commands, date and time, create FCB, chain program, direct disk I/O, abbreviated commands and more! Available for North Star (either single or double density), TRS—80 Model I (offset 4200H) or II, Versafloppy I, Tarbell I, or Osborne I.

TPM-II - $125 An expanded version of TPM which is fully CP/M 2.2 compatible but still retains the extra features our customers have come to depend on. This version is super FAST. Extended density capability allows over 600K per side on an 8" disk. Available preconfigured for Versafloppy II (8" or 5"), Epson QX-10, or TRS-80 Model II.

CONFIGURATOR I

This package provides all the necessary programs for customizing TPM for a floppy controller which we do not support. We suggest ordering this on single density (8SD).

Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), LINKER ($80), DEBUG I ($80), QED ($150), ZEDIT ($50), TOP I ($80) BASIC I ($50) and BASIC II ($100)

$815 Value NOW $250

CONFIGURATOR II

Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QSAL ($200), QED ($150), ZTEL ($80), TOP II ($100), BUSINESS BASIC ($200) and MODEM SOURCE ($40) and DISASSEMBLER ($80)

$1485 Value NOW $400

MODEL I PROGRAMMER

This package is only for the TRS-80 Model I. Note: These are the ONLY CDL programs available for the Model I. It includes: TPM I ($80), BUSINESS BASIC ($200), MACRO I ($80), DEBUG I ($80), ZDDT ($40), ZTEL ($80), TOP I ($80) and MODEM ($40)

$680 Value NOW $680

MODEL II PROGRAMMER

This package is only for the TRS-80 Model II. It includes: TPM-II ($125), BUSINESS BASIC ($200), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QED ($150), ZTEL ($80), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM ($40), MODEM SOURCE ($40) and DISASSEMBLER ($80)

$1445 Value NOW $375

BASIC I - $50, a 12K+ basic interpreter with 7 digit precision.

BASIC II - $100, A 12 digit precision version of Basic I.

BUSINESS BASIC - $200, A full disk extended basic with random or sequential disk file handling and 12 digit precision (even for TRIG functions). Also includes PRIVACY command to protect source code, fixed and variable record lengths, simultaneous access to multiple disk files, global editing, and more!

ACCOUNTING PACKAGE - $300, Written in Business Basic. Includes General Ledger, Accounts Receivable/Payable, and Payroll. Set up for Hazeltine 1500 terminal. Minor modifications needed for other terminals. Provided in unprotected source form.

MACRO I - $80, A Z80/8080 assembler which uses CDL/TDL mnemonics. Handles MACROs and generates relocateable code. Includes 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical ops, local and global symbols, linkable module generation, and more!

MACRO II - $100, An improved version of Macro I with expanded linking capabilities and more listing options. Also internal code has been greatly improved for faster more reliable operation.

MACRO III - $150, An enchanced version of Macro II. Internal buffers have been increased to achieve a significant improvement in speed of assembly. Additional features include line numbers, cross reference, compressed PRN files, form feeds, page parity, additional pseudo-ops, internal setting of time and date, and expanded assembly-time data entry.

6502X - $150, A 6502 cross assembler. Runs on the Z80 but assembles 6502 instructions into 6502 object code! Similar features as our Macro assemblers.

DEVELOPER I

Includes: MACRO I ($80), DEBUG I ($80), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)

$440 Value NOW $150

DEVELOPER II

Includes: MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), BUSINESS BASIC ($200), QED ($150), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM SOURCE ($40) and DISASSEMBLER ($80) ZTEL (80)

$1280 Value NOW $350

DEVELOPER III

Includes: QSAL ($200), QED ($150), BUSINESS BASIC ($200), ZTEL ($80) and TOP II ($100)

$730 Value NOW $300

COMBO

Includes: DEVELOPER II ($1280), ACCOUNTING PACKAGE ($300), QSAL ($200) and 6502X ($150)

$1930 Value NOW $500

LINKER - $80, A linking loader for handling the linkable modules created by the above assemblers.

DEBUG I - $80, A tool for debugging Z80 or 8080 code. Disassembles to CDL/TDL mnemonics compatible with above assemblers. Traces code even through ROM. Commands include Calculate, Display, Examine, Fill, Goto, List, Mode, Open File, Put, Set Wait, Trace, and Search.

DEBUG II - $100, A superset of Debug I. Adds Instruction Interpreter, Radix change, Set Trap/Conditional display, Trace options, and Zap FCB.

6502X - $150, A 6502 cross assembler. Runs on the Z80 but assembles 6502 instructions into 6502 object code! Similar features as our Macro assemblers.

QSAL - $200, A SUPER FAST Z80 assembler. Up to 10 times faster than conventional assemblers. Directly generates code into memory in one pass but also to offset for execution in its own memory space. Pascal like structures: repeat...until, if...then...else, while...do, begin...end, case...of. Multiple statements per line, special register handling expressions, long symbol names, auto and modular assembly, and more! This one uses ZILOG Mnemonics.

QED - $150, A screen editor which is both FAST and easy to learn. Commands include block delete, copy, and move to a named file or within text, repeat previous command, change, locate, find at start of line, and numerous cursor and window movement functions. Works with any CRT having clear screen, addressable cursor, clear to end of line, and clear to end of screen.

ZTEL - $80, An extensive text editing language and editor modelled after DEC's TECO.

ZEDIT - $50, A mini text editor. Character/line oriented. Works well with hardcopy terminals and is easy to use. Includes macro command capability.

TOP I - $80, A Text Output Processor for formatting manuals, documents, etc. Interprets commands which are entered into the text by an editor. Commands include justify, page number, heading, subheading, centering, and more.

TOP II - $100, A superset of TOP I. Adds: embedded control characters in the file, page at a time printing, selected portion printing, include/merge files, form feed/CRLF option for paging, instant start up, and final page ejection.

ZDDT - $40, This is the disk version of our famous Zapple monitor. It will also load hex and relocatable files.

ZAPPLE SOURCE - $80, This is the source to the SMB ROM version of our famous Zapple monitor. It can be used to create your own custom version or as an example of the features of our assemblers. Must be assembled using one of our assemblers.

MODEM - A communication program for file transfer between systems or using a system as a terminal. Based on the user group version but modified to work with our SMB board or TRS-80 Models I or II. You must specify which version you want.

MODEM SOURCE - $40, For making your own custom version. Requires one of our Macro Assemblers.

DISASSEMBLER - $80, Does bulk disassembly of object files creating source files which can be assembled by one of our assemblers.

HARDWARE

S-100 — SMB II Bare Board $50, "System Monitor Board" for S-100 systems, 2 serial ports, 2 parallel ports, cassette interface, 4K memory (ROM, 2708 EPROM, 2114 RAM), and power on jump. When used with Zapple ROM below, it makes putting a S-100 system together a snap.
Zapple ROM $35, Properly initializes SMB I/II hardware, provides a powerful debug monitor.

IBM PC — Big Blue Z80 board $595, Add Z80 capability to your IBM Personal Computer. Runs CP/M programs but does not require CP/M or TPM. Complete with Z80 CPU, 64K add on memory, serial port, parallel port, time and date clock with battery backup, hard disk interface, and software to attach to PC DOS and transfer programs. Mfr'd by QCS.
50% Discount on all CDL software ordered at the same time as a Big Blue (and for the Big Blue).

APPLE II — Chairman Z80 board $425, Add Z80 capability to your Apple II/II Plus computer. Runs CP/M programs with our more powerful TPM. Includes 64K memory add on (unlike the competition this is also useable by the 6502/DOS as well as the Z80), TPM, QSAL assembler, QED Screen Editor, and Business Basic. Mfr'd by AMT Research.
Apple Special $175, Buy the Apple Z80 Developer at the same time as the "Chairman" and pay only $175 instead of $325.

APPLE Z80 DEVELOPER

Includes: 6502X ($150), MACRO II ($100), MACRO III ($150), QSAL ($200), QED ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), ZDDT ($40) and BUSINESS BASIC ($200)

VALUE: $1250 $325

$175 when purchased with AMT "Chairman" Board

DISK FORMATS

When ordering software specify which disk format you would like.

CODE	DESCRIPTION
8SD	8" IBM 3740 Single Density (128 bytes/26 sectors/77 tracks)
8DD	8" Double Density (256 bytes/26 sectors/77 tracks)
8XD	8" CDL Extended Density (1024 bytes/8 sector/77 traceks = 616K)
5SD	5.25" Single Density (TRS80 Model I, Versafloppy I, Tarbell I)
5EP	5.25" Epson Double Density
5PC	5.25" IBM PC Double Density
5XE	5.25" Xerox 820 Single Density
5OS	5.25" Osborne Single Density
5ZA	5.25" Z80 Apple (Softcard compatible)

TPM INFO When ordering TPM I or II, in addition to Disk Format, please specify one of the following codes:

	CODE	DESCRIPTION
TPM I:	NSSD/H	North Star Single Density for Horizon I/O
	NSSD/Z	North Star Single Density for Zapple I/O
	NSDD/H	North Star Double Density for Horizon I/O
	NSDD/Z	North Star Double Density for Zapple I/O
	TRS80-I	TRS-80 Model I (4200H Offset)
	TRS80II	TRS-80 Model II
	VI8	Versafloppy I 8"
	VI5	Versafloppy I 5.25"
TPM-II:	VII8	Versafloppy II 8" (XD)
	VII5	Versafloppy II 5.25"
	TRS80II	TRS-80 Model II (XD)

Prices and Specifications subject to change without notice.
TPM, Z80, CP/M, TRS80 are trademarks of CDL, Zilog, DRI and Tandy respectively.

ORDERING INFORMATION:

VISA/MasterCard/C.O.D.
Call or Write With Ordering Information....

MasterCard
VISA

OEMS:

Many CDL products are available for licensing to OEM's. Write to Carl Galletti with your requirements.

Dealer Inquiries Invited.

For Phone Orders ONLY Call Toll Free...

1-(800) 458-3491 (Except Pa.)

Ask For Extension #15

For information and Tech Queries call
(609) 599-2146

Computer Design Labs

342 Columbus Avenue/Trenton, NJ 08629

CIRCLE 84 ON READER SERVICE CARD

IF YOU KNOW WHAT YOU WANT...

EDIX/WORDIX	SUPERCALC	CONDOR 3	VOLKSWRITER	VISICALC
$149 (ea.)	$185	$488	$147	$169
CROSSTALK	EASYWRITER II	PEACHPAK	FINAL WORD	DB MASTER
$119	$259	$359	$223	$148

IBM PC

	List	Now
Advanced Operating Systems		
The Programmer	200	150
American Training International, Inc.		
ATI-Power for PC DOS	75	56
Applied Software Technology		
Versaform	389	288
Ashton-Tate		
dBase II [4]	700	429
Aspen Software		
Grammatik	75	56
Proofreader	50	38
Byrom Software		
BSTAM (Micro to micro comm.)	200	149
BSTMS (Micro to mainframe comm.)	200	149
Comshare		
Master Planner	325	253
Condor Computer Corp.		
Condor 1	195	147
Condor 3	650	488
Continental Software		
Home Accountant Plus	150	112
1st Class Mail	125	98
CP Aids		
Master Tax	1695	1371
Standard Tax	595	446
DataMost		
Write On	130	94
Real Estate Investment Program	130	95
Denver Software		
EASy (Executive Acctg. System)	750	562
Digital Research		
CBASIC 86	200	150
CIS COBOL 86	800	600
Concurrent CP/M-86	350	294
Level 2 COBOL 86	1600	1200
Pascal MT+86/w SSP	600	468
SID86	150	94
SPP86	200	168
Eagle Software		
Money Decisions	199	149
Emerging Technology		
Edix	195	149
Wordix	195	149
Howard		
Real Estate Analyzer II	250	187
Innovative Software		
T.I.M. III	495	357
Fast Graphs	295	221
ISM		
MatheMagic	100	75
Graphmagic	90	65
Combo Pack	160	120
IUS		
Easy Speller II	225	159
Easy Filer	400	281
Easywriter II	350	259
Easy Planner	250	188
AR/AP/GL (ea)	600	450
Financial Management Series (all 3)	1495	1121
Lexisoft		
Spellbinder	495	281
Lifeboat		
Lattice C Compiler	500	415
Lifetree		
Volkswriter	195	147
Lightning		
Master type	50	38
Mark of the Unicorn		
Mince	175	153
Final Word	300	223
Metasoft		
Benchmark Word Processor*	500	367
Benchmark Mail list	175	154
Microfocus		
CIS COBOL 8086	850	637
Micro Lab		
The Tax Manager	250	188
MicroPro		
Infostar [5]	495	327
Report Star	350	231
Wordstar [4]	495	327
Spellstar	250	165
Mailmerge	250	165
Wordstar/Mailmerge [4]	695	426
Microsoft		
Flight Simulator	50	38
Multiplan		212
MicroStuf		
CrossTalk	195	119
North American Business Systems		
The Answer	250	188
Northwest Analytical		
Statpak	495	369
OASIS		
The Word Plus	150	112
Organic Software		
Datebook	395	331
Milestone (CP/M 86)	395	331
Textwriter III (Text formatter)	125	97
PBL Corporation		
Personal Investor (128K)	145	105
Peachtree		
Series 4 Peachpak (GL/AR/AP) [5]	595	359
Ryan-McFarland		
RM/COBOL		
• Full Dev. System	950	713
• Runtime only	250	188
Scripps Data		
Job Cost Systems	495	371
Select Information Systems		
Select with SuperSpell (Word Proc.)	595	369
Software Publishing		
PFS File		109
PFS Report		99
Sorcim		
Spellguard	195	146
Supercalc	295	185
Superwriter	395	296
Southeastern		
Data Capture	120	90
Star		
Legal Timekeeping, Billing, & Accts	950	771
Structured Systems		
GL, AR, AP, Payroll, Inventory, OE (ea.)	1250	790
Analyst	250	188
Letteright	200	150
NAD	100	75
QSORT	100	75
Supersoft		
Optimizer	200	149
Personal Data Base	125	93
Synergistic		
Data Reporter	250	188
Versa Computing		
Graphics Hardcopy System	25	19
VisiCorp		
VisiCalc (256K)	250	169
Desktop Plan I	300	229
VisiTrend/Plot	300	215
Visidex	250	189
VisiFile	300	229
Visischedule	300	234
Business Forecasting Models	100	78
Woolf Software Systems		
Move It	150	99

dBASE Corner

- dBASE II (relational DBMS) $429
- User's Guide $29
- Both $449
- Quickcode (program generator) $195
- dUTIL (programmer's utilities) $59
- ABSTAT (statistics package) $359
- D B Plus (short utility) $95

Technical questions, M-F/9-5 (415) 324-0311

IBM PC HARDWARE

	List	Now
Kraft		
Joystick	70	52
Microsoft		
RAM Card 64K	350	253
RAM Card 128K	525	381
RAM Card 192K	700	508
RAM Card 256K	875	634
QUADRAM		
Quadboard 64K	595	428
• 256K	995	670
TG Products		
Joystick	59	43
Versa Computing		
VersaWriter Graphics Tablet	299	249

CP/M

8" standard SSSD and Apple 5¼". Please call to confirm price & availability for other formats, available as special orders.

	List	Now
Adventure International		
Adventures (#1-12)	129	97
American Training International, Inc.		
ATI-Power for CP/M	75	54
Anderson-Bell		
ABSTAT		359
Artificial Intelligence		
(Requires CBASIC 2)		
Medical	995	749
Dental	995	749
Ashton Tate		
dBase II [4]	700	429
Aspen Software		
Grammatik	75	56
Proofreader	50	38
Balcones		
The Boss: Financial Accounting System	2495	1496
Byrom Software		
BSTAM (Micro to micro comm.)	200	144
BSTMS (Micro to mainframe comm.)	200	144
Caxton		
Cardbox	245	177
Centa Systems		
CP/M Northstar (Horizon only)	179	150
CP/M Micropolis	189	156
Computer Control		
FABS (B-TREE file access)	195	146
Ultrasort II	195	146
Computer Development, Inc.		
Image	495	356
Comshare Target Software		
Plannercalc	50	38
Masterplanner	325	263
Condor Computer Corp.		
Condor 1	195	147
Condor 3	650	488
Designer Software		
Palantir	425	319
Dictronics		
Random House Electronic Thesaurus	150	113
Digital Research		
Access Manager	300	225
MAC (macro assembler)	90	75
SID (symbolic debugger for 8080)	75	63
ZSID	100	88
CBASIC (language)	150	94
CB80	500	395
CIS COBOL	800	595
Display Manager	400	295
Pascal Mt+ w/SPP	500	395
DJR Associates		
FMS 80	995	625
FMS 80-1	495	312
FMS 80-2	495	312
Ecosoft		
MicroStat (stat package)	295	248
Epic Computer Corp.		
Supervyz	150	94
Faircom		
MICRO B+ (Keyed file accessing)		
For CBASIC 2	260	211
For MBASIC, COBOL, PL1/80, FORTRAN, Pascal MT+, CB 80	260	211
Fox and Geller		
dGRAPH	295	195
Quickscreen	149	125
Quickcode	295	195
dUTIL	99	59
FYI		
Superfile	195	129
Infocom		
Deadline	60	43
Zork I, II, III (ea.)	50	38
Star Cross	50	38
Human Soft		
D B Plus	125	95
ISM		
Mathemagic	99	72
Lexisoft		
Spellbinder	495	281
Lifeboat		
T-Maker II	275	215
Mark of the Unicorn		
Mince (text editor)	200	136
Final Word	300	229
Micro Pro		
Infostar [5]	495	327
Report Star	350	231
Wordstar [4]	495	327
MailMerge	250	165
Wordstar/Mail-Merge [4]	695	426
Spellstar [4]	250	165
Wordstar/Spell-star [4]	695	426
Word/Spell/Mail [4]	845	558
Calcstar	145	96
Datastar	295	195
MicroSoft		
BASIC 80	350	252
BASIC Compiler	395	284
FORTRAN 80	500	360
COBOL 80	750	540
MACRO 80	200	156
EDIT 80	120	94
μ Math/μ Simp	250	194
μ LISP/μ Star	200	156
M-Sort	195	151
Multiplan	275	179
Microstuf		
CrossTalk	195	119
Northwest Analytical		
Statpak	495	369
OASIS		
The Word Plus	150	112
Organic Software		
Datebook	295	229
Milestone	295	229
Peachtree		
Series 4 Peach Pak (GL,AR,AP)	595	359
Pearlsoft		
Personal Pearl	295	184
Pickles and Trout		
CP/M TRS 80 Mod II	185	169
Select Information Systems		
Select with Super-spell	595	369
Software Dimensions		
Accounting Plus™		
GL, AR, AP, Inv each	645	402
All four	2580	1435
Sorcim		
Pascal/M (Z80 or 8080)	395	284
Pascal/M (8086, 8088)	495	356
ACT 80 (Z80, 8080, 8085)	175	126
ACT 86/88 (8086, 8088)	175	126
TRANS 86	125	90
Spellguard	195	146
SuperCalc	295	185
Superwriter	395	249
Supersoft		
C Compiler	250	188
Ada Compiler	300	225
Woolf Software Systems		
Move it	100	66

Formats for CPM 8", Apple 5¼ & others. Price the same in most cases.

HARDWARE

	List	Now
Corona		
5MB Hard Disk [6]	2495	1895
Daisywriter		
Printer [10]	1395	1235
Hayes Microcomputer Products		
Datacom Pascal Package	50	36
Hayes Stack Chronograph (RS-232) [4]	249	199
Hayes Stack Smart-Modem (RS-232) 300 Baud [4]	279	209
Hayes Stack Smart-Modem (RS-232) 1200 Baud [4]	699	509
NEC		
12" Green Hi-Res CRT [10]	285	167
7710 Printer [10]	3085	2295
3530 Printer [10]	2290	1735
8023 Printer [10]	795	520
Novation		
J-Cat [4]	149	119
Smart-Cat 103 [4]	259	207
Smart-Cat 103/212 [4]	595	476

® =Manufacturer's trademark.

APPLE DOS

	List	Now
Applied Software Technology		
Versa Form:		
VS-1 Apple II/64K	389	249
VH-1 (Corvus)	495	360
VP-1 Versaform Pascal/Interface	245	180
ARTSCI (Softape)		
Magic Window II (40/70/80)	150	109
Magic Window	100	75
Magic Mailer	70	49
Magic Words	70	49
Magic Pack Combo (Contains all 3)	225	156

	List	Now
Avant-Garde Creations		
Ultra Plot/D.I.F./Data Graph	99	71
Beagle Brothers		
Alpha Plot	40	28
DOS Boss	24	17
Utility City	30	22
Apple Mechanic	30	22
Broderbund Software		
General Ledger (with A/P)	495	305
Payroll	395	275
Computer Station		
Combined Enhanced Graphics Software	55	40
Station Master	175	136
Comshare Target Software		
Image Maker	175	126
Continental		
Home Accountant	75	52
1st Class Mail	75	54
CPA #1 — GL	250	159
CPA #2 — AR	250	159
CPA #3 — AP	250	159
CPA #4 — PR	250	159
All four [5]	1000	609
CPA #5 — Property Management	495	352
Crane Software		
Menu Generator	40	29
Creative Curriculum, Inc.		
Speed Reading Courseware	99	75
Dakin 5/Level 10		
Depreciation Planner	395	299
Budget Planner	150	112
Business Bookkeeping System	395	299
Datamost		
Write On II	130	90
Real Estate Analysis Program	130	90
Tax Beater	130	90
Datasoft, Inc.		
Micropainter (48K)	35	25
BASIC Compiler (32K)	100	75
LISP Interpreter	125	94
Decision Support		
The Accountant	129	97
• with DBCalc	149	107
Delta Software		
Bookkeeper Master Program	90	68
Bookkeeper Check Writer	40	29
Denver Software		
Pascal Tutor	125	97
Pascal Programmer	296	223
Financial Partner	250	188
Eastside		
Wildcard	130	109
Eduware		
Compu-Read	30	22
Statistics	30	22
Spelling Bee with Reading Primer	40	30
Compu-Math/Fractions	40	30
Compu-Math/Decimals	40	30
Compu-Math/Arithmetic Skills	50	37
Algebra 1	40	30
Fractions (New version)	49	37
Decimals (New version)	49	37
Algebra 2	40	30
Counting Bee	30	22
Compu-Spell/System	30	22
Compu-Spell/Data Disk Levels 4 thru 8 (ea)	20	15
Metri-Vert	16	12
Uni-Solve	25	19
PSAT Word Attack Skills	49	37
SAT Word Attack	49	37
Hayden Software		
Pie Writer/Standard	150	108
• Double Vision	150	108
• Multi 80-col (for Smarterm, Videoterm, & Superterm)	150	108
Hayes		
Terminal Pgm.	100	75
Highlands Computer Services		
E-Z Ledger	60	37
Howard Software		
Creative Financing 1983	195	149
Real Estate Analyzer II	195	146
Information Unlimited Software (IUS)		
Datadex	150	108
Datadex/Hard Disk	300	216
Professional Easywriter	175	126
Professional Easymailer	150	108
Easy Mover	100	75
Tellstar Level I	40	30
Tellstar Level II	80	58
FORTH Development System	140	101
Insoft		
GraFORTH II	75	56
Electric Duet	30	23
TransFORTH II	125	94
ALD System II	75	56
International Software Marketing		
Mathemagic (Advanced programmable calculator package)	90	65
Graphmagic	90	65
Kensington Microware Ltd.		
Format II	250	169
Lightning Software		
Mastertype	40	30
Link Systems		
Datafax	199	149
Datalink	100	75
Link Index	195	149
Link Video	55	41
Link Disk	70	51
LJK Enterprises		
Letter Perfect with Mail Merge	150	112
Data Perfect	100	78
Lower Case Character Generator (accessory)	25	19
Lotus		
Executive Briefing System	199	149
Micro Lab		
The Invoice Factory	200	144
The Merger	50	37
The Learning System	150	108
Painter Power	40	30
US Constitution Tutor	30	22
English SAT #1	30	22
VisiBlend	50	37
VisiFactory	75	54
The Asset Manager	200	144
The Tax Manager	180	129
Data Factory 5.0	300	216
Wall Streeter	300	216
Payroll Manager	300	216
Microsoft		
TASC (Applesoft compiler)	175	119
Typing Tutor II	25	19
Time Manager	150	116
Adios μMath-80	125	97
Adios μMath/μSimp	250	194
M-Sort	195	151
μLisp/μStar	200	156
Multiplan	275	179
MUSE		
Supertext 40/56/70	125	79
Supertext 40/80	175	126
Form Letter Module 48K	100	72
Data Plot 48K**	60	43
The Voice 48K	40	29
**Requires Applesoft, ROM or Apple II Plus		
Omega		
Locksmith	100	75
Inspector	60	49
On-Line Systems		
The General Manager	230	168
Screenwriter II	130	85
The Dictionary	100	70
Expeditor II+ (Applesoft compiler)	135	94
LISA	80	56
EPFIV (Extended Programming Facility IV)	80	56
LISA Educational Package	120	83
Optimized Systems Software, Inc.		
Speed Read Plus	60	43
Peachtree		
Series 40 GL, AR, AP, Inventory, Payroll (ea)	400	239
Penguin Software		
Complete Graphics System	70	50
Special Effects	40	30
Graphics Magician	60	45
Additional Fonts and Character Sets	20	15
Complete Graphics System II (Apple Tab Ver.)	120	86
Special Effects (Apple Tab Ver.)	70	50
Program Design, Inc.		
The New Step-by-Step	80	58
Quality Software		
GBS-DB Apple Dev. Prog.	700	504
Sensible Software		
Super Disk Copy III	30	23
Multi Disk Catalog III	25	19
Disk Recovery ("The Scanner")	30	23
Applesoft Plus Structured BASIC	25	19
Disk Organizer II	30	23
DOS Plus	25	19
Quickloader	25	19
Image Printer-Letter Quality	40	30
Image Printer-Epson	40	30
Image Printer-NEC Dot Matrix	40	30
The Bug	50	37
Build Using	30	23
Edit Soft	30	23
Quadrant 6112	35	26
Sensible Speller	125	94
• for Supertext	125	94
• for Apple CP/M	125	94
• Pascal	125	94
BEST (Applesoft Program Optimizer)	40	30
Select Information Systems		
Select (with Super-spell)	595	356
Silicon Valley		
Word Handler	199	145
List Handler	90	68
Dictionary	125	94
Sirius		
Pascal Graphics Editor (48K II/II+)	100	69
Sof/Sys, Inc.		
Executive Secretary	250	169
Executive Speller	75	56
Softech Microsystems		
UCSD P-System Software Set [4]	625	450
BASIC Compiler and Runtime Package	225	169
Xenofile	50	39
Softeach	125	90
Software Dimensions		
Accounting Plus™ GL, AR, AP, Inv ea	395	239
All four	1250	699
Software Publishing		
PFS:File	140	99
PFS:Report	125	79
PFS:Graph	140	99
Southeastern Software		
Data Capture 4.0	65	45
Data Capture 4.0/Videx, Smarterm, Superterm	90	65
Southwestern Data		
Z-Term Professional	150	99
ASCII Express	80	59
ASCII Express Professional	130	89
STC		
Professional Time and Billing	395	284
Language Reading Development Program	175	126
Apartment Manager	325	234
Stoneware		
DBMaster	229	148
Utility Pak #1	99	65
Utility Pak #2	99	65
DBMaster/Hard Disk	499	359
Stat Pac	99	71
Micro Memo	40	30
Electronic Price Sheet	100	72
Compucube	30	22
Graphics Processing System (Standard)	60	45
Graphics Processing System (Professional)	179	129
Synergistic Software		
The Linguist	40	30
The Directory Manager	25	19
The Star Gazer's Guide	30	22
Higher Graphics II	35	26
Mailing List Data-Base	50	37
Program Line Editor	40	30
Modifiable DataBase	80	59
The Data Reporter	220	158
Integer BASIC Compiler	150	108
Global Program Line Editor	65	49
Data Reporter—Hard Disk	220	158
Versa Computing		
VersaWriter Expansion Pac-I	40	30
VersaWriter Expansion Pac-II	40	30
VersaWriter Graphics Tablet (accessory)	299	252
VisiCorp (Personal Software)		
VisiCalc	250	169
VisiFile	250	169
VisiLink	250	169
Desktop Plan II	250	194
VisiPlot	200	156
VisiTrend/Plot	300	234
VisiDex	250	194
VisiTerm	100	65
VisiSchedule	300	234
Business Forecasting Model (req. Visi Calc)	100	78

APPLE HARDWARE

	List	Now
Hayes		
Micromodem II [4]	379	259
• with terminal pgm	409	289
Kensington		
System Saver (Fan)	90	65
Micro-Sci		
A2 35T Disk Drive [5]	479	329
Microsoft		
RAMcard (memory board)	100	69
Softcard (CP/M board)	345	219
Premium System: Softcard, 16K RAMcard, Videx Videoterm 80 col. Card, Softswitch, CP/M User Guide [5]	695	489
Personal Computer		
Appli-Card (6MHz)	595	429
Videx		
Keyboard Enhancer II	149	99
Videoterm w/Soft-Switch	375	239

APPLE ///

	List	Now
Applied Software Technology		
Versa Form VS-3	495	371
DataMost		
Write On	130	99
Denver Software		
EASy (Executive Accounting System)	725	562
Insoft		
TransForth ///	125	94
ALD System ///	75	56
Link Systems		
Link Index	195	146
Link Video	55	42
Link Sampler I	60	45
Datafax	249	187
Datalink	149	112
Micro Lab		
Data Manager ///	750	562
Software Publishing Corp.		
PFS: File	175	120
PFS: Report	125	86
PFS Graph	175	120
VisiCorp		
Desktop Plan ///	300	234
Visischedule ///	300	234
Visicalc Advanced III	400	312
Business Forecasting Model	100	78

APPLE IIe

Most Apple II products also run on the IIe (call for details). Products below are specially designed for the IIe.

	List	Now
Software Publishing		
PFS File	125	99
PFS Report	125	99
PFS Graph	125	99
VisiCorp		
Visicalc Enhanced		202

TELEPHONE ORDERS

Outside Calif.: **(800) 222-8811**
Inside Calif.: **(415) 324-3730**

Order lines are manned 6:30 - 6 Monday thru Friday and 9 - 5 Saturday. Other lines are open 9 - 5 Monday thru Friday

Technical Support *(415) 324-0311*
Order Status *(415) 324-0306*
Sales Manager *(415) 324-0305*

MORE CP/M®, APPLE®, IBM PC®, UNIX™, UCSD p-System™ software, hardware, etc: call for quote.

PROGRAMMING INTERNATIONAL

505 Hamilton Ave. • Suite 107
Palo Alto, CA 94301

TERMS: All prices subject to change without notice and availability. Cashier's check/MO/bank transfer. Allow time for company or personal checks to clear. Prices reflect cash prepaid discount. VISA/MASTERCARD/COD/PO's + 3%. CA residents add sales tax. All sales final for games & special orders.

SHIPPING: $2 per item for UPS surface ($6 for Blue Label) within continental USA, except where shipping cost is specified in square brackets.

RETURNS: Must have authorization number, obtained at 415-324-0305. Unauthorized returns will be refused; damaged goods will be refused. All returns subject to 15% restocking fee. No return after 30 days.

WHY PAY MORE?

CIRCLE 25 ON READER SERVICE CARD

HP-GL: Add Graphics to Any Computer System

by Steve Leibson

Generally, when we think about computer languages, it is the traditional ones that come to mind. Such names as Basic, Pascal, Forth, and Assembler are commonplace in the pages of *Microsystems*. HP-GL (Hewlett-Packard Graphic Language) was developed by the Hewlett-Packard company to control its line of computer plotters (Hewlett-Packard, 16399 W. Bernardo Drive, San Diego, CA 92127-1899). Any programming language that can output ASCII characters can drive an HP-GL plotter. This article shows you how to use HP-GL to add sophisticated graphic capability to your system.

The first plotter to use HP-GL appeared in 1977. It was the Hewlett-Packard 9872 plotter. Since then HP has introduced a wide range of plotters and other graphic output devices that accept HP-GL commands. Plotters are now available—from the 7470A ($1,575), which plots on 8½" x 11" paper or transparency film, to the 7585B plotter (about $22,900) with a 36" bed. Nicolet Zeta plotters are also compatible with HP-GL (Nicolet Zeta Corp., P.O. Box 4003, Concord, CA 94524), and are available with EIA RS-232C and IEEE-488 (GPIB, HP-IB interfaces). Most microcomputers are sold with one or the other of these standard interfaces, making high-quality graphics just a connection away.

An I/O language

HP-GL is an interface language, as opposed to a programming language. You control the peripheral through characters sent to the device. Most printers have primitive interface languages involving escape sequences to activate special features such as underlining and special character sets. Any programming language that can send characters out a serial or IEEE-488 port may be used to control an HP-GL peripheral.

The syntax for HP-GL commands is very simple. A syntax diagram is shown in Figure 1. The command starts with a two-letter instruction mnemonic such as "PA" for Plot Absolute. Any needed command parameters follow the instruction mnemonic. Parameters are separated by commas or spaces (spaces are ignored on the 9872 plotter). The command ends with a termination character that may be either a semicolon, a line feed, or both (the line feed can be used only on the HP-IB interface). One exceptional terminator is needed for the label command: an ASCII ETX character (CHR$(3)). This special terminator allows semicolons and line feeds to appear in the label string of the label command.

Steve Leibson, 4040 Greenbriar Blvd., Boulder, CO 80803

The Hewlett-Packard 7585 drafting plotter, which takes 36½" x 48½" paper

HP-GL has been developed over the past eight years. The syntax diagram in Figure 1 is for the newer 9490A plotter; Figure 2 illustrates the more restrictive HP-GL syntax of the 9872.

Parameters may take one of four formats: integer, scaled decimal, decimal or label. A number in integer format is between −32768 and +32767. If the parameter must be an integer, the plotter truncates any fractional part of the parameter that may have been included. A number in scaled decimal format is between −32768.0000 and +32767.9999, with an optional decimal point and fraction. Scaled decimal format is used when user-unit scaling is active. It applies to all HP-GL instruction parameters that are interpreted as user units. Decimal format numbers are of the range −128.000 to +127.9999 (with decimal point and fraction optional). Label fields may contain any ASCII text.

Hewlett-Packard has incorporated a sophisticated graphics extension to the Basic interpreters in its computers. This extension, called AGL (A Graphics Language), translates statements written in HP extended Basic into one or more HP-GL commands. You don't need AGL extensions to run HP-GL plotters, however. HP-GL commands can be generated on non-HP computers with PRINT statements.

I am going to present two programs using HP-GL written on an IBM PC in Microsoft Basic. The first, a very simple program, draws a circle, and the second, with just a little more complexity, plots some data in a very fancy manner. These programs will illustrate some of the more useful HP-GL commands and how to generate them.

Drawing a circle

Listing 1 is the circle-drawing program, adapted from a program fragment in the 7470A Interfacing and Programming manual. Figure 3 is the circle drawn by this program. Most of the statements in Listing 1 are plain vanilla Basic. We will look closely at those lines peculiar to IBM's version of Microsoft Basic and the lines that output HP-GL commands.

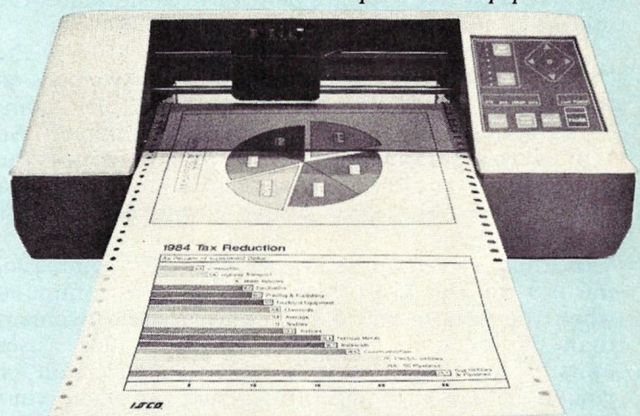
Line 1100 connects the IBM serial port (COM1:) to file #1 so that we can print to it. The extra parameters of the OPEN statement configure the port for 9600-baud operation, with no parity, eight bits per character, one-stop bit, 10-second timeouts on the Clear-to-Send and Data-Set-Ready lines, and an automatic line feed generation with every carriage return. These are the parameters that work with my 7470A. Some Basics do not require line 1100 (such as North Star's). You can print to the 7470A at 9600 baud only if you have implemented either the hardwire or software handshake—otherwise the data will be sent too quickly and the plotter will lose some of the commands.

The plotter is initialized on line 1130. Here we encounter out first HP-GL command. It is "IN", the INitialize plotter command that has no parameters. This places the plotter in its initial power-on state and returns all plotter internal parameters to their defaults. It is a very good idea to send the "IN" command to the plotter at the beginning of each run to place it in a known state. No telling what the last program did to it.

A semicolon terminator separates the "IN" command from the second command. This is the "IP" or Input P1 and P2. The lower left corner of the plot is called P1 and the upper right is P2; they define the area for scaling. The "IP" command sets the absolute positions of these two points on the paper, defining the plotting rectangle. All lines drawn by the plotter will fall in this area when AGL is being used. Otherwise, P1 and P2 do not restrict pen motion. The parameters of the "IP" command are in "plotter units," which are 0.025 mm. An 8½" x 11" sheet of paper is approximately 7,560 by 10,300 plotter units.

In our circle program, we set up the lower left and upper right points at 2000,1000 and 8000,7000 respectively. This gives us a square area to plot on. As

The Nicolet Zeta 8: An 8-pen desktop plotter



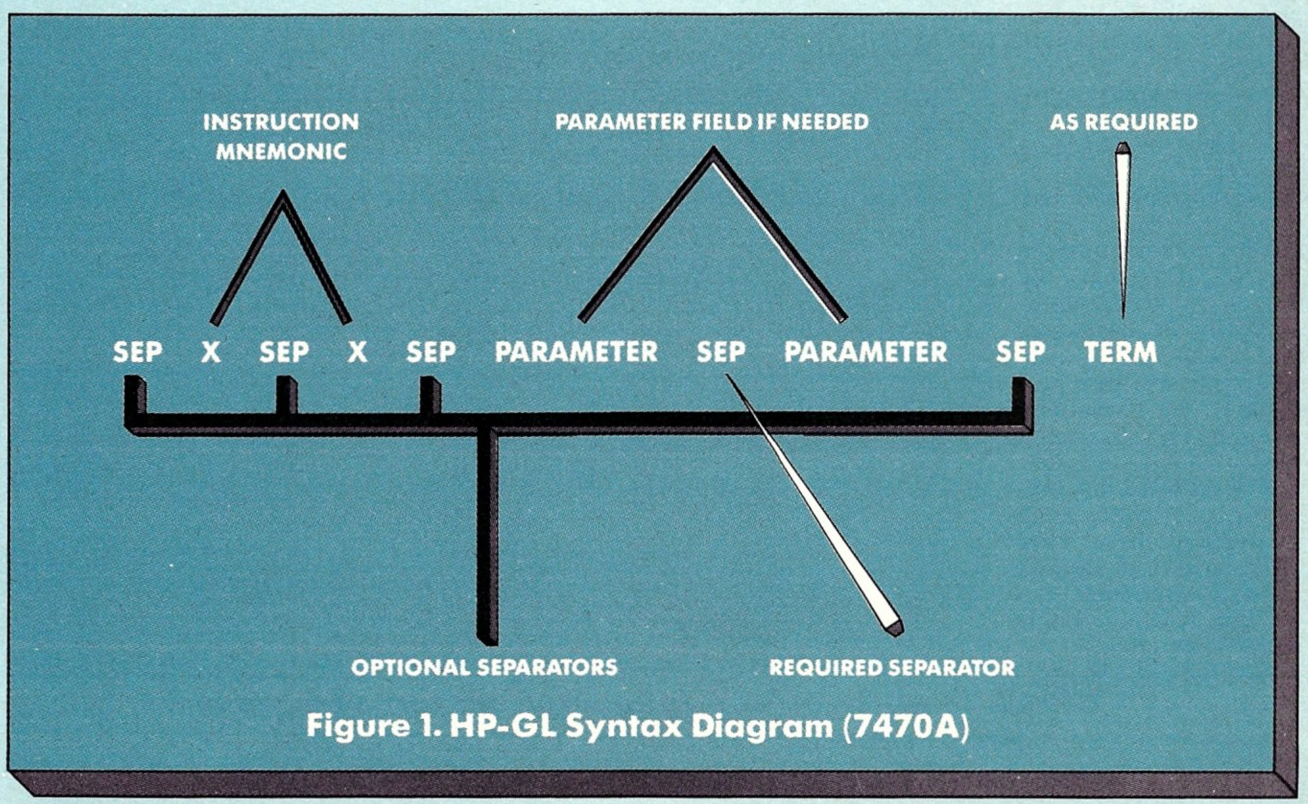
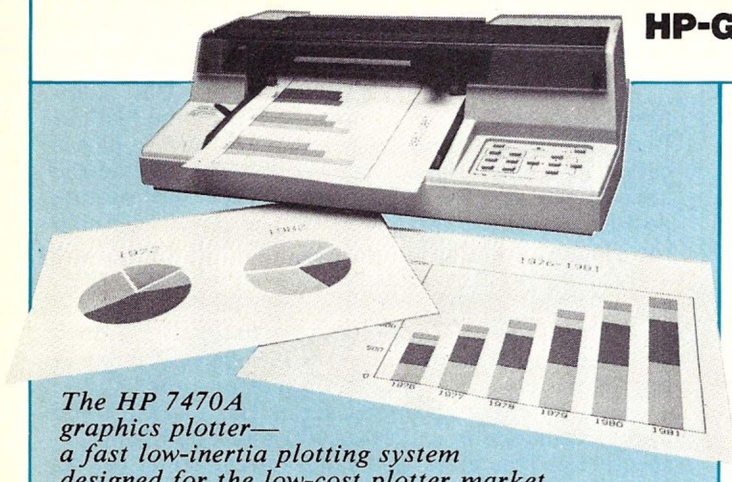


Figure 1. HP-GL Syntax Diagram (7470A)



The HP 7470A graphics plotter— a fast low-inertia plotting system designed for the low-cost plotter market

we shall soon see, a square plotting area simplifies the scaling of the plot.

The next HP-GL command on line 1130 is the "SP" or Select Pen command. There are two pens on the 7470A. You pick up the first with an "SP1" command and the other with an "SP2" command. An "SP" with a parameter of 0 or no parameter puts the pen away. The plotter knows which pen it has and replaces it in the proper holder. Larger HP plotters have eight pens.

Scaling is accomplished with the "SC" command, the last on line 1130. You use the scale command when you wish to plot data in units more convenient than plotter units. Scaled units are called "user units." Thus the plotter will make the conversion for you. A little distributed intelligence can make your job much easier. Here we scale the square plotting area so that it is two user units by two user units. Since we are going to plot a circle with a radius of 1, it should conveniently fit in the center of our square plotting area.

I have included a semicolon at the end of the scale command as a terminator. Since this is the last command in the PRINT statement, the line feed that will automatically be output at the completion of the statement will also terminate the command when the HP-IB is being used. Double termination is allowed by HP-GL syntax: Better safe than sorry. Line 1130 also shows that several HP-GL commands can be sent with a single PRINT statement.

The main body of the circle-drawing program is the FOR/NEXT loop found from line 1190 through 1250. We draw a circle by computing the endpoints of 60-line segments and plotting them. The actual plotting command, Plot Absolute (PA) is spread over three program lines: 1220 through 1240. Line 1220 outputs the instruction mnemonic, line 1230 outputs an endpoint to plot to, and line 1240 tells the plotter to put the pen down ("PD") when it gets to the specified location. This command is on three lines so that the variables x and y could be sent with the PRINT USING syntax, and to illustrate that a single HP-GL command need not be sent with a single PRINT statement.

The first time that the "PA" command is output, the plotter has not put the pen down (pen up is the power-up default). Thus the plotter will first move to the desired point and then put the pen down because the "PD" follows the coordinate parameters. This is equivalent to a MOVE statment in some graphics-extended Basics. For all subsequent "PA" commands, the plotter will draw a line from where it was to where it ends up. This is equivalent to a DRAW statement in some Basics.

I had to put a fudge factor in the FOR/NEXT parameter because the plotter didn't draw the last line segment of the circle. When loop counter T reached 2*PI, the loop was satisfied and the last "PA" wasn't sent. By telling the loop counter to go just a bit farther, the problem went away.

To demonstrate the labeling capabilities of the 7470A, I printed some text in the circle. Line 1270 is the character SIze command. The parameters define the width and heighth of the label characters in cen-

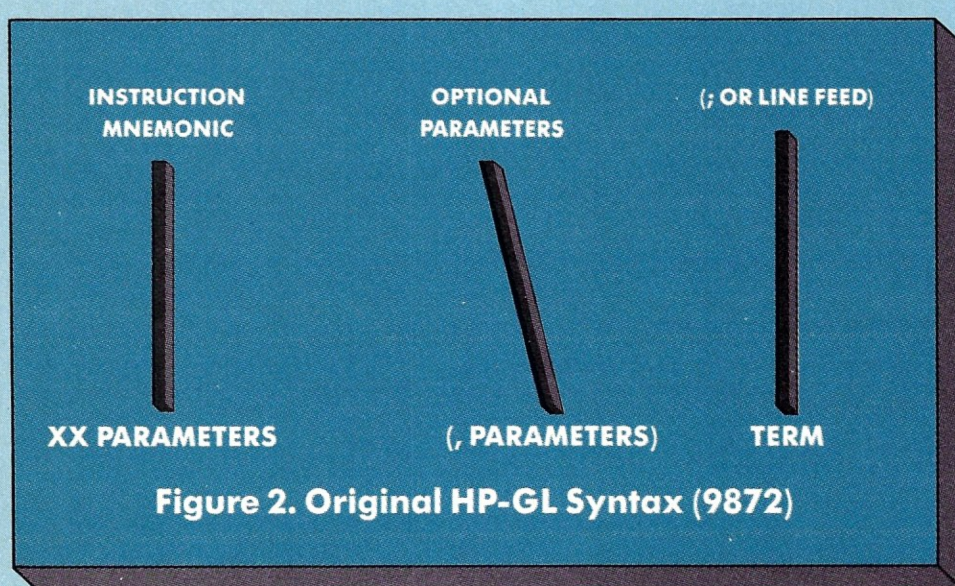


Figure 2. Original HP-GL Syntax (9872)

HP-GL is an interface language, as opposed to a programming language. You control the peripheral through characters sent to the device.

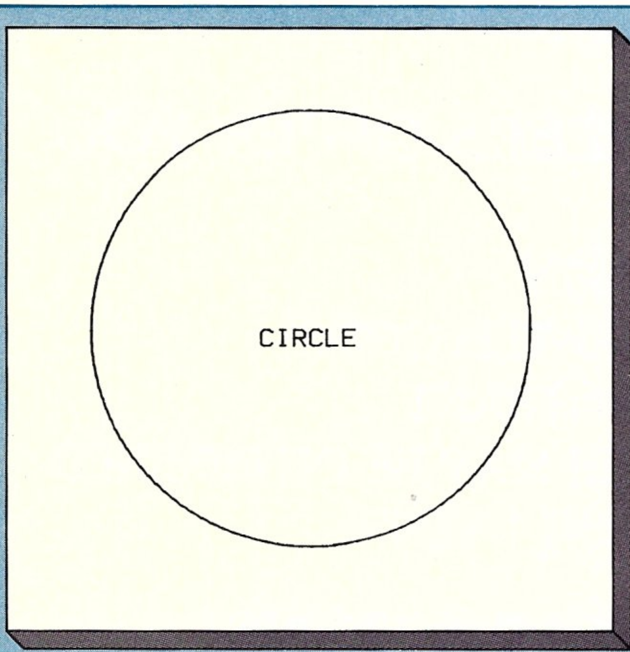


Figure 3. *Circle drawn by the program in Listing 1.*

timeters. Line 1280 moves the pen to location 0,0 after first lifting the pen up (PU). Remember, we left the pen down after plotting the circle. (Yes, I did forget to do this the first time!)

Line 1290 is a very useful command, the Character Plot ("CP") command. It allows you to place the pen at a location based on the size of the characters you are about to plot. Here we back up three character spaces and down half a space to center the label. Line 1300 is the LaBel command ("LB") which tells the plotter to print all subsequent characters until an ETX is received. The ETX is sent using the CHR$(3). Finally, we put the pen away with "SP0" so it won't dry out.

A square deal

Earlier I mentioned that it was easier to deal with a square plotting area. Suppose we had left out the "IP" command and used the plotter's default, which is rectangular. Listing 2 shows the only change we will make to the program, the deletion of the "IP" command in line 1130. When this is done, we get the drawing of Figure 4. The circle is now an elipse. This happened because the plotter-unit scales for x and y are equal, but the user units are not. Thus one user unit in the x direction is larger than one user unit in the y direction. Sometimes, this can be used to intentionally stretch or distort a plot.

Plotting data

Listing 3 is a program for plotting data. It uses more HP-GL commands and therefore more features of the 7470A. Figure 5 is a plot produced by this program. Line 1140 again initializes the plotter, selects a pen, and sets up a plotting area.

Line 1150 sets up our scales. The x axis is going to represent months and is therefore scaled into 12 user units. The y axis is scaled for our data. If the maximum and minimum limits on the data are unknown, it is a fairly simple matter to write the program to determine these and use variables for the scaling parameters. Line 1160 draws the axes and then raises the pen.

On line 1200, we again find a SIze command to set the label character size, along with a Tick Length command ("TL") which sets the size of the tick marks on the axes. Tick marks are specified as a percentage of the horizontal and vertical size of the plotting area. The first Tick Length parameter specifies the length of the mark above the x axis and to the right of the y axis, while the second parameter specifies the length below and to the left. A grid can be drawn using tick marks of 100% length.

Ticks are drawn by using the Plot Absolute command to position the pen, and the X Tick ("XT") and Y Tick ("YT") commands to draw the ticks. Since the pen was left up, the Tick commands will draw a tick mark and then return the pen to the up position. Also, since the pen is in the general vicinity, we use the Character Plot and LaBel commands to label the ticks at the same time. The x axis is ticked and labeled by lines 1210 through 1270, while lines 1330 through 1380 do the y axis. Lines 1280 and 1390-1400 label the axes.

At the line 1410, we finally get through with the gingerbread and plot some data. We are going to plot three sets of data and take advantage of another 7470A feature to do this. Lines 1450 through 1480 set the Line Type ("LT"). Instead of a solid line, each set of data is plotted with a unique dash pattern,

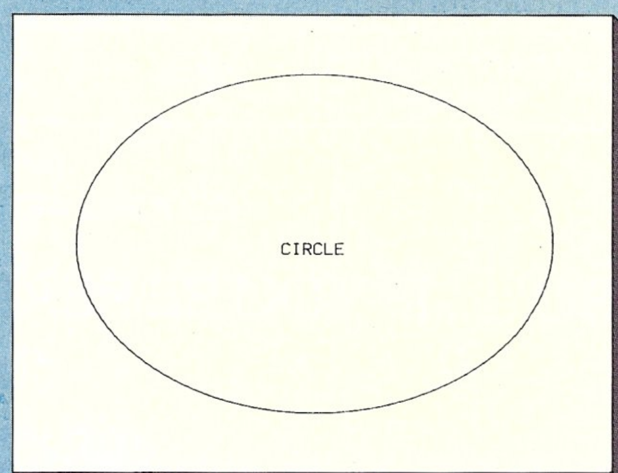
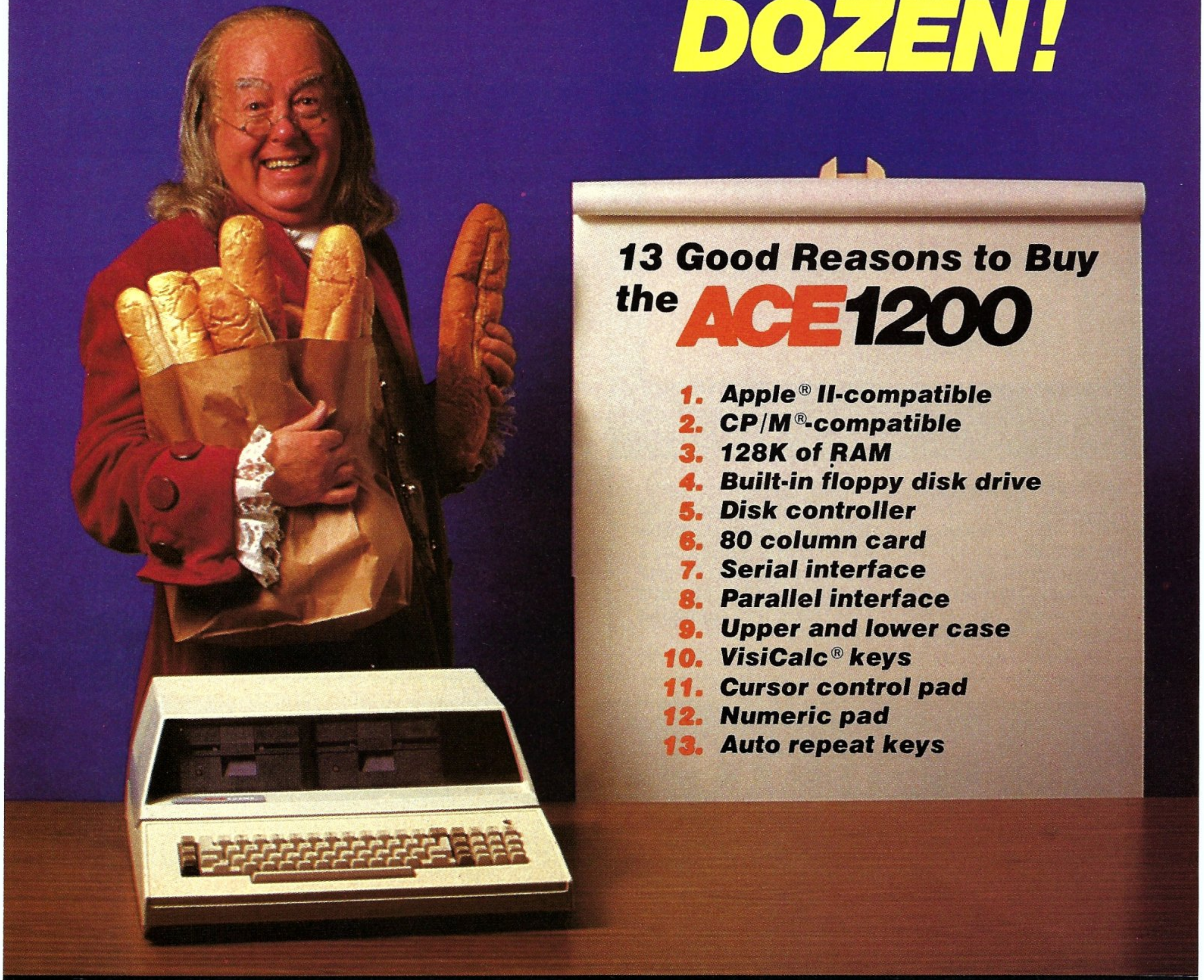


Figure 4. *Circle in Figure 3 redrawn using the plotter's default, which makes a user unit on the x axis larger than one on the y axis.*

HP-GL commands can also be generated on non-HP computers by using PRINT statements.

FRANKLIN'S BAKER'S DOZEN!



Extras can more than double the price of your personal computer. Not so with the Franklin ACE 1200. It's the professional computer system that includes the extras—and a long list of exclusive Franklin features that make it the most extraordinary value on the market today.

The ACE 1200 has everything you'll need to add a color or black and white monitor, modem, printer, back-up disk drive and other accessories. You can choose from the enormous selection of Apple programs and peripherals because the ACE 1200 is hardware- and software-compatible with the Apple II. And, with the built-in CP/M card, you can run both Apple II and CP/M programs. Franklin's CP/M operates three times as fast as many competing systems, drastically reducing processing time for most business applications.

The Franklin ACE 1200—the most extraordinary value on the market today. Call or write today for the name of your local authorized Franklin dealer.

Franklin ACE is a trademark of Franklin Computer Corporation.
Apple is a registered trademark of Apple Computer Inc.
CP/M is a registered trademark of Digital Research Inc.
VisiCalc is a registered trademark of Visi Corp.

FRANKLIN
COMPUTER CORPORATION

2138 Route 38, Cherry Hill, NJ 08002 609-482-5900 Telex: 837-385

CIRCLE 227 ON READER SERVICE CARD

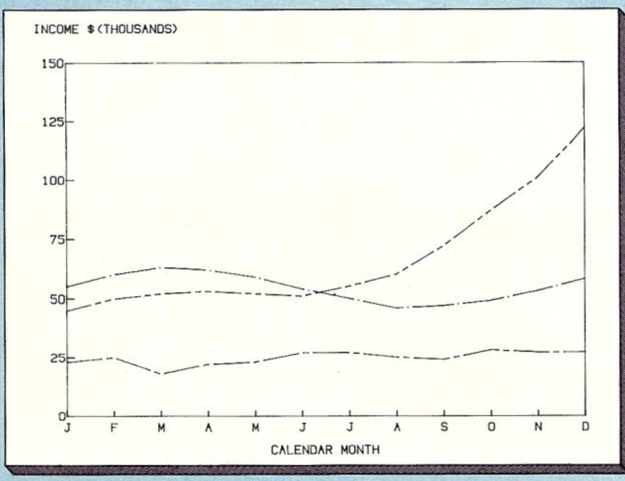


Figure 5. *Plot produced by the program given in Listing 3.*

line types 4 through 6. These line types are preprogrammed. There are 6 of them. The first parameter of the "LT" command specifies the line type, while the second specifies the pattern repeat length as a percentage of the diagonal distance between P1 and P2. Lines 1490 through 1550 plot the data. Again I have spread one HP-GL Plot Absolute command over three Basic PRINT statements.

One more note. As I developed the programs in the listings, the plots would frequently go awry. The best indication that things were not going as they should be was a flashing error light on the 7470A control panel. When this happened, I would note where in the plot the error occurred and would then redirect the PRINT statments from the suspect program lines to my screen. Then I could see what the computer was actually sending to the plotter.

Usually, the problem was caused by the automatic formatting of the numeric parameters by the PRINT statement. For example, when a number becomes small enough, the PRINT statement switches from fixed-point format to floating point, which gives the plotter indigestion. That is the reason you find PRINT USING statements for numeric output in my programs.

Because of HP-GL's ASCII orientation, this type of troubleshooting is easy to do. Simply print the offending output and compare it to the syntax diagram of Figure 1 to find your problem.

Some final words

I hope that these two simple programs have shown how simple it is to produce very fine graphic output with your computer and HP-GL. There are many commands we haven't covered. The 7470A has 42 HP-GL commands for plotting, and the RS-232C version adds 14 escape sequences to control the interface. Larger HP plotters have even more commands. If you want or need graphics, HP-GL may just be your ticket.

VEDIT
CUSTOMIZABLE
FULL SCREEN EDITOR

The Professional's Editor for Program Development Word Processing Source Code Translations

Widely acclaimed as an editor, VEDIT has evolved to be much more. Only VEDIT offers the combination of a versatile full screen editor integrated with a powerful command language. For the first time you'll be able to perform complex, yet useful, text manipulations that are virtually impossible with other editors or word processors. Plus, its customizability and hardware support ensure that VEDIT will be perfectly matched to your individual needs and to any microcomputer you are ever likely to own.

With two modes of operation, VEDIT never compromises its speed or ease of use for its power and sophistication. As one reviewer (Bradford Thompson, BYTE) wrote: 'If this review gives you an appetite for simplicity while editing, then VEDIT is well worth considering.' Its command language, based on TECO, is virtually a text oriented programming language, allowing command macros to be created, loaded and saved on disk. Yet its simplicity allows even a novice to perform tasks beyond the capabilities of any word processor.

VEDIT cuts programming time in half - with multiple file handling, macro capability and special features for Pascal, PL/1, 'C', Cobol, Assembler and other languages. And it can help with source code translations (example ZILOG to/from INTEL translator macros are included). A complete line of translators will be available by the year's end.

Word processing is a snap with word wrap, paragraph and print functions. Command macros free you from tedious search/replace operations. Hundreds of search/replace on dozens of files can be performed by VEDIT without waiting or intervening.

VEDIT easily configures to your favorite keyboard layout. Use any function or cursor keys you wish. It optimally supports nearly every 8080, Z80 and 8086 computer.

Go ahead and expect a lot from VEDIT. Its performance and our support will satisfy your most exacting needs.

To order, please specify your 8080, Z80 or 8086 microcomputer, operating system and disk format.

COMPARE VEDIT'S FEATURES

- True Full Screen Editing
- Horizontal scrolling
- Edit files one disk in length
- Automatic Disk Buffering
- Compact (only 16K) and Fast
- Display of line and column #
- Set/Goto text markers
- 'Undo' key to restore line
- Automatic Indent/Undent
- Adjustable Tab positions
- Repeat function key
- Text Move and Copy
- 10 Scratchpad Buffers
- Load/Save buffers on disk
- Powerful command macros
- Directory display
- Edit additional (small) files simultaneously
- Insert another disk file
- Unlimited file handling
- Recovery from 'Full Disk'
- Change disks while editing
- Word wrap, format paragraph
- Simple Printing
- 150 page indexed manual
- Startup command file
- Menu driven installation
- Program CRT function keys
- Support newest CRT terminals
- Support smart CRT functions
- Flexible Memory Mapped support
- Customizable keyboard layout

IBM PC, Displaywriter ° Zenith Z100 and Z89 ° NEC APC ° DEC Rainbow and VT180 ° Televideo 802
TRS-80 I, II and 16 ° Xerox 820 ° Apple II Softcard ° SuperBrain ° NorthStar
MP/M ° CP/M-86 ° MP/M-86 ° Concurrent CP/M-86 ° Cromix ° Turbo DOS ° MSDOS ° PCDOS

VEDIT - Disk and Manual
8080, Z80 or IBM PC $150
CP/M-86 or MSDOS $195
Manual only $18

CompuView
PRODUCTS, INC.

1955 Pauline Blvd., Suite 200 ° Ann Arbor, Michigan 48103 ° (313) 996-1299

CIRCLE 32 ON READER SERVICE CARD

Listing 1

```
1000 REM ***********************************************************
1010 REM
1020 REM DRAW A CIRCLE USING HP-GL COMMANDS
1030 REM
1040 REM   ADAPTED FROM HP BASIC TO IBM/MICROSOFT BASIC BY
1050 REM   STEVE LEIBSON    JANUARY 30, 1983
1060 REM
1070 REM ***********************************************************
1080 REM
1090 REM FIRST, CONNECT BASIC TO THE SERIAL PORT
1100 OPEN "COM1:9600,N,8,1,CS10000,DS10000,LF" AS #1
1110 REM
1120 REM INITIALIZE PLOTTER, SET LOWER LEFT, UPPER RIGHT AND SCALE
1130 PRINT #1,"IN;IP2000,1000,8000,7000;SP1;SC -2,2,-2,2;"
1140 REM
1150 REM NOW DRAW THE CIRCLE
1160 REM
1170 DEFSNG X,Y
1180 PI=3.1416
1190 FOR T=0 TO 2*PI+.1 STEP PI/60
1200   X=COS(T)
1210   Y=SIN(T)
1220 PRINT #1,"PA";
1230 PRINT #1,USING "###.###";X;Y;
1240 PRINT #1,"PD"
1250 NEXT T
1260 REM NOW FOR A NICE LABEL
1270 PRINT #1,"SI.19 .27;"
1280 PRINT #1,"PA PU 0 0;"
1290 PRINT #1,"CP -3 -.5;"
1300 PRINT #1,"LBCIRCLE",CHR$(3);
1310 PRINT #1,"SP0;"
1320 END
```

Listing 2

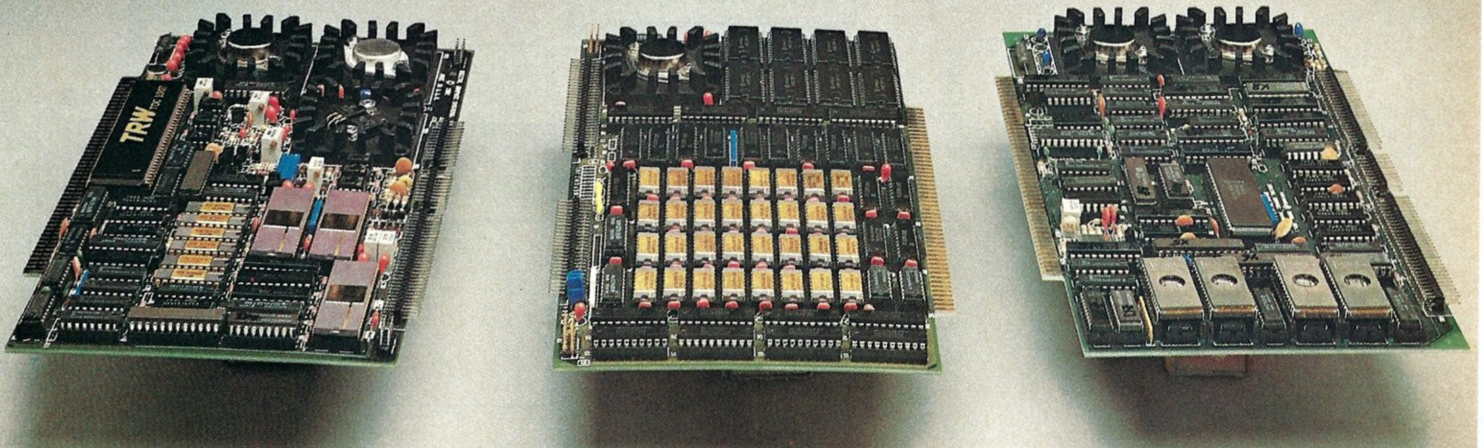
```
1130 PRINT #1,"IN;SP1;SC -2,2,-2,2;"
```

Listing 3

```
1000 REM ***********************************************************
1010 REM
1020 REM MAKE A PLOT USING HP-GL COMMANDS
1030 REM
1040 REM   ADAPTED FROM HP BASIC TO IBM/MICROSOFT BASIC BY
1050 REM   STEVE LEIBSON    JANUARY 30, 1983
1060 REM
1070 REM ***********************************************************
1080 REM
1090 REM FIRST, CONNECT BASIC TO THE SERIAL PORT
1100 OPEN "COM1:9600,N,8,1,CS10000,DS10000,LF" AS #1
1110 REM
1120 REM INITIALIZE THE PLOTTER AND DRAW THE AXES
1130 REM
1140 PRINT #1,"IN;SP1;IP1250,750,9250,6250;"
1150 PRINT #1,"SC1,12,0,150;"
1160 PRINT #1,"PA 1,0 PD 12,0 12,150 1,150 1,0 PU;"
1170 REM
1180 REM LABEL THE X-AXIS
1190 REM
1200 PRINT #1,"SI.2,.3;TL1.5,0;"
1210 FOR X=1 TO 12
1220   PRINT #1,"PA ";
1230   PRINT #1,USING "##";X;0;
1240   PRINT #1," XT;"
1250   READ A$
1260   PRINT #1,"CP-.33,-1;LB";A$;CHR$(3)
1270 NEXT X
1280 PRINT #1,"PA6.5,0;CP-7,-2.5;LBCALENDAR MONTH";CHR$(3)
1290 DATA "J","F","M","A","M","J","J","A","S","O","N","D"
1300 REM
1310 REM AND NOW THE Y-AXIS
1320 REM
1330 FOR Y=0 TO 150 STEP 25
1340   PRINT #1,"PA 1,",Y,"YT;"
1350   PRINT #1,"CP-4,-.25;LB";
1360   PRINT #1,USING "####";Y;
1370 PRINT #1,CHR$(3)
1380 NEXT Y
1390 PRINT #1,"PA1,150;CP-4,2;LBINCOME $";CHR$(3);CP -9,-1;"
1400 PRINT #1,"LB(THOUSANDS)";CHR$(3)
1410 REM FINALLY, PLOT SOME DATA
1420 REM
1430 PRINT #1,"LT3;"
1440 FOR P=4 TO 6
1450   PRINT #1,"LT";
1460   PRINT #1,USING "##";P;
1470  PRINT #1,",8;"
1480   PRINT #1,"PU"
1490   FOR X=1 TO 12
1500     READ Y
1510     PRINT #1,"PA";
1520     PRINT #1,USING "####";X;Y;
1530     PRINT #1,"PD"
1540   NEXT X
1550 NEXT P
1560 PRINT #1,"SP;"
1570 DATA 55,60,63,62,59,54,50,46,47,49,53,58
1580 DATA 23,25,18,22,23,27,27,25,24,28,27,27
1590 DATA 45,50,52,53,52,51,55,60,72,87,101,122
1600 END
```

The Image Solution

The CAT 1600 Series lets you take full advantage of your color graphics potential.



Plug this powerful color video graphic system into your IEEE-696 bus and watch your computer open its eyes. Exercise your creativity developing new ways to study your world and discovering the flexibility of video imaging. Our real time frame grabber gives you instant availability of the image to be processed. The CAT 1600 is the creative link between machine, man and the world.

Resolution is the name of the game, and we've got it. Physically you're looking at 512 × 512 pixels up to 24 bits deep. And that's real color. Now, center on a pixel, any pixel, and roam the screen through an image space as large as 1K × 2K. Zoom in and explore a close-up of 32:1, not in the usual quantum leaps of integer zooms, but in smooth logarithmic steps of 1.1%. A smooth zoom... that's human engineering.

At the heart of the matter is a dedicated 8086 image processor. It blazes a 16 bit wide path through the various memories, lookup tables and image parameters as it executes high level commands from your host processor. Up to 48K of static RAM makes the image processor useful for downloading custom programs from the host.

When it came to adequate memory, we didn't forget. 768KB of dynamic memory gives you plenty of image. Our PROMs have a library of 64K organized into over 130 sophisticated graphics commands such as continuous live digitization, character and shape generation, global image manipulation and animation effects, to relieve the host computer from low level primitives.

At your disposal is a palette of 16.7 million colors and 256 shades of gray. Quantized lines or free-hand sketching completes the picture. Use a variety of pen widths, brush strokes, even airbrush! Now imagine what you can do with a superb quality image captured in real time from a color video camera. Contact us for an eye opening demonstration: 935 Industrial Avenue, Palo Alto, CA 94303. 415/856-2500.

DIGITAL GRAPHIC SYSTEMS, INC.

What You See Is What You Get.

CIRCLE 249 ON READER SERVICE CARD

A Laboratory Application of Microcomputer Graphics

by Kalle B. Gehring and John W. Moore

In the summer of 1980 we developed a Pascal graphics and instrument interface program for one of our Z80/S-100 based microcomputers. The computer interfaces to a stopped-flow spectrophotometer and replaces a storage oscilloscope and polaroid camera. A TV monitor connected to a set of Cambridge Development Laboratory (CDL) graphics boards displays kinetic traces, experimental parameters, and the results of analysis.

The stopped-flow interface (Figure 1) shows the value of having a computer graphics system connected to a laboratory instrument. Although data points can be read by the computer without the graphics system, the user has no way of knowing whether the data is valid. The graphical representation of the kinetic traces allows users to evaluate the experiment as it progresses and to correct experimental errors before they are saved for analysis. In our stopped-flow interface it is also necessary to see the kinetic trace in order to decide what part of the trace should be used for the analysis. Our system uses a lightpen to read the endpoints used in analysis from the TV screen.

Kalle B. Gehring and John W. Moore, Chemistry Dept., Eastern Michigan University, Ypsilanti, MI 48197

Besides simply replacing an oscilloscope, the interface program provides permanent storage of the kinetic traces that can later be recalled on the TV monitor. A second program performs a least-squares analysis on the traces to find the kinetic rate constants. Previously kinetic traces were digitized by hand from a polaroid photograph and entered into a hand-held calculator for analysis. The computer interface not only provides greater throughput, but digitizes the kinetic traces more accurately and can statistically combine the results of several hours of data collection.

A stopped-flow spectrophotometer measures the rate of a chemical reaction from the rate of disap-

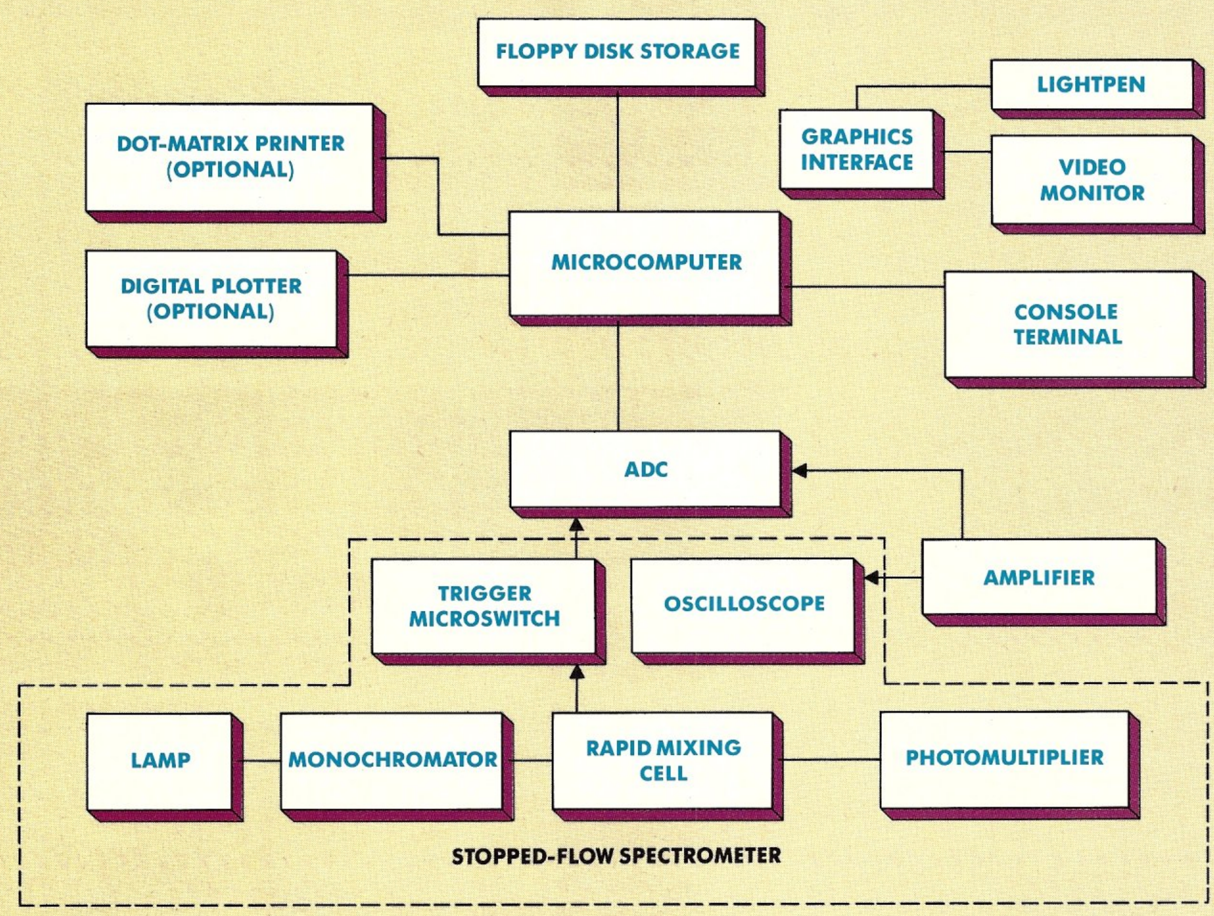


Figure 1. Block diagram of a microcomputer-interfaced stopped-flow spectrophotometer.

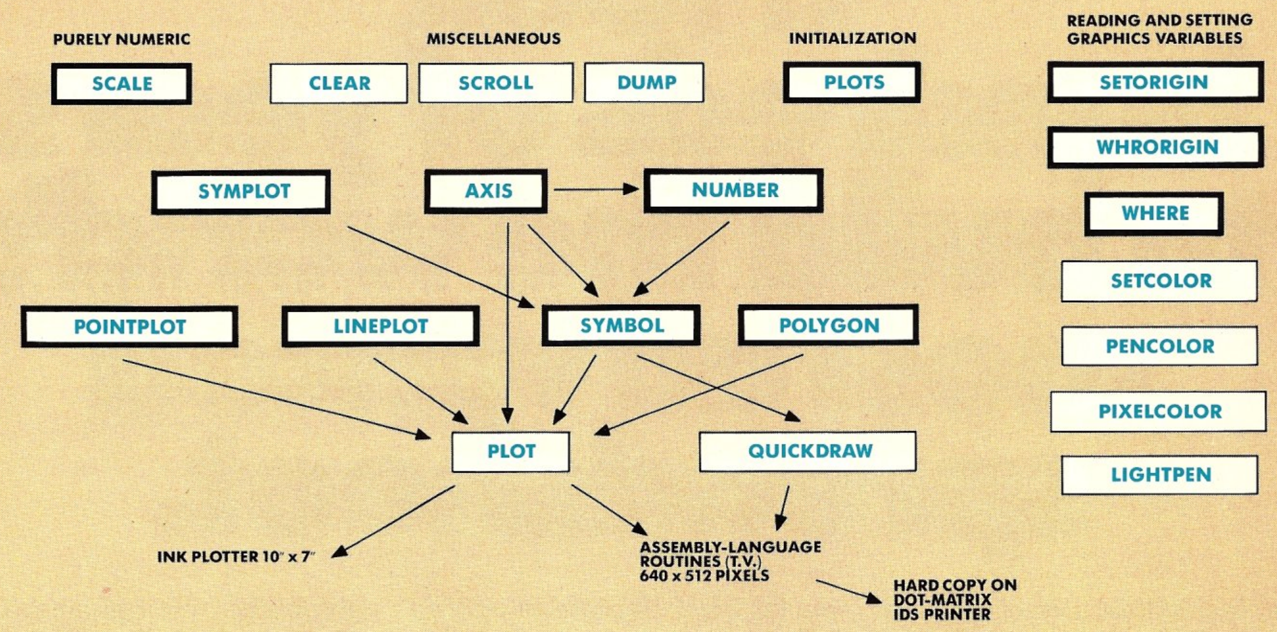


Figure 2. *Flowchart of the graphics subroutines. Boldface routines use both the TV graphics boards and the plotter. Except for QUICKDRAW, all the functions draw lines by calling PLOT, which allows programs to preview plots on the TV monitor before plotting. This software is available for a nominal fee through the USUS library. The figure was drawn by a HIPLOT ink plotter.*

pearance or appearance of a colored reactant. In the stopped-flow apparatus, two solutions flow through a Y-shaped tube, where they mix and start to react. At the bottom of the Y tube is an observation cell and a photomultiplier tube. The time course of the reaction is studied by abruptly stopping the flow, so that instead of flowing out of the tube the solutions react in the observation cell. Our spectrophotometer, a Beckman DU monochromator, is connected to a linear-log photometer that provides a 10V signal that reflects the intensity of light passing through the reacting solution. This voltage signal is digitized by a multichannel Tecmar analog-to-digital (ADC) board connected to the S-100 bus of the Z80. When the flowing solutions are stopped, the stopped-flow apparatus closes a circuit to signal the computer to start the kinetic trace. The ADC board polls this circuit and starts converting the photometer signal when the circuit is closed.

Our system was developed using UCSD Adaptable Pascal II.0 (see the November 1982 *Microsystems* for a description of implementing UCSD Pascal on our system). UCSD Pascal provides an editor, linker, and Z80 assembler as well as the Pascal compiler and p-machine. The graphics software is in a UCSD Pascal Unit that allows the graphics subroutines to be compiled separately from the interface programs. This allowed the graphics subroutines to be written before the interface program and allows the graphics system to be used by other application programs. A set of Z80 assembler routines are associated with the Graphics Unit for the low-level control of the graphics boards. The two interface programs, Data Collection and Data Analysis, are both Pascal programs that call procedures in the Graphics Unit. The Data Collection program also calls an assembly-language routine to do the time-critical analog-to-digital conversions.

The Graphics Unit was developed along the lines of the Fortran Calcomp plotting routines. The software was originally written for a Houston Instruments digital plotter and was later upgraded for the TV boards. Programs can switch easily between the plotter and TV monitor because output from the Graphics Unit is funnelled through one subroutine, PLOT. PLOT chooses to draw on the plotter or to call an assembly-language subroutine to draw on the TV monitor. The stopped-flow interface does not use the plotter because of the availability of hard copy of the TV screen on a dot-matrix printer. Both the Data Collection and Analysis programs allow the experimenter to save a picture of the kinetic trace by calling (through the Graphics Unit) the assembly-language subroutine that scans the graphics boards memory and prints it on an Integral Data Systems (IDS) 445 printer. Figure 2 shows a flowchart of the Pascal subroutines in the Graphics Unit. The higher-level subroutines draw graphs and text, while the lower-level routines draw lines, clear the screen, and read variables from the graphics boards.

Graphical representation of the kinetic traces allows users to evaluate the experiment as it progresses and correct experimental errors before they are saved for analysis.

It took hundreds of years to build the world's most advanced spreadsheet.

Way back in 1534, a Venetian scholar wrote a text that standardized the techniques of modern bookkeeping. And up till 1979, there weren't a lot of real improvements in the way people did spreadsheets. You still had to enter all your figures by hand. Run all the totals yourself. And when you had to make a few changes, you had to run the totals all over again.

Then along came a program that let you create spreadsheets on a personal computer. Without touching a pencil. Or adding anything by hand. Best of all, this new program gave you the freedom to look at your balance sheet a dozen different ways if you felt like it. Because all you had to do was change one figure, and the computer would refigure everything else for you. Automatically.

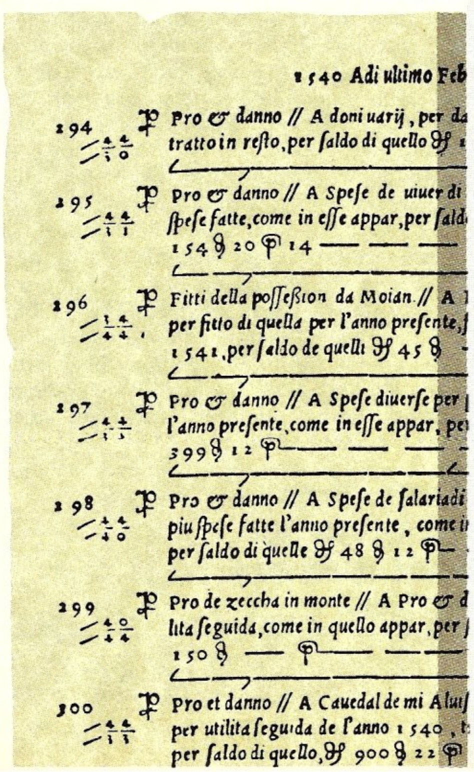
Like most breakthrough ideas though, this original spreadsheet program had a few things that definitely needed improvement. For instance, all the columns had to be the same width. And the entire system took about 20 hours to learn.

So we developed SuperCalc,® which was such a giant step in the right direction that it became a best-seller. Over 250,000 copies in only 18 months.

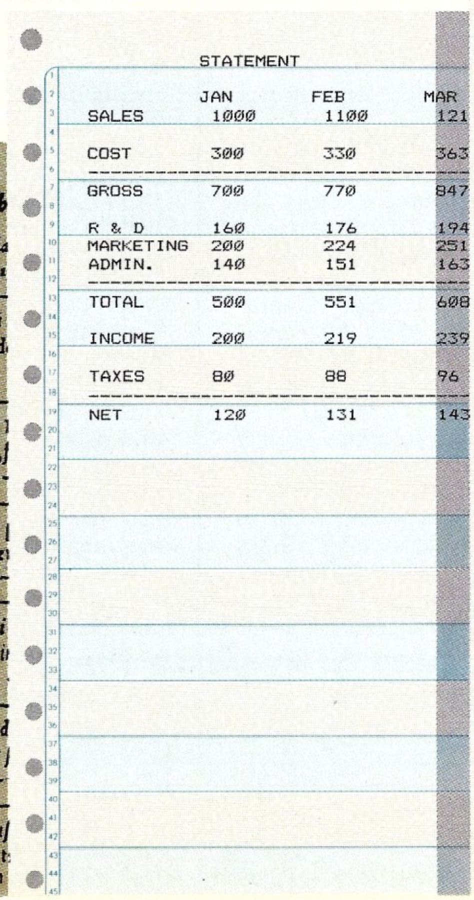
Today, we've gone even farther. After talking with thousands of customers, we've found a lot of important new features you'd like to have. And we've built them into the world's most advanced spreadsheet.

SuperCalc².

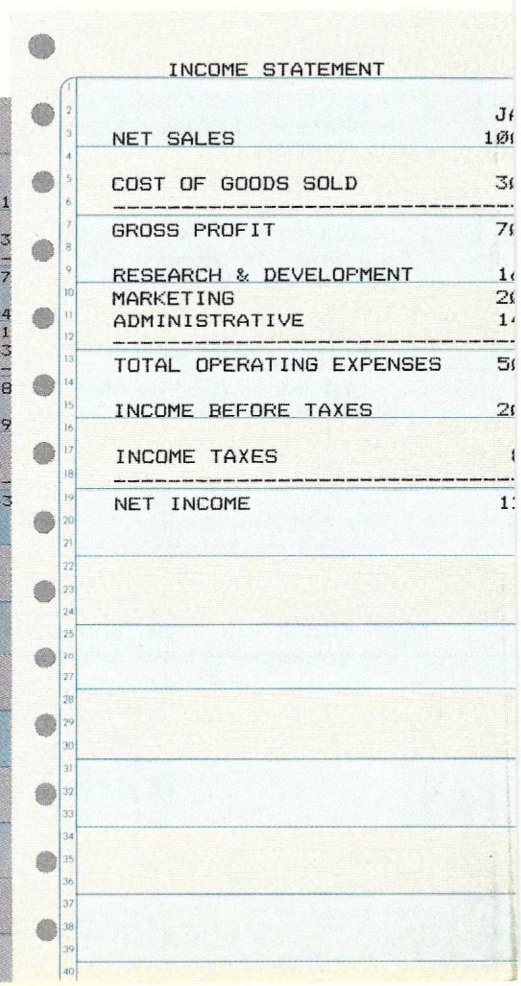
1534



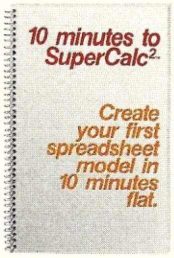
1979



1981



It'll take you 10 minutes to learn it.



With SuperCalc² you get all kinds of features that have never been possible before. Like custom formatting capabilities. Automatic consolidation. Sorting. And more. Take a look at the spreadsheet below, and you'll see just how much we're giving you.

Best of all, we made it just as easy to use as the original SuperCalc. There's a self-paced starter guide that shows you exactly what to do, step by step. And an AnswerKey™ that gives you instant help right on the screen. Because of these features, you can actually create your first spreadsheet in 10 minutes flat. Even if you've never used a computer before.

SuperCalc² is available for the IBM PC and XT. Where it can run in full color. With full memory. Even on hard disk. It's also available for CP/M, CP/M-86, MS DOS and other operating systems, so you can use it with most popular personal computers.

But with all its advanced features, SuperCalc² is the same price as the original SuperCalc. Only $295, suggested retail. We're also offering SuperCalc for only $195. And an upgrade to SuperCalc² whenever you're ready.

So. If you've always thought there had to be a better way to do your spreadsheets, stop by your closest computer store. And take a good look at SuperCalc². After just 10 minutes, you won't want to do one more spreadsheet the old way.

1983

Combines any number of spreadsheets—automatically!

Calendar/date capabilities.

Percentages.

```
        COMPANY CONSOLIDATED MONTHLY PAYROLL

  Today's Date                        4/15/83      Deduction Percentages
  Payroll Start Date                  4/ 1/83      Fica             6.700%
  Days this period                         15      SDI                 .8%
  Recalculate YTD Y/N ?  N
                                    Gross       Total        Net          YTD
   Emp# Employee name   Status      Salary      Deduct       Pay        Gross
  ============================================================================
   34   Adams           M       $1,100.00  ( $82.50) $1,017.50    $6,200.00
   49   Beckette        S         $750.00  ( $56.25)   $693.75    $5,250.00
   84   Johnson         S         $850.00  ( $63.75)   $786.25    $5,950.00
   92   Jones           M         $900.00  ( $67.50)   $832.50    $6,300.00
   12   Samson          S         $560.00  ( $42.00)   $518.00    $3,920.00
   19   Santos          M         $650.00  ( $48.75)   $601.25    $4,550.00
   45   Smith           S         $700.00  ( $52.50)   $647.50    $4,900.00
  ============================================================================
  Total # employees             7

  Total Gross Salaries (100's)      $55.1
  Total Deductions (100's)          $-4.1
  Total Net Pay (100's)             $51.0
```

User-set decimal places.

Floating $ signs.

Embedded commas.

Negative numbers in ( ).

Arrange reports numerically or alphabetically, like this

Numbers in 100s or 1000s.

SuperCalc²™

SORCIM®

2310 Lundy Avenue
San Jose, CA 95131
(408) 942-1727

CIRCLE 57 ON READER SERVICE CARD

Both the interface programs are menu-driven in much the same way as UCSD Pascal. At each point in the program the top line of the console presents the available commands, and the user types just the first character to select a command. An inexperienced user is guided through the experimental procedure by the order of the available commands.

The Data Collection program emulates the oscilloscope screen through the CDL graphics boards and presents prompts on the console. In the experimental procedure, the photometer is first calibrated with 0% and 100% transmittance through the reaction cell. After a sweep speed is selected the computer waits for a signal from the stopped-flow apparatus that the reaction is starting and then records 101 conversions of the reaction trace. The 0%, 100%, and kinetic traces as well as the experimental parameters are displayed by the Graphics Unit on the TV monitor (Figure 3). After each kinetic trace the user is given the option of saving the trace as part of a disk file, discarding it, or printing it on an IDS dot-matrix printer connected to the graphic system.

The actual A to D conversions are done by a Tecmar multichannel ADC board with 12-bit resolution. It is controlled and read by a Z80 assembler routine that uses an interrupt timer on a Cromemco TUART board to provide accurate timing. A to D conversions are initiated at rates from 5000 hertz to 20 hertz. Since each kinetic trace is sampled at 101 points, reactions occurring in 20 milliseconds to 5 seconds can be studied.

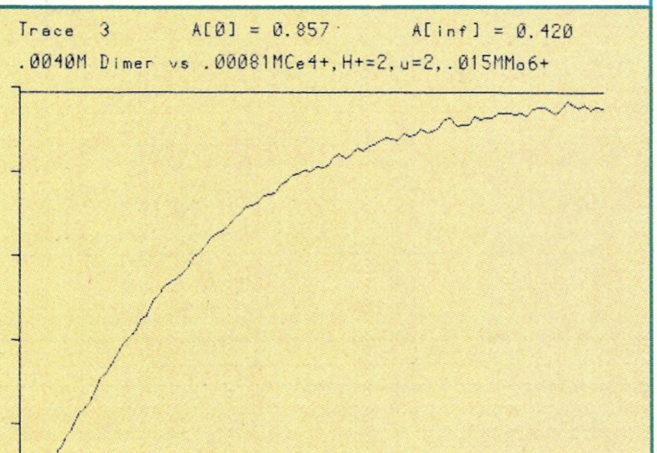


***Figure 3.** Example kinetic trace from an experiment. The TV screen was "DUMPed" to the dot-matrix printer, a process that takes about 2½ minutes. The axes are absorbance versus time. Two traces are shown: the top one is the final (t = infinity) absorbance, and the lower trace is the change in absorbance during the reaction.*

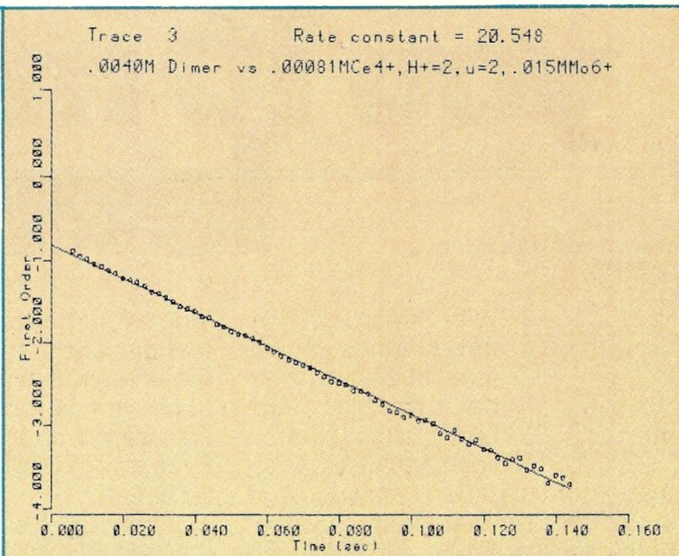


***Figure 4.** Sample kinetic analysis. The data from the previous figure were analyzed and a rate constant determined; this figure shows the data points and the line of best fit as displayed on the TV monitor. (The second line of text above the trace is a description of the experiment.)*

The Analysis program is run after data collection is complete. It reads kinetic traces previously saved as disk files and performs an analysis according to either first- or second-order reaction kinetics. A graph of the data and a line of best fit is displayed on the TV monitor and can be printed. Figure 4 shows a sample graph. At the end of the analysis the kinetic rate constants of several traces can be weighted-averaged and a table of the results saved on disk. In addition to the menu-style input from the console, the Analysis program uses a lightpen connected to the CDL graphics boards to select the section of the kinetic trace to be analyzed.

Our graphics software (with documentation) is available in a variety of disk formats for a small distribution fee through UCSD Pascal User's Group (USUS) library. Write to the USUS library, care of Dr. James Gagne, Datamed Research, 1433 Roscomare Rd., Los Angeles, CA 90024, (213) 472-8825. Anyone interested in obtaining a listing of our two interface programs should write to us.

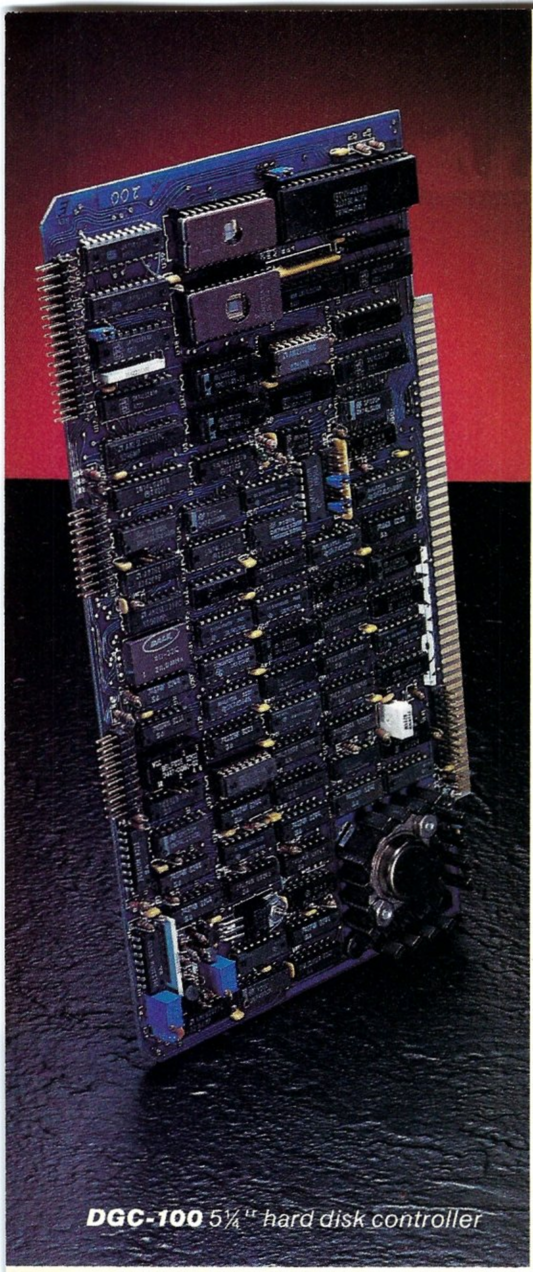
Our TV graphics hardware consists of a set of three Cambridge Development Laboratory graphics boards. The boards contain all of the graphics memory for 512 x 640 pixels and provide a composite video signal for a TV monitor. The boards are controlled through eight I/O ports that implement a number of powerful hardware functions. For displaying text, a screen-scrolling function and a specialized "byte writing" function are available. For our purposes, we were pleased by the high speed of the boards and special features such as the lightpen that can read a po-

The CDL graphics boards and the Tecmar ADC board have worked without any problems, transforming our Z80/S-100 computer into an advanced digital storage oscilloscope.

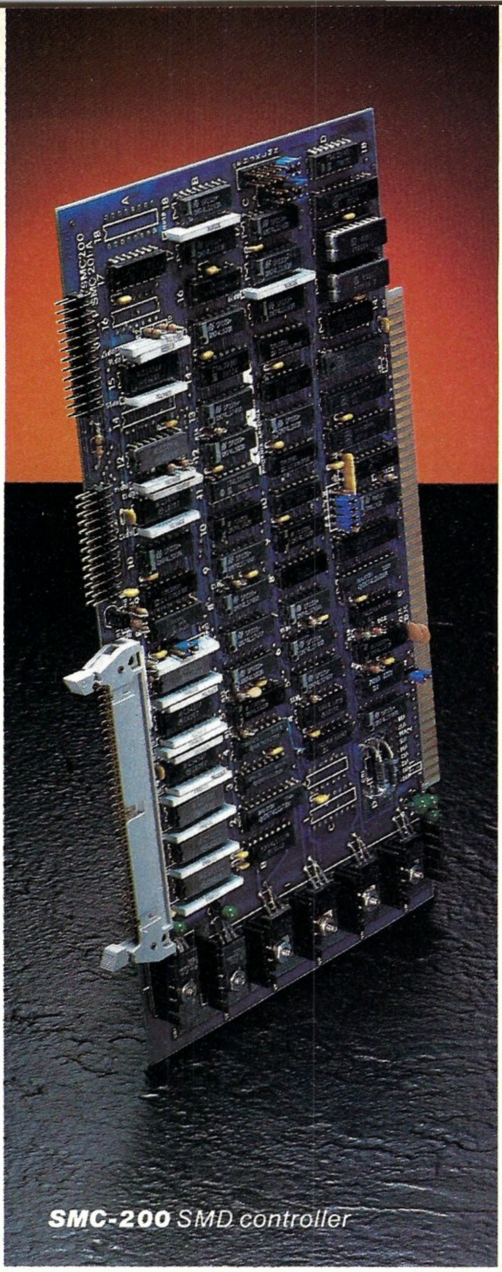
sition from the TV monitor. Up to 16 colors are available with the CDL boards, but this entails purchasing three additional boards for the necessary graphics memory.

The Houston Instruments HIPLOT digital plotter is 10″ by 7″ flatbed ink plotter. It is connected to the computer through a 4800-baud serial port that sends 11 different ASCII characters to control the pen movements. The plotter has a maximum resolution of 200 steps per inch and a maximum speed of 480 steps per second. Our plotter has worked very well connected to the graphics software, although it is not used by the stopped-flow interface. Figure 2 shows the high quality of drawings possible with this inexpensive plotter.

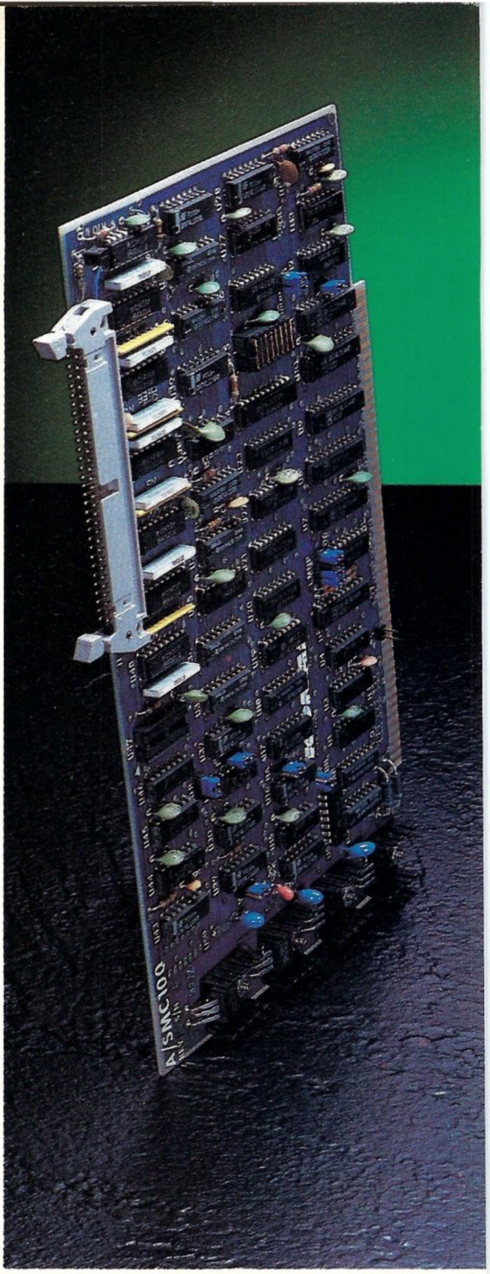
The stopped-flow spectrophotometer interface has been in operation for more than two years with very good results. The CDL graphics boards and the Tecmar ADC board have worked without any problems and have transformed our Z80/S-100 computer into advanced digital storage oscilloscope. The high-resolution graphics displays the experimental data collected from the ADC board, and, with the lightpen, provides an easy means of entering parameters used in the data analysis. Experiments can be saved on 8″ floppy disks for permanent storage, and the graphics boards memory can be printed to give hard-copy graphs of kinetic traces and analyses. To borrow a chemist's term, the rate limiting step in the laboratory is no longer the analysis of the experiments, but preparing and doing the experiments.



DGC-100 5¼" hard disk controller



SMC-200 SMD controller



A SPECTRUM OF THAT PUT YOU

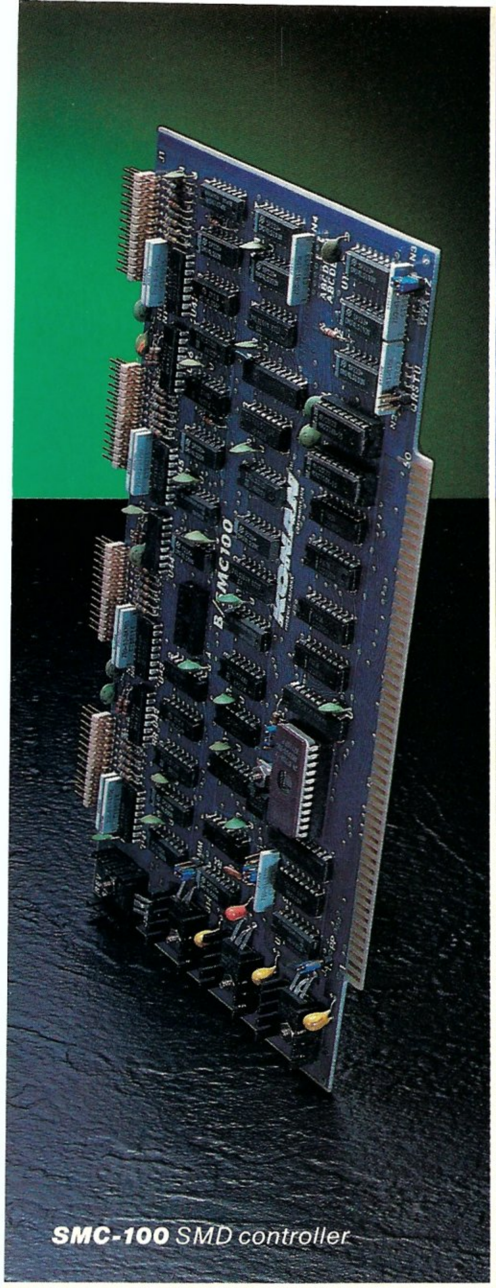
Konan's board array of feature-packed, versatile controllers makes your design decision simple and cost-effective. The Konan S-100 controller line provides you with the assurance that you have designed-in the high performance and quality you demand. And highly qualified, technical support and prompt deliveries go a long way to ensure you'll meet your production schedule... on time! The spectrum includes:

The DGC-100 *A high performance 5¼" hard disk controller... with full sector buffering, error correction, automatic flaw mapping... all standard features. And the versatile DGC-100 supports the known spectrum of Seagate ST-506 compatible drives.*

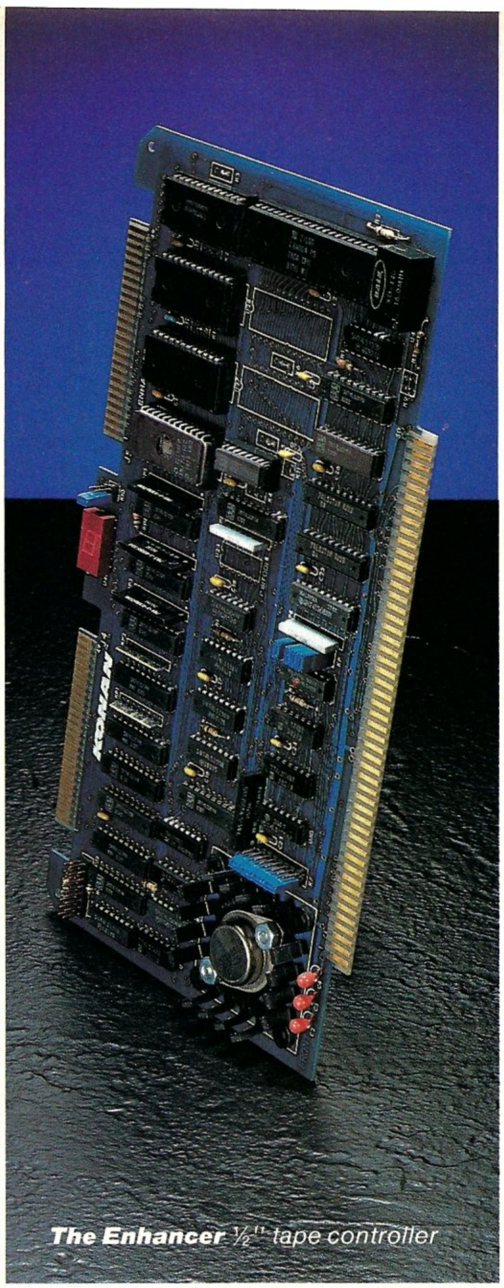
The SMC-200 *A versatile disk controller that interfaces S-100 bus microcomputers with all fixed or removable media disk drives with SMD interfaces. And each SMC-200 will control up to 2 drives and up to 1200 Megabytes... using only one bus slot.*

The SMC-100 *A cost-efficient, versatile controller that interfaces S-100 bus microcomputers with all fixed or removable media disk drives with SMD interfaces. Each SMC-100 will control up to 4 drives and up to 2400 Megabytes... using only 2 bus slots.*

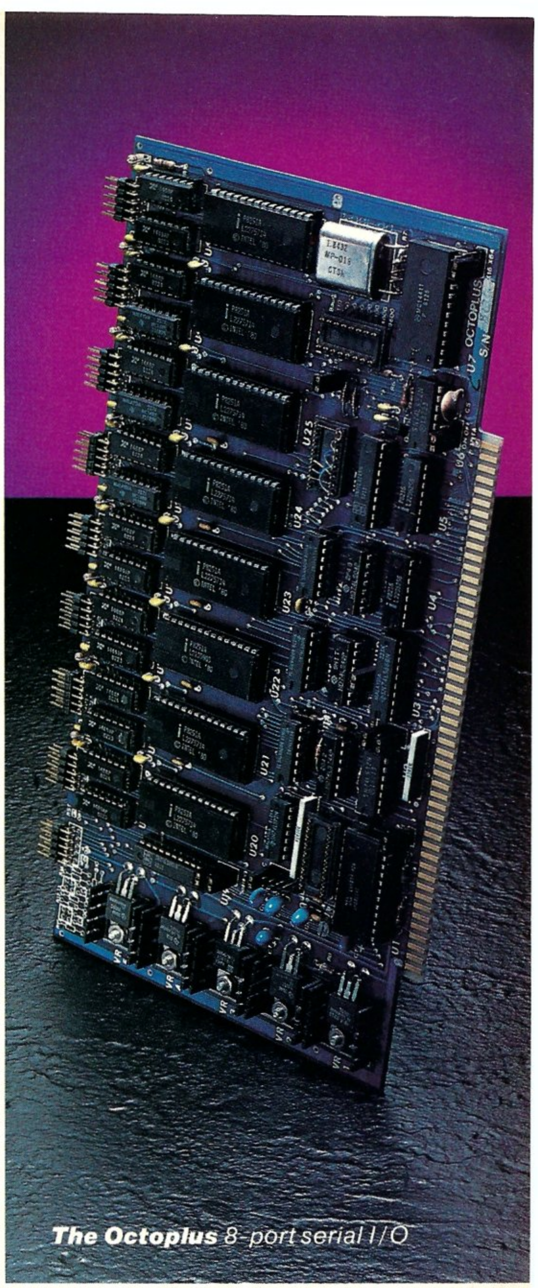
The Enhancer *An intelligent, 9-track IBM compatible, ½" tape and high speed printer and reel-to-reel*



SMC-100 SMD controller



The Enhancer ½" tape controller



The Octoplus 8-port serial I/O

S-100 SOLUTIONS IN CONTROL.

tape controller. The Enhancer interfaces to both Pertec Micro Formatter and CDC 92144 compatible tape drives. And the Enhancer features an 8-bit parallel, high speed printer port.

The Octoplus A versatile, low cost, quality solution for multi-user or data acquisition systems. The Octoplus features selectable serial data rates from 110 to 19.2 K Baud, selectable real-time interrupts from 3.125 ms to 100 ms, and a general purpose, 8-bit parallel input port.

For more information on how Konan's spectrum of S-100 solutions can put you in control...call or write:

KONAN T.M.

in control

1448 North 27th Avenue/Phoenix, Arizona 85009
(602) 257-1355
TWX/TELEX 9109511552

CIRCLE 242 ON READER SERVICE CARD

GrafTalk

A graphics package for producing pie, bar, and line plot graphs

by Joseph W. Long

GrafTalk is a business-oriented graphics package which can be used to produce pie, bar, and line plot graphs. It is intended for a wide range of users: Both people with relatively little computer knowledge and experienced graphics programmers can easily and profitably use GrafTalk. GrafTalk is a superb package: it is easy to install, easy to use, beautifully documented and very powerful. Using its extensive tutorial, it is possible to produce graphs within a few minutes of installing it. Exploring all of the power of GrafTalk will take some time, but it's an enjoyable process because of the very high quality of the documentation and the extreme power of the package.

It should be possible to run GrafTalk on nearly any CP/M-based system that drives a graphics device through a serial or parallel output port. Systems using special graphics boards or memory-mapped graphics may or may not work with GrafTalk. For this type of system it would be a good idea to contact Redding Group, Inc., before you buy. At this time, only seven systems (Altos—all models, Radio Shack Model II, Xerox 820 Superbrain I, North Star Advantage, IBM PC with Baby Blue board, Zilog MCZ-2 plus directions for any Z80 computer with an SIO, 8251 or 2661 chip) are supported in the sense that specific directions are included for them with GrafTalk. My system (Horizon, ADM3 + RG512 Graphics, DMP4) was quite easy to get going, but again, I would suggest contacting Redding Group before you buy to avoid any surprises.

Installation

I installed GrafTalk correctly the first time I attempted it, using the "INSTALL" program which, together with the documentation provided, makes the installation quite straightforward. The installation documentation is adequate, but does not have the high polish of the rest of the package. A person with little or no microcomputer experience might have some problems trying to follow the INSTALL documentation.

The installation program is menu-driven and allows the installation of a console device, up to three graphics devices and, in addition, changing of various default settings (more on these default settings later). GrafTalk supports a number of video terminals and a variety of graphics hardware, including video graphics devices, digital plotters, and printers with graphics capabilities (see Table 1).

The documentation includes an example installation dialog, but it is placed at the end of the installation manual, and I must confess that I didn't find it until I had successfully completed installing my terminal and video graphics. My first attempt at installation led to the sign on prompt and a "FOR HELP TYPE HELP" message, but no keyboard response, so I redid the installation with the same result. The problem is that GrafTalk needs about 10 seconds after the prompt appears to bring itself all the way up. After my third try I went for coffee, and when I returned everything was working. Ten seconds seems like forever when you are bringing up new software! This delay is not mentioned in the documentation.

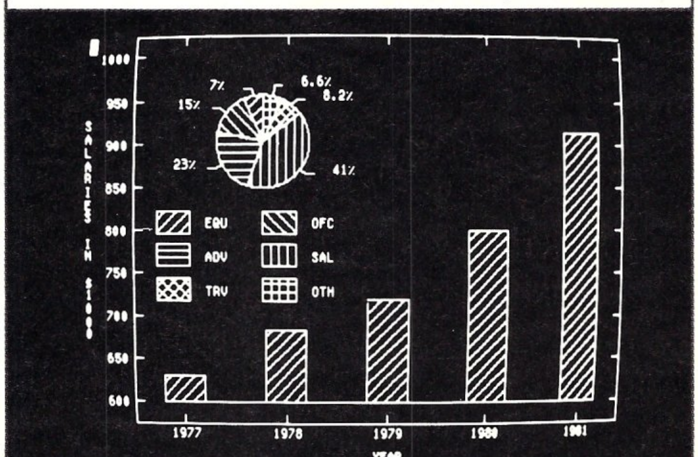
Joseph W. Long, Dept. of Chemistry, Broome Community College, Binghamton, NY 13902

Structure

GrafTalk operates in two modes. There is a command processor and a text editor. The command processor allows you to run graphics programs created using the text editor. A variety of utility type commands is also available in the command mode for clearing video graphics, clearing the terminal screen, switching plotters, etc. These are all mnemonic two-letter commands that are easy to remember.

There is an extensive HELP file which I found myself gradually relying on more and more—it is often quicker than looking in the manual if you don't need a detailed explanation. A powerful feature included in the command mode is that almost all of the graphics commands used to create graphs are available in the command mode. This means that plots can be made interactively, so it is very easy to quickly explore the graphics commands and what they do. This capability, along with the outstanding tutorials in the manual, make learning GrafTalk a pleasure.

The text editor included with GrafTalk is an excellent piece of software. It is video oriented and has a command/status line across the bottom of the screen. I found it very easy to adapt to (I use WordMaster for most text editing). The editor is a subset of a larger editor called "SCATE" marketed by Redding Group. I intend to look into SCATE further because I am highly pleased with the portion included with GrafTalk. The editor allows two workspaces, so that



A video plot of one of the sample programs from the tutorial.

Plotter

DMP-3, 4, 6, 7, 29
TEK 4662
HP 7220, 7225, 7221, 7470
CALCOMP 81
GOERZ 281
STROBE 100
WATANABE 4675
IBM XY/750

Graphics Video
(Tektronix 4010 Compatible)

* ADDS VIEWPOINT
* ADM 3A, 3A+, 5
ID-100
MICROANGELO MA512
TAB 132/15-G w/TB600 board
* TELEVIDEO 910, 910+, 912, 920, 925, 950
* VT 100
* VT 132

Screen Dump Devices
(From Screen or Disk)

Anadex 9501
NEC 3510, 3520, 5510, 5520, 7510, 7520
Paper Tiger IDS 460
Epson, MX80, MX100
Diablo 1640, 1650, 630
IDS PRISM

Graphics Printers

Diablo 1640, 1650, 630
NEC

TABLE I
GRAPHICS DEVICES SUPPORTED

*These screens require retrofits from Digital Engineering, Datatype and/or Selanar.

two graphics programs may be worked on at the same time, which I often found to be convenient in developing new programs. Working with the editor is very easy because the program currently in the workspace can be executed using a simple escape command. After the program is executed you are automatically returned to the edit mode, ready to continue development. If you go to the command mode and then reenter the editor, the program in the work space is still there, ready to go.

You can use SCATE as a text editor for nongraphics work: I was able to write a Basic program, leave GrafTalk, and load and run the program using Microsoft Basic. You cannot use the editor without loading GrafTalk first, but I think of this capability as a nice little "free bonus." Also, GrafTalk can read files created with other text editors, which could be very helpful if you have large data files, since they may not have to be retyped.

Using GrafTalk

Producing graphs with GrafTalk is child's play. All plotting parameters have default values, so that a graph can be produced with very few commands. Many of the default settings will of course not produce the graph that you want, but you do have a firm place to start in learning to get exactly what you are after. I think that this feature is one which makes GrafTalk such a beautiful package.

To produce a graph, GrafTalk needs some data, which can be entered in the immediate (i.e., command) mode (see Listing 1) or put in a disk file via the text editor. The command "PIE R2" will produce a pie graph of the data in row 2 of the data (Figure 1). The commands "BAR R2 R1" and "PLOT C2 C3 VS C1" produce the plots in Figures 2 and 3.

From this simple start, GrafTalk will do as much for you as you are willing to learn. Everything can be changed, in nearly any way you might like. Titles, axis names, tick mark names, and any other alphanumeric labeling in any position are easy to add. The size of the plot may be changed and positioned anywhere on the plotting surface. The automatic scaling of data can be defeated and the number and position

Intended for a wide range of users, GrafTalk is a superb package: easy to install, easy to use, beautifully documented, and very powerful.

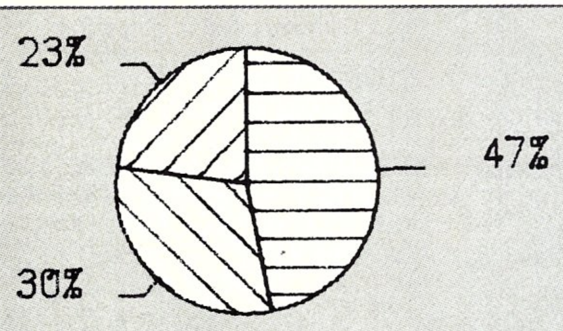


FIGURE 1

Note: The plots in Figures 1-3 were done using GrafTalk's "REGION" command, which controls the size and position of a graph on the plotting surface, and the "TEXT" command for lettering.

of tick marks can be specified. The sequence of shading patterns and the patterns themselves can be varied. Even the size and aspect ratios of alphanumeric text can be changed. Colors are supported for devices with that capability, and color sequences may be varied or defeated. The list is endless.

As I mentioned above, this is a very powerful package indeed. All of the above graphing parameters and many more are available and easy to use because of the tutorial examples, most of which are illustrated, and the reference documentation, where each command is thoroughly explained. One could really learn to use the package by ordering the manual only and carefully going through the tutorial and reference sections.

An especially useful feature of GrafTalk is that a plot may be rapidly developed using a video graphics device; then, when everything is just the way you want it, hard copy may be run on a plotter or graphics printer. Having available a fast video graphics display is of great help in quickly developing a plot. I would hate to have to do all of my plotting development using just the digital plotter, although most GrafTalk users seem to do it that way.

If you do have only a digital plotter, there is the one out that GrafTalk provides. You can set the plotting area to be quite small during the development process, and when the graph is correct you can reset the plot size. Because of the area effect, cutting the plot size to one third speeds up plotting by a factor of almost 10.

GrafTalk provides several ways to control the size and position of a graph on the plotting surface. The "REGION" command causes a cursor to appear on the graphics device, which can be moved to the position you want to be the lower left limit. You then move the cursor to the upper right limit, and all subsequent plotting will be done in that area. There is even a hi-low speed feature for this movement. A similar mode, "JOYSTICK", allows you to position titles, legends, and other text where you want, if the default locations are not right.

GrafTalk has other even more powerful ways of setting plot sizes: there are three graphics coordinate systems: NORMALIZED, WORLD, and USER. There are also USER and WORLD WINDOWS which I won't attempt to explain further here, as they are somewhat complicated. The manual has an explanation in this section entitled "Notes to the hopelessly confused," which is encouraging since this material is covered in a section labeled "Advanced Plotting Methods," and many users of GrafTalk will probably never have need of these commands. Still, they are there for those who want them.

Data files may contain up to three ignored characters. This means that dollar signs, commas, and % signs (these are the default characters and can be changed) will be ignored: $23,000 and 14.05% are read as 23000 and 14.05 by GrafTalk (see Listing 2). It should be possible to use many existing data files without modifying them to be readable by GrafTalk. GrafTalk has a "MARKER" command that allows data files to be broken up into sections; a program can access data in any desired section. Data files may contain up to 20 rows (35 rows in version 2.0, released in April 1983) and an "unlimited" number of columns. With more than 20 (35) rows, however, data can be plotted only as columns.

Another highly useful feature is a marker that allows you to mark a beginning and endpoint in a program you are developing—only the program state-

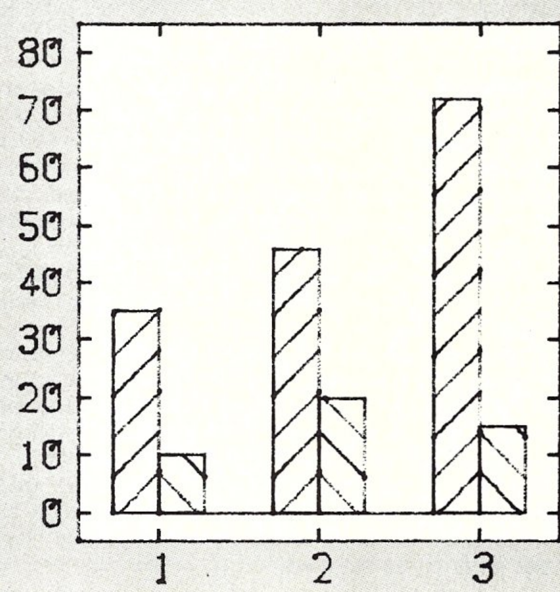


FIGURE 2

GrafTalk's text editor allows two workspaces, so that two graphics programs may be worked on at the same time. This is a convenient feature for developing new programs.

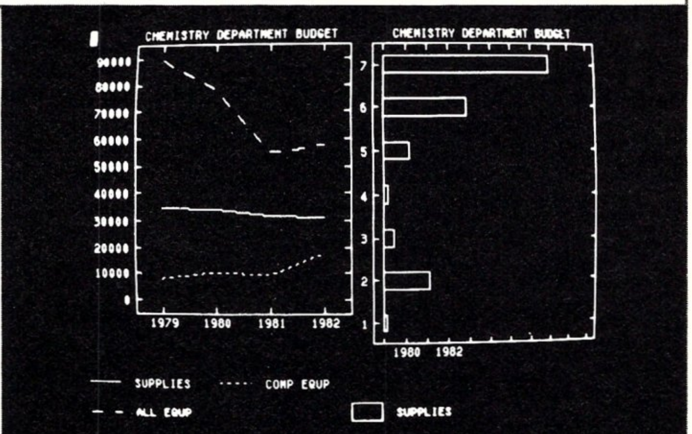
ments positioned inside the limits chosen will be executed. This can really speed up development.

There is one aspect of GrafTalk that takes a little getting used to. A lot of plotting information (titles, for example) is stored in disk files. These files are not cleared when a new plot is run. This means that if you run a plot from a command file, then run another command file without the TITLE instruction, the title from the first file will be drawn on the second plot. The "INITIALIZE" command (which clears all such files) at the beginning of a program will prevent this, as would a "NO TITLE" command in the second plot above. I found this feature rather puzzling and annoying until I figured out what was happening, but after using GrafTalk a lot, I have found that it's very handy.

Scientific plots

Since it is a business-oriented graphics package, I was initially skeptical about whether I would have any real use for GrafTalk beyond playing with it. A little reflection revealed two ways in which the package would be useful within my department (Chemistry). The first use is just for what it was designed. In preparation of our budgets and in other dealings with the college administration, graphical presentation of data for justifying equipment requests and other similar uses will have a great impact and therefore be very helpful to the department. This will be especially advantageous as long as our department alone on the campus has this graphics capability!

My personal interest in graphics is in scientific plotting. Much to my surprise, I have found that it is not difficult to do such work. This is because GrafTalk recognizes data files generated by other software. I was able to write a simple program in Micro-



A video plot of some of the data from the data file given in Listing 2 (the axes are mislabeled in the left-hand plot).

soft Basic that writes data to a disk file in row column fashion (see Listing 3). A Basic program may take data and process it in ways of interest to physical scientists and engineers: take logs, do curve fits, and so on. The manipulated data may then be written to a disk file for access by GrafTalk. The main disadvantage to this approach is that it is somewhat time-consuming to switch from Basic to GrafTalk and back, but it does work well (see Figure 4).

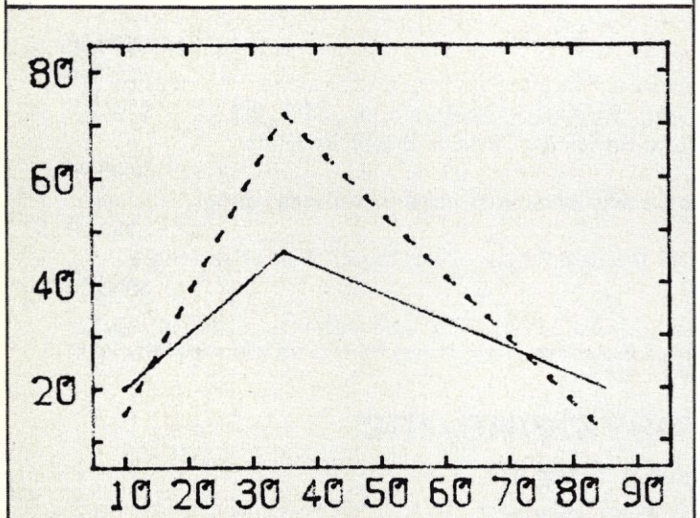


FIGURE 3

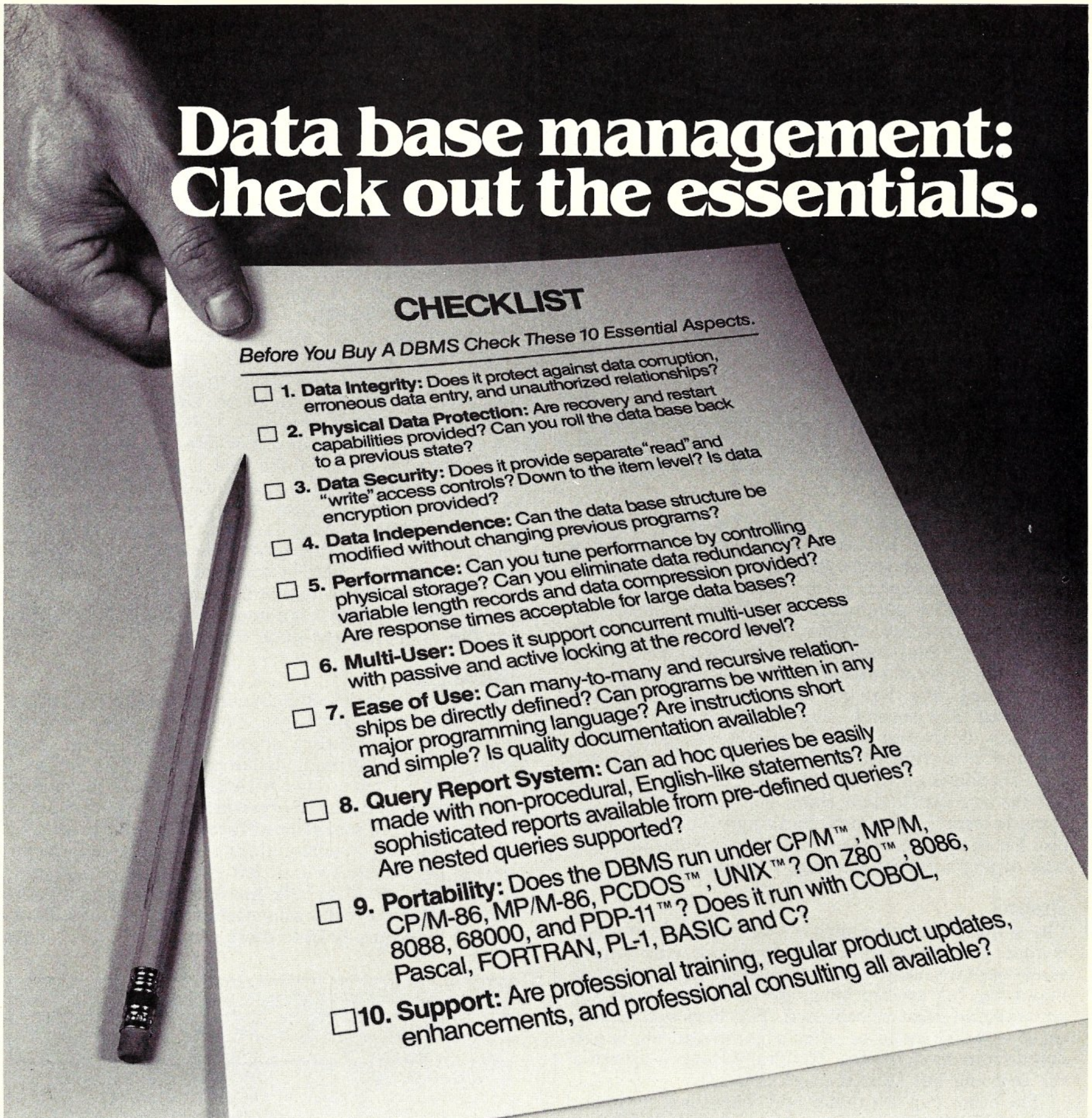
Comparisons

There is another graphics package called Grafpak (Mycroft Labs, Tallahassee, FL). Despite the similar names, GrafTalk and Grafpak are quite different. GrafTalk is business oriented and is intended for use by applications people who know little about graphics except that they want a plot. Grafpak is a set of Fortran plotting subroutines that are most easily used in scientific plotting and require a good knowledge of Fortran.

The newest versions of Grafpak include programs that generate pie, bar, and line plots automatically, but they are not really in the same league with GrafTalk for business plotting. On the other hand, for scientific plotting, Grafpak is superior, and it also includes a program (EZPLOT.COM) that does interactive scientific plotting. The best solution would be to own both of these packages because of the way they complement one another.

Redding Group also markets "GrafLib," a set of Fortran subroutines that allows you to develop your own graphics software. Although it doesn't say so anywhere, it seems that GrafTalk was written using the routines in the GrafLib package. GrafLib is very different from GrafTalk; it is designed for experienced Fortran programmers only. For example, you cannot use it to quickly patch GrafTalk's capability into Basic programs. I plan to spend some time with GrafLib and review it for *Microsystems.*

Scientific plotting is not difficult to do because GrafTalk recognizes data files generated by other software. Thus data processed for scientific use may be written to a disk file for access by GrafTalk.



There is only one DBMS that satisfies all these criteria. That's MDBS III, the only authentic DBMS running on a wide variety of microcomputers and the PDP-11.

ISE™ PROFESSIONALS KNOW THE DIFFERENCE

In U.S. & Canada contact:
ISE-USA
85 W. Algonquin Rd.
Arlington Heights, IL 60005
(800) 323-3629/(312) 981-9200

Elsewhere contact:
ISE
PO Box 600
Lafayette, IN 47902
(317) 463-4561/2581

If you can't answer "YES" to these questions, send for our comprehensive booklet, "How to Evaluate and Select a Data Base Management System." Sold previously for $5.00, but it's yours *FREE* if you return this coupon now. But hurry, supplies are limited.

MAIL TO: Your local ISE affiliate.

Name________________ Title________________

Company________________

Address________________

City________________ State________ Zip________

Phone________________
(area code) (ext.)

Z80 is a registered trademark of Zilog; PDP-11 of Digital Equipment Corp; CP/M of Digital Research; UNIX of Western Electric; PCDOS of IBM.

CIRCLE 44 ON READER SERVICE CARD

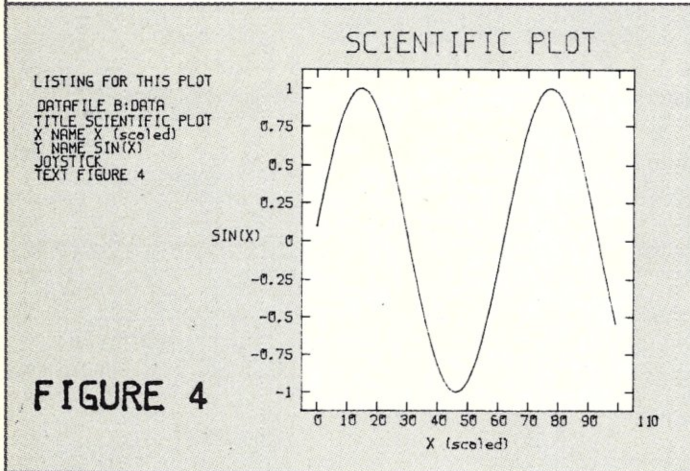


FIGURE 4

Documentation

As I have mentioned above, GrafTalk's documentation is excellent. It comes in a three-ring binder, is well organized and very easy to work with. There is no index, but there is a very complete table of contents that makes commands easy to find. The manual is divided into three sections: the first section includes the user's manual, which contains a brief overview of GrafTalk, and the tutorial, which contains many example programs that are very carefully and clearly explained. The second—the reference section of the manual—deals with each GrafTalk command in detail and cross-referenced to related commands. The final section is on the use of SCATE. The manual is written in clear English; there is no jargon and everything is clearly explained. Nothing is left as an exercise for the reader. GrafTalk's documentation is the best of any software package I have seen.

Bugs

I have found only two bugs in GrafTalk, one of which is that the line continuation character (the ampersand) does not work when it is encountered in a command file. My system hangs up when it finds an ampersand and must be rebooted. This does not occur in the command mode or when a command file is executed from a workspace. Redding Group is aware of the bug and has fixed it in version 2.0.

The other bug is a minor one: There is a command "NUMBER" which you can use to print numbers anywhere on the plotting surface. NUMBER has two arguments: the number to be printed and the number of significant figures (this one is optional). I found that until a number of significant figures *is* specified the first time, the numbers plotted are wrong. After that, NUMBER works correctly without the optional argument.

Support

I have had only a few occasions to contact Redding Group with problems, but each time they were immediately able to give me the information I needed. One problem was in getting a DMP-4 plotter working with GrafTalk. It seemed that I needed some handshaking between the plotter and my North Star, and I wasn't sure exactly how to get it working. Even though the North Star Horizon is not officially supported as of yet, they were able to tell me exactly what the problem was and how to correct it (they were right, too). I then called Lifeboat Associates, which also distributes GrafTalk. They gave me the same advice, which was a pleasant surprise. Those folks really do know about the software they sell!

As this review was about to be submitted, I ran into an interesting snag with GrafTalk. I have a North Star 18MB hard disk CP/M that just came back from being repaired. For some reason, GrafTalk will not work under North Star hard disk CP/M. (It does work with North Star non-hard-disk CP/M.) Lifeboat Associates cannot help because they do not support the North Star hard disk. North Star can not help because they have never heard of GrafTalk. I am hoping that there is someone out there who has solved this problem and will contact me. The CP/M software bus concept is a good one, but it *does* lead to problems of this sort upon occasion.

Summary

GrafTalk is an excellent business graphics package that does everything right. It is an example of what all computer software ought to be: powerful, very easy to use, and well documented. I have enjoyed everything I have done with GrafTalk. I never had the feeling that I was overwhelmed by how much I had to learn to make the software do useful work, or that I was lost in a mountain of mediocre documentation never able to find that elusive bit of information. This sets GrafTalk apart from other software that is also powerful and useful, but too difficult to use to be of any help to most people.

GrafTalk Overview	
Purpose:	Business graphics package
Manufacturer:	Redding Group Inc. 609 Main St. Ridgefield, CT 06877 (203) 431-4661
Distributor:	WESTICO 25 Van Zant St. Norwalk, CT 06855 (203) 853-6880
System requirements:	48K CP/M 180K disk drives graphics device
Version 2.0: Version 2.0 update:	\$450 \$100

GrafTalk is an excellent business graphics package that does everything right. It is an example of what all computer software ought to be.

GrafTalk continued . . .

```
LISTING 1

SMALL DATA TABLE ENTERED
WHILE IN COMMAND MODE
.
.@DATA
.10 20 15
.35 46 72
.85 20 11
.@END
```

```
LISTING 2

SOMEWHAT LARGER DATA FILE ENTERED USING SCATE
Note commas, dollar signs in first row of data.

A>TYPE DEPT.BUD

THESE FIGURES REPRESENT THE
CHEMISTRY DEPARTMENT'S BUDGET
FOR THE LAST FOUR YEARS, BROKEN DOWN BY
CATAGORIES

                      FY1981   FY1982   FY1979   FY1978

TEMP SERVICE          $2,000   $2,100   $1,900   $1,877
SUPPLIES               35000    34000    32000    31500
EXPENSE                 4500     5400     6500     7000
COMPUTER SUPPLIES       3100     2400     2000     2900
COMPUTER EQUIPMENT      8200    10500     9500    17600
EQUIPMENT              89000    78000    55000    57600

TOTAL                 151800   132400   106900   118477

A>
```

```
LISTING 3

BASIC PROGRAM TO GENERATE SIN CURVE DATA
AND PLACE IT IN A DISK FILE READABLE BY GRAFTALK

1000 REM           PROGRAM TO CREATE A DATA FILE
1010 REM           ACCESSABLE FROM GRAFTALK
1020 REM           FOR PLOTTING PURPOSES
1030 REM
```

GrafTalk continued . . .

```
1040 REM
1080 REM-------------------LOAD ARRAY-------------------
1090 DIM A(100,2)
1100 FOR J=0 TO 100
1110   A(J,1)=J : A(J,2)=SIN(.1*J)
1120 NEXT J
1130 REM-------------------END LOAD-------------------
1140 REM
1150 REM
1160 REM-------------------PRINT ARRAY-------------------
1170 FOR J=1 TO 100
1180   PRINT A(J,1) , A(J,2)
1190 NEXT J
1200 PRINT
1210 REM-------------------END PRINT-------------------
1220 REM
1230 REM
1240 REM-------------------SAVE ARRAY IN DISK FILE-------------------
1250 OPEN "O", #1 , "DATA"
1260 FOR J=1 TO 100
1270   PRINT #1, A(J,1) , A(J,2) , : REM TRAILING COMMA REQUIRED TO PROPERLY
1280                                 REM FORMAT DISK FOR GRAFTALK
1290   PRINT #1 , CHR$(141) :        REM INSERTS A <CR> INTO DISK FILE FOR
1300                                 REM GRAFTALK
1310 NEXT J
1320 REM-------------------END SAVE-------------------
1330 END
```

HARDWARE REVIEW

J.E.S. Graphics Super Compuprism Graphics Board

by Thomas Ceska

The Super Compuprism from J.E.S. Graphics is an S-100 graphics board, although its S-100 signals do not conform to the recently adopted IEEE-696 specifications. It has a 288 x 192 x 4 matrix display, with each pixel in the 288 x 192 plane capable of being assigned any of 16 grey levels or 16 colors. The board contains 32K of dynamic memory. The cost of the bare board is $50; the kit version is $350; and the assembled and tested version is $395. I purchased the bare board and was able to populate it for about $150, making the total cost about $200. The major expense was the 16 150 ns 16K x 1 memory chips ($40), which have since dropped by half in price.

I will give my impressions of the documentation first, before I go into more detail about my experiences in getting the board working. The kindest thing I can say about the documentation is that it is adequate. The instructions are 19 pages of photocopies stapled together. All of the documentation had been typed at one time, but subsequent revisions are handwritten, with occasional scribbles on at least half of the 19 pages. This documentation describes the assembly of the minimal Compuprism. The Super Compuprism is an enhanced Compuprism with additional integrated circuits piggybacked on existing chips. The documentation for enhancing the Compuprism to Super Compuprism consists of five continuous sheets of dot matrix output, all printed in capitals. The text was not paginated. The documentation was not one to inspire confidence, but, as I said, it was adequate. Undeterred, I ordered the parts.

The instructions call for assembling the board to its minimal Compuprism configuration first, followed by testing before modifying the board. The board has plated-through holes, and parts placement on the board is marked. For the most part, parts placement is unambiguous. A few of the capacitor locations are unclear, and the orientation of the resistor packages is not indicated, but with a little thought the correct locations and orientations can be deduced. I socketed all of the ICs and I did not have any difficulty in assembling the board up to this point. When I plugged the board into my chassis and applied power I could not get the display to stabilize, nor could I read or write to the board. After a few calls to J.E.S. Graphics, I deduced that the board was using $\phi 1$ (S-100-25) as the system clock. After modifying the board by cutting the trace to pin 25 and adding a jumper from the cut trace to $\phi 2$ (S-100-24), the board worked fine. I should mention that help over the telephone from J.E.S. Graphics was courteous and forthcoming. Although the source of my problem was not pointed out by J.E.S., I was able to deduce what the difficulty was after the helpful discussions.

At this stage the Compuprism board has 16K of memory, and the display resolution is 144 horizontal by 192 vertical elements. The pixels are short horizontal bars rather than dots. The next step in assembly was to expand the Compuprism board from 16K to 32K of memory, add bank select control, and add the 16 grey-level options (switchable from 16 colors). This involves piggybacking a handful of chips to existing ones and adding two handfuls of jumpers. On the 47 original ICs, 16 more were added piggyback. Needless to say, the board began to look like a rat's nest, even though I tried to keep the jumpers neat and short (see Figure 1).

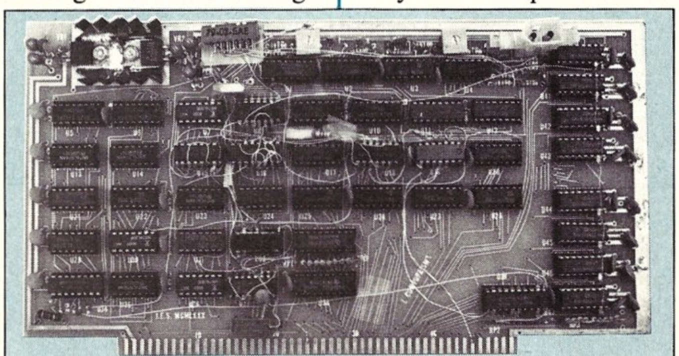


Figure 1. The Super Compuprism, with the bank select and 16 grey-level options installed: 16 integrated circuits are piggybacked on existing chips.

But how does it work? Have a look at Figure 2 for an example. The board works quite well, although there are problems with the stability of the display. More on this later. The programs for manipulating the data are relatively straightforward and are not shown here. The routines to write and read the memory on the board are shown in Listing 1. The pixels are 1 nibble each (4 bits), thus one byte contains 2 pixels. Writing pixels into the 288 x 192 array is simple. The documentation provides a hex dump of a short routine to test the board. Although a few routines illustrating pixel manipulation would have been useful, I found that writing the subroutines I used was not difficult.

Disadvantages

The memory store and fetch times on this board are quite slow, and the documentation claims that memory operates at a quarter the speed of a normal memory board. The Compuprism forces a wait state until the board enters horizontal retrace. Thus the display is stable while its memory is being accessed. The manual also mentions that the Compuprism may not

Thomas Ceska, 934 Stewart Ave., Apt. 12, Ithaca, NY 14850

be able to perform op-code fetch cycles when used at 4 MHz. Since the memory is slow, it seems unwise to execute a program in its memory. I use the board as a large temporary data matrix for some of my programs.

I mentioned before that I found the display oscillated up and down on occasion. The problem is intermittent, and I traced it to a design flaw in the vertical synchronization circuitry. Horizontal and vertical timing are determined by 74123 one-shots. For horizontal retrace, the accuracy of the one-shot timing is quite good; however, for vertical synchronization, a relatively long time is required before retrace. A small error in the timing of each retrace will result in the vertical displacement of the display lines: thus an oscillation of the display is observed. A digital counter for vertical synchronization may be required for a stable display.

Figure 2. Example of the graphics capability with the Super Compuprism. The data for this Mona Lisa was taken from the book, Digital Image Processing, *by R.C. Gonzalez and P. Wintz, Addison-Wesley Publishing Co., New York, 1977. The image has been enlarged by a factor of two. The data set comprised a 64 x 64 matrix, but the display is 128 x 128 in size. The resolution of the board is actually better in both dimensions than the photograph indicates. A total of 15 grey levels are used in the image.*

Compuprism does not conform to the proposed IEEE-696 specifications for the S-100 bus. I found that I had to reroute the pin 25 trace to pin 24. The board is not capable of 24-bit addressing or 16-bit data transfer, but it works quite well with older S-100 systems.

The question of reliability and ease of maintenance is of paramount importance to most computer users. The board which I populated operated as advertised and I was quite pleased with it, even though it does have its shortcomings. The board worked well for about eight months, but then one day I was unable to read or write to its memory. Pulling out my logic probe, I probed it to the point where no signal was coming out to the monitor. I traced the problem to a burned-out IC that was underneath a piggybacked chip. After desoldering, replacing the chip, and resoldering all of the jumpers, the board was working again.

However, recently I find I am having difficulty with memory read and write operations, and I suspect that the same IC has blown again. Thus I fixed the problem, but not the cause of the problem. Since a third of the ICs are buried under other ICs and jumpers, the board is extremely difficult to debug and fix. Because of the large number of ICs on this board, the amount of power drawn is quite considerable. This may be an underlying factor in the reliability of the board.

Advantages

The Super Compuprism has a resolution of 288 horizontal by 192 vertical pixels. This is relatively high for microcomputer-based graphics boards. Sixteen different grey levels or colors (switch selectable) can be produced by the board, and each pixel can be independently set. (I haven't used the board in its color mode, so I don't know what the quality of the color display is like.) The graphics display is easy to manipulate, and the subroutines that write to the board are straightforward.

The board is low priced when compared to boards with greater resolution and pixel depth. For example, Scion's MicroAngelo with a 512 by 480 by 1 resolution costs $1100.

Conclusion

Advantages:

1. Relatively high resolution
2. 16 independent grey levels or colors
3. Simple to program
4. Low cost

Disadvantages:

1. Slow memory
2. Oscillating display
3. Non-IEEE-696/S-100
4. Reliability?

The minimal Compuprism graphics board is probably reliable and definitely easily maintainable. However, with a resolution of 144 by 192, I do not think it is a very useful board. The Super Compuprism is potentially very useful and useable with its 288 by 192 array. I think if J.E.S. redesigned the board to remove the design flaws I mentioned, made room on the board for all ICs, and conformed to the S-100 IEEE-696 specifications, they would have an excellent product. However, in its present configuration, with the reliability and maintenance problems I encountered, I cannot recommend the board to the general user. Hardware hackers may be able to resolve difficulties more easily and should consider this board if they need graphics capability.

The board has a resolution of 288 by 192 pixels; each pixel can be independently set to one of sixteen grey levels or colors.

$49.99

CP/M™ will never be the same again!

CLIP™, our Command Line Interpretive Program, makes CP/M more efficient than ever before. It replaces CP/M's command processor with a user friendly "shell" containing over 50 commands. CLIP runs as a normal CP/M program using only 5K of memory when running programs and 25K when interpreting commands.

Editing Keys

CLIP adds editing keys to all standard CP/M programs—just like a word processor! Single keystrokes allow you to move forward and backward to insert or delete text. The command interpreter remembers your last 10 commands, which can be retrieved, edited, and reentered effortlessly.

Make Your Own Commands

Sequences of commands may be placed in a file to create a "macro" command.

Macro commands can:
- Take arguments and switches
- Execute programs
- Ask questions
- Conditionally execute commands
- Loop
- READ and WRITE files
- Call upon other macros

Built-In Text Editor

CLIP contains a simple editor for creating macros, programs, or data files. English-like commands simplify any editing task.

Automatic File Searching

Keep commonly accessed files on one disk and user number. CLIP searches for files when not found on your current disk and user number—saves disk and directory space.

Built-In Calculator

The functions +, −, ×, ÷, Modulus, AND, OR, XOR, and NOT are available with I/O memory registers. Results may be passed to programs and macro commands.

A Unix™-like Environment with I/O Redirection and Pipes

CLIP can direct screen output from a program or macro to a file for viewing or printing, or cause program input to be taken from a file instead of the keyboard. Pipes force the output from one program into the input of another automatically.

Sophisticated command language

- Conditional command execution
- Command file tracing
- Multiple commands per line
- File I/O from the command line
- Extensive string manipulation
- Descriptive error messages

CLIP is equivalent to a high level programming language, yet simple enough for non-programmers, too!

Software Tools Package Just $25.00 with CLIP!

We have many software tools that compliment and enhance CLIP.

BC	- Binary file comparison
BFE	- Binary file editor
DMP	- Hex file dumper with ASCII
HC	- Horizontal file concatenate
LCASE	- Convert to lower case
PAD	- Pad lines with spaces
SC	- Source file comparison
SR	- Search for a pattern in a file
SRT	- Simple in memory sorter
UCASE	- Convert to upper case
UNIQ	- Extract unique entries from a file
WC	- Count words, lines and bytes
XC	- Extract columns from a file
XW	- Extract words from a file

Data Encryption Program $25.00 with CLIP

A twelve character password encrypts a file such that no one may read it but you.

CLIP; a Help compiler; over 60K of help information; example macros; user's manual and binder—all for $49.95.

Software Tools $25.00.

File Encryptor $25.00.

User's manual $25.00 (credit upon purchase of CLIP).

Visa, Mastercard, check or money order accepted.

CLIP, the tools and the file encryption program have been designed for the Z80™, running CP/M 2.2. Distribution is on 8 inch SS/SD or Osborne 1, Kaypro, or Apple. Call for availability of other formats.

Dealer inquiries invited.

Thoughtware, Inc.

800-821-6010/602-327-4305

P.O. Box 41436, Tucson, AZ 85717

CIRCLE 58 ON READER SERVICE CARD

Listing 1. *Primitive assembly language routines to read and write to the graphics board, both in a byte mode and in a pixel mode.*

```
                   ;          SOURCE FILE FOR PROM CHIP
                   ;          RAM SOURCE ADDRESS = 4000H
                   ;          REM FIRST FILL 4000 4FFF FF
                   ;
47C2                          ORG       47C2H
                   ;
                   ;          PUTS BYTE OF DATA IN C INTO ADDRESS HL
                   ;          OF GRAPHIC EXTENDED MEMORY. DESTROYS 'A' REG.
47C2  3EFF         EXMPUT:    MVI       A,0FFH            ;SELECT GRAPHIC MEMORY
47C4  D37F                    OUT       7FH               ;VIA THIS PORT.
47C6  71                      MOV       M,C               ;PUT BYTE THERE
47C7  97                      SUB       A                 ;BIT 7 = 0
47C8  D37F                    OUT       7FH               ;RETURN TO NORMAL MEM
47CA  C9                      RET
                   ;
                   ;          GETS BYTE OF DATA FROM GRAPHIC BOARD EXTENDED MEMORY
                   ;          HL = ADDRESS, DATA "ETURNED IN C, 'A' REG DESTROYED.
47CB  3EFF         EXMGET:    MVI       A,0FFH            ;BIT 7 = 1
47CD  D37F                    OUT       7FH               ;SELECT GRAPHICS BOARD
47CF  4E                      MOV       C,M               ;GET BYTE
47D0  97                      SUB       A                 ;BIT 7=0
47D1  D37F                    OUT       7FH               ;RETURN TO NORMAL MEMORY
47D3  C9                      RET
                   ;
                   ;          CLEARS GRAPHICS BOARD SCREEN RAM: 27,648 BYTES
                   ;          ENTRY @ SETGB USED FOR BACKGROUND IF BIT 7=1 (E.G.,88H)
47D4  3EFF         CLEARGB:   MVI       A,0FFH            ;FF=BLANK SCREEN
47D6  E5           SETGB:     PUSH      H                 ;SAVE REGISTERS
47D7  D5                      PUSH      D
47D8  C5                      PUSH      B
47D9  210000                  LXI       H,0000            ;START CLEARING HERE
47DC  110100                  LXI       D,0001            ;NEXT LOCATION
47DF  01FF6B                  LXI       B,27647           ;ONE LESS TO CLEAR
47E2  D37F                    OUT       7FH               ;SELECT GRAPHIC MEMORY
47E4  77                      MOV       M,A               ;CLEAR FIRST LOCATION
47E5  EDB0                    DB        0EDH,0B0H         ;THEN CLEAR THE REST (WITH LDIR)
47E7  97                      SUB       A                 ;BIT 7 = 0
47E8  D37F                    OUT       7FH               ;RETURN TO NORMAL MEMORY
47EA  C1                      POP       B
47EB  D1                      POP       D
47EC  E1                      POP       H                 ;RESTORE GEGISTERS
47ED  C9                      RET
                   ;
                   ;          DESELECTS SORCERER 48K, SELECTS GRAPHICS BOARD MEMORY
                   ;          PUTS DESIRED DOT ONTO GRAPHICS DISPLAY
                   ;          HL = ADDRESS OF DOT, C = INTENSITY
                   ;          NOTE: CORRECT NIBBLES IN C ARE PREDETERMINED.
47EE  3EFF         GRDOT:     MVI       A,0FFH            ;BIT 7 = 1
47F0  D37F                    OUT       7FH               ;SELECT GRAPHIC MEMORY
47F2  7E                      MOV       A,M               ;GET 2 PIXELS
47F3  A1                      ANA       C                 ;SET DESIRED PIXEL
47F4  4F                      MOV       C,A               ;SAVE THIS INTENSITY
47F5  77                      MOV       M,A               ;PUT BYTE IN MEMORY
47F6  97                      SUB       A                 ;BIT 7 = 0
47F7  D37F                    OUT       7FH               ;RETURN TO NORMAL MEMORY
47F9  C9                      RET
                   ;
47FA  C300D0                  JMP       0D000H            ;DISK BOOT ADDRESS
47FD  C3C8E0                  JMP       0E0C8H            ;COLD START ADDRESS
                   ;
4800                          END

                                            1
```

HUNT ELECTRONICS
4808 N. NEB AVE.
TAMPA, FL. 33603
813-236-5948

65K CMOS STATIC RAM-$260
MODEL T kit

- PHANTOM
- 24 BIT EXT. ADD.
- 150 NSEC RAM
- IEEE 696 / S-100
- WAIT STATE
- VERY LOW POWER

BARE BOARD $ 64.50 w/ PARTS exc. RAM $105.00
BOARD w/ PARTS ass & tested $135.00
150 nsec CMOS RAM ea. $ 5.00
gold RAM sockets - prime parts - full buffering - solder mask character screen - full manual w/assy instructions

RS 232 EPROM PRGMR. $210
MODEL 27XX

- STAND-ALONE OPERATION 110V POWER
- USE WITH ANY ASCII SERIAL DATA SOURCE
- CURRENT SOURCE, EBCDIC, & BI SYNC optional

2716/32 EPROM MEM. $160
MOTHERLODE ass&tst

- SOFT CONTROL OF ON-BOARD PHANTOM GEN.
- 24 BIT EXTENDED ADD. • FULL BUFFERING

BARE BOARD $55 KIT OF ALL PARTS exc. MEM $115

DUAL DISK PWR. CNTLR. $35

- INDIVIDUAL DISK ON / OFF
- FULL ISOLATION
- KIT ONL
- TTL INPUT

ADD $3.00 handling M.O. UPS COD

CIRCLE 194 ON READER SERVICE CARD

LYNC

COMMUNICATIONS SOFTWARE

WHY BUY 3 PROGRAMS?
LYNC HAS ALL 3 IN ONE!

1) FILE TRANSFER 2) REMOTE OPERATION
3) COMMUNICATE WITH TIME-SHARE

Easy set up on most systems.
Menu driven install program.
Command Mode of operation
Help Screen for Lync & Term

Z80 based CP/M $155
PC DOS $155 TRSDOS $75

ISA

We're Not the Biggest, Just the Best.

International Software Alliance (805) 966-3077
1835 Mission Ridge Santa Barbara, CA 93103

** DEALER AND OEM INQUIRIES INVITED **

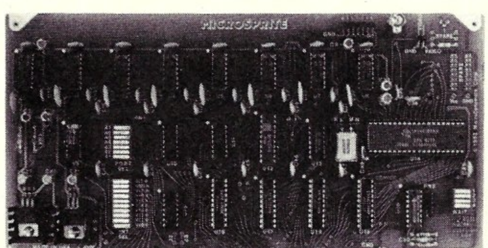
FORMATS: 8" Standard; IBM PC DOS; TRS-80 Mod II with CP/M; VectorGraphic; Cromemco; North Star; Osborne; Xerox 820 (8" or 5"); SuperBrain; NLS KayPro; TeleVideo; TRS-80 Model I or III; Zenith/Heathkit; Victor 9000; Sanyo; Altos; Fujitsu; Otrona.

CP/M is a trademark of DIGITAL RESEARCH. LYNC is a trademark of MIDNIGHT SOFTWARE.

CIRCLE 231 ON READER SERVICE CARD

S-100 COLOR GRAPHICS!

MICROSPRITE



- Display consists of backdrop and pattern planes plus 32 sprite planes; each pixel in a plane can be colored or transparent to reveal the underlying plane(s); 16 colors (including transparent) are available.
- Four display modes:
  1) TEXT—twenty-four 40-character rows in two colors. 2) MULTICOLOR—64H x 48V pixels; each pixel can be one of 16 colors. 3) GRAPHICS I—256H x 192V pixels; each 8 x 8 group of pixels can contain two different colors; all 16 colors can be on screen simultaneously. 4) GRAPHICS II—256H x 192V pixels; each horizontal group of 8 pixels can contain two different colors; all 16 colors can be on screen simultaneously.
- Composite video output connects directly to color monitor or RF modulator.
- On-board 16K RAM occupies *no* system memory space; board uses only two I/O ports.
- Uses powerful Texas Instruments TMS9918A Video Display Processor; DIP switch selection of 0-4 wait states, I/O port numbers and vertical retrace interrupt options.
- Comprehensive documentation includes user's manual with listings of demonstration software and TI's manual for the TMS9918A video display processor.

MicroDynamics Corporation
P.O. Box 17577, Memphis, TN 38117
2233 Cornwall St., Germantown, TN 38138
(901) 755-0619
DEALER AND OEM INQUIRIES INVITED

SPECIAL INTRODUCTORY PRICE!
$249.95
(assembled & tested)
We pay UPS ground shipping in the continental U.S. UPS air add $2.00. COD add $1.50. Outside U.S. add $15.00. TN residents add 6% sales tax. VISA and MASTERCARD welcome.

CIRCLE 62 ON READER SERVICE CARD

MICROSTAT® - Release 3.0
MICROSTAT® + baZic® = PERFORMANCE

The best just got better! MICROSTAT has been the leader in the statistics field for microcomputers since 1979, and the new release 3.0 outperforms and is noticeably faster than previous versions. Just a few of the features include:

GREATER ACCURACY
BCD with up to 14 digit precision;

PROGRAM ENHANCEMENTS
Missing data capabilities and many more;

FASTER EXECUTION
Calculation time greatly reduced;

DYNAMIC FILE ALLOCATION
Data can be inserted, added, or deleted;

SPECIAL PRICE:
For a limited time get MICROSTAT plus baZic complete with program disk and documentation for each for $395.00, save $50.00!

The MICROSTAT - baZic version requires: a Z80 CPU, CP/M™ and 48K of memory. Available formats: 8" SD disk or 5¼" North Star only. Check with your dealer for other formats. Also available for: Microsoft's Basic-80™, North Star DOS and IBM. For more information, call or write:

ECOSOFT INC.
P.O. Box 68602
Indianapolis, IN 46268-0602
(317) 255-6476

Master Card VISA

MICROSTAT is a registered trademark of ECOSOFT, INC.
baZic is a registered trademark of MICROMIKES, INC.
CP/M is a registered trademark of DIGITAL RESEARCH
Basic-80 is a registered trademark of MICROSOFT

CIRCLE 154 ON READER SERVICE CARD

The ZCPR2 System

An introduction to a Z80 enhanced replacement, in the public domain, for the CP/M CCP

by Richard Conn

Welcome to the world of ZCPR2—which you can best become acquainted with via a package of programs now in the public domain known as the ZCPR2 system. It is divided into two parts—ZCPR2 itself and the ZCPR2 utilities. ZCPR2 replaces the Console Command Processor (CCP) of CP/M, which itself performs only two main functions: (1) accepting command lines from either the user or a running SUBMIT command file ($$$.SUB) and (2) processing those command lines.

ZCPR2 runs in place of the CP/M 2.2 CCP. It provides functions similar to those provided by the CCP, but also offers many enhancements to those functions (see the section comparing ZCPR2 and the CP/M CCP).

ZCPR2 does not stand alone, but functions as a part of the **ZCPR2 system,** which also includes a set of utility programs. More than 30 such programs make up the ZCPR2 system, with about 15 more files containing on-line documentation that tells you how to use it. The latter are called HLP (Help) files, and one of the ZCPR2 utilities, HELP, can be used to index quickly into these HLP files and read them.

The ZCPR2 system is available through the Special Interest Group in Microcomputers (SIG/M) of the Amateur Computer Group of New Jersey (ACG-NJ), as well as a growing number of Remote CP/M (RCPM) Systems, the Compuserve Timesharing Network, the ARPA Network of the DoD (for the DoD Community only), local computer clubs, and a variety of other sources. It is free of charge (except for copying fees or connect time, or whatever your source of supply charges) and covers 10 8" IBM 3740 floppy disks. The ZCPR2 system is divided among these disks:

1 Disk	—	Source Code to ZCPR2 and some key programs
3 Disks	—	Source Code to the ZCPR2 Utilities
1 Disk	—	COM Files of the ZCPR2 Utilities
2 Disks	—	HLP Files on the ZCPR2 System
3 Disks	—	Manuals on the ZCPR2 System

The ZCPR2 system documentation consists of the following four manuals:

The Installation Manual—How to install the ZCPR2 system on your microcomputer (48 pages).

The Concepts Manual—The basic ideas of what ZCPR2 and the ZCPR2 system is and does (65 pages).

The User's Guide—How to use ZCPR2 itself, all of the ZCPR2 utilities, and all of the ZCPR2 subsystems (140 pages).

Richard Conn, 93 Wedgewood Circle, Eatontown, NJ 07724

The Rationale—Design ideas behind the ZCPR2, including why things were done the way they were, and how some things tick internally to ZCPR2 (20 pages).

Many of the ZCPR2 utilities are written in M80 assembly language (using the Microsoft M80 Assembler) and call routines from SYSLIB 2.6. SYSLIB is a library of more than 130 subroutines written in assembly language. It provides basic support for a variety of functions, including directory access, disk I/O, math, sorting, and number conversion, and can be used on any CP/M 2.2 system. SYSLIB 2.6 is in the public domain and covers another three disks that contain:

- the source code to all of the SYSLIB modules
- on-line documentation on SYSLIB (HLP files)
- SYSLIB itself as a relocatable object file
- the manuals on SYSLIB

The SYSLIB manuals are:

The User's Guide—Tutorial on SYSLIB and how to use it, with many examples (56 pages).

The User and Reference Manual—In-depth documentation on all 130 subroutines within SYSLIB (112 pages)

What does ZCPR2 buy me?

ZCPR2 was designed with one primary purpose in mind: to increase productivity—a goal attained by giving the user an environment responsive to his needs and configurable to his specific desires, while still remaining compatible with CP/M 2.2 and allowing him to run most, if not all, of his CP/M-based software with little or no modification. For instance, WordStar, dBASE II, WordMaster, ASM, BDS C, and most of the commercial software packages run under a ZCPR2 system just as they run under CP/M 2.2—they can't tell the difference.

How does ZCPR2 act more responsively to the user's needs? First, it gives him a number of commands designed to make his job easier. For instance, under CP/M, the ERA command will not erase read-only files. The BDOS traps any attempt to erase such a file and gives an error message, forcing the user to issue a command to change the protection on the file and then reissue the ERA command. In other words, instead of simply helping the user along, CP/M forces the user to bend to it in a manner selected by the person who originally designed it.

ZCPR2 is different in that respect. In the case of ERA, for instance, if the user wishes to erase a read-only file, he just uses the ERASE command instead. It will tell the user that the file is read only, and ask him if he wants to go ahead and erase it. Since the ERASE command accepts a list of files, the command could be:

```
ERASE file1,file2,file3,...
```

where each "filen" could contain wildcards. Alternatively, the ERASE command could be issued with an inspect option,

```
ERASE *.* I
```

in which case all files in the directory are displayed one at a time, and the user is asked in each case if he wants to erase it.

REN, DIR, and TYPE also have transient counterparts under ZCPR2: RENAME, XDIR, and PAGE, respectively. These alternate commands are merely four of the 31 utilities available, all of which are designed to enlarge system capabilities and improve human engineering.

A second way in which ZCPR2 is more responsive is in the way it handles these commands. The user no longer has to be aware that ASM.COM is on drive A in user 0 and XDIR.COM is on drive B in user 4, for instance, because ZCPR2 can be made to search for the COM file when the user issues a command. The user need never concern himself with where his COM files are—he is free to concentrate on the problem at hand. Granted, on small disk systems, this is only a minor problem of remembering to prefix, say, XDIR with A:XDIR, and so on. Under ZCPR2 he does not even have to remember that.

Each command has a number of options. Under the ZCPR2 idea of bending the system to the user, you are not forced to remember these commands and their options. You can always ask for help when you are on-line. This is the third way in which ZCPR2 is more responsive to the user.

Help is obtained in two ways under ZCPR2. The simplest way is to type the name of the command followed by two slashes. For instance, typing:

```
ERASE //
```

will print out one screen full of documentation on the ERASE command, giving you information on what its parameters and options are. This is nice for jogging your memory, but, if you don't feel that this is enough, you can issue the command:

```
HELP or HELP ZCPR2
```

and receive page upon page of documentation on all the commands in the ZCPR2 system. Invoking the "HELP ZCPR2" command calls up a menu of broad topics. By striking one letter, you are indexed into that topic and, frequently, given a menu of subtopics. By striking another letter, you are indexed into that subtopic, where you will usually find the information you want. Of course, you can ask for help relating to the Help subsystem itself and receive information on how to use it.

If productivity is the primary goal of the ZCPR2 system, learning is the secondary one. ZCPR2 is released to the public domain, and anyone can acquire the complete source code to the system and see how it works internally. I am excited by all of its novel ideas, and feel that as people start using ZCPR2, they will pick up ideas from it and see CP/M in a different light. Key among these is how responsive a system can be to its users, and how this can be accomplished. Having the complete source code to ZCPR2 and SYSLIB allows the user to delve into the way things work. It provides a nice laboratory environment in which one can learn by example and then push on with his own ideas.

ZCPR2 compared to CCP

A dissertation that compares ZCPR2 to CP/M exhaustively lies outside the scope of this article. The reader is invited to read the *Concepts Manual for ZCPR2* and the *Rationale Manual for ZCPR2* for more detail, but here is a brief summary of the main points (see Table 1).

Only two major differences are noted between the memory images of the two systems: ZCPR2 resides in place of the CP/M CCP, and the ZCPR2 BIOS is modified (optionally). ZCPR2 can be implemented in over 4 million different configurations; a minimum configuration has a standard BIOS containing no modifications at all, and a maximum configuration has a modified BIOS which may be 3K or 4K larger than the standard, thereby reducing your Transient Program Area (TPA) correspondingly.

Table 1: CP/M and ZCPR2-based executing memory images

```
Address           CP/M System                ZCPR2 System
High Memory ->  ----------------------     ----------------------
                | BIOS               |     | Modified BIOS      |
BDOS+0E00H  ->  ----------------------     ----------------------
                | CP/M 2.2 BDOS      |     | CP/M 2.2 BDOS      |
CCP +0800H  ->  ---------------------- -   ----------------------
                | CP/M 2.2 CCP       | T | ZCPR2              |
CCP Base    ->  ---------------------- P ----------------------
                | Scratch Area       | A | Scratch Area       |
100H        ->  ---------------------- -   ----------------------
                | CP/M Buffers et al |     | ZCPR2 Buffers et al |
  0H        ->  ----------------------     ----------------------
```

Note that, depending on how you want to configure your ZCPR2 system, you may have to sacrifice some of your TPA to make room for some of the advanced features. A second difference lies in the resident commands under ZCPR2, which are:

- DIR can also display system files if needed
- ERA has a verify option that allows you to approve the files before they are erased; ERA also displays the names of all the files it is erasing.
- REN can delete, with your approval, any non-read-only file it is renaming.
- TYPE is a command that pages, stopping the display after filling your screen; it can also be made to

ZCPR2 runs in place of the CP/M 2.2 CCP. It provides functions similar to those of CCP, and also offers many enhancements.

print a file without paging (as in normal CP/M).

- LIST is a command like TYPE, but prints the file on the LST: device without paging.
- SAVE accepts both decimal and hexadecimal arguments, and you can specify the number of 128-byte sectors or 256-byte blocks to be saved; the hex argument feature is especially nice since DDT gives all its values in hex.
- GET loads a file anywhere in memory for you.
- GO re-executes the COM file currently residing in the TPA.
- JUMP allows you to call a subroutine at any memory address.

Another important feature of ZCPR2 is the DU form, applicable to all the ZCPR2 resident commands and utilities that are provided as part of the ZCPR2 system. You can use "DU:"—where "D" is a disk letter and "U" is a user number—in all the places where you simply used "D:" before in these three ways:

1. DU can be used to log into a different disk and/or user area. The USER command is no longer needed or available.
2. DU can be used in front of file names and in places where D was used before.
3. DU can prefix commands to indicate to ZCPR2 where to go first to find the COM file.

Examples of these features are given in Table 2. Note that the ZCPR2 prompt tells you what disk and user area you are on.

Table 2. Examples of use of the DU form in ZCPR2

A>1:	A>dir 2:	A>b4:xdir
A1>b2:	A>type c31:read.me	A>6:myprog
B2>c:	A>save 15h b4:image.cpm	A>b:doit
C2>a0:	A>era b2:thatfile	A5>b2:run
A>	A>ren c:file1=file2	C3>a:doit

The last feature of ZCPR2 itself that I want to discuss here is the multiple command line. With CP/M, you had to issue one command, then wait for it to finish before issuing another. Under ZCPR2, you can issue a group of commands on one line and then go off while your system does the work. The following command line is perfectly valid:

```
A>b:;dir a7:*.txt;dir c22:;c7:;era *.com;dir
```

Major features of the ZCPR2 system

ZCPR2 itself, obviously, is a major feature of the ZCPR2 system. To recap, some of its features are:

1. Extensions to the resident commands
2. The use of the DU form
3. COM file search
4. Multiple command line

The ZCPR2 utilities combine in various ways to provide even more features, first among which is the DIR form, used in addition to the DU form by the ZCPR2 utilities.

This allows you to assign names to user areas. For instance, you can assign the name "JEFF" to disk A/user 7, "HELP" to disk A/user 24, "WORK" to disk B/user 0, and "ROOT" to disk A/user 15. Once you have done this, commands like the following are possible:

Command Line	Function
A>cd root	Log into the directory "ROOT"
A15>xdir help:	Display files in "HELP"
A15>erase jeff:myfile	Erase MYFILE from "JEFF"
A15>print work:file1	Print FILE1 in "WORK"

Four ZCPR2 utilities are provided to deal directly with named directories (the DIR form), and the rest of the ZCPR2 utilities (like XDIR) can reference them. CD is used to log into a named directory with password protection; PWD is used to print the name of the current directory or the names of all directories you can reference; MKDIR is used to create named directories; and LD defines a global set of directories (see the manuals for more detail).

The named directory feature once more bends the system to the needs of the user. The main application of this feature is in large disk environments (like hard disks), where the user can divide his work into functional areas and refer to them by name, rather than remembering arbitrary disk and user numbers.

Another major feature of the ZCPR2 system is the **path.** A path is a specific sequence of directories, disks, and user areas followed by the ZCPR2 system during the search for a file. ZCPR2 itself uses a path when looking for a COM file, and the utilities can be set up to use any number of paths for a variety of purposes. HELP, for instance, can follow its own path when looking for HLP files. A path can also be expressed symbolically in terms of DU and DIR forms, as in the following example:

```
A$ A0 BASE ROOT
```

This path extends from disk A/current user (the user you are now logged into) to disk A/user 0 to the directory named "BASE" to the directory named "ROOT". A command called PATH is available, which allows you to display and alter paths in the ZCPR2 system. You can use PATH to change the sequence of directories through which ZCPR2 looks for COM files.

The multiple command line is a third feature of the ZCPR2 system; I mention it again because it can be used in another way by some of the ZCPR2 utilities. The buffer, which stores the multiple command line, can also be used by one program to chain to another program or group of programs. For instance,

ZCPR2 was designed to increase productivity—a goal attained by giving the user an environment responsive to his needs while remaining compatible with CP/M 2.2.

Bring the flavor of Unix to your Z80 CP/M system with Unica

"Unicum: a thing unique in its kind, especially an example of writing. Unica: the plural of unicum."

The Unica: a unique collection of programs supporting many features of the Unix operating system never before available under CP/M. The Unica are more than software tools; they are finely crafted instruments of surgical quality. Some of the Unica are:

bc	-	binary file compare, display differences in hex
cat	-	catenate files (vertically)
cp	-	copy one or more files, even between users
dm	-	disk mapper, reports free blocks and directory space
fid	-	file identification by unique numbers (CRC's)
hc	-	horizontal file catenation and column permutation
ln	-	create file links (multiple names for one file)
ls	-	intelligent directory lister, optional multi-columns
mv	-	move (rename) files, even between users
rm	-	remove (delete) files, with optional verification
sc	-	source file compare, with resynchronization
sfa	-	set/reset file attributes, optional verification
sp	-	spelling error corrector, with 80,000 word dictionary
sr	-	search multiple files for a pattern
srt	-	in-memory file sorter, optional duplicate line omission
tee	-	pipe fitting (copy input stream to multiple outputs)
tr	-	transliterate (translate character codes)
wc	-	word counter, counts characters, words, and lines
wx	-	word extractor, copies each word to a separate line

Each Unicum understands several flags ("options" or "switches") which control program alternatives. No special "shell" is needed; Unica commands are typed to the standard CP/M command interpreter. The Unica package supports several Unix-like facilities, such as filename user numbers:

sc data.bas;2 data.bas;3

(compares files belonging to user 2 and user 3);
Wildcard patterns:

rm -v *tmp*

(types each filename containing the letters TMP and asks whether to delete the file);
I/O redirection:

ls -a >proj.dir

(writes a directory listing of all files to file "proj.dir");
Pipes:

dm b: | sr free >lst:

(creates a map of disk B:, extracts those lines in the map which contain the word "free", and prints them on the listing device).

The Unica are written in XM-80, a low level language which combines rigorously checked procedure definition and invocation with the versatility of Z80 assembly language. XM-80 includes a language translator which turns XM-80 programs into source code for MACRO-80, the industry standard assembler from Microsoft. It also includes a MACRO-80 object library with over forty "software components", subroutine packages which are called to perform services such as piping, wildcard matching, output formatting, and device-independent I/O with buffers of any size from 1 to 64k bytes.

The source code for each Unicum main program (but not for the software component library) is provided. With the Unica and XM-80, you can customize each utility to your installation, and write your own applications quickly and efficiently. Programs which you write using XM-80 components are not subject to any licensing fee.

Extensive documentation includes tutorials, reference manuals, individual spec sheets for each component, and thorough descriptions of each Unicum.

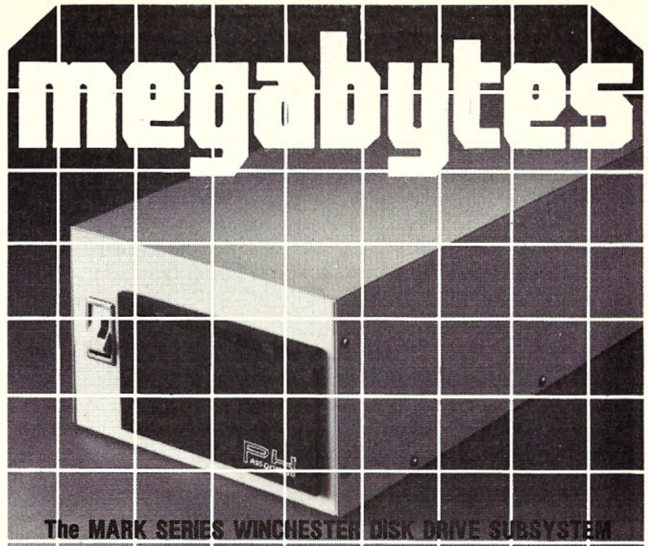
Update policy: each Unica owner is informed when new Unica or components become available. At any time, and as often as you like, you can return the distribution disk with a $10 handling fee and get the current versions of the Unica and XM-80, with documentation for all new or changed software.

The Unica and XM-80 (which requires MACRO-80) are priced at $195, or $25 for the documentation. The Unica alone are supplied as *.COM executable files and are priced at $95 for the set, or $15 for the documentation. Software is distributed only on 8" floppy disks for Z80 CP/M version 2 systems. All orders must be paid in advance; no COD's or purchase orders, please. Quantity discounts are available. Shipment outside of the US or Canada costs an additional $20. Bank checks must be in US funds drawn on a US bank.

Knowlogy
P.O. Box 283-A
Wilsonville, Oregon 97070

Visa/Mastercard customers call (503) 639-3420 for next day shipment.

CP/M is a trademark of Digital Research. Unicum and Unica are trademarks of Knowlogy. Unix is a trademark of Bell Telephone Labs. XM 80 is a trademark of Scientific Enterprises. Z80 is a trademark of Zilog Inc.



The MARK SERIES WINCHESTER DISK DRIVE SUBSYSTEM

- **CAPACITY** - 20, 33 and 46 megabytes.
- **SIZE** - 5¼ floppy size (6x4x2).
- **SPEED** - Twice as fast as the the competition (30 milliseconds).
- **COMPATIBILITY** - Hardware and Software interfaces for TRS-80,® Apple II,® IBM-PC,® CP/M,® Turbodos,® S-100, any 8 bit parallel port, any Z-80® computer such as Osborne,® Xerox,® Televideo,® Northstar Advantage,® NEC,® AVL-Eagle.®
- **EXCEPTIONAL WARRANTY** - Comprehensive 1 year parts and labor.
- **NOTHING ELSE TO BUY** - Subsystems include disk, chassis with power supply, controller, cables, hardware adapter and software pkg.
- **OEM quantity one prices start as low as $2664.**

PH ASSOCIATES
8720 Old Courthouse Road, Vienna, Virginia 22180
703-281-5762

CIRCLE 52 ON READER SERVICE CARD

Professionals Prefer Q/C.

For only $95, Q/C is a professional, fully-supported C compiler for CP/M. Q/C supports a large subset of C, and is upward compatible with the UNIX Version 7 C compiler from Bell Labs. The Q/C library includes over 50 input/output and other support functions, all written in C.

When you buy Q/C, you get a working compiler that generates assembly language. You also receive the complete source code for the Q/C compiler and the function library. The Q/C compiler is written in C, with a few functions hand-coded in assembler to enhance performance. Most compiler options can be customized to suit your taste by using the configuration program we supply.

What really sets Q/C off from the competition is our 138-page *User's Manual*. The tone of the manual is informal and personal. Jim Colvin (the author of Q/C) tells you how to use the compiler, and clearly describes each library function. There's even a chapter that explains in detail the "internals" of Q/C.

Q/C is a fully-supported professional product. We continue to develop and enhance Q/C, and provide updates at a nominal cost. Write or call for details of Q/C Version 2.0.

THE CODE WORKS
5266 Hollister
Suite 224
Santa Barbara, CA 93111
(805) 683-1585

CP/M is a trademark of Digital Research.
UNIX is a trademark of Bell Laboratories.

CIRCLE 64 ON READER SERVICE CARD

one of the ZCPR2 subsystems is the MENU system, which presents a menu to the user and runs a command line based on the menu item selected by him. MENU does this by placing that command line into the multiple command line buffer followed by a semicolon and the command "MENU" (this returns control to the MENU program when done).

More than 30 utilities are provided as a part of the ZCPR2 system, divided into the following classes:

1. The I/O subsystem
2. Directory display
3. Disk utility
4. Library utility
5. Menu subsystem
6. File compare progs
7. File copy utility
8. Command file processors
9. Named-directory programs
10. ZCPR2 residents
11. ZCPR2 alternates
12. On-line documentation
13. Miscellaneous

And this is only the beginning—if it has sparked your interest, I recommend the *Concepts* manual.

What are the gotchas?

With any piece of software, there is always something that someone won't like about it. ZCPR2 is no exception. Although neither the ZCPR2 system nor Mod 0.2, the latest modification to it, contains any known bugs, there are things to watch out for.

The first problem that everyone encounters is that of installation. The system does not come up simply by your typing a command. You (or someone who is knowledgeable in assembly language) have to install ZCPR2. It can be configured in over 4 million ways—and this does not even include the utilities, which also have to be installed.

Installing ZCPR2 necessitates reading the *Concepts* and *Installation* manuals in detail, understanding what is discussed, and then writing some assembly language code. ZCPR2 comes as two files, the body and the header. Configuring ZCPR2 involves selecting options and setting values in the header: you need to know 8080 assembly language to do this. You should not need to touch the body. ZCPR2 must then be assembled with the MAC assembler (which is *not* public domain and must be purchased from Digital Research, Inc.).

When you have configured ZCPR2, you may also have to modify the code in the cold boot routine in your BIOS. The extent of this modification depends on what configuration options you have selected.

Finally, you need to install the utilities of the ZCPR2 system that you want to use. This is relatively easy, however, since an installation program, GENINS, is provided.

GENINS permits you to select the principal op-

tions for the utilities, save these options away in a save buffer, and then move through each utility, copying the save buffer into it and setting whatever special options are necessary. GENINS is menu driven, and, once you are familiar with it, it usually takes approximately 15 minutes to install all 31 utility programs.

Installation, then, is the main "gotcha" to the ZCPR2 system. It takes a certain knowledge of assembly language and access to a commercial assembler (MAC) install it.

A second "gotcha" is the overall size. The ZCPR2 system can be fully implemented only on systems with a relatively large disk capacity. As a rule, more than 500K per disk is recommended in order to implement the ZCPR2 system as it was intended to be used. If you have less capacity than this, you will have to configure the system so that some options are left out. This can still give you many good features, but you may not realize the full capability of the ZCPR2 system as described in the manuals. Features such as paths, multiple command lines, and menus cost little and add much to a system. These combine to form a nice ZCPR2 system subset.

A third "gotcha" is that to implement a full ZCPR2 system, you need to modify the cold boot routine in your BIOS. You can implement a "nice" configuration of ZCPR2 without doing this, and you can realize many good features of ZCPR2 without BIOS modification. However, BIOS modification allows several more outstanding features of the ZCPR2 system to come into play.

Finally, a fourth "gotcha" revolves around the issue of support and assistance with the ZCPR2 system. The support you receive will have to come from your local computer club, people you can contact, the company you bought your ZCPR2 system from (if you elected to buy it rather than use the public domain distribution system), or from me (the author) if you can get hold of me and I'm willing to help at the time.

One important point to remember is that we are dealing with public domain here, and not a business. You are not necessarily paying money for this software, and there is no full-time, dedicated support organization to help you, such as you would find in dealing with a company. There are a great many good people who enjoy becoming involved in local computer clubs and helping others in such matters, but they are doing it for no pay and for altruistic reasons. You cannot make demands on them, but you can ask them for help, and they are usually glad to lend a hand.

Closing words

The ZCPR2 system is capable of growth and evolution. As we use it and learn from it, some ideas will be discarded; others will be adopted and expanded upon. It is not a panacea—but neither is any software. It is a good tool, however, that can meet many needs. I hope you investigate it further, read the manuals, and enjoy using it.

Implementing Console Key-Pressed Interrupts

Type console commands and/or program inputs before they are requested

by John M. Potochnak

The first machine I ever used permitted type-ahead. I sorely missed it when I brought up my CP/M-based micro. The result was its implementation under the grudging auspices of CP/M version 2.2. My definition of type-ahead is the ability to type console commands and/or program input before they are requested. To do this you must process console key-pressed interrupts and buffer the typed characters until the next character requestor gets around to eating them.

Processing interrupts

Step 1 is to enable interrupts for your console keyboard. I have both a serial console (connected through a UART) and a video board. My software will handle either. It uses Z80-mode vectored interrupts, so the correct routine is called whenever a key is pressed.

Step 2 is to buffer the characters somewhere. My buffer is 256 bytes long. This may seem excessively long, but you'll see how that can be used to re-do the CP/M SUBMIT facility into one that is not only faster but far more versatile for small (buffer-size) command files.

Step 3 is to rewrite the console status (CONST) and console get character (CONIN) BIOS routines. CONST now checks to see if anything is in the buffer; CONIN sets a character from the buffer instead of from the device.

The next step is to apply the following BDOS patch.

```
SGCCP   EQU    0980H        ; Where SYSGEN drops the CCP
SGBDOS  EQU    SGCCP+0806H  ; Where SYSGEN drops the BDOS

ORG     SGBDOS+014AH        ; Prohibit BDOS XON/XOFF
NOP                         ;   processing
NOP
NOP
```

This will prohibit the BDOS from doing input while it is outputting to the console. This is necessary, otherwise the standard control-S to stop/start console output will eat up your typed-ahead characters. The control-S feature is then duplicated in the BIOS—fortunately, it is trivial.

The code I use to accomplish the preceding functions accompanies this article. It works with the SD SBC-200 CPU board and the SD VDB-8024 video display board.

Immediate mode character processing

There are a few advantages to processing some characters immediately (i.e., at interrupt level). I have implemented these for my VDB-8024 video board. Its keyboard is wired so that the high-order bit of the key character can be set by holding down a super-shift key. This gives me an additional 128 function keys.

My function keys are as follows:

a) Clear input buffer and warm boot. This allows you to trap out of looping programs or simply abort something.

b) Dummy up an RST instruction. This only works if DDT or my debugger is loaded. When the RST is simulated, the debuggers treat it as if a breakpoint has been hit. This allows you to determine the PC of a looping program or get back into DDT if your breakpoints are not hit.

c) Immediately output a status line. My status line consists of the wall clock time, disk statistics, program state, and the current PC. These are useful for timing operations or judging program progress.

d) Clear the input buffer. Useful if you make a boo-boo.

Handling XON/XOFF in the BIOS

I use control-S (XOFF) and control-Q (XON) to handle stopping and starting console output. I like this better than the standard method of using only control-S to stop and anything to start. This way you can type as many control-S's or control-Q's as you want and always produce the correct effect. Also, this scheme cooperates with type-ahead in that characters other than control-S and control-Q may always be typed ahead.

There is a "gotcha" with handling your own XON/XOFF. You must be able to disable this processing sometimes. My text editor uses these control characters, so they must be passed while editing.

Originally I had a byte which the text editor changed to zero, saying "don't process control-S and control-Q." The byte was also checked to prevent the BIOS output routine from doing auto-linefeed. Auto-linefeed had to be done by the BIOS because my original terminal was too dumb to linefeed at the end of a line.

I now have my own BIOS entry point, which is called by the text editor. This was necessary when the BIOS was broken into two segments to allow most of it to live in another RAM bank. The warm boot code resets this parameter back to normal. Programs that change it need not worry about the next guy unless they "RET" to the CCP.

Ah, the CCP! Alas, she is still half-duplex. I

John M. Potochnak, 106 Birchwood Terrace, Wayne, NJ 07470

looked into the possibility of modifying it but decided that the return on investment (ROI) was much too low. The only commands that don't permit type-ahead are TYPE and DIR. I found that I used neither when typing ahead, so why bother? Additionally, there would have been problems with terminating a TYPE/DIR prematurely. The immediate warm boot would have been required.

Who cooperates with type-ahead

Surprisingly, almost all my software worked without modification in the type-ahead environment. The CP/M utilities, compilers and the editor appear to be happy. The "STAT filename $xxx" command was the only exception I encountered: It aborts if anything is typed ahead.

Of course, any program that does character input, constantly checking for keys pressed, will eat the characters. This may or may not produce the desired results. I found it to work in the majority of cases.

One super advantage is the ability to type full speed at the word processor and text editor. I no longer have to worry about whether they are listening or performing the last function. I don't have to worry about them doing disk I/O. I just type.

Things of which to beware

Beware of race conditions in the character buffering routines. I was bitten when the function key that cleared the input buffer was first installed. It would make the getter and putter indices into the type-ahead buffer equal. This means that nothing is in the buffer. On rare occasions, the character fetcher called via the BIOS jump vector would have just read one of these indices but not the second. It would then be interrupted by the clear input buffer function key, and when it resumed execution, it would use the old value of the first index. The result was the appearance of old characters.

Watch out for devices that require interrupts to be off for long periods of time. My nemesis was the SD Versafloppy II disk controller. It uses programmed I/O transfers. This would turn off interrupts for as much as one-sixth of a second. During heavy disk I/O, I lost a considerable amount of type-ahead. I fixed this by building a little DMA adapter for doing the disk I/O.

Replacing SUBMIT

One nifty feature of having this buffer in the BIOS is the ability to retire the CP/M SUBMIT/XSUB team. I replaced them with a utility called JAM. JAM is a C program that reads a command file, does parameter substitution like SUBMIT, then jams it into the type-ahead buffer (via a BIOS jump vector) and exits.

The limit on what can be jammed is the buffer size in the BIOS. I have never exceeded 256 characters; consequently my current buffer size is totally adequate.

There are several advantages to this scheme. One is that it is faster, as you are not constantly reading files. There is a noticeable difference in the time between commands. Second, it works at the character level. This means that anything which reads console input, even through the BIOS jump vectors, will work. Finally, it works on all disk drives, not just the A: drive.

To implement the JAM program, the STCHAR routine (see listing), which stuffs characters in the type-ahead buffer, will be called outside of interrupt level. Interrupts must be disabled if this is done; otherwise a race condition will arise.

Is it worth the effort?

JAM is nice, but SUBMIT is functional. The "RST" function key is also nice: I've even used it to find a loop in a program. I use the status line function key on a regular basis to time programs. These things are all nice, but I don't believe that any one individually would make it worth the effort.

The two items that make it all worth while are the behavior of the word processor/text editor and the ability to type several commands in advance and leave the machine without having to generate a command file. With the word processor/text editor, you simply type at your own speed, ignoring disk I/O and the processing the programs must do. This alone would make it worth it. The other items are simply gravy.

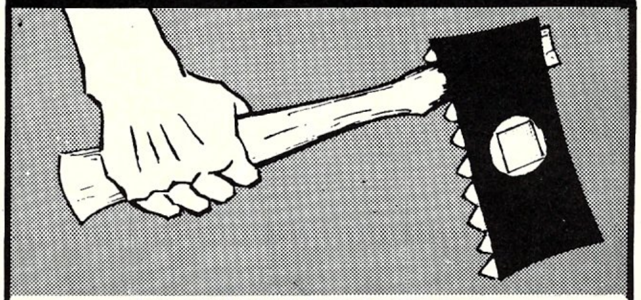
```
                           PSECT   ABS
)F600                      ORG     0F600H                                  18-Jul-82 21:29:29

                 ; ***************************************************************
                 ; *                                                             *
                 ; *                    Type-ahead equates                       *
                 ; *                                                             *
                 ; ***************************************************************
)0038            RST7      EQU     0038H              ; DDT/DEBUG RESTART LOCATION
)00FF            MB        EQU     0FFH               ; BANK SELECT PORT
)007D            USTS      EQU     7DH                ; USART STATUS
)007C            UDAT      EQU     7CH                ; USART DATA
)0079            CTC1      EQU     79H                ; VDB KEYBOARD INTERRUPT CHANNEL
)0078            CTC0      EQU     78H                ; BAUD RATE CTC PORT
)0001            VDAT      EQU     01H                ; VDB DATA PORT

                 ; ***************************************************************
                 ; *                                                             *
                 ; *               Z80 vectored interrupt table                  *
                 ; *                                                             *
                 ; ***************************************************************
)F600                      DEFS    (($+7).AND.0FFF8H)-$ ; FORCE INT VECTORS ON
                                                    ;  AN EIGHT BYTE BOUNDRY
)F600            IVCTC:
F600  CDF8                 DEFW    DMAINT             ; DMA CHIP END-OF-BLK INT PROCESSOR
F602  26F6                 DEFW    KBINTV             ; VDB KEYBOARD ON CHANNEL 1
F604  13F6                 DEFW    KBINTU             ; SBC-200 USART RCV DATA ON CHANNEL 2
F606  0000                 DEFW    0                  ; NOTHING ON CHANNEL 3 (YET)

                 ; ***************************************************************
                 ; *                                                             *
                 ; *  These instructions go into the GOCPM routine in the BIOS   *
                 ; *                                                             *
                 ; ***************************************************************
F608  3EC9       GOCPM:    LD      A,0C9H             ; RET
F60A  323800               LD      (RST7),A           ; IN CASE HE TYPES FUNC-BS W/O A DEBUGGER
F60D  3236F8               LD      (XNFFLG),A         ; XON OUTPUT
F610  322FF8               LD      (CICHR),A          ; ANYTHING NON-ZERO TO PROCESS XON/XOFF

                 ; ***************************************************************
                 ; *                                                             *
                 ; *   Here on vectored interrupt caused by UART received char   *
                 ; *                                                             *
                 ; ***************************************************************
```

I no longer worry about whether the word processor and text editor are listening or performing the last function, or about them doing disk I/O. I just type.

Key-Pressed Interrupts continued . . .

```
F613  ED7332F8  KBINTU: LD      (SPSAV),SP      ; PRESERVE SOMEONE'S STACK
F617  317BF8            LD      SP,LOCSTK       ; OURS
F61A  F5                PUSH    AF
F61B  E5                PUSH    HL
F61C  DB7C              IN      A,(UDAT)        ; GET CHARACTER
F61E  E67F              AND     7FH             ; LOSE PARITY
F620  FE1E              CP      '^'-40H         ; CONTROL-^ IS THE EQUIVALENT OF SS-BREAK
F622  284E              JR      Z,SSWB
F624  1830              JR      KBINT           ; SKIP TO CHARACTER STORE

                ; ************************************************************
                ; *                                                          *
                ; * Here on vectored interrupt caused by VDB-8024 received char *
                ; *                                                          *
                ; ************************************************************
F626  ED7332F8  KBINTV: LD      (SPSAV),SP      ; PRESERVE SOMEONE'S STACK
F62A  317BF8            LD      SP,LOCSTK       ; OURS                  18-Jul-82 21:29:29
F62D  F5                PUSH    AF
F62E  E5                PUSH    HL
F62F  DB01              IN      A,(VDAT)
F631  EE80              XOR     80H             ; COMPLEMENT SUPER SHIFT BIT
F633  F256F6            JP      P,KBINT         ; SUPER-SHIFT NOT PRESSED IF NOT SIGN BIT
F636  FEE1              CP      80H+'a'         ; UPPER CASE LOWER CASE
F638  3806              JR      C,NOTLWR
F63A  FEFB              CP      80H+'z'+1
F63C  3002              JR      NC,NOTLWR
F63E  D620              SUB     'a'-'A'         ; CONVERT TO UPPER CASE
F640  FE80      NOTLWR: CP      80H             ; SS-BREAK
F642  282E              JR      Z,SSWB          ;  YES, WARM BOOT NOW
F644  FE88              CP      88H             ; SS-BACKSPACE
F646  2840              JR      Z,SSDDT         ;  YES, BLINDLY RESTART DDT
F648  FEDE              CP      '^'+80H         ; SS-^
F64A  285B              JR      Z,SSSTAT        ;  YES, INSTANT STATUS
F64C  FEDC              CP      '\'+80H         ; SS-\
F64E  282F              JR      Z,SSCLBF        ;  YES, CLEAR INPUT BUFFER
F650  FEAD              CP      '-'+80H         ; SS-MINUS?
F652  2002              JR      NZ,KBINT
F654  3E5F              LD      A,5FH           ; YES, CHANGE TO UNDERSCORE (NONE ON KEYBOARD)

F656  67        KBINT:  LD      H,A             ; PRESERVE THE CHARACTER
F657  3A2FF8            LD      A,(CICHR)       ; SEE IF WE SHOULD DO XON/XOFF
F65A  B7                OR      A
F65B  7C                LD      A,H             ; GET CHARACTER BACK
F65C  2808              JR      Z,NOX           ; NO
F65E  FE13              CP      'S'-40H         ; CONTROL-S?
F660  2856              JR      Z,XOFF          ;  YES, XOFF OUTPUT
F662  FE11              CP      'Q'-40H         ; CONTROL-Q?
F664  2853              JR      Z,XON           ;  YES, XON OUTPUT
F666  CDBEF6    NOX:    CALL    STCHAR          ; PUT CHAR IN THE BUFFER
F669  E1        KBIGNR: POP     HL              ; RESTORE EVERYTHING
F66A  F1                POP     AF
F66B  ED7B32F8          LD      SP,(SPSAV)      ; CURRENT PROCESS' STACK
F66F  FB                EI                      ; RE-ENABLE INTERRUPTS
F670  ED4D              RETI


                ; ************************************************************
                ; *                                                          *
                ; *             Here to handle the function keys             *
                ; *                                                          *
                ; ************************************************************
                ; HANDLE WARM BOOT BY SS-BREAK OR CONTROL-^
F672  FD2A32F8  SSWB:   LD      IY,(SPSAV)      ; USER'S STACK
F676  21CDF8            LD      HL,WBOOT        ; WHERE WE WANT TO LAND
F679  FD7500            LD      (IY),L
F67C  FD7401            LD      (IY+1),H

F67F  AF        SSCLBF: XOR     A
F680  32CBF8            LD      (KBTAKR),A      ; NO CHARS IN INPUT BUFFER
F683  32CCF8            LD      (KBPUTR),A
F686  18E1              JR      KBIGNR          ; RESTORE REGISTERS AND 'RETURN' TO WARM BOOT

F688  3A38F8    SSDDT:  LD      A,(CBANK)       ; SEE WHICH MEMORY BANK IS ON BELOW INT LVL
F68B  3239F8            LD      (DDT),A         ; MAY HAVE TO WAIT TO RESTART DDT
F68E  B7                OR      A
F68F  20D8              JR      NZ,KBIGNR       ; MUST WAIT TIL OUT OF THE BIOS
F691  ED7330F8          LD      (KBTEMP),SP     ; SAVE OUR STACK
F695  ED7B32F8          LD      SP,(SPSAV)      ; USER'S STACK          18-Jul-82 21:29:29
F699  213800            LD      HL,RST7         ; WHERE RST IS FOR DDT/DEBUG
F69C  E5                PUSH    HL
F69D  ED7332F8          LD      (SPSAV),SP      ; SAVE FOR RETI
F6A1  ED7B30F8          LD      SP,(KBTEMP)     ; OUR STACK
F6A5  18C2              JR      KBIGNR

F6A7  3E01      SSSTAT: LD      A,1             ; FORCE BANK 1
F6A9  D3FF              OUT     (MB),A
F6AB  3236F8            LD      (XNFFLG),A      ; IF WE CALL AT IRP LVL, MUST NIX XOFF
F6AE  CDCDF8            CALL    $STATLN         ; PRINT A STATUS LINE
F6B1  3A38F8            LD      A,(CBANK)       ; BACK TO WHATEVER
F6B4  D3FF              OUT     (MB),A
F6B6  18B1              JR      KBIGNR


                ; SET XON/XOFF FLAG
F6B8  AF        XOFF:   XOR     A
F6B9  3236F8    XON:    LD      (XNFFLG),A      ; XON/XOFF
F6BC  18AB              JR      KBIGNR          ; AND DUMP CHARACTER

                ; ************************************************************
                ; *                                                          *
                ; *         Store character in the type-ahead buffer         *
                ; *                                                          *
                ; ************************************************************
F6BE  F5        STCHAR: PUSH    AF              ; PRESERVE FOR LATER
F6BF  3ACCF8            LD      A,(KBPUTR)      ; INDEX FOR PUTTER (US)
F6C2  F5                PUSH    AF              ; SAVE, WE MAY STORE CHAR HERE
F6C3  21CBF8            LD      HL,KBTAKR       ; ADDRESS OF TAKER INDEX
F6C6  3C                INC     A               ; STEP TO NEXT PUT INDEX
F6C7  BE                CP      (HL)            ; WILL WE PASS TAKER?
F6C8  2003              JR      NZ,KBINT0       ; NO, OK TO STORE CHAR
F6CA  F1                POP     AF              ; DUMP PUTTER INDEX
F6CB  F1                POP     AF              ;  AND CHARACTER
F6CC  C9                RET                     ; AND IGNORE THIS INTERRUPT
F6CD  32CCF8    KBINT0: LD      (KBPUTR),A      ; SAVE INDEX FOR NEXT TIME
F6D0  F1                POP     AF              ; GET INDEX FOR THIS CHAR
F6D1  212FF7            LD      HL,KBBUFR       ; ADDRESS OF THE KB BUFFER
F6D4  85                ADD     A,L             ; COMPUTE LOC FOR CHARACTER
F6D5  6F                LD      L,A
F6D6  3001              JR      NC,KBINT1
F6D8  24                INC     H
F6D9  F1        KBINT1: POP     AF              ; GET THE CHARACTER
F6DA  77                LD      (HL),A          ; SAVE IN THE BUFFER
F6DB  C9                RET


                ; ************************************************************
                ; *                                                          *
                ; *  This code used at cold boot time to initialize the UART *
                ; *  and setup the CTC for vectored interrupts on key pressed *
                ; *                                                          *
                ; ************************************************************
F6DC  000000    DINIT:  DEFB    0,0,0           ; GET USART INTO KNOWN STATE
F6DF  40                DEFB    40H             ; RESET USART
F6E0  4E                DEFB    4EH             ; 4FH FOR 150, 300 BAUD
F6E1  37                DEFB    37H             ; DTR TRUE
F6E2  450D              DEFB    45H,13          ; CTC0 TO 9600 BAUD
F6E4  00                DEFB    IVCTC.AND.0F8H  ; INT VECTOR FOR CTC
F6E5  C501              DEFB    0C5H,1          ; CTC CHANNEL 1 SETUP FOR KB INTS
                                                                        18-Jul-82 21:29:29
                ; ************************************************************
                ; *                                                          *
                ; *  Following code for interrupts from VDB-8024 only, serial *
                ; *  console accessed via the UART is enabled by a program   *
                ; *                                                          *
                ; ************************************************************
F6E7  3EF6              LD      A,IVCTC.SHR.8   ; HIGH ORDER ADDRESS OF VECTORS
F6E9  ED47              LD      I,A             ; SET INTERRUPT VECTOR HIGH
F6EB  ED5E              IM      2               ; SET VECTORED INTERRUPT MODE
F6ED  21DCF6            LD      HL,DINIT        ; USART/CTC SETUP
F6F0  017D06            LD      BC,600H+USTS    ; XFER 6 BYTES
F6F3  EDB3              OTIR
F6F5  017803            LD      BC,300H+CTC0    ; 3 BYTES
F6F8  EDB3              OTIR
F6FA  DB01              IN      A,(VDAT)        ; CLEAR KEY PRESSED ON COLD BOOT
F6FC  017902            LD      BC,200H+CTC1
F6FF  EDB3              OTIR

                ; ************************************************************
                ; *                                                          *
                ; * This loop goes at the beginning of the CONOUT routine, it *
                ; *  implements XON/XOFF                                     *
                ; *                                                          *
                ; ************************************************************
)F701           $CONOUT:
F701  3A36F8            LD      A,(XNFFLG)      ; CHECK FOR XON
F704  B7                OR      A
F705  28FA              JR      Z,$CONOUT

                ; ************************************************************
                ; *                                                          *
                ; *         Check for character in the input buffer          *
                ; *                                                          *
                ; ************************************************************
F707  F3        $CONST: DI                      ; INSURE NO FUNNIES (SS-\)
F708  3ACBF8            LD      A,(KBTAKR)      ; INDEX OF TAKER
F70B  21CCF8            LD      HL,KBPUTR       ; ADDRESS OF THE PUTTER INDEX
F70E  96                SUB     (HL)            ; SEE IF CHARACTER READY (IE NOT =)
F70F  FB                EI
F710  C8                RET     Z               ; EQUAL, NOTHING YET
F711  3EFF              LD      A,0FFH          ; ELSE GOT SOMETHING
F713  C9                RET
```

UNIPROM



THE VERSATILE EPROM HANDLER

- Reads programs 2704, 2708, 2758, 2508, 2516, 2716 (1supply) 2532, 2732, INTEL'S 2732A and the 8755A (INTEL/NEC).
- Reads/erases/programs Hitachi 48016 EEPROM'S
- No personality modules required.
- All signals are S-100 compatible (can adapt to most other buses)
- Port mapping occupies NO memory space.
- Bus clock rates exceeding 6mhz.
- All software is 8080/8085/Z80 compatible.
- Software "user" friendly.
- All software is fully CP/M® & CDOS compatible.

Board (A&T) with extensive documentation .. $199.00
Disk software (8" or NORTHSTAR 5.25") $38.00
EPROM-based software with source listing ... $55.00
Expansion console, 24 pin $50.00

CDOS is a registered trademark of CROMENCO
CP/M® is a registered trademark of DIGITAL RESEARCH INC.
NORTHSTAR is a registered trademark of NORTHSTAR COMPUTER, INC.

Featured in July, August '82 Microsystems

CER-TEK

6020 Doniphan (915) 581-6697 El Paso, TX 79932

ELECTRONICS MANUFACTURING SINCE 1975

CIRCLE 76 ON READER SERVICE CARD

THE V.I.P.* IS HERE!

*Versatile Instrumentation Peripheral

Perfect for Use in Process Control and Data Acquisition Systems or in Standalone Instrumentation

High Performance and Resolution

Extremely Cost Effective

- ★ Single Card S100 A/D, D/A, Digital I/O
- ★ Instrumentation Amplifier Input Channel
- ★ D/A Voltage and Current Source Output
- ★ Full 12 Bit Resolution
- ★ Software Configurable Analog Circuitry
- ★ 25 μs A/D Conversion Time
- ★ 48 High Voltage, High Current Digital Drivers, 100mA per Output
- ★ Drives LED Display and Decimal Points with Software Selectable ROM Decoding
- ★ Two Double Pole Utility Relays
- ★ 8 Bit TTL-Compatible Input
- ★ Self-Test Capability

$595

Fully assembled and tested. Documentation and applications software provided. Check, MO, MC, Visa accepted. MA residents add 5% sales tax, $4 shipping and handling charge.

AUTOMATED CONTROL SYSTEMS
1105 Broadway
Somerville, MA 02144
(617) 628-5373

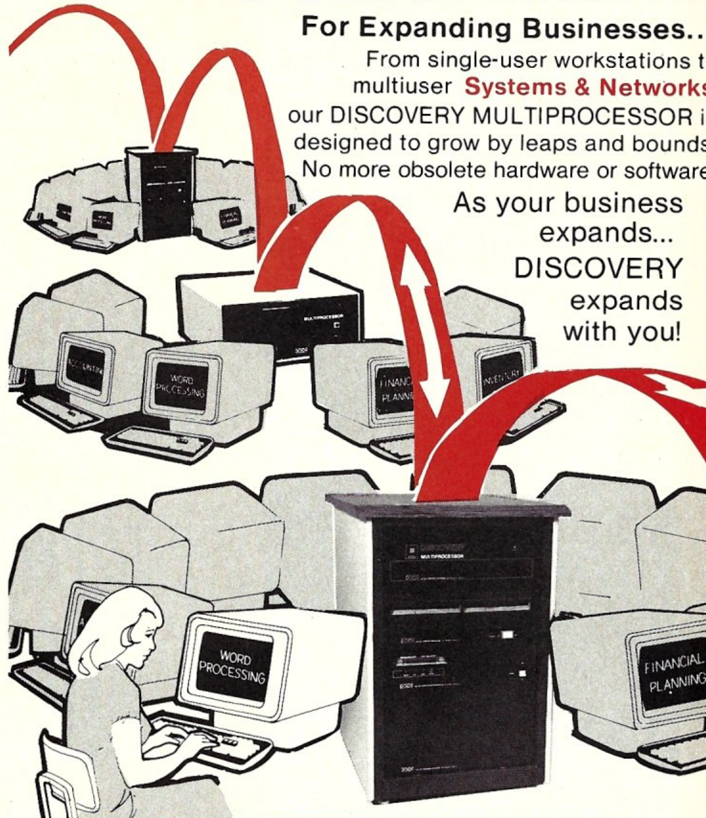
CIRCLE 200 ON READER SERVICE CARD

The ACTION Solution

For Expanding Businesses...

From single-user workstations to multiuser **Systems & Networks**, our DISCOVERY MULTIPROCESSOR is designed to grow by leaps and bounds. No more obsolete hardware or software.

As your business expands... DISCOVERY expands with you!



The **DISCOVERY 500,** a fully integrated desktop computer with 5¼" hard and floppy disks, supports up to 7 users. It is the ideal, low cost turnkey business system. The full size DISCOVERY supports up to 16 users with a wide variety of disk and tape subsystems. And remember, all DISCOVERY users have their own dedicated memory and 8-bit or 16-bit CPU, running CP/M-80* or CP/M-86*. **Action's** own multiuser multiprocessor operating system, the **dpc/os**©† makes it easy.

The **dpc/net**™ low-cost **local area networks** of multiple DISCOVERYs provide the ultimate in performance. Up to 150 users in 10 DISCOVERY systems can be on-line simultaneously with full resource sharing. **For the first time,** mainframe capability at micro prices.

*CP/M is a reg. TM of DIGITAL RESEARCH CORP.
†**dpc/os** is a reg TM of ACTION COMPUTER ENTERPRISE, INC.

Dealer, Distributor & OEM inquiries are invited.

Take ACTION! Call us NOW...at (213) 351-5451

Action Computer Enterprise, Inc.
430 N. Halstead St., Pasadena, CA 91107 USA
TWX 910-588-1201 ACTION PSD ○ (213) 351-5451

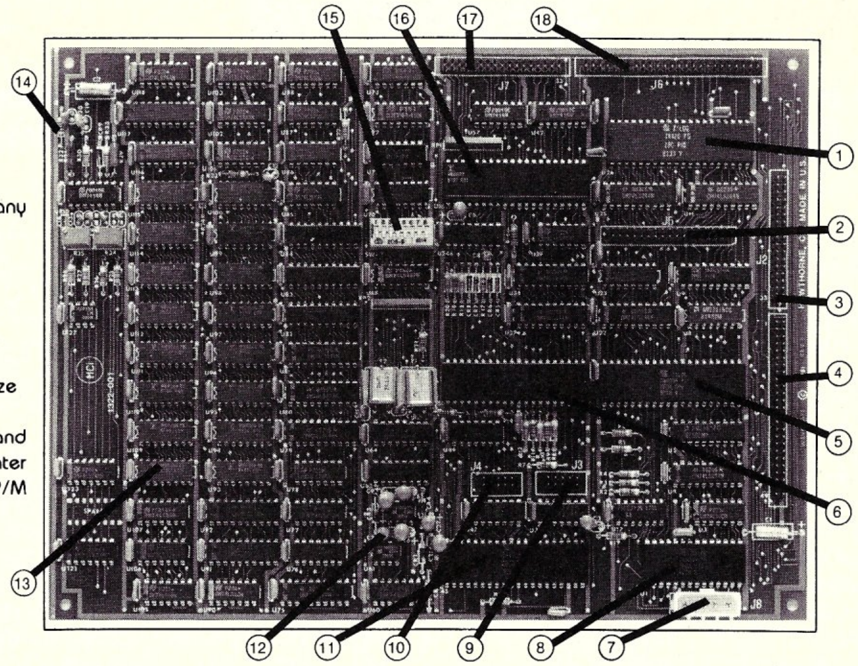
In Europe: A.C.E. (Europe) Eindhoven, Netherlands TELEX 51767 ACE E NL Tel. 040-452658
In Canada: CESCO Electronique LTEE ○ Montreal, Canada ○ (514) 735-5511
In Asia: Action Computer Enterprise (Asia) Hong Kong TWX 75332 PACIC HX Tel. 5-441692

CIRCLE 197 ON READER SERVICE CARD

Key-Pressed Interrupts continued. . .

```
                 ; ***********************************************************
                 ; *                                                         *
                 ; *       Read a character from the type-ahead buffer       *
                 ; *                                                         *
                 ; ***********************************************************
>F714            $CONIN:
F714  CD07F7             CALL    $CONST          ; CHECK STATUS
F717  28FB               JR      Z,$CONIN        ; NO, WAIT
F719  3ACBF8             LD      A,(KBTAKR)      ; INDEX FOR TAKER (US)
F71C  212FF7             LD      HL,KBBUFR       ; ADDRESS OF THE BUFFER
F71F  85                 ADD     A,L             ; COMPUTE CHAR LOCATION
F720  6F                 LD      L,A
F721  3001               JR      NC,CONIN0
F723  24                 INC     H
F724  3ACBF8     CONIN0: LD      A,(KBTAKR)      ; BUMP TAKER ADDRESS
F727  3C                 INC     A
F728  F3                 DI                      ; NO RACES PLS            18-Jul-82 21:29:29
F729  32CBF8             LD      (KBTAKR),A
F72C  7E                 LD      A,(HL)          ; GET THE CHARACTER
F72D  FB                 EI
F72E  C9                 RET

                 ; ***********************************************************
                 ; *                                                         *
                 ; *                 Data items for type-ahead               *
                 ; *                                                         *
                 ; ***********************************************************
>F72F            KBBUFR: DEFS    256             ; KEYBOARD INPUT BUFFER
>F82F            CICHR:  DEFS    1               ; NON-ZERO FOR SPECIAL CHAR PROCESSING IN BIOS
>F830            KBTEMP: DEFS    2               ; FOR TEMPORARY STACK STORAGE
>F832            SPSAV:  DEFS    2               ; SAVE CALLER'S STACK HERE DURING INTERRUPTS
>F834            BANKSV: DEFS    2               ; SAVE STACK DURING BANK SWITCHING
F836  FF         XNFFLG: DEFB    0FFH            ; NON-ZERO IF CONSOLE OUTPUT XON'D
F837  00         BANK:   DEFB    0               ; MEMORY BANK INTO WHICH TO DO DISK IO
F838  00         CBANK:  DEFB    0               ; MEMORY BANK IN CONTROL NOW (FOR DDT RST7)
F839  00         DDT:    DEFB    0               ; NON-ZERO TO CAUSE DDT RESTART
F83A  00         UTERM:  DEFB    0               ; NON-ZERO TO ACTIVATE PARALLEL TERMINAL
>F83B                    DEFS    40H             ; LOCAL STACK SPACE FOR INTERRUPTS
>F87B            LOCSTK  EQU     $
>F87B                    DEFS    50H             ; FOR BANK SWITCHING (OLD STACK MAY NOT BE
>F8CB            BANKSP  EQU     $               ;  ADDRESSABLE)
F8CB  00         KBTAKR: DEFB    0               ; KEYBOARD INDEX FOR TAKER
F8CC  00         KBPUTR: DEFB    0               ; INDEX FOR PUTTER (DEFB TO FORCE ALLOCATION)

                 ; ***********************************************************
                 ; *                                                         *
                 ; *          Dummy entries to make assembler happy          *
                 ; *                                                         *
                 ; ***********************************************************
>F8CD            DMAINT:
>F8CD            WBOOT:
>F8CD            $STATLN:
F8CD  C9                 RET

                         END
```

GET THE BEST OUT OF CP/M® WITH WAVE MATE'S BULLET



- Cost Effective
- Highest Performance of any 4MHZ Z80 Board
- 128 K Usable RAM
- Enhanced Software to Optimize CP/M
- DMA Track Buffered Disk Controller
- Only 8 x 10.7 inches in size
- Power only 5v @ 1.5 a.
- Use any Serial Terminal and Centronics compatible printer
- Capable of supporting MP/M

1. Printer/Winch Port
2. Printer Conn.
3. Winch. Conn.
4. Exp. Data Bus
5. CPU Chip
6. DMA Chip
7. Power
8. Clock Timer Cont.
9. First Serial Port
10. 2nd Serial Port
11. DART
12. Charge Pump
13. 128K RAM
14. Data Separator
15. Boot Disk Setter
16. Floppy Disk Contr.
17. 5¼" Floppy Conn.
18. 8" Floppy Conn.

WAVE MATE INC.
14009 S. Crenshaw Blvd.
Hawthorne, CA 90250
(213) 978-8600 Telex: 194369

WAVE MATE INTERNATIONAL
159 Chee de Vleurgat
1050 Brussels, Belgium
Tel: (02) 649 10 70 Telex: 24050

CIRCLE 177 ON READER SERVICE CARD

A> INFORMATION MANAGEMENT

DBPACK - Data Base Management: Indexing, Sort & Search, Tabulation, Address Labels, Word Processor Interfaces, and lots more!

FILER - Files Compression, Archiving and Cataloging: Substantial Disk Space Savings; Date-stamping & Descriptive Comments ...

A> COMMUNICATION SOFTWARE

COMCOM - Communication Program: Communicate with Time-sharing systems and other computers; Advanced Features; Easy to Use.

CPMCPM - File Transfer Program: Transfer Files (any type) between different CP/M-based systems.

A> For CP/M® Computers

CP/M is a registered trademark of Digital Research

Configured for a wide variety of systems. Disk formats include 8-inch, Osborne, Xerox ...

Call or write for information

COMPU-DRAW
1227 Goler House
Rochester, NY 14620
Phone: (716)-454-3188

Dealer inquiries invited

MasterCard, Visa & American Express cards welcome. Separately ordered documentation may be returned for full refund within 10 days!

It's the writing on the wall

CIRCLE 207 ON READER SERVICE CARD

THEY SAY IT ALL... WE DO IT ALL!

ANNOUNCING THE C86™ C COMPILER —THE COMPILER THAT SPEAKS THE LANGUAGE OF THE FUTURE!

Kernighan and Ritchie's book, *The C-Programming Language*, is the key source for C. Just as fundamental is the C86™ C Compiler.

The C86™ C Compiler is especially designed for the IBM® Personal, IBM® Display Writer, CP/M-86® and MS-DOS®.

For further information on the C-programming language and the C86™ C Compiler, please contact:

C86 is a trademark of Computer Innovations, Inc.; CP/M-86 is a trademark of Digital Research; IBM and MS-DOS are registered trademarks of International Business Machines, Inc.

Computer Innovations, Inc.
75 Pine Street
Lincroft, New Jersey 07738
Telephone: (201) 530-0995

CIRCLE 209 ON READER SERVICE CARD

Customize Your Word Processor Keyboard

Use a translator and an interrupt handler to generate Escape sequences without losing characters

by H. Bradford Thompson

Does your favorite editor of word processor use all the special keys on your terminal? Or do you wear out your left pinky entering control characters while special-function and cursor-control keys sit idle? And if the latter, have you wondered why? After all, how much can it take to activate those special keys? Why not just detect whatever they send, look it up in a table, and send the processor program what it expects for the action desired?

Of course, if it were that simple more editors would come fully customizable, and I wouldn't have a subject for this article. So I'll first describe the problems and then discuss their solution for WordStar with the Zenith/Heath Z-19/H-19 terminal. The added code is placed in areas that WordStar designated for user installation patches, plus space in the CP/M (Customer-adaptable) Basic Input Output System. The programs have been kept as general as possible and should be adaptable to other editors and terminals.

With the resulting system you can assign to single keys entire sequences of editor instructions and small routines, thus creating your own instructions. As a simple example, chemists commonly want single-number subscripts (as for instance, in H_2O). In WordStar the subscript requires the sequence

```
Ctrl-P, V, 2, Ctrl-P, V
```

for a total of seven finger-contacts with four different keys. I just typed that subscript 2 with two strokes:

$$f_1, 2.$$

My use of the Z-19's keypad and other special keys is summarized in Figure 1.

The problem

The special-function keys of a terminal commonly produce a sequence starting with a nonprinting character. This character is usually the ESC, or Escape, and that character and what follows is called an escape sequence. Counting the keypad at right[1], the Z-19 has 21 special keys. Ten of these send one escape sequence unshifted and a different sequence with the shift depressed. We can thus assign 31 different special functions to these keys. Nineteen of these sequences consist of ESC followed by one character, while 12 consist of the ESC, a question mark, and a third character. For example, the right-arrow key shifted sends ESC C. Unshifted[1], the same key sends ESC ? v.

H. Bradford Thompson, Dept. of Chemistry, The University of Toledo, 2801 W. Bancroft St., Toledo, OH 43606

We must, then, intercept and identify these sequences and translate them into what we wish the Editor (or word processor, but we'll say Editor for short) to see. For versatility, we will include a way of calling short special programs. This is the task of the Translator, which is the first of the two programs described below.

The Translator alone will work pretty well. However, suppose an escape sequence arrives from the terminal while the Editor is still coping with the previous arrival. The ESC may well get lost, leaving the Editor with only the following letter. The result, while rarely catastrophic, can be pretty annoying. WordStar normally can handle input at typing speed, but can easily be overloaded by using the Z-19 repeat key. Incidentally, this is not a problem unique to my Translator. If you own the full-screen editor VEDIT and a Z-19 terminal, try holding down the REPEAT and either the up or the down arrow. All goes well for a while, but soon after the cursor hits bottom (or top) and VEDIT has to scroll the display, you will find stray letters in your text. These are simply the "followers" of missed Escapes. VEDIT does its best: It checks for incoming characters whenever feasible. It just can't keep up with the REPEAT key.

There is another time when a Translator can lose part of an Escape Sequence: when the Editor takes time for a disk operation. WordStar prints WAIT at the top of the screen. That's good advice. However, loss of an ordinary character or two is pretty innocuous. Loss of part of an escape sequence, plus the possibility of tacking on something else, is much more serious.

You can avoid these problems by using the REPEAT with care (or not all), and by heeding all WAITs or just quitting while your disks whir. However, we set out to use the full capacity of our keyboard, so why quit now? A solution exists in a well-designed interrupt-driven keyboard input handler. Such a handler, written for Lifeboat Associates' BIOS for CP/M and a North Star Horizon computer, is the second program to be described below. Unless your computer doesn't permit user-introduced interrupts, this program ought to suggest what must be done, and may be adaptable directly.

Does the use of ESC to introduce sequences mean you lose the ESC key for its usual purposes? No, you don't have to!

The Translator

WordStar contains a provision for a user-designed keyboard input routine. The Translator fills this role. Whenever WordStar requests an input character, the Translator first checks to see if it has any characters to send to complete some previous request, and if not, takes in one character from the keyboard. It checks to see if this is an ESC. If it is not, the character is sent along to WordStar. If an ESC is received, the Translator stalls for slightly longer than the time needed for the keyboard to send one more character (for example, 1.3 milliseconds at 9600 baud). If nothing is received in that time, the ESC came from the ESC key and is sent on. If another character is received within this stall period, translation is in order. Those so inclined can follow the process in Listing 1.

Listing 1 includes, besides the Translator, the character translation tables (Tables 1-5), some minor patches, and a patch loader for convenience in loading it all into WordStar. First we will examine a line in Translation Table 1:

```
80H+'J', XK, 25 .
```

which translates into

```
J*  CTRL-K  CTRL-Y
```

where J* stands for a character J with the parity bit set to 1. This says to the Translator: "The sequence ESC J (which is generated by the ERASE key) is to be translated as the sequence Ctrl-K, Ctrl-Y (which is WordStar input for 'Delete Marked Block')." The Translator uses Table 1 as follows: To the character following an ESC, a set parity bit is added, and Table 1 is searched for a match. The character sequence following the matched character in the table is then supplied to WordStar. On receiving, for example, an ESC J, the Translator would feed WordStar a Ctrl-K and then a Ctrl-Y. The next character (E*) in Table 1 has a set parity bit: the Translator realizes this is the start of another sequence, so the present string of output characters is complete.

Possible translations are not restricted to WordStar's single-key instructions. An example is "Delete Word Left", which is not in WordStar's repertoire (although "Delete Word Right" is). I more often want to delete the word I just typed, so a single keystroke is translated into two commands: "Move Left One Word", then "Delete Word Right".

If more than a simple translation is required, special routines may be written and included. An example is the Table 1 sequence

```
80H+'S', 0, SPR1, XP, 22, 0, XP, 22.
```

This translates ESC, S, produced by the key F_1, and handles the single-character subscript. The first 'character' after the S* is a zero byte. The Translator interprets this as meaning that the next byte is the relative address of a special routine. The special routine at relative address SPR1 reads one more character from the keyboard, plants that character in the sixth position in the sequence, and then resumes the sequence with the first XP, thus sending WordStar

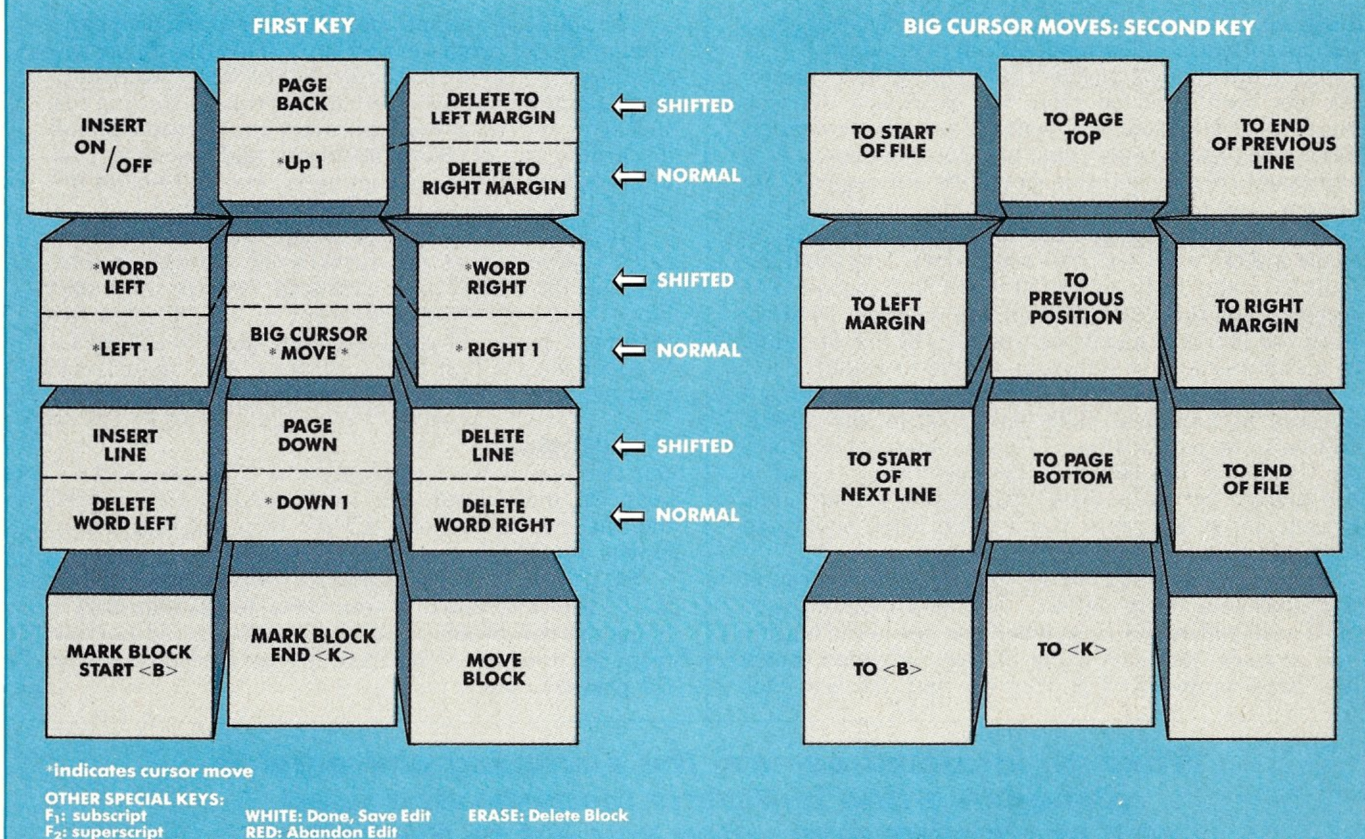
```
Ctrl-P Ctrl-V  n  Ctrl-P Ctrl-V
```

where N is the one-character subscript.

Another special routine leads into secondary tables. For example, the last item in Table 1:

```
80H+'?', 0, SPRT, TABL2L
```

handles the Z-19 three-character escape sequences beginning with ESC ?. The special routine reached via SPRT interprets the next byte (TABL2L) as the



Possible use of Z19 keypad

length of the table immediately following, gets one more keyboard-input character, and looks it up. Table 2 contains the third characters of three-character sequences, plus the desired translations.

A more elaborate use of the secondary-table routine is made for two-key sequences beginning with the center ("HOME") key of the keypad. This key plus one of its neighbors is used for large cursor movements (see Figure 1, right side). The two-key combinations are very natural-feeling, and none of these cursor moves are ones with which it is logical to use the repeat key. Table 3 and Table 4 of the Translation Tables check on the "ESC, ?" sent by the second key; Table 5 completes the translation. These cursor moves include two not in WordStar's repertoire that I find very useful: "To End of Previous Line" and "To Start of Next Line".

WordStar provides space for user console-routine patches. The Translator fits nicely into this space. It could, however, be part of the CP/M BIOS (Basic Input-Output System), provided there is no conflicting use of the special keys by other programs.

The keyboard interrupt handler

We noted above that an interrupt-driven keyboard input handler greatly improves Translator operation. I had written interrupt software for larger computers, but this was my first attempt with the Z80 processor. Contrary to much I had read, I found the Z80 interrupt structure easy to work with.

Of course, the computer in use must allow user-designed interrupts. This may involve hardware as well as software adjustment: What is required is that an interrupt request be generated whenever a character has been received from the keyboard. On the North Star Horizon, four types of console-service interrupt request signals come to a common point: It is necessary to connect this point via an appropriate jumper, and to disconnect all but the one type of request desired[2]. The jumper was installed so as to cause a RST 18H (in Z80 code; RST 3 in 8080 assembler code) to be executed when this request was received and the Z80 interrupt was enabled, Mode 0. Both the Horizon hardware and CP/M are designed to make this sort of interrupt simple to install.

The program to service the interrupts thus created is given in Listing 2. The listing shows the USER section to be patched into Lifeboat Associates' BIOS for CP/M for the Horizon. The second serial port in my system serves a NEC Spinwriter printer. The console input interrupt portion has been separated from the rest for easy transplantation.

When an interrupt occurs the incoming character is placed in a "ring buffer." This is a memory area used as a continuous ring: when you get to the bottom you go back and start using the top, provided that it has been emptied. The routine that fetches each character for program use will zero out the space thus emptied, so the input routine can always tell if the next buffer location is free. When the buffer fills, we just have to throw away input. This is unlikely to be a real loss. Normal typing will never fill up the buffer.

The input routine also watches for ESC characters, and notes when an incoming escape sequence is in progress. Now suppose that we hold down a special key and the repeat key for so long that we fill the buffer. We may find ourselves with too little room for all of a sequence, and a partial sequence is just what we don't want! So in this case we backtrack to the ESC and discard it and all that followed.

In Listing 2 the ring buffer is installed by cleaning out the memory occupied by the once-only code executed on a CP/M "Cold Boot" (i.e., startup from scratch). This saves space but has disadvantages. You may prefer to move the buffer to the unused area following the patch.

When designing any interrupt-driven handler, one must watch out for possible effects of interrupts on parts of the software that are timing-sensitive. An example in my system is the floppy-disk handler, which requires the attention of the Z80 with the interrupts off.[3] The Lifeboat BIOS contains code that disables and enables the interrupts accordingly. However, this does not get us off the hook: suppose that the start of an escape sequence is lost while the disk handler is at work, but the end of the sequence comes in after the interrupts have been enabled. In that case we are in trouble.

The solution is to intercept the call to the disk handler, and take proper action after it. What if characters are arriving from the keyboard when we return from handling the disk? We detect this by stalling for slightly more than one character-time, reading and discarding any keyboard interrupt, and repeating this until nothing further is coming in. For 9600-baud input, this will cost us a very few milliseconds.

Finally, how do we intercept that disk-routine call? Lifeboat does not furnish the listing for that part of its BIOS, but a search reveals only one DI . . . EI ("Disable Interrupt" . . . "Enable Interrupt") pair, and right between them is a CALL. We just replace that CALL with a CALL to our own routine, which has the original CALL in its middle.

Conclusion

So, that's it! I have used parts of this system for a year or more, and the final version for several months. As far as I can tell the bugs are out, and it makes WordStar more fun to use.

So if you have WordStar, a North Star, and a Z-19, use it directly. If you have something else, I hope you can profit from my experience and borrow from my routines. You have my permission to copy for personal use.

When an interrupt occurs, the oncoming character is placed in a ring buffer—a memory area used as a continuous ring: when you get to the bottom, you go back and start using the top.

```
SUBTTL WSPATCH  --  CONSOLE PATCH FOR WORDSTAR   28 SEPT 82
;
; ----------------------------------------------------------
; ¶                                                        ¶
; ¶ SPECIAL-KEY TRANSLATION PATCH -- For Wordstar          ¶
; ¶                                                        ¶
; ¶ Functions:                                             ¶
; ¶   Adaptation to keyboard of particular terminal, &     ¶
; ¶      optimal use of special keys.                      ¶
; ¶   Addition of specialized features.                    ¶
; ¶                                                        ¶
; ¶ Parts:                                                 ¶
; ¶   Patch Segment Loader                                 ¶
; ¶   Translator:  Traps ESC sequences and translates      ¶
; ¶      or handles them as needed.                        ¶
; ¶   Translation Tables                                   ¶
; ¶                                                        ¶
; ¶ Brad Thompson                                          ¶
; ¶                                                        ¶
; ¶ Developed and tested on:                               ¶
; ¶   Northstar Horizon w/56k memory & two double sided    ¶
; ¶      double density 5" floppy disk drives              ¶
; ¶   Zenith Z19 Terminal                                  ¶
; ¶ CP/M 2.2. Lifeboat Version for Northstar               ¶
; ¶                                                        ¶
; ¶ This package is copyrighted by the author.             ¶
; ¶                                                        ¶
; ----------------------------------------------------------
;
;    <<<<< SPECIAL FEATURES ADDED TO WORDSTAR >>>>>
;
;ONE-KEY SUPER- AND SUBSCRIPT SETUP:
;
;        A one-character subscript is entered by striking
;the f1 key and then the subscript (x).  This sends the
;sequence °P°Vx°P°V, resulting in insertion of °Vx°V in
;the text.  The F1 key can also be used to initiate and
;terminate a multiple-character subscript by striking f1
;and then a space.  This inserts only °V.  The F2 key
;operates similarly to produce superscripts.
;
PAGE 62
```

```
;
;    <<<<< HOW TO ADD THIS PACKAGE TO WORDSTAR >>>>>
;
;Start with an "installed" file WS.COM -- that is. a file
;created using the program INSTALL.  Use DDT as follows:
;
;        A>REN WSX.COM=WS.COM     <-- Rename WS for safety
;        A>DDT WS.COM
;        DDT VERS 2.2
;        NEXT  PC
;        3F00 0100
;        -IWSPATCH.HEX            <-- I. then patch file
;        -R                         (This identifies, then
;        NEXT  PC                   reads patch)
;        4100 3F00                <-- First no. is upper
;                                   limit of file.
;        -A100                    <-- Put in Jump to patch
;        0100  JMP 3F00             into start of WS
;        0300                     <-- End assemble by <CR>
;        -°C                      <-- Exit from DDT
;        A>SAVE 64 WS.COM
;
;NOTES:
;
;Size of file to SAVE is calculated from first two digits
;of NEXT value after -R.  Example using 4100 above:
;4x16 + 1 = 65.  If and only if NEXT ended in 00, we can
;subtract 1, giving 64.
;
;The JMP patched onto Wordstar is to cause the program to
;start with the Loader in this package.  The first thing
;the loader loads is the original address.  The address
;WSTART in this module should match the contents of loc.
;101-2 in your Wordstar.
```

```
Macros:

Symbols:
BIOS    D300   BUFLEN  007E   CBAUD   0060   CHRLEN  0061
CINSET  DAB8   CIPORT  0002   CLINK   DAB1   CLINKA  0018
CONSRV  DAE3   CSMASK  0002   CSPORT  0003   DCALLA  D7AA
DEL2    DB71   DELOOP  DB67   DROUTN  D7FF   DRSET   DB5F
EILOOP  DB64   ESC     001B   ESCZAP  DAFE   ESCZLP  DB01
H19LUP  DA87   H19SET  DA95   H19STL  001C   HCINIT  DA00
HCONIN  DA06   HCONOU  DA09   HCONST  DA03   HLIST   DA0C
HPRST   DA15   HPUNCH  DA0F   HREADR  DA12   INBEND  DACD
INDON1  DAF9   INDON2  DB0F   INDONE  DB0C   INGET1  DB2C
INGET2  DB16   INITO   DA4F   INPBUF  DA4F   INPTRO  DAE4
INPTR1  DB3D   INTNO   0003   INTRY   DB19   IOBYTE  0003
IPBLUP  DAD4   LNKLEN  0007   LOCALL  DAE0   MOVPTR  DB4D
MSIZE   0038   OFFSET  4D00   PARLEN  F400   PRETN   DA30
PROST1  DA47   PRTOUT  DA3C   ROMTOP  EC00   UCNOUT  DA25
UCON1   DB3F   UCONIN  DB3C   UCSTAT  DB56   USER    DA00
USRLEN  0179   VERS    DA1D   Z80MHZ  0004

No  Fatal error(s)
```

```
                  ;
0000'                     ASEG              ;Absolute mode for Macro-80
                          .Z80              ;Sorry:  I like Zilog code!
                  ;
                  ;System-dependent constants:
                  ;
0005              BDOS    EQU     0005H             ;CP/M entry point
D300              BIOS    EQU     0D300H            ;Get from CP/M listing
D306              CONST   EQU     BIOS+6
                  ;       Note:   CP/M Function 11 (Console Status) does
                  ;               not work properly -- reason unknown.
0004              Z80MHZ  EQU     4                 ;Processor clock rate
0060              CBAUD   EQU     96                ;Input Baud/100
00C0              ECOUNT  EQU     Z80MHZ*4615/CBAUD        ;26-cycle loop.
                                                  ;  counter for 1.2-char. wait.
                  ;Locations in Wordstar:
                  ;
02BD              UCONI   EQU     02BDH             ;Special console input
02E0              CINSET  EQU     02E0H             ;Patch area MORPAT
7849              WSMEM   EQU     7849H             ;WS loc. MEMORY
035C              PRGMEM  EQU     035CH             ;WS pointer to MEMORY
0292              TRMINI  EQU     0292H             ;Terminal setup patch
3F00              CPLOAD  EQU     3F00H             ;Immed. above WS proper
2D08              WSTART  EQU     2D08H     ;Get via DDT from WS loc 101.2
                                            ;  (Before patching, of course!)
                  ;
                  ;Misc
                  ;
001B              ESC     EQU     1BH
                  ;
                  PAGE 62
```

Customize Your Keyboard continued . . .

Notes

1. The Z-19 keypad can be used unshifted for numbers and shifted as special keys, or in "alternate mode" as special keys both shifted and unshifted. I have chosen the latter since I don't enter that many numbers. By simply changing the translation table in this article, you could (1) use the keypad unshifted for numbers and shifted for whatever you desire, or (2) use function keys to change the keypad mode. Be my guest.

2. For North Star users: (1) install a header on socket 1A of the motherboard, with a jumper connecting pins 3 and 14; (2) break the links marked J9, J10, and J11 on the motherboard adjacent to sockets 1A and 2B. See the lower right-hand corner of sheet 1 of the motherboard schematics in your manual for position.

3. If you have a separate processor handling your disks, lucky you! You can read this solely to learn how the poor folk live.

```
                 ; --------------------------------------------------------
                 ; ¶                                                      ¶
                 ; ¶                   PATCH LOADER                       ¶
                 ; ¶                                                      ¶
                 ; --------------------------------------------------------
                 ;
                 ;
                 ; Each separate patch (of consecutive locations to be
                 ;   overlaid in Wordstar) is preceded by two words, the
                 ;   first giving the starting address for the patch and
                 ;   the second giving the patch length.    When a word
                 ;   starting with 00xxH is encountered in place of a
                 ;   loading address, loading ceases and a jump is made
                 ;   to 0100H.
                 ;
                 ; The loader and patches are then loaded above Wordstar
                 ;   proper.  A JP CPLOAD is inserted at locations
                 ;   100 - 102, and the patched version is saved.
                 ;
                         ORG     CPLOAD
3F00    21 3F14          LD      HL,PATCH1
3F03    5E       LDRLUP: LD      E,(HL)          ;Get patch destination
3F04    23               INC     HL
3F05    56               LD      D,(HL)
3F06    7A               LD      A,D             ;00xx means all patched
3F07    B7               OR      A               ;if so, then
3F08    CA 0100          JP      Z,100H          ;  start patched WS
3F0B    23               INC     HL
3F0C    4E               LD      C,(HL)          ;Get patch length
3F0D    23               INC     HL
3F0E    46               LD      B,(HL)
3F0F    23               INC     HL
3F10    ED B0            LDIR                    ;load this patch
3F12    18 EF            JR      LDRLUP
                 ;
                 ;PATCH 1:  Replace JP at 100, displaced to reach loader
                 ;
3F14    0100     PATCH1: DW      0100H           ;First patch destin.
3F16    0003             DW      3               ;length
                         .PHASE 0100H
0100    C3 2D08          JP      WSTART          ;Put back overwritten JP.
                         .DEPHASE
                 ;
                 ;PATCH 1A:  Initialize Z19 terminal
                 ;
                 ; This goes in std WS terminal-initalization space.
                 ;
3F1B    0292     PAT1A:  DW      TRMINI          ;Second patch destination
3F1D    0005             DW      5               ;length (for loader)
                         .PHASE TRMINI
0292    04               DB      4               ;length (for WS)
0293    1B 75            DB      ESC,'u'         ;Assure unshifted keypad
0295    1B 3D            DB      ESC,'='         ;Alternate keypad
                         .DEPHASE
                 PAGE 62
                 ; --------------------------------------------------------
                 ; ¶                                                      ¶
                 ; ¶                    TRANSLATOR                        ¶
                 ; ¶                                                      ¶
                 ; --------------------------------------------------------
                 ;
                 ;This patch enables use of the special keys on the Z19
                 ;  for cursor movement and other functions.  The precise
                 ;  functions assigned are determined by contents of
                 ;  the Translation Tables, which this program interprets.
                 ;
                 ;Conventions for patching Wordstar are given in Appendix
                 ;  E of the Wordstar User's Guide.  See Wordstar 3.0
                 ;  USER1 Listing, pages 14, 19.
                 ;
                 ;Method:  Each input request is received at UCONI.  A
                 ;  CALL to BDOS is made, and any normal character is
                 ;  passed on.  On receipt of an ESC, the next input
```

```
                 ;  character is read, and sought in List 1.
                 ;
                 ;  When an input escape sequence has been identified,
                 ;  the first character of the corresponding Wordstar
                 ;  sequence is returned.  Succeeding Wordstar input
                 ;  requests are given the rest of the sequence until
                 ;  it is complete.
                 ;
                 ;Special routines can be called in response to ESC
                 ;  sequences, either within or in place of output
                 ;  character sequences.
                 ;
                 ;
                 ;PATCH 2:  Jumps to console status and input routines
                 ;
3F24    02BA             DW      UCONI-3         ;Patch destination
3F26    0006             DW      6               ;Patch length
                         .PHASE  UCONI-3
02BA    C3 02E0          JP      CISTAT          ;Direct call to BIOS
02BD    C3 02EB          JP      CIN             ;Character-input routine
                         .DEPHASE
                 ;
                 ;PATCH 3:  Main Body of Translator
                 ;
3F2E    02E0             DW      CINSET          ;Load in WS MORPAT area
3F30    0075             DW      P3END-$-2       ;Patch length
                         .PHASE  CINSET
                 ;
                 ;Status check routine
                 ;
02E0    2A 02FE  CISTAT: LD      HL,(POINTR)
02E3    7E               LD      A,(HL)
02E4    B7               OR      A               ;Char. left?
02E5    FA D306          JP      M,CONST         ;  if not, check via CP/M
02E8    3E FF            LD      A,0FFH          ;  if so, send back sig.
02EA    C9               RET
                 PAGE 62
                 ;
                 ;Routine to service input request from Wordstar
                 ;
02EB    2A 02FE  CIN:    LD      HL,(POINTR)     ;Get list pointer
02EE    7E       CINRUN: LD      A,(HL)          ;Char from last time?
02EF    B7               OR      A
02F0    FA 0300          JP      M,READ1         ;  if not, go get one.
02F3    23       GOBAKI: INC     HL              ;Advance pointer
02F4    28 32            JR      Z,SPROUT        ;Zero -> special routine
02F6    22 02FE          LD      (POINTR),HL
02F9    E6 7F    GOBACK: AND     7FH             ;Trim parity bit.
02FB    28 03            JR      Z,READ1         ;Don't send back nulls!
02FD    C9               RET                     ;  And give char. to WS
02FE    7849     POINTR: DW      TABL1           ;Start out at "no char."
                 ;
                 ;Read input
                 ;
0300    CD 0346  READ1:  CALL    CHARIN          ;Get any waiting char.
0303    FE 1B            CP      ESC             ;Check for ESC
0305    C0               RET     NZ              ;  and return if it isn't
                 ;
                 ;Wait to see if this is a sequence, & if so service it
                 ;
0306    01 00C0          LD      BC,ECOUNT       ;Char.-time counter
0309    0B       ECLOOP: DEC     BC              ;Countdown.
030A    78               LD      A,B             ;  Could also be done
030B    B1               OR      C               ;  with CPI; JP PE,ECLOOP
030C    20 FB            JR      NZ,ECLOOP       ;(26-cycle loop)
030E    0E 0B            LD      C,11            ;CP/M Console Status
0310    CD D306          CALL    CONST           ;  Call
0313    B7               OR      A               ;Sequence still coming?
0314    3E 1B            LD      A,ESC           ;  (Get ESC, in case.)
0316    C8               RET     Z               ;No.  Return the ESC
0317    21 7849          LD      HL,TABL1        ;Set up search
031A    01 0038          LD      BC,TABL1L
```

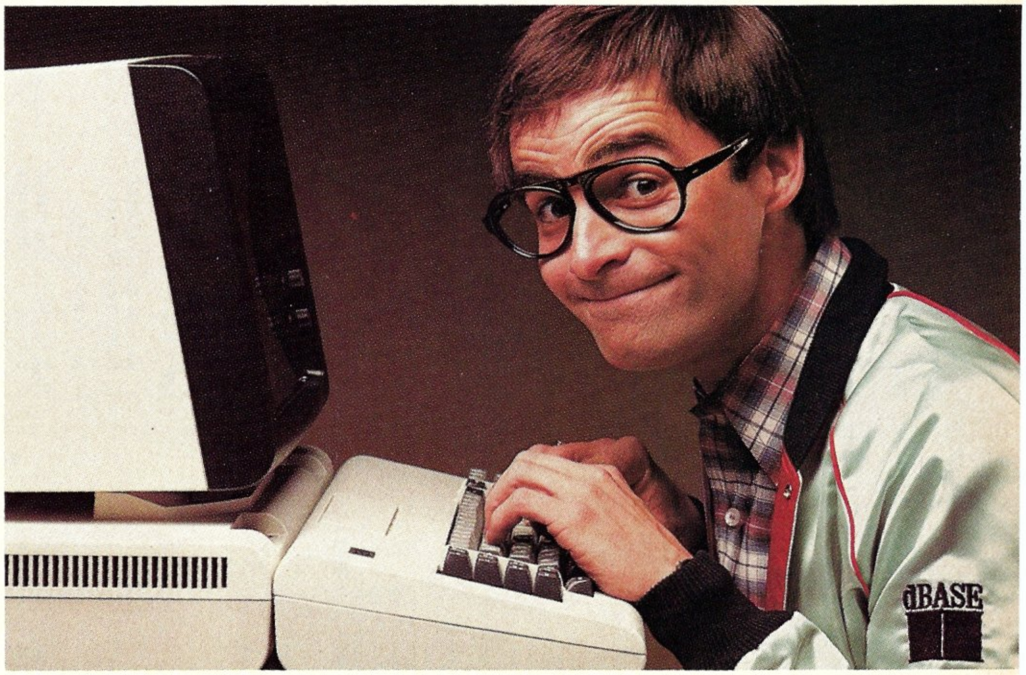
```
031D   CD 0346      TBLKUP: CALL    CHARIN          ;Get char. for search
0320   C6 80                ADD     A.80H           ;Add search char. marker
0322   ED B1                CPIR                    ;Search; if futile
0324   20 D3                JR      NZ.GOBACK       ;  send char. back
0326   18 C6                JR      CINRUN          ;Return with first char.
                                                    ;  of translation.
                    ;
                    ;Jump to special routine
                    ;
0328   7E           SPROUT: LD      A.(HL)          ;Use first byte after null
0329   32 032D              LD      (SPRJP+1),A     ;  to direct:
032C   18 00        SPRJP:  JR      SPRBAS          ;Special routine jump
032E                SPRBAS  EQU     $
                    ;
                    ;Routine to jump to secondary a list
                    ;
0000                SPRT    EQU     $-SPRBAS
032E   23                   INC     HL              ;Move ahead to get
032F   4E                   LD      C,(HL)          ;  length of second list
0330   18 EB                JR      TBLKUP
                    ;
                    ;Subscript. Superscript special routine
                    ;  This routine gets a single character and places it
                    ;  between °V's or °T's.  If a space is typed as the
                    ;  character, only the first °V or °T is produced.
                    ;
0004                SPR1    EQU     $-SPRBAS
0332   CD 0346              CALL    CHARIN          ;Get single character
0335   FE 20                CP      20H
0337   FA 02E0              JP      M.CINSET        ;Restart on ESC or Ctrl.
033A   20 02                JR      NZ,ONECHR       ;If it's a space,
033C   3E 80                LD      A.80H           ;plant a terminator
033E   23           ONECHR: INC     HL              ;Move ahead, to
033F   23                   INC     HL
0340   23                   INC     HL
0341   77                   LD      (HL),A          ;  put it in sequence
0342   2B                   DEC     HL              ;and move back to
0343   2B                   DEC     HL              ;  get first output char
0344   18 A8                JR      CINRUN
                    ;
                    ;Subroutine to input one char. using direct console I/O
                    ;
0346   E5           CHARIN: PUSH    HL
0347   C5                   PUSH    BC
0348   0E 06        CHARLP: LD      C,6             ;CP/M Dir.Console input
034A   1E FF                LD      E.0FFH
034C   CD 0005              CALL    BDOS
034F   B7                   OR      A               ;Set flags
0350   28 F6                JR      Z.CHARLP        ;Loop until char. present
0352   C1                   POP     BC
0353   E1                   POP     HL
0354   C9                   RET
                    ;
                            .DEPHASE
3FA7                P3END   EQU     $
                    ;
                    PAGE 56
                    ; ---------------------------------------------------
                    ; ¶                                                 ¶
                    ; ¶               TRANSLATION TABLES                ¶
                    ; ¶                                                 ¶
                    ; ---------------------------------------------------
                    ;
                    ;PATCH 4A:  Reset start of Wordstar working storage
                    ;
3FA7   035C                 DW      PRGMEM
3FA9   0002                 DW      2
                            .PHASE  PRGMEM
035C   78D6                 DW      TABLZZ+2
                            .DEPHASE
                    ;
```

```
                    ;PATCH 4:  Load in space stolen from WS text buffer
3FAD   7849                 DW      WSMEM
3FAF   008C                 DW      P4END-$-2
                            .PHASE  WSMEM
                    ;
                    ;Format for each item in TABL1 to TABL5:  Search char.
                    ;first, and has parity bit = 1.  Remaining char.s are
                    ;the translation to be sent back.  Sequence ends before
                    ;the next character with a set parity bit.
                    ;
                    ;If any character of translation is zero. next bit is
                    ;relative address of a special routine, which is called
                    ;at that time.
                    ;
0010                XP      EQU     16              ;for °P
0011                XQ      EQU     17              ;For °Q
000B                XK      EQU     11              ;for °K
                    ;
7849                TABL1:                  ;One-char. after ESC. no '?'
7849   C1 12                DB      80H+'A',18      ;Shift/up = page back
784B   C2 03                DB      80H+'B'.3       ;Shift/down = page forw'd
784D   C3 06                DB      80H+'C',6       ;Shift -> = word right
784F   C4 01                DB      80H+'D'.1       ;Shift <- = word left
7851   C0 16                DB      80H+'@',22      ;Insert Character toggle
7853   CF 16                DB      80H+'O'.22      ; ""
7855   CA 0B 19             DB      80H+'J'.XK,25   ;Erase = Delete block
7858   C5 0B 19             DB      80H+'E'.XK,25   ;Shift ERASE = Del.block
785B   CD 19                DB      80H+'M',25      ;Shift DL = Delete line
785D   CC 0E                DB      80H+'L'.14      ;Shift IL = Insert line
785F   D3 00 04 10          DB      80H+'S',0,SPR1.XP,22
7863   16
7864   00 10 16             DB        0.XP,22       ;f1 = single subscript
7867   D4 00 04 10          DB      80H+'T',0,SPR1,XP,20
786B   14
786C   00 10 14             DB        0.XP.20       ;f2 = superscript
786F   CE 11 1F             DB      80H+'N'.XQ,31   ;Shift DC = Del.to l.marg
7872   D2 0B 44             DB      80H+'R'.XK,'D'  ;White = Update file
7875   D1 0B 51             DB      80H+'Q'.XK,'Q'  ;Red = Abort, keep old.
7878   80 00 00 00          DB      80H,0.0.0.0     ;  --  space for more --
787C   00
787D   BF 00 00 28          DB      80H+'?'.0.SPRT.TABL2L
                                                    ;Advance to second list

0038                TABL1L  EQU     $-TABL1
                    ;
                    ;Some keys on the Z19 generate a sequence of the form
                    ;  ESC ? x .  These are handled via a special routine
                    ;  keyed by SPRT (see above).  The character x is then
                    ;  sought in TABL2 below.
7881                TABL2:                  ;'?' and one char. after ESC
7881   F4 13                DB      80H+'t',19      ;<--
7883   F6 04                DB      80H+'v'.4       ;-->
7885   F8 05                DB      80H+'x'.5       ;Cursor up
7887   F2 18                DB      80H+'r'.24      ;Cursor down
7889   F7 16                DB      80H+'w',22      ; = Insert character toggle
788B   F9 11 59             DB      80H+'y'.XQ,'Y'  ;DC = Delete rest of line
788E   F3 14                DB      80H+'s',20      ;DL = Delete word right
7890   F1 01 14             DB      80H+'q'.1.20    ;IL = Delete word left
7893   F0 0B 02             DB      80H+'p',XK,2    ;0 = mark block start
7896   EE 0B 0B             DB      80H+'n'.XK,11   ;. = mark block end
7899   CD 0B 16             DB      80H+'M',XK,22   ;ENTER = move block
789C   80                   DB      80H             ;  --  space for more --
789D                        DS      8
78A5   F5 00 00 04          DB      80H+'u'.0.SPRT.TABL3L
0028                TABL2L  EQU     $-TABL2
                    ;
                    ;The HOME key is used as leadin to a separate series of
                    ;  big cursor movements, via the last item above.  The ESC
                    ;  and ? generated by the key next struck are handled in
                    ;  TABL3 and TABL4, leading into TABL5.

                    TABL3:                                  ;Char. after HOME
```

```
78A9    9B 00 00 04                DB      80H+ESC,0,SPRT,TABL4L     ;ESC
0004                       TABL3L  EQU     $-TABL3
                           ;
78AD                       TABL4:                  ;Char. after HOME, ESC
78AD    BF 00 00 23                DB      80H+'?',0,SPRT,TABL5L     ;?
0004                       TABL4L  EQU     $-TABL4
                           ;
78B1                       TABL5:                  ;Char. after HOME. ESC. ?
78B1    F4 11 53                   DB      80H+'t',XQ,'S'   ;<-- = to left margin
78B4    F6 11 44                   DB      80H+'v'.XQ,'D'   ;--> = to line end
78B7    F8 11 45                   DB      80H+'x',XQ,'E'   ;UP = to screen top
78BA    F2 11 58                   DB      80H+'r'.XQ,'X'   ;DN = to screen bottom
78BD    F1 11 53 18                DB      80H+'q'.XQ.'S',24
                                                    ;IL = start of next line
78C1    F9 05 11 44                DB      80H+'y',5.XQ,'D';DC = end of prev. line
78C5    F0 11 42                   DB      80H+'p'.XQ,'B'   ; TO <B>
78C8    EE 11 4B                   DB      80H+'n',XQ,'K'   ; TO <K>
78CB    F5 11 50                   DB      80H+'u'.XQ,'P'   ;5-5 = last position
78CE    F7 11 52                   DB      80H+'w',XQ,'R'   ; To beginning of file
78D1    F3 11 43                   DB      80H+'s'.XQ,'C'   ; To end of file
0023                       TABL5L  EQU     $-TABL5
78D4    80                 TABLZZ: DB      80H
                           ;
                                   .DEPHASE
403D                       P4END   EQU     $
403D    0000                       DW      0000H
                                   END     CPLOAD

Macros:

Symbols:
BDOS    0005   BIOS    D300    CBAUD   0060    CHARIN  0346
CHARLP  0348   CIN     02EB    CINRUN  02EE    CINSET  02E0
CISTAT  02E0   CONST   D306    CPLOAD  3F00    ECLOOP  0309
ECOUNT  00C0   ESC     001B    GOBACK  02F9    GOBAKI  02F3
LDRLUP  3F03   ONECHR  033E    P3END   3FA7    P4END   403D
PAT1A   3F1B   PATCH1  3F14    POINTR  02FE    PRGMEM  035C
READ1   0300   SPR1    0004    SPRBAS  032E    SPRJP   032C
SPROUT  0328   SPRT    0000    TABL1   7849    TABL1L  0038
TABL2   7881   TABL2L  0028    TABL3   78A9    TABL3L  0004
TABL4   78AD   TABL4L  0004    TABL5   78B1    TABL5L  0023
TABLZZ  78D4   TBLKUP  031D    TRMINI  0292    UCONI   02BD
WSMEM   7849   WSTART  2D08    XK      000B    XP      0010
XQ      0011   Z80MHZ  0004


No  Fatal error(s)

                           TITLE USER W/KB INPUT INTERRUPT - VERS. M
                           ;USERIN                                   MARCH 28. 1982
                           ;
                           ; -------------------------------------------------------
                           ; ¶  USER AREA ROUTINES -- For Lifeboat BIOS for CP/M   ¶
                           ; ¶      on Northstar.  See Lifeboat Users'Notes        ¶
                           ; ¶                                                     ¶
                           ; ¶  These routines use interrupts for input from the   ¶
                           ; ¶  keyboard. for use with VEDIT and Wordstar, to      ¶
                           ; ¶  allow keyboard customization without glitches.     ¶
                           ; ¶                                                     ¶
                           ; ¶  Brad Thompson                                      ¶
                           ; ¶                                                     ¶
                           ; ¶  Developed and tested on:                           ¶
                           ; ¶    Northstar Horizon w/56k memory                   ¶
                           ; ¶    Two dual density double sided 5" floppy drives   ¶
                           ; ¶    Zenith Z19 terminal                              ¶
                           ; ¶                                                     ¶
                           ; ¶  This patch is copyrighted by the author.           ¶
                           ; ¶                                                     ¶
                           ; -------------------------------------------------------
                           ;
0038                       MSIZE   EQU     56              ;Memory size. kBytes
                           ;
```

```
D300                       BIOS    EQU     MSIZE*400H-0D00H
DA00                       USER    EQU     BIOS+700H
4D00                       OFFSET  EQU     2700H-USER       ;Use this with DDT -R
0003                       IOBYTE  EQU     3
                           ;
0000'                              ASEG                     ;Absolute segment
                                   .Z80                     ;Zilog Z80 coding
                                   ORG     USER-1           ;Origin of this program
                           ;
D9FF    10                         DB      10H              ;Interrupt-Enabled flag
                           ;
                           ;The following jump table is entered from the main body of
                           ;  BIOS.  The sequence of jumps must not be changed.
DA00    C3 DA4F            HCINIT: JP      INITO            ;Initialize devices. RAM
DA03    C3 DB56            HCONST: JP      UCSTAT           ;Status)
DA06    C3 DB3C            HCONIN: JP      UCONIN           ;Input )-Console port =
DA09    C3 DA25            HCONOU: JP      UCNOUT           ;Output)  left serial port
DA0C    C3 DA3C            HLIST:  JP      PRTOUT           ;Printer out: right port
DA0F    C3 DA3C            HPUNCH: JP      PRTOUT           ;  also right port out
DA12    C3 DA30            HREADR: JP      PRETN            ;Right port in
DA15    C3 DA47            HPRST:  JP      PROST1           ;Printer status: right pt.
DA18    00 00 00                   DB      0.0.0            ;Reserved
DA1B    0179                       DW      USRLEN           ;Length of this user seg.
                           ;
DA1D    48 42 54 2E        VERS:   DB      'HBT.1.14'       ;HBT's USER Version number
DA21    31 2E 31 34

                           ;
                           PAGE 60

                           ;
                           ;CONSOLE ROUTINES
                           ;
                           ;For input see interrupt handler
                           ;
DA25                       UCNOUT:                 ;Console (left) serial port output
DA25    DB 03                      IN      A.(3)            ;Free?
DA27    E6 01                      AND     1
DA29    CA DA25                    JP      Z.UCNOUT
DA2C    79                         LD      A.C
DA2D    D3 02                      OUT     (2),A
DA2F    C9                         RET

                           ;
                           ;PRINTER ROUTINES
                           ;
DA30                       PRETN:                           ;Printer return signals
                                                            ;Used for XON/XOFF

DA30    DB 05                      IN      A.(5)
DA32    E6 02                      AND     2
DA34    CA DA30                    JP      Z.PRETN
DA37    DB 04                      IN      A.(4)
DA39    E6 7F                      AND     7FH
DA3B    C9                         RET
                           ;
DA3C                       PRTOUT:                          ;Printer on right port
DA3C    DB 05                      IN      A,(5)
DA3E    E6 01                      AND     1                ;Free?
DA40    CA DA3C                    JP      Z.PRTOUT
DA43    79                         LD      A.C
DA44    D3 04                      OUT     (4),A
DA46    C9                         RET
DA47    DB 05              PROST1: IN      A.(5)            ;Right ser. out status
DA49    E6 01                      AND     1
DA4B    C8                         RET     Z                ;NO KEY
DA4C    3E FF                      LD      A.0FFH
DA4E    C9                         RET                      ;KEY
                           ;
                           ;SYSTEM INITIALIZATION ROUTINES
                           ;
DA4F                       INITO:
DA4F    DB 02                      IN      A.(2)            ;Start cleaning up
                           ;
                           ;First read. rewrite all RAM to set parity
                           ;
```

All you dBASE II™ hotshots are about to get what you deserve.



You've written all those slick dBASE II programs.

Business and personal programs. Scientific and educational applications. Packages for just about every conceivable information handling need.

And everybody who sees them loves them because they're so powerful, friendly and easy to use.

But that's just not good enough.

Uh-uh.

Because now you can get the gold and the glory that you really deserve.

Here's how.

We've just released our dBASE II RunTime™ application development module.

And it can turn you into an instant software publisher.

The RunTime module condenses and encodes your source files, protecting your special insights and techniques, so you can sell your code without giving the show away.

RunTime also protects your margins and improves your price position in the marketplace. If your client has dBASE II, all he needs is your encoded application. If not, all you need to install your application is the much less expensive RunTime module.

We'll tell the world.

With your license for the dBASE II RunTime module, we provide labels that identify your program as a dBASE II application, and you get the benefit of all the dBASE II marketing efforts.

We'll also provide additional "how to" information to get you off and running as a software publisher sooner.

And we'll make your products part of our Marketing Referral Service. Besides putting you on our referral hotline, we'll publish your program descriptions and contact information in *dBASE II Applied*, a directory now in computer stores world-wide.

Go for it.

But we can't do any of this until we hear from you.

For details, write RunTime Applications Development, Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230.

Or better yet, just call (213) 204-5570. And get what you deserve today.

dBASE™ II

ASHTON·TATE

CIRCLE 13 ON READER SERVICE CARD

```
EC00                ROMTOP  EQU     0E800H+400H     ;First address past ROM
F400                PARLEN  EQU     0F400H          ;No. of locs to set
DA51    21 EC00             LD      HL,ROMTOP
DA54    54                  LD      D,H
DA55    5D                  LD      E,L
DA56    01 F400             LD      BC,PARLEN
DA59    ED B0               LDIR                    ;Set all parity bits
DA5B    3E 41               LD      A,41H
DA5D    D3 C0               OUT     (0C0H),A        ;Arm parity error check
                    ;Action on parity error depends on wiring.  See memory
                    ;instructions
                    ;
                    ;Initialize IOBYTE
DA5F    3E 81               LD      A,81H           ;LIST-->LPT, CONSOLE-->CRT
DA61    32 0003             LD      (IOBYTE),A      ;  See CP/M Alt. Guide p.16
DA64    AF                  XOR     A               ;Reset USARTS
DA65    D3 06               OUT     (6),A

                    ;
                    ;Set up ports
DA67    F3                  DI                      ;Disable interrupts
DA68    3E 4E               LD      A,4EH           ;For 1 stop bit
DA6A    D3 03               OUT     (3),A           ;To first serial port
DA6C    D3 05               OUT     (5),A           ;To second serial port
DA6E    3E 37               LD      A,37H           ;See INTEL 8251 data sheet
DA70    D3 03               OUT     (3),A           ;First port
DA72    D3 05               OUT     (5),A           ;Second port
DA74    DB 02               IN      A,(2)           ;Clear input buffers
DA76    DB 04               IN      A,(4)           ;  on both serial ports
DA78    3E 30               LD      A,30H           ;Reset parallel port
DA7A    D3 06               OUT     (6),A           ;  PI flag
DA7C    3E 60               LD      A,60H           ;  PO flag
DA7E    D3 06               OUT     (6),A

                    ;
                    ;Initialize the H19
DA80    21 DA95             LD      HL,H19SET       ;Text for setup
DA83    06 1C               LD      B,H19STL        ;Length of text
DA85    0E 02               LD      C,2             ;Port
DA87    DB 03       H19LUP: IN      A,(3)
DA89    E6 01               AND     1
DA8B    28 FA               JR      Z,H19LUP        ;Wait until ready
DA8D    ED A3               OUTI                    ;Send
DA8F    20 F6               JR      NZ,H19LUP       ;Message done?
DA91    DB 02               IN      A,(2)           ;One last clean-up
DA93    18 23               JR      CINSET          ;Set up interrupt
                    ;
                    ;Text for H19 initialization
                    ;
DA95    00 00 00    H19SET: DB      0,0,0           ;Spare
DA98    1B 45               DB      1BH,'E'         ;Clear screen on boot
DA9A    42 49 4F 53         DB      "BIOS version HBT 1.14",0DH,0AH
DA9E    20 76 65 72
DAA2    73 69 6F 6E
DAA6    20 48 42 54
DAAA    20 31 2E 31
DAAE    34 0D 0A
001C                H19STL  EQU     $-H19SET
                    ;
                    PAGE 60

                    ; -------------------------------------------------------
                    ; ¶                                                     ¶
                    ; ¶        INTERRUPT-DRIVEN CONSOLE INPUT ROUTINE       ¶
                    ; ¶                                                     ¶
                    ; -------------------------------------------------------
                    ;
                    ;The following are defined to match the system and
                    ;  hardware:
0003                INTNO   EQU     3       ;No. of interrupt wired for
                                            ;  console input
0018                CLINKA  EQU     INTNO*8 ;Interrupt Link Address
0002                CIPORT  EQU     2       ;Console input port
0003                CSPORT  EQU     3       ;Console status port
0002                CSMASK  EQU     2       ;Input char. ready bit
```

```
0060                CBAUD   EQU     96      ;Input baud rate (in 100Hz units)
0004                Z80MHZ  EQU     4       ;Processor clock rate
0061                CHRLEN  EQU     Z80MHZ*2330/CBAUD ;51-cycle loop, counter
                                            ;  for 1.2-character wait.
001B                ESC     EQU     1BH
                    ;
                    ;The following describe the only DI ...... EI segment in
                    ;  the Lifeboat BIOS:
D7FF                DROUTN  EQU     BIOS+4FFH ;Target of (transplanted) CALL
D7AA                DCALLA  EQU     BIOS+4AAH ;Address of same
                    ;
DA4F                INPBUF  EQU     INITO   ;Input buffer origin.  If this
                    ;  routine follows immediately after the BIOS once-only
                    ;  initialization segment. INPBUF can be set to the first
                    ;  byte of that segment.  Otherwise, substitute
                    ;          INPBUF  EQU     CLINK
                    ;  This will provide a 27-byte buffer, which is normally
                    ;  more than adequate.  Alternatively. the buffer can be
                    ;  placed between this patch and USER+1FFH.
                    ;
                    ;Developed and tested on 56K N* Horizon w/two quad dens.
                    ;  drives; Lifeboat CP/M; Zenith Z19 terminal.
                    ;
                    ;The following interrupt links are planted on page 0, as
                    ;  used in Z80 Mode 0.
                    ;
DAB1                CLINK:
                            .PHASE  CLINKA
0018    F5                  PUSH    AF      ;Console input interrupt link
0019    C5                  PUSH    BC
001A    D5                  PUSH    DE
001B    E5                  PUSH    HL
001C    C3 DAE3             JP      CONSRV
                    .DEPHASE
                    ;
0007                LNKLEN  EQU     $-CLINK
                    ;
007E                BUFLEN  EQU     INBEND-INPBUF   ;Console buffer length
                    ;
                    PAGE 60
                    ;
                    ;INITIALIZATION:
                    ;
                    ;  Must be executed during BIOS initialization. before
                    ;  first console input
                    ;
                    ;Plant interrupt link
                    ;
DAB8    01 0007     CINSET: LD      BC,LNKLEN       ;Enter here
DABB    11 0018             LD      DE,CLINKA
DABE    21 DAB1             LD      HL,CLINK
DAC1    ED B0               LDIR                    ;Put link in place
                    ;
                    ;Divert CALL to disk routine via handler below.
                    ;
DAC3    21 DAE0             LD      HL,LOCALL       ;Get CALL to local routine
DAC6    11 D7AA             LD      DE,DCALLA       ;Put in where disk
DAC9    0E 03               LD      C,3             ;  call was
DACB    ED B0               LDIR
                    ;
DACD    21 DA4F     INBEND: LD      HL,INPBUF       ;Clear out input buffer
DAD0    01 007E             LD      BC,BUFLEN
DAD3    AF                  XOR     A
DAD4    77          IPBLUP: LD      (HL),A          ;Clear one byte
DAD5    ED A1               CPI                     ;  and count
DAD7    EA DAD4             JP      PE,IPBLUP
                    ;
DADA    36 FF               LD      (HL),0FFH       ;Place Buffer End Marker
DADC    ED 46               IM      0               ;Interrupt Mode 0
DADE    FB                  EI                      ;And cock the trigger!
DADF    C9                  RET                     ;FINAL return from INIT.
                    ;
```

EXTRA

EXTRA

S-100 World News

MACROTECH International Corporation 22133 Cohasset Street, Canoga Park, California 91303 • 213-887-5737

Image achieved by DGS' CAT 1600 Series color video graphic workstation. Picture courtesy of Digital Graphic Systems, Inc. See story below.

GRAPHICS: NOW MAX-IMIZED

CANOGA PARK—March 30, 1983—The decreasing costs and increasing density of memory made possible the present boom in digital graphics. Graphic systems designers are now able to take another major step with the introduction of MAX-M, a one megabyte memory board for $1983. As large size system memory and multi-megabyte Virtual Disk, MAX-M opens up major new low cost implementations.

Wayne Maw, Director of R&D for RGB Dynamics, Salt Lake City, Utah, reports, "My application is dependent on speed. With the Macrotech dynamic board, I have the needed speed." The RGB system is a Z80-based, high resolution color directory system for shopping malls, due for April release.

Empirical Research Group of Kent, Washington, creates a state-of-the-art high resolution color video graphics system by integrating their fast 68000 computer, Macrotech system memory, and the color video image processor from Digital Graphic Systems, Inc., Palo Alto, California. Radcliffe Goddard of Digital Graphics states, "High speed image processing requires large system memory to provide instantaneous display frame paging."

The demand for MAX-M by the graphics industry was nearly instantaneous following the initial Macrotech announcement.

MAX—256K to 1M S-100 Memory

CANOGA PARK—March 30, 1983—Mike Pelkey, Macrotech International president, today released details of the revolutionary MAX line of S-100 memory boards. Pelkey stated: "IEEE-696 now has a new standard for dynamic memory. The MAX product line offers 256K to 1M, at a price that ranges down to less than $0.00023 per bit." Pelkey continued, "The MI product line now includes our ultra fast (70 ns) 128K static memory, with battery backup capability, plus the 150 ns dynamic memories—in every 128K step from 256K through 1M (1024K) bytes, and add-on kits to permit field upgrade of sizes."

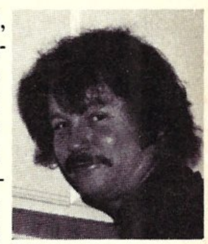
The extreme density of the MAX family is made possible through the use of proprietary PALs (programmable array logic). Also stated as available for add-on to any size MAX is Macrotech's popular M³ memory mapping architecture. M³ permits the 16-bit address space of an 8-bit processor to be dynamically mapped in 4K pages into as much as 16 megabytes of physical memory.

Parity error detection and 8/16 bit data transfer capabilities are provided as standard on the MAX series memory board.

PRICE INDEX

	SIZE	P/N	PRICE
Static Memory	128K	128-ST	$1232
Dynamic Memory	256K	MAX-256	$1108
24-bit	384K	MAX-384	1292
Addressing	512K	MAX-512	1647
	768K	MAX-768	1815
	896K	MAX-896	1899
	1M	MAX-M	1983
With 16-bit M³ Addressing option, add $91			
	FROM/TO	**P/N**	**PRICE**
Upgrade Kits	256K/384K	MKT-2/3	$ 192
	256K/512K	MKT-2/5	692
	256K/768K	MKT-2/7	876
	256K/896K	MKT-2/8	967
	256K/1M	MKT-2/M	1060
	384K/512K	MKT-3/5	600
	384K/768K	MKT-3/7	784
	384K/896K	MKT-3/8	876
	384K/1M	MKT-3/M	968
	512K/768K	MKT-5/7	284
	512K/896K	MKT-5/8	376
	512K/1M	MKT-5/M	468
	768K/896K	MKT-7/8	192
	768K/1M	MKT-7/M	284
	896K/1M	MKT-8/M	192
M³ option		MKT-M3	121
Software (provided on 8″ disk) Virtual Disk for MP/M II* and CP/M 2.2, CP/M 3.0* Bios modules, CP/M memory tests			$ 25
Manuals (sold separately)			
128/ST			$ 15
MAX Technical Manual			15

Software for M³ Available



BURBANK—March 30, 1983—"M³ bank switching for 8-bit processors is much more useful with the new creative systems programs," states Dan West of Westcom Systems Inc. MP/M II* disk intensive applications are greatly improved with the new Virtual Disk routines now available through Macrotech OEM's and dealers for their M³ memory boards.

Westcom Systems, as the software consulting firm for Macrotech, has also provided subroutine listings to easily incorporate M³ mapping into the new CP/M 3.0* (CP/M Plus*) Bios module. The advantages of CP/M 3.0* with disk buffering, hashed directories, and user program expansion go hand in hand with Macrotech's flexible "bank switched" memory capabilities.

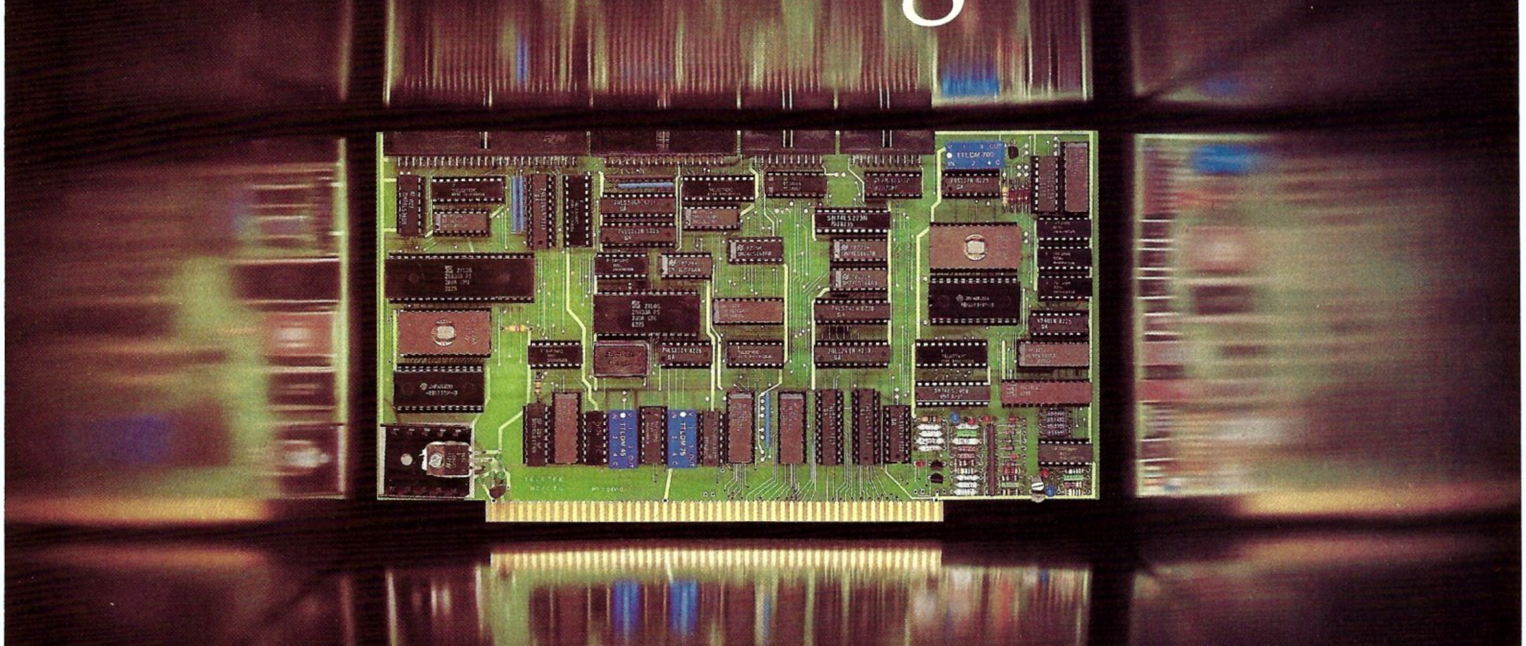
All Macrotech software and manuals are available through Dan West's Compuserve account #70250,102.
Leave comments/questions as E-Mail.

These new techniques can combine the above features with custom needs of the future, such as printer buffering, multi-page display and memory-intensive graphics displays.

The software listings are included in the Macrotech memory board manuals and are optionally available on 8″ diskettes.

```
DAE0   CD DB5F      LOCALL: CALL    DRSET           ;To be transplanted
                    ;
                    ;Routine to zero the console input buffer.  Used here
                    ;  and during disk operations.
                    ;
                    ;
                    PAGE 60
                    ;
                    ;CONSOLE INPUT INTERRUPT SERVICE ROUTINE
                    ;
DAE3   21 DA4F      CONSRV: LD      HL,INPBUF       ;Get buffer pointer
DAE4                INPTRO  EQU     CONSRV+1        ;  Address altered: points
                                                    ;    to first vacant slot
DAE6   01 0000              LD      BC,0            ;Zero ESC SEQ counter
DAE9   CD DB2C              CALL    INGET1          ;Get the waiting char.
DAEC   FE 1B                CP      ESC             ;If it's not an ESC
DAEE   C2 DB0C              JP      NZ,INDONE       ;  we're done
DAF1   CD DB16              CALL    INGET2          ;Wait for and get the next
DAF4   FE 3F                CP      '?'             ;If it's a '?'
DAF6   CC DB16              CALL    Z,INGET2        ;  wait for, get the last
DAF9   79           INDON1: LD      A.C             ;Compare no. of char. read
DAFA   B8                   CP      B               ;  vs stored
DAFB   CA DB0C              JP      Z,INDONE        ;  If equal. we're done.
DAFE   2A DAE4      ESCZAP: LD      HL,(INPTRO)     ;Otherwise go back,
DB01   AF           ESCZLP: XOR     A               ;  and zero out the seq:
DB02   CD DB4D              CALL    MOVPTR
DB05   0D                   DEC     C               ;C = No. of char. stored
DB06   C2 DB01              JP      NZ,ESCZLP
DB09   C3 DB0F              JP      INDON2
                    ;
DB0C   22 DAE4      INDONE: LD      (INPTRO).HL     ;Reset input pointer
DB0F   E1           INDON2: POP     HL
DB10   D1                   POP     DE
DB11   C1                   POP     BC
DB12   F1                   POP     AF
DB13   FB                   EI
DB14   ED 4D                RETI
                    ;
                    ;SUBROUTINE TO TAKE. STORE ONE INPUT CHARACTER
                    ;
DB16                INGET2:                 ;Enter here to time possible 2nd
                                            ;or 3rd entry in an ESC sequence
DB16   11 0061              LD      DE.CHRLEN       ;For 1.2-char. time loop
DB19   DB 03        INTRY:  IN      A.(CSPORT)      ;Read port status
DB1B   E6 02                AND     CSMASK          ;Check in-ready bit
DB1D   20 0D                JR      NZ,INGET1       ;  If ready. go get it
DB1F   1B                   DEC     DE              ;Count down to time
DB20   7A                   LD      A.D             ;Check the countdown
DB21   B3                   OR      E               ;  (0 only if D and E 0)
DB22   20 F5                JR      NZ, INTRY
DB24   D1                   POP     DE              ;Discard calling addr.
DB25   05                   DEC     B               ;Is B=1?
DB26   CA DB0C              JP      Z.INDONE        ;  Yes.  Leave the ESC.
DB29   C3 DAFE              JP      ESCZAP          ;  No. Zap the ESC Seq.
                    ;
DB2C   DB 02        INGET1: IN      A.(CIPORT)      ;Read the waiting char.
DB2E   04                   INC     B               ;Count char's read
DB2F   E6 7F                AND     7FH             ;Scrap the parity bit
DB31   57                   LD      D,A             ;  and store in D
DB32   7E                   LD      A.(HL)          ;Is there space in
DB33   B7                   OR      A               ;  the buffer?
DB34   7A                   LD      A.D             ;Retrieve character
DB35   C0                   RET     NZ              ;No.
DB36   0C                   INC     C               ;Yes. Count char's stored
DB37   CD DB4D              CALL    MOVPTR          ;Store character
DB3A   7A                   LD      A.D
DB3B   C9                   RET
                    PAGE 60
                    ;
                    ;CONSOLE INPUT FOR CP/M FROM BUFFER
                    ;
DB3C   21 DA4F      UCONIN: LD      HL,INPBUF       ;Char. waiting in buffer?
DB3D                INPTR1  EQU     UCONIN+1        ;  address changes, is
                                                    ;  buffer-unload pointer
DB3F   7E           UCON1:  LD      A,(HL)
DB40   B7                   OR      A               ;  Set flags
DB41   28 FC                JR      Z,UCON1         ;Wait for char.via int'rpt
DB43   4F                   LD      C,A
DB44   AF                   XOR     A               ;00 to clear buffer slot
DB45   CD DB4D              CALL    MOVPTR          ;Store, advance HL pointer
DB48   22 DB3D              LD      (INPTR1),HL     ;  and store it too.
DB4B   79                   LD      A.C
DB4C   C9                   RET
                    ;
                    ;MOVPTR:  BUFFER POINTER ADVANCE. DATA STORE SUBROUTINE
                    ;
                    ;  Used during interrupt to advance INPTRO, store data
                    ;  Also when deleting partial ESC seq. which cannot be
                    ;     completed because buffer is full
                    ;  Used during fetch (above) to advance INPTR1, store
                    ;    zeros to replace digits fetched
                    ;
DB4D   77           MOVPTR: LD      (HL),A          ;Feed buffer (data or 00)
DB4E   23                   INC     HL              ;Move up the pointer
DB4F   7E                   LD      A.(HL)          ;Look for 0FFH
DB50   3C                   INC     A               ;  at end of buffer
DB51   C0                   RET     NZ
DB52   21 DA4F              LD      HL,INPBUF       ;If so, set back to start
DB55   C9                   RET
                    ;
                    ;CONSOLE INPUT STATUS FOR CP/M
                    ;
DB56   2A DB3D      UCSTAT: LD      HL,(INPTR1)     ;Check buffer contents
DB59   7E                   LD      A,(HL)
DB5A   B7                   OR      A
DB5B   C8                   RET     Z               ;No key
DB5C   3E FF                LD      A.0FFH
DB5E   C9                   RET                     ;Key
                    ;
                    PAGE 60
                    ;
                    ;ROUTINE TO ELIMINATE INTERRUPT INTERFERENCE WITH DISK
                    ;
                    ;       (And Vice Versa)
                    ;
                    ;This intercepts BIOS disk-handlers that operate with
                    ;  the interrupt off.
                    ;
DB5F   CD D7FF      DRSET:  CALL    DROUTN          ;FILLED (See above)
                    ;
                    ;Before enabling interrupt. check for characters waiting
                    ;  and coming in.  Any found may be the end of an ESC
                    ;  sequence. and are discarded.  The constant CHRLEN
                    ;  will cause a wait of 1.3 character periods after last
                    ;  character.
                    ;
DB62   F5                   PUSH    AF
DB63   C5                   PUSH    BC
DB64                EILOOP:                         ;Wait 1.3 char. periods:
DB64   01 0061              LD      BC,CHRLEN       ;Load no. of counts
DB67   DB 03        DELOOP: IN      A,(CSPORT)      ;Get KB status
DB69   E6 02                AND     CSMASK          ;  and test for char.
DB6B   28 04                JR      Z,DEL2
DB6D   DB 02                IN      A.(CIPORT)      ;If so. take care of it
DB6F   18 F3                JR      EILOOP          ;  and check for another
DB71   0B           DEL2:   DEC     BC              ;Count down one
DB72   78                   LD      A,B
DB73   B1                   OR      C               ;Any bit in BC?
DB74   20 F1                JR      NZ.DELOOP       ;If so, count on
DB76   C1                   POP     BC
DB77   F1                   POP     AF
DB78   C9                   RET
                    ;
0179                USRLEN  EQU     $-USER
                    ;
                            END     USER
```

Bored Waiting? Here's The Board You've Been Waiting For.



A hard disk and cartridge tape controller together on one board? Magic? Not really. It's Teletek's HD/CTC. The hard disk and cartridge tape drive controller provide the support necessary to interface both rigid-disk drives and a cartridge tape deck to the S-100 bus.

- A Z-80A CPU (optionally Z-80B) providing intelligent control of the rigid-disk and cartridge tape drives.
- Support of 5¼" rigid-disk drives with transfer rates of 5 megabits per second. Minor changes of the on-board components allow the support of other drive types/sizes and transfer rates up to 15 megabits per second. (Interface to disk drive is defined by software/firmware on-board.)
- Controller communications with the host processor via 2K FIFO at any speed desirable (limited only by RAM access time) for a data block transfer. Thus the controller does not constrain the host processor in any manner.
- Two 28-pin sockets allowing the use of up to 16K bytes of on-board EPROM and up to 8K bytes of on-board RAM.
- Individual software reset capability.
- Conforms to the proposed IEEE-696 S-100 standard.
- Controller can accommodate two rigid-disk drives and one cartridge tape drive. Expansion is made possible with an external card.

Teletek's HD/CTC Offers A Hard Disk Controller, Plus Cartridge Tape Controller, All On One Board.

TELETEK

9767F Business Park Drive Sacramento, CA 95827 (916) 361-1777 Telex #4991834 Answer back-Teletek

CIRCLE 220 ON READER SERVICE CARD

GENESIS COMPUTER CORPORATION

introduces...

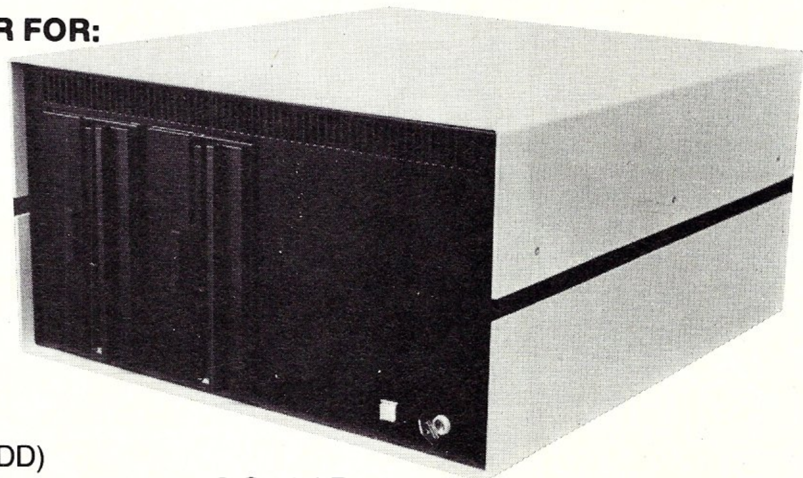
the Price & Performance Leader

the GENESIS 8D S-100 micro

A COMPLETE DUAL 8" DISK DRIVE S-100 MICROCOMPUTER FOR ONLY... **$2395.00**

THE IDEAL MICROCOMPUTER FOR:

★ Business
★ Education
★ Industry
★ Development
★ Networking



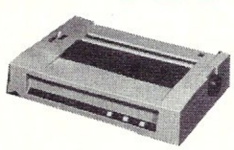
FEATURES:

★ Z80A CPU 4MHz
★ 64K Dynamic RAM
★ 2 1.024 MB 8" disk drives (DSDD)
★ Floppy Disk Controller
★ CPM 2.2© Operating System (MPM© and TURBODOS® Available)
★ 10 Slot IEEE 696 Mainframe
★ 2 Serial Ports
★ 2 Parallel Ports
★ DMA
★ Keylock ON/OFF Switch

COMPLETE THE SYSTEM WITH OUR INEXPENSIVE TERMINAL AND PRINTER COMBINATION FOR ONLY

$925

ADDS VIEWPOINT



C. ITOH PROWRITER

GENESIS COMPUTER CORPORATION

1444 LINDEN ST.
P.O. BOX 1143
BETHLEHEM, PA 18018
(215) 861-0850

SCHOOLS AND BUSINESSES ASK ABOUT OUR LEASE/PURCHASE ARRANGEMENTS

© Registered Trademark Digital Research

® Registered Trademark Software 2000

CIRCLE 173 ON READER SERVICE CARD

WordStar Arrow Keys for the H-19 Terminal

by Norman Dresner

While my own computer was being serviced, I borrowed a friend's computer, which uses an H-19 terminal. I could tolerate the 9600-baud output from WordStar, but I was homesick for my own keyboard, which has WordStar-compatible arrow keys. The code patch given here will allow the direct use of the arrow keys on the H-19's numeric keypad.

Only the arrow key functions are implemented—all other keypad editing functions are ignored by this code. These could be implemented by extending the length of the patch, but I needed the patch area for printer-related code.

The terminal initialization string (TRMINI) puts the H-19 into the shifted-keypad mode. A side effect of this is that the numbers 1-9 on the auxiliary numeric keypad keys can only be accessed by simultaneously depressing the shift key. Since these keys are dual use, only one function can be used without the shift key, and I wanted the arrow keys. The terminal deinitialization string (TRMUNI) restores the H-19 to the normal mode.

The user-input routine (OURINP) accepts a character from the console. If it is not an ESC, it is passed to WordStar immediately. If it is an ESC, the code waits for a second character, assuming that either a special function key or a shifted keypad key has been depressed. Since the first ESC is always ignored, to enter an ESC when WordStar demands one, e.g., after a ^U, it is necessary to hit the ESC key twice.

To install the patch, enter the program shown in Listing 1. If you use WordStar, you can't use the arrow keys yet. Assemble it, using ASM. Install the patch and save the modified WordStar program as follows:

(x:)DDT (y:)WS.COM	x&y drives as appropriate
...	DDT's ID
-IAKPAT.HEX	identify patch file. This assumes it is located on the default disk.
-R	read the patch in
-^C	exit to CP/M
z>SAVE 56 (y:)WS.COM	create new file with patch

The method used here is applicable to any terminal that transmits a unique code, or sequence of characters, for the arrow keys. If no other user patches are installed in the WordStar patch area (MORPAT), other H-19 keys could also be decoded. If more space is needed, the user can SYSGEN a CP/M system 1K or 2K smaller, and use the terminal initialization code (INISUB @ 02Ah) to jump to a routine in the WordStar patch area that will read the extended patch into memory above the CP/M system. This technique is preferable to the MicroPro recommended approach of installing the patch at the end of the WordStar code and overlay space, because this greatly enlarges the size of the .COM save file.

The requirement to hit the ESC key twice could be eliminated by a timing loop after the first byte to determine whether a human hit an ESC key or whether the terminal is transmitting the first character of an ESC sequence, but at the expense of a longer patch.

Norman Dresner, CompuMagic, Inc., Box 780, Severn, MD 21144

Note: H-19 is a product of Heath/Zenith; WordStar is a registered trademark of MicroPro International Corp.

```
WordStar Arrow Keys for the H-19 Terminal

LISTING 1.  WordStar patch AKPAT.ASM for H-19 terminal arrow keys
;
; Patch for WordStar version 2.26 to allow H-19 Arrow Keys to be
; used for cursor positioning.  The technique is to use the
; Shifted Key-Pad Mode to get automatically the H-19 arrow
; keys.  However, these are ESC sequences, not the WS-required
; arrow keys.  Code for translation is used and linked to WS
; at the User CONsole Input routine.
;   The technique is simple.  If a character is received which
; is not an escape, it is returned to WS unmolested.
;   If an ESC is received, the subroutine waits for the next
; character.  If it matches one of the AK's, it is translated
; and given to WS.
;   However, if another ESC-code is received, a bell is rung
; and the subroutine waits for a valid character.
;
ESC     EQU     01BH
UCA     EQU     041H
BDOS    EQU     00005H
CONIN   EQU     001H
DIRIO   EQU     06H
CONOUT  EQU     002H
BELL    EQU     007H
        ORG     0292H           ; Terminal Initialization String
TRMINI: DB      03              ; # of bytes
```

```
                DB      ESC,'x'  ; set terminal for shifted
                DB      '6',0    ; keypad
;               DB      0,0,0    ; nulls for later
                DB      0
TRMUNI:         ORG     029BH    ; Terminal Termination String
                DB      03       ; # of bytes
                DB      ESC,'y6' ; reset to normal keypad
                ORG     02BDH    ; User Console Input Routine
                JMP     OURINP   ; go to code extension
                ORG     02E0H    ; User Patch Area==MORPAT
OURINP:         CALL    GETCON   ; get console input
                CPI     ESC      ; is it an ESC code
                RNZ              ; if not, exit
                CALL    GETCON   ; get next character
                CPI     ESC      ; does he really mean it
                RZ
                SUI     UCA      ; reduce by "A"
                JC      INPERR   ; if n.g., diagnose
                CPI     04       ; upper-limit+1 of acceptable
                JNC     INPERR   ; diagnose if n.g.
                LXI     H,XTAB   ; translate it to WS
                MOV     E,A
                MVI     D,0
                DAD     D
                MOV     A,M
                RET              ; exit smiling
```

WordStar Arrow Keys for the H-19 Terminal

```
GETCON:         MVI     E,0FFH   ; we want input
                MVI     C,DIRIO  ; naked, no less
                CALL    BDOS
                ORA     A        ; do we have one?
                RNZ              ; yes
                JMP     GETCON   ; no, try again
INPERR:         MVI     E,BELL
                MVI     C,CONOUT
                CALL    BDOS
                JMP     OURINP
;
;   TRANSLATE TABLE          ^E ,^X ,^D ,^S
;
XTAB:           DB      05H,18H,04H,13H
                END
```

More WordStar Mods for the Z-19

Other enhancements for the Heath/Zenith Z-19 CRT terminal

by Bill Machrone

This program modifies WordStar to use the Heath/Zenith Z-19's function keys and the keypad for greater ease in editing. The 25th line is used to explain the use of each of the function keys. The keypad is put into its shifted state and its functions are equivalent to the legends on the keys.

This patch method uses WordStar's normal keyboard entry code and modifies the keycode dispatch tables internal to WordStar. Each dispatch is a four-byte table entry:

1: The first stroke of a multicharacter command or a single-character command
2: The second stroke of a multicharacter command or a null
3, 4: Actual address of the routine in memory, or a displacement into the overlay file.

There are some limitations to this technique, the first being that no key (except Escape) can have two meanings, both as a lead-in character and a one-character instruction. Also, there is a limit to how fast WordStar can look up and dispatch the entries generated by the keyboard. Thus, I've defined the arrow keys on the keypad to mean "move to the limits of the screen or line," equivalent to ^QE, ^QS, ^QD and ^QX. WordStar, without a modified input routine, cannot handle the two characters generated by the arrow keys at the repeat rate, which would be desirable for character-at-a-time cursor movement.

I don't consider this a drawback, as I prefer not removing my hands from the typing position to do cursor positioning. WordStar's standard "cursor-control diamond" arrangement is fine with me, while the right arrow key is probably the one I use most, returning me to the end of the line after I have backed up to correct errors or insert words. The other keys on the keypad perform the closest analogue to their Z-19 function: "IC" toggles the insert modes, "DC" is the same as pressing ^G, "IL" inserts a blank line, and "DL" deletes a line as incontrovertably as ^Y. I was not able to implement an equivalent to "home," so that key simply repeats the cursor to top of screen function. The functions performed by the top row of keys are defined under XTAB in the listing.

A quirk in the Z-19's control program makes the keypad a little tricky: If you inadvertently press the control key and one of the keypad functions, the defined function is performed *locally* by the terminal and no transmission is made to WordStar. Thus, if you press control and IC/7, the terminal goes into its local insert mode, and WordStar's insert mode is not toggled on or off. The resulting screen will make you think your system is ready for the scrap heap. Actually, your file isn't affected—you can close it with a ^KX, reset the terminal, and start over.

Bill Machrone, 121 North Avenue, Fanwood, NJ 07023

```
title 'Z19PAT3  Terminal initialization for WordStar 3.XX'
; By:
;       Bill Machrone
;       121 North Avenue
;       Fanwood, NJ 07023

; Version 1.1 03-Nov-80.  Puts indent and find on line 25.
; Arrow keys on keypad move cursor to extremes on screen
; rather than one position.

; Version 2.0 02-Mar-81.  New addresses for WS version
; 2.26.  Minor corrections to 1.1

; Version 3.0 11-Nov-81.  New addresses for WS version
; 3.0.  Disable 25th line on Z19 at end of session.

; Attributes for Heath/Zenith Z19 terminal, so that
; function keys defined in WS patch area are explained
; on the 25th line of the screen.
; Code below is intended to be assembled, then
; overlaid onto installed WS.COM with DDT or equivalent.

FALSE   equ     0
TRUE    equ     not FALSE
ESC     equ     1bH
BDOS    equ     0005

REVID   macro           ;Reverse video.
        db      ESC,'p'
        endm

REVOFF  macro           ;Reverse video off.
        db      ESC,'q'
        endm

SPACE   macro
        db      ' '
        endm

CURSOR  macro col, line
        db      ESC,'Y'
        db      line + 31, col + 31
        endm

LDIR    macro           ;Save the bother of Z80.LIB
        db 0EDH,0B0H
        endm

; The extra key dispatches are placed in XTAB and they
; also overlay get/set user place markers 6-9.  Anyone
; who tries to keep more than 5 place markers straight at
; one time should probably have his head examined, but
; thanks anyway, MicroPro.

MEMORY  equ     3800H   ;End of WS.COM; see note below.

TRMUNI  equ     029BH   ;WS de-initialization routine.
PBGMEM  equ     035CH   ;WS modified working storage jump.
INISUB  equ     02A4H   ;Initialization subroutine.
UNISUB  equ     02A7H   ;De-init subroutine.
```

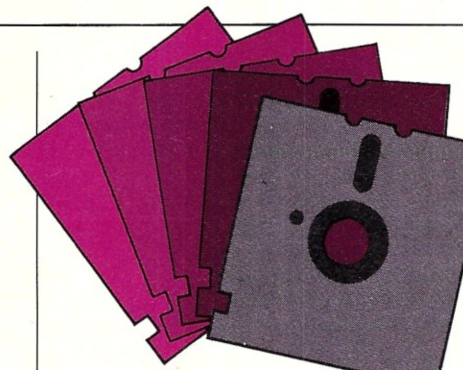
```
MORPAT   equ   02E0H       ;Extra room for de-init.
XTAB     equ   0649H       ;Up to nine extra dispatches.
EXTCUR   equ   04E9H       ;Cursor to markers 6-9
EXTCUR1  equ   0589H       ;Set markers 6-9.
EXTCUR2  equ   053DH       ;"Home" key dispatch.
COLDST   equ   2D08H       ;Cold start jump @ 100H

; Equates for dispatch addresses inside WS

CRU      equ   643EH       ;Cursor up.
CRD      equ   6424H       ;Cursor down.
CRF      equ   63A0H       ;Cursor right/forward.
CRB      equ   6365H       ;Cursor left/backwards.
CLL      equ   63F1H       ;Cursor to left side.
CRR      equ   63D3H       ;Cursor to right end of line.
CTT      equ   6478H       ;Cursor to top of screen (Home).
CBB      equ   6483H       ;Cursor to bottom of screen.
INLIN    equ   6549H       ;Insert line.
DELIN    equ   0242H       ;Delete line.
DERIT    equ   67E4H       ;Delete character at cursor.
TOGL     equ   6543H       ;Toggle insert mode.
RELMAR   equ   021CH       ;Margin release.
SETRUL   equ   0236H       ;Margin from file line.
CENTER   equ   0234H       ;Center text on line.
BFIL     equ   6B18H       ;Jump to beginning of file.
EFIL     equ   6B1EH       ;Jump to end of text.
UPRED    equ   0204H       ;Save and re-edit file.
PARAT    equ   0248H       ;Temporary indent (^OG).
VFIND    equ   0212H       ;Find prompt (^QF)

org EXTCUR
        db   ESC,01        ;Cursor to top of screen.
        dw   CTT
        db   ESC,02        ;Cursor to bottom of screen.
        dw   CBB
        db   ESC,03        ;Cursor to end of line
        dw   CRR
        db   ESC,04        ;Cursor to left of screen.
        dw   CLL
org EXTCUR1
        db   ESC,0CH       ;Insert line.
        dw   INLIN
        db   ESC,0DH       ;Delete line.
        dw   DELIN
        db   ESC,0EH       ;Delete character (^G)
        dw   DERIT
        db   ESC,00        ;Insert toggle.
        dw   TOGL
org EXTCUR2                ;Overlay alternate delete (1F)
        db   ESC,08        ;Curs to top.
        dw   CTT

; The following patches correct or modify a number of areas
; to save the bother of using INSTALL.

org 190H                   ;Modify the Signon Message.

        db   'Zenith Z-19 Terminal

org 2AAH                   ;MicroPro's initialization is wrong.
```

```
        db   0FFH          ;Patches USELST properly for Z19.
        db   0,0,0         ;Reserved by WS.
        db   1,1           ;Set timing delays to 1 ms.

org XTAB
        db   ESC,13H       ;Margin release=f1.
        dw   RELMAR
        db   ESC,14H       ;Mar from file line=f2.
        dw   SETRUL
        db   ESC,15H       ;Center text=f3.
        dw   CENTER
        db   ESC,16H       ;Find=f4 (^QF)
        dw   VFIND
        db   ESC,17H       ;Indent=f5
        dw   PARAT
        db   ESC,10H       ;Beginning of text=blue.
        dw   BFIL
        db   ESC,11H       ;End of text=red.
        dw   EFIL
        db   ESC,12H       ;Save & re-edit=white.
        dw   UPRED

org TRMUNI
        db   08            ;Number of characters to send.

; Overlay any existing patches for closing session.

        CURSOR 1,25        ;Get to 25th line
        db   ESC,'E'       ;and erase it.
        db   ESC,'u'       ;Normalize keypad.

org UNISUB
        jmp  MORPAT        ;Set up jump for de-init.

org MORPAT
        mvi  c,09H         ;Print string.
        lxi  d,CURSUP
        call    BDOS
        ret

CURSUP:
        CURSOR 1,24        ;Avoid clearing screen so
        db  esc,'yl'
        db  '$'            ;end-session err msgs remain.


; The following code modifies WS's jump from 100H into
; its initialization routines.  Control is given to
; the 25th line routine (below) and then the jump is
; restored.  The routine branches there and WS
; initializes normally.  The reason for this is that
; you would otherwise have to increase the size of WS.COM from
; 14K to about 30K, drastically increasing the load time
; to handle only 150 or so extra bytes.

;In this version, the screen intialization is buried in
; WS's communication area.

; After patching, "SAVE 63 WS.COM".
```

WordStar Mods for the Z-19 continued . . .

```
org     100H
        jmp     MEMORY

org MEMORY
INITIALIZE:
        mvi     c,09H           ;Function prints string.
        lxi     d,MESSAGE
        call    BDOS

; Re-patch WS entry point.

        lxi     b,0003          ;Number of characters
        lxi     d,100H          ;destination
        lxi     h,TRUJMP                ;Overlay characters
        ldir                    ;You can't use an 8085.
        jmp     100H            ;Cold start.

TRUJMP: db      (jmp)
        dw      COLDST          ;WS cold start jump.
        ret                     ;Back to WS.

MESSAGE:
        db      ESC,'x','1'     ;Enable 25th line.
        db      ESC,'x','6'     ;Shift the keypad.
        CURSOR 1,25             ;Put the cursor at 1 25, col 1
        REVID
        db      'mar rel=f1'
        REVOFF
        SPACE
        REVID
        db      'mar/line=f2'
        REVOFF
        SPACE
        REVID
        db      'center=f3'
        REVOFF
        SPACE
        REVID
        db      'find=f4'
        REVOFF
        SPACE
        REVID
        db      'indent=f5'
        REVOFF
        SPACE
        REVID
        db      'begin=blue'
        REVOFF
        SPACE
        REVID
        db      'end=red'
        REVOFF
        SPACE
        REVID
        db      'save=white'
        REVOFF
;End of line 25 print.

        db      ESC,'H'         ;Home the cursor.
        db      '$'             ;End of output.

end INITIALIZE
```

Software Directory



Program name: GrafTalk 2.0
Hardware system: Z80 or 8080, with CP/M or MP/M
Language: Object code
Description: GrafTalk is a device-independent business graphics package that uses English-language commands. The version 2 release incorporates all the features of version 1 for producing bar, pie, line, and symbol plots. It also includes these new features: access to a menu mode at the user's choice, with either standard menus or customized menus created by the user; a built-in mini spreadsheet capability that allows viewing and manipulation of the data set; several new types of bar charts; increased flexibility in number and types of access lines; increased flexibility in data sets (including larger data sets, subsets, etc.); a built-in editor and a large range of graphics and text commands. GrafTalk accepts data as in an ordinary table of numbers and supports multiple plots on a single surface, user-adjustable names, labels and legends, and an on-line help facility.
When released: April 1983 (see p. 74 for a review of version 1).
Price: $450; Version 1 upgrade, $100.
Included with price: Disk containing software and drivers for many screens, printers, and plotters; manual (with full-color tutorial sections).
Where to purchase it:
Redding Group, Inc.
609 Main St.
Ridgefield, CT 06877
CIRCLE #117 ON READER SERVICE CARD

Program name: MODEM 8-16
Hardware systems: CompuPro 8/16 and MP/M 8-16 V2F
Minimum memory size: 192K
Language: Object code
Description: Permits bidirectional file transfer and asynchronous communication via modem or RS-232 interface between any terminal and another mainframe, minicomputer, of CP/M and MP/M 8-16 based system. The system's other operations are not affected and the program can support as many as six modems simultaneously. Under MODEM 8-16, each terminal within a GCS multiuser system can operate either in "terminal" mode as a remote station of the external

Software Directory continued . . .

system, or in "command" mode, which allows each terminal to specify parameters under which information will be exchanged and understood by the external system.

Other features include the ability to store sequences of commands as a file, and simultaneous communications with the operating system while in use. The program's large command set also allows users to automate complex communications procedures as well as control and customize the entire telecommunications process.

When released: February 1983
Price: $295
Included with price: Disk and manual
Where to purchase it:
Gifford Computer Systems
1922 Republic Ave.
San Leandro, CA 94577
(415) 895-0798

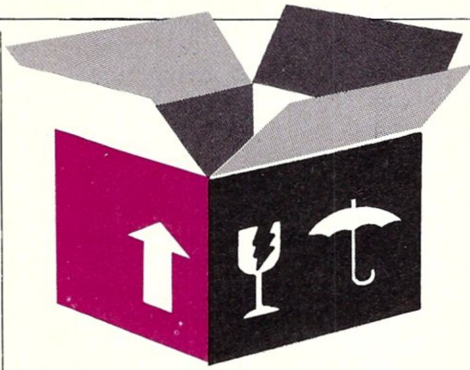
CIRCLE #118 ON READER SERVICE CARD

New Products

Dual CRT terminal

The IMS CRT stand-alone terminal has a separate microprocessor for each of its major functions: keyboard control and screen handling. It can be customized to emulate nearly any conventional CRT.

The IMS CRT has all the capability of a CRT but includes a general-purpose Z80 microcomputer complete with two high-speed RS-422 serial ports, one RS-232 printer port, 64K of RAM, and a battery-backed-up realtime calendar clock.



In expanded mode the terminal becomes a node processor in a multiuser, multiprocessor network environment.

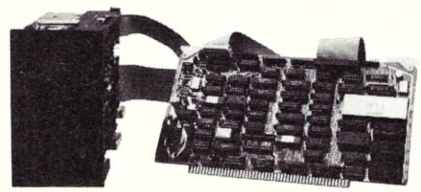
Price: basic CRT, $1095; expanded CRT, $1795.

IMS International, 2800 Lockheed Way, Carson City, NV 89701; (702) 883-7611.
CIRCLE #119 ON READER SERVICE CARD

5¼″ hard disk S-100 subsystem

Advanced Digital Corp. has announced a 5¼″ Winchester hard disk subsystem for S-100 microcomputers built around ADC's HDC-1001 hard disk controller board. It is available in 5, 10, 20, and 40MB configurations. The HDC-1001 is IEEE-696 compatible, features an on-board microprocessor, and has error-correction capabilities. Other ECC features include up to 9-bit single-burst correction; multiple-burst detection, programmable correction/detection span, process check/syndrome bits in serial fashion; and supports read/write, short/long features.

The Model HDC-1001 controller provides control for up to 4 drives and up to 8 R/W



New Products continued . . .

heads. It has built-in data separation and features data rates of up to 5 MB/sec, 256-sector addressing range, CRC generation/verification on ID fields, unlimited sector interleave capability, automatic retries on all errors and automatic restore and re-seek on seek error.

The subsystem comes complete with controller board, connector cable, CP/M BIOS disk and several hard disk drives, depending on the capacity. For example, the 5MB version will feature the industry-proven Seagate ST 506 drive. The 10 and 20MB versions will feature the highly reliable miniscribe drive.

Price: 5 MB, $1800; 20MB, $2000.

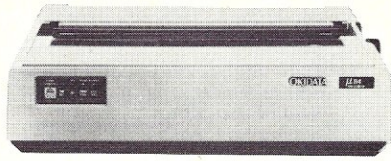
Advanced Digital Corp., 12700 B Knott Ave., Garden Grove, CA 92641; (714) 891-4004.

CIRCLE #120 ON READER SERVICE CARD

New Products continued . . .

Upgraded Microline 84 printer

Upgrades and enhancements to the high-performance Microline 84 dot matrix printer are contained in the "Step 2" Microline 84. Along with correspondence-quality printing, the "Step 2" models feature full compatibility with WordStar, Visicalc, and other popular software packages, thus eliminating the need for modifications when using these types of



software in the correspondence mode.

Other improvements include a switch-selectable interface, select/deselect control, Spanish character set and forward feed superscript/subscript printing.

New Products continued . . .

The "Step 2" models, control-code-compatible with the entire Microline family of Okidata printers, offer 200 cps bidirectional printing and a 50 cps correspondence-quality printing mode with proportional character spacing.

A new superspeed RS-232C interface is now standard on all serial Microline 84s. This interface provides switch-selectable transmission speeds up to 19.2K baud and supports normal, correspondence, and graphics print data. Both the X-ON/X-OFF and printer Ready/Busy protocols are provided. A monitor mode and a self-test mode provide extensive diagnostic capabilities.

Okidata Corporation, 111 Gaither Drive, Mt. Laurel, NJ 08054; (609) 235-2600.
CIRCLE #121 ON READER SERVICE CARD

Portable S-100 systems

Computershop's Star-Lite professional portable computer uses the S-100 bus, Z80 CPU, and 4 open slots expandable to graphics, hard disk, or hundreds of other applications. The Star-Lite offers sided drives totalling 400K disk storage with optional 800K disk storage and 26 extra user-programmable function keys.

Retail price is $2,695 and includes word-processing, spreadsheet, and CP/M; optional software includes database filers, spelling checkers, and other CP/M compatible software. A 9″ green screen provides half-density, underlining, and blinking functions, and there is a

contoured programmable non-capacitance numeric keypad with cursor control and 26 function keys. One RS-232 serial and one Centronics parallel port are built in; one additional serial port may be added without using any open slots. Special-purpose ports may be added by using the open slots.

Computershop, Inc., 139 First St., Cambridge, MA 02141; (617) 542-4295.
CIRCLE #122 ON READER SERVICE CARD

Microsystems Mart

classified ads

CLASSIFIED RATES: Per Word, 15 Word Minimum. REGULAR: $1.00. EXPAND-AD: $1.50. GENERAL INFORMATION: Prepayment discounts available. Payment must accompany order except credit card — Am. Ex., Diners, MC, VISA (include exp. date) — or accredited ad agency insertions. Copy subject to publisher's approval; must be typewritten or printed. First word set in caps. Advertisers using P.O. Boxes MUST supply permanent address and telephone number. Orders not acknowledged. They will appear in next available issue after receipt. Send order & remittance to: Classified Advertising, MICROSYSTEMS Magazine, 1 Park Avenue, New York, NY 10016. Direct inquiries to: Kenneth L. Novotny (212) 725-3927.

COMPUTER NETWORKS

THE CRITICAL CONNECTION! Now you can buy an inexpensive ATARI-400, and have it use the disk drives, printer, and keyboard on your CP/M computer. Complete package $175.00 DEALER INQUIRIES WELCOME. USS ENTERPRISES, 6708 Landerwood Lane, San Jose, CA 95120, (408) 997-0264.

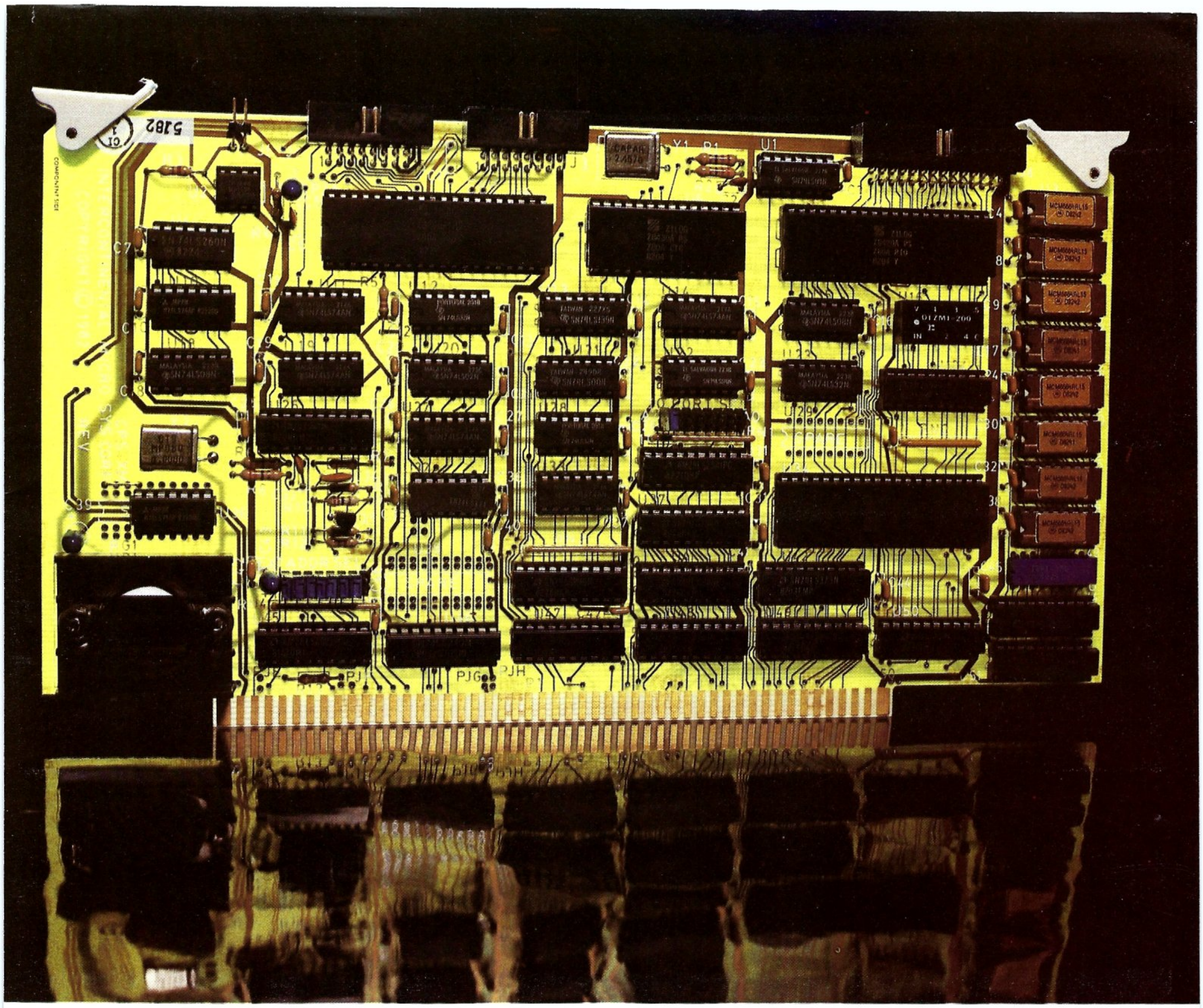
BUSINESS OPPORTUNITIES

COMPUTER CASH—101 exciting, new, sparetime home businesses. Report. $4.50, Associated Research, Box 22485MI, Houston, TX 77227.

INDEXING AND MICROFORM AVAILABLE

This magazine is indexed in THE MAGAZINE INDEX available from Information Access Corporation, 404 Sixth Avenue, Menlo Park, California, and in microform from Xerox University Microfilms, Ann Arbor, Michigan and Bell and Howell, Micro Photo Division, Wooster, Ohio.

ADVERTISERS INDEX



Leading Edge Z-80, S100 Distributed Processing.

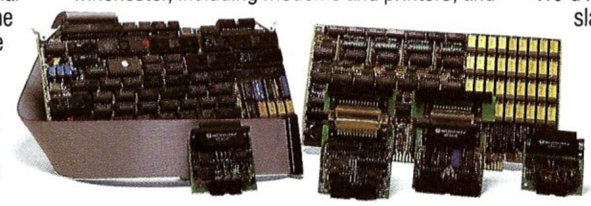
$475.00

Memory transfer rates of 517Kbytes/second, direct memory access, memory mapping and host to slave requests via interrupt control make the CPS-MX fast. And easy to integrate. Fully compatible with TURBOdos™, Intercontinental Micro System's slaves are available in four versions: synchronous or asynchronous serial port, 4Mhz or 6Mhz. The choice is yours. The CPS-MX also allows the bus master to utilize slave memory at the user's discretion. The slave then acts as a 64K RAM card.

The CPS-MX is also easy to integrate with Intercontinental's full line of S100 products: CPZ-48000 SBC single board computer with 64K on board RAM, 4 I/O channels, memory management, on board floppy controller, DMA and vectored priority interrupts; and 256K bank selectable or contiguous memory. A complete line of personality boards allow easy interface to anything from a floppy to a winchester, including modems and printers, and don't take up any S-100 bus space.

Best of all is the price. The CPS-MX starts at $475.00. That's right, up to 65% less than what you have been paying for products that may not measure up.

Call Intercontinental Micro Systems today. We'd like to send you information on the S-100 slave alternative.



INTERCONTINENTAL MICRO SYSTEMS CORPORATION

1733 South Douglass Road, Suite E, Anaheim, California 92806 (714) 978-9758 Telex: 678401-TAB-IRIN