# TABLE OF CONTENTS

**Why I Wrote A Debugger** ....................................... 4
**Xerox Monitor Modifications** .................................... 26
**Direct BDOS Calls in CP/M** .................................. 30
**RS-232C: The Interface** ......................................... 35
**Review: Media Master** .......................................... 45
**Color Speedometer For The Kaypro** ......................... 49
**The 32-Bit Super Chips** ........................................ 50
**The Definicon DSI-32 Co-Processor** ....................... 52
**Programming The 32032: Setting Out** .................... 58
**Turbo Memory Assignments** .................................. 74
**Modula II (excerpts from a talk by Niklaus Wirth)** .... 76
**Cheap DC To DC Converter** ..................................... 81
**Extending RATFOR** ............................................ 85

**Regular Features**

- The S-100 Bus ................................................. 11
- In The Public Domain ........................................ 13
- C'ing Clearly ................................................ 16
- The Xerox 820 Column ....................................... 21
- 86 World ...................................................... 38
- The Kaypro Column .......................................... 46
- FORTHwords .................................................. 63
- Pascal Procedures ........................................... 70
- On Your Own .................................................. 86
- Technical Tips ............................................... 90
- Culture Corner ............................................... 97
- Future Tense ................................................ 98
- Tidbits ....................................................... 98
- The Last Page ............................................... 104
Jim Ferguson, the designer of the "Big Board" distributed by Digital Research Computers, has produced a stunning new computer that Cal-Tex Computers has been shipping for a year. Called "Big Board II", it has the following features:

- **4 MHz Z80-A CPU and Peripheral Chips**
The new Ferguson computer runs at 4 MHz. Its Monitor code is lean, uses Mode 2 interrupts, and makes good use of the Z80-A DMA chip.

- **64K Dynamic RAM + 4K Static CRT RAM + 24K E(P)ROM or Static RAM**
"Big Board II" has three memory banks. The first memory bank has eight 4164 DRAMs that provide 60K of user space and 4K of monitor space. The second memory bank has two 2Kx8 SRAMs for the memory-mapped CRT display and space for six 2732As, 2Kx8 static RAMs, or pin-compatible EEPROMs. The third memory bank is for RAM or ROM added to the board via the STD bus. Whether bought as a bare board or assembled and tested, it comes with a 2732 EPROM containing Russell Smith's superb Monitor.

- **Multiple-Density Controller for SS/DS Floppy Disks**
The new Cal-Tex single-board computer has a multiple-density disk controller. It can use 1705 or 8877 controller chips since it generates the side signal with TTL parts. The board has two connectors for disk signals, one with 34 pins for 5.25" drives, the other with 50 pins for 8" drives.

- **Vastly Improved CRT Display**
The new Ferguson SBC uses a 6845 CRT controller and SMC 8002 video attributes controller to produce a display rivaling the display of quality terminals. There are three display modes: Character, block-graphics, and line-graphics. The board emulates an ADM-31 with 24 lines of 80 characters formed by a 7x9 dot matrix.

- **STD Bus**
The new Ferguson computer has an STD Bus port for easy system expansion.

- **DMA**
The new Ferguson computer has a Z80-A DMA chip that will allow byte-wise data transfers at 500 Kbytes per second and bit-serial transfers via the Z80-A SIO at 800 Kbits per second with minimal processor overhead. When a hard-disc subsystem is added, the DMA chip makes impressive disk performance possible.

**Terms:** Orders paid for with a cashier's check or bank card will be shipped within three working days. Orders paid for with a personal check will be shipped within three weeks. Add $5 for packing & shipping in North America.

**CAL-TEX COMPUTERS, INC.**
12788 HWY. 9 • BOULDER CREEK, CA 95006 • (408) 338-2572
Short And...

You may notice that this editorial is unusually short (at least for me). No, it isn't that Gary and Becky have finally found their delete keys; it's just that I've been very busy. I've spent a good deal of time speaking to groups around the Northwest — the Piper Cherokee that I rent has been getting a workout. (It's so old that its radio has a hand-crank.)

Plus, I've spent a number of hours poring over turgid tomes (not morbid gnomes) on Modula II. The language is downright neat. Niklaus Wirth has done himself proud.

The Modula II article in this issue is really a chat with Niklaus Wirth rather than a dry review of the language.

If you're familiar with Pascal, you should really enjoy Niklaus' ideas about it and Modulas. Even if you're just getting started with programming, I think you'll come away with a better understanding of computer languages in general. (A special thanks to Ken and Pam Benedict — plus all you Sacramento Micro C readers who invited me to come down for Niklaus' talk.)

New Processors

This issue has really turned into a new-processor issue. We have three articles covering the new 32-bit processors (mostly the 32032). Arnie Henden takes a close look in the FORTH column at how these fancy devices are designed. Plus, the Slicer column is broadening to cover the whole gamut of Intel chips (see below). Of course, Tidbits is still playing the "what's new" game. Only the editorial remains aloof — a quiet haven for those of you whose sensibilities have been battered.

86 World

We have renamed and revised the Slicer Column to "86 World." Actually, the column has been changing for quite some time. Laine has written some really interesting stuff on 8086 assembly language and operating systems, but since they've been hidden under the name of Slicer, very few of you have noticed them. (Watch for the new 80186-based systems coming to the SOG.)

So we've taken Laine out of the closet and turned him loose on the whole field of 8086, 80186, 80286, and 80386 ... operating systems, assemblers, compilers, boards, you name it.

86 World is a particularly appropriate name for this column. You see, Laine's off to see the world. Right after SOG IV, he'll be spending a couple of years in Turkey where he will be designing computer systems to help automate agriculture. He'll continue doing his column during his stay there.

I'm sure he'll welcome cards, letters, and visits from anyone — especially visits from mobile hackers of the feminine set.

Running Off Is Running On

The response to the Pascal Runoff is absolutely super. A week after the first announcement of the contest, we had six entries — neat stuff, too — and submissions (projects for the underwater Navy) are continuing to roll in. So take your keyboard in hand and submerge yourself in that one great program to end all programs. (Get a picture of your family to take with you so you'll be able to recognize them after you've finished.)

A Thorne By Any Other Name

Thorne EMI has found out what happens when you take reasonably good software packages and shoot them in the feet.

(continued on page 94)
In Defense Of Well Written Prose

Ever since the Issue 22 editorial, I've been thinking about how thoroughly hung up one can become over some absurdly simple thing. I believe we readers could have eventually deduced the Thompson Algorithm for the half-fold then three-fold for reducing 8 1/2" X 11" sheets to fit small envelopes, but I really can't claim credit for brilliance. As a matter of fact, I remembered how to do it.

I'm 73 years old, and when I was young there were no Size 10 envelopes (not where I worked, anyhow). If you didn't half-fold-three-fold, you couldn't mail a letter, so most of us puzzled out some solution to this environmental challenge. The Thompson Algorithm for Insertion Procedure, though, certainly does clear up what could be a tricky problem for someone younger.

This leads to another topic. One of the biggest defects in the documentation of computer programs is an assumption on the part of the writer that the user actually knows something. Simple directions do not have to be given in a condescending or a cutesy-pootsy manner. A little simple expository prose about the most elementary things is often received with great appreciation, even by the cognoscenti. I am not one of that elite group, but, as I said above, I'm 73, and I've been at this since card-sorters and the IBM 650. I spend a lot of time calling long-distance about things that should have been put in simple English in the first few pages of the manual.

Even in my professional work, I am surprised how often I am hung up on some simple matter and can't find the answer in books which ought to contain the information. The following happened just a few nights ago as I was writing a report about a patient.

I came across a respiratory rate of 18 breaths per minute in the record. That's pretty fast breathing for a patient lying quietly in bed. I thought, so I wrote "tachypnea." That's medical hog-Graeco-Latin for fast breathing. Then I got to thinking, this report is for a nonphysician. I ought to document the normal range of respiratory rate. I looked in two brand-new textbooks of medicine (which I had just recently purchased for $75 each), but the information was not there. I looked in a two-volume loose-leaf text of medicine for which I annually pay $100 for monthly updates. No luck. I used to teach physiology. I looked in three fat textbooks of human physiology. No go. There was lots of talk about tachypnea in all of these works, but not word one about a formal definition of what IS tachypnea, in numbers, that is.

(Just so you won't lie awake all night wondering, I found the following in a German textbook: mean respiratory rate, men, resting: 11.7 breaths per minute; 95% range: 10.1-13.1; Fruhmann, G., Zeitsch Exp Med, 1964;138:1.)

I suppose if you can't find some of the elementary facts of life in six medical textbooks, it's too much to hope to find the simple but necessary facts in a program document.

George R. Meneely, MD
514 Southfield Rd.
Shreveport LA 71106

Editor's note:
Thanks, George. Your letter is a breath of fresh air (11.7 times a minute).

Using WordStar With Patches

I have WordStar version 3.3 and found a piece of public domain software called ANYCODE which is a patch that steals calls to WordStar's print routine and allows a custom interpretation of printer control codes imbedded in your text. However, I found that the area the patch was designed to go into was already being used in some way to support the Kaypro 4. I then tried to add the patch outside of WordStar and enlarge the WordStar load module to include it. WordStar, being much brighter than I, knows how big it is supposed to be. Hence, the patch works only under certain circumstances and is entirely destroyed under other conditions.

How do I arbitrarily increase the size of WordStar to include patches? Or is there a better way?
Roger G. Fordham
4050 E. Sacaton St.
Phoenix AZ 85044

High On Micro C

It was with great joy that I greeted the arrival of Issue 23. I leaped through it and read the articles of immediate interest, then tucked it into my bag for the flight home. The flight was delayed (something about not being able to screw the right wing back on properly), so much so, that by the time we queued up for the takeoff, we had already been in the plane three hours. Out of kindness (or perhaps to keep from getting lynched) the flight attendants opened up the bar. When we arrived we were so well lubricated that we had not only forgiven, but forgotten everything.

I awoke with a jolt that night when I remembered my issue of Micro C was still in the seat pocket on the airplane, between the plastic safety card and the funny little bag with the twist tie across the top.

I had a lousy weekend.

I come to you now with a tear in my eye and a check in my hand and ask that you send me another issue #23.
I will never drink and read Micro C again.

Julian R. Bryttan
1926 Prairier Square #311
Schaumburg IL 60195

Editor's note:
It staggers me to think what Micro C drives some folks to do. If you are now subscribing because of an issue #23 you found in an airline pocket, you might send Julian something as a thank you (anything but a drink).

Kaypros In Zaire

I have found your publication very interesting and most helpful, but a
LETTERS

little overwhelming at my present stage. I suppose this would be true of anyone who read 11 issues in a little over a week. Much of it is way over my head, but I have found that just reading, even when I do not completely understand, is a great help in expanding my horizons. I have learned a lot about my Kaypro II from the tips and suggestions in Micro C, and have even done a few of the modifications.

There are about 10 Kaypro users here in Kinshasa, Zaire, and all are as new to computers as I am. We have a very informal Kaypro users group, but only one of us is technically oriented and he's leaving soon, so we are really hurting. It's kind of like the blind leading the blind.

Most of what I get now is from manuals, but as you have pointed out, manuals are not the best way to learn about computers or programs. So I have a question for you. I would like to upgrade my Kaypro II to a 4 and install the Pro-8 ROM for possible future expansion. Since I already have the Pro-Monitor II on the machine, do I have to do the 4 mod first before I install the 8, or can I go directly to the 8?

Gordon L. Bottemiller
C.B.Z.O.
P.B.4728
Kinshasa II, Rep. of Zaire

Editor's note:
You obviously have an 83 Kaypro II (with a real Kaypro II processor board). You will have to do the II to 4 upgrade in order to run the Pro-8 monitor. You won't have to change your drives right away, however.
Also, don't let your problems with Micro C drive you to drink. It'll only make things worse (see the previous letter).

Disk Box Stash
I recently replaced two full-height SSDD drives with two half-height DSQD 'TEACs (you guessed it: from Cal. Digital). This left a large empty cavern where drive B used to be. I found this space to be a perfect place to stash two boxes of diskettes during transportation. Furthermore, by using a blank drive bezel (leftover from an IBM PC upgrade) which snaps perfectly into place, everything can be hidden away safe and sound (kind of like an Osborne pocket).

Roy Trevino
709 Kofa Avenue
Parker AZ 85344

Editor's note:
Look closely at your screen, Roy. If the letters are really tiny and the system displays only 52 columns, I'd take out the disk holder.

Turbo Utilities For CP/M-80
I am programming in Turbo Pascal under CP/M 80. I would like to contact others who are using CP/M 80 so as to exchange ideas, tips, and utilities. There seems to be a flood of IBM Turbo utilities, but very little for CP/M 80. Thanks for your help.

Ralph E. Freshour
Box 7000-309
Redondo Beach CA 90277

Editor's note:
There's also a Turbo section on the Micro C bulletin board. There isn't much there yet, but if you'll log on and share ideas and programs, there'll be more. The number is 503-382-7643. It's 300 and 1200 baud, 8 bits per char, no parity, 1 stop bit, 24 hours.

Kaypro Lockup Problem
Last year I managed to speed up my old (1982) Kaypro II from 2.5 to 5MHz, with a toggle switch between the two speeds. I even added a muffin fan in the back.

For many months all went well. But yesterday, while in Perfect Writer/PluPerfect at 5MHz for less than an hour, I got keyboard lockup! No way could I save the partial letter in memory. I toggled to 2.5MHz, reset, and everything seemed to be OK again.

Today, it's locking up on both speeds. What could it be??

I can't remember all the mods I've made with your help, so what I'm asking for is a list of all the new chips in best "final" configuration — what you have learned from subscribers and experience to be the best new components. For example, U67 should have a Z80B SIO made by Zilog, etc., etc.

I'm sure I have installed most of these, but perhaps not all, and such a list would enable me to see what remains to be modified.

Harald S. Gibson
444 -36th St.
Manhattan Beach CA 90266-3208

Editor's note:
Your problems don't seem to be speed sensitive (at least your computer-related problems). A number of people have written to us about problems with power connectors (e.g., between the power supply and the main board, the drives, and the video monitor). Some good contact spray cleaner and exercising of all the plugs (pushups) wouldn't hurt at all.

Also, you might want to try firmly seating (or reseating) all the socketed chips on your board, especially the larger ones. (Keep your seat and read the following letter.)

In Search Of Exo
I have recently purchased a used Exo single board computer and did not get the wiring and interface documentation with it. It also had an excellent set of business programs (AP, AR, GL and Payroll) for which I would like to get the source code and documentation. The programs were written in C-BASIC and are the best I've seen. I wonder if any readers might know how to contact Exo, if they are still around, or might have the documentation.

J. E. McMurray
P.O. Box 6772
Columbia SC 29260

(continued on page 88)
If you are using a Z80-based CP/M system and are programming in assembly language (or you plan on trying your hand at this sport), then a debugger should be part of your toolkit.

What Is A Debugger, Anyway?
"Debugger" is the slang term for a program monitor, a special routine that lets you interrupt the execution of a program and observe the condition of the CPU registers — examine the contents of memory — modify the contents of the CPU registers and memory — and perform other tests to help you find the source of programming errors.

Why Use A Debugger?
When you're writing and testing a program in a high-level language and you encounter unexpected results or strange answers, it's usually easy to track down the problem. In many cases, the language provides you with messages that tell you the kind of mistake you've made and its location in the program.

With an interpreted language such as MBASIC, for example, you can break into the program at any point and display the values of variables. With a compiled language, the compiler will usually indicate syntactic or structural errors, and you can easily include "print" or "write" statements in your source to check the progress of your program.

But with assembly language programs, you don't get any help beyond identification of language errors that prevent the assembler from doing its job.

Once you begin executing your assembled program, you're on your own. Sometimes, you may simply get unexpected results. Other times, a program may get "lost." That is, it may be misdirected to an area of memory that does not contain an executable instruction, and the results can be bizarre, amusing, or destructive (usually it's all three).

You could write diagnostic messages into the program, but that's quite tedious to do in assembly code, even if you knew exactly where to put the diagnostic message.

Why Not DDT?
After all, it comes with every CP/M system. There is one strong reason against using DDT: a debugger must be written specifically for the system CPU. DDT was written for the 8080 and is of limited use in Z80 systems. In particular, the Z80 has more registers than the 8080 and it instruction set is quite a bit larger. What you need is something that will do for the Z80 what DDT does for the 8080.

Enter DEBUG!
Shortly after I began using Micro C's CROWECMP assembler to do some experimenting with the video controller chip in my Kaypro 4-84, I wished I had a Z80 debugger to help with the task. I shopped around and quickly decided that, as a hobbyist just wanting to experiment, I could not justify the cost of one of the professional assemblers/debuggers. After mulling the situation over for a while, I decided that writing a debugger shouldn’t be too difficult and that it might be an interesting exercise. The results of my efforts — called DEBUG! — serve as illustration for the remainder of this article.

The Assembled Listing
In order to use a debugger effectively, you must have an assembler listing of the program you're debugging. Figure 1 shows the assembler listing for a short demonstration program.

First, let's take a look at the information you get from the assembled listing. The leftmost column shows the addresses at which the instructions or storage areas are located. The second column shows the assembled machine code for the mnemonic instruction, with addresses in low byte-high byte order (for example, 312401, the first executable instruction (31), is the Z80 machine code meaning "put the value 0124H into the SP register"). The third column contains labels. The fourth and fifth columns contain the source code instructions, and anything to the right of a semicolon is a comment.

Following the source code is the symbol table which shows the location of all the labels. (Labels are to assembly language what line numbers are to BASIC. They are places you can access or jump to.)

The line-by-line comments (they follow the semicolons) describe what happens in each step of the program. If you can't yet read assembly code well enough to be able to understand
all the details, don't worry about it.

The point of the program is this: the load (LD) instructions put hex values $11, 2233$, etc., into the A, BC, etc., registers. We fill them with these values so they are easy to spot when we use DEBUG! to look at the register status.

Similarly, the two PUSH instructions put (push) the contents of the BC and DE register pairs onto the stack so the stack will contain something easy to identify.

Then the EX and EXX instructions exchange the Z80's regular and prime registers. This moves the previously loaded values to the second (prime) set of registers. This second register set appears in the upper portion of the register display. (Note: We will specify the main register set as letters such as A and F, while the second, or prime set will be followed by single quotes such as A' and F'.)

Next, 2 is loaded into the A register and 3 into the BC register pair. Loading 3 into BC puts the 3 in C and a 0 in B — kind of a cheap trick to make the 3 stand out. Finally, the ADD instruction combines the contents of the A and C registers and places the result in A. So, if all goes well, the DEBUG! display will show 5 in the A register.

Displaying DEBUG!

Figure 2 is a screen dump of the DEBUG! display with execution halted at address 011AH, just after the addition and just before the jump to a warm boot. The RST $38H, the second instruction in the program, halted program execution and invoked DEBUG! More about that later. While it was stopped, I set a breakpoint (roughly equivalent to a STOP in BASIC) at 011AH, then issued a “Go” command to have the program continue execution to the breakpoint.

What It Means

The upper portion of the display shows the contents of the CPU registers and the register indirects. The contents of the addresses are displayed next to the addresses in the registers.

For example, DE contains the value F8AEH. Address F8AEH contains the value 08H, address F8AFFH contains the value EDH, address F8B0H contains the value 7BH, and so on, through the end of the 16-byte line following each register. The stuff to the right of the numbers is the ASCII representation of the memory contents.

A period indicates that a byte does not have an ASCII representation.

This kind of display is the same for all the registers except AF and A'F'.

(Notes: The contents of a register pair may not be an address, e.g., if you are using the B and C registers to hold two 8-bit integers, you wouldn't care what 16-bit “address” BC was pointed to, or the contents of that “address.”)

In this example, BC contains 0003H, a value for calculation, as explained above. The special registers SP and PC, however, always contain addresses, and the interpretation of the contents of the addresses (the indirect values) is special to those two registers.

Stack Pointer

The SP (Stack Pointer) register shows the current address of the stack. Our program loads the stack pointer with 0124H, but we’ve pushed four bytes onto the stack so the SP shows 0120H (the address at which the next byte pushed will be stored).

Because the SP decrements with each push, the addresses above its current value show what’s on the stack. And, sure enough, the display shows the contents of the BC and DE registers, now on display as the alternate register set (stored low byte first).

Program Counter

The PC (Program Counter) register shows the address of the next instruction to be executed. The breakpoint is set at 011AH; the PC shows 011AH. According to the assembled program listing, the instruction at 011AH is JP 0 (C3000), and the indirects following the PC display are C3 00 00.

The AF Registers

The AF and A'F' registers are not register pairs in the same sense as the others.

The A (Accumulator) register is a one-byte register whose contents can be manipulated in a variety of ways. Conversely, the F (Flag) register is a special register used by the CPU to indicate conditions resulting from cer-

(continued on page 7)
Attention Modem Buyers!!!

We Guarantee Lowest Price and Best Support Around!!

SIGNALMAN MARK 12
This 300-1200 baud Hayes compatible smartmodem comes with RS-232 cable, 12v power supply and phone cable.

SPECIAL: $219.95

WESTRIDGE 64/20
This 300 baud auto-dial/auto-answer modem is designed for Commodore 64 computers. It comes complete with software on disk and phone cable.

NOW ONLY: $59.95

SIGNALMAN MARK 10
This 300 baud smartmodem is Hayes compatible and comes with RS-232 cable, 9v power supply and phone cable.

NEW PRICE: $99.95

VOLKSMODEM 300
Now at its lowest price ever this 300 baud answer/originate modem only needs a cable and battery and it's ready to communicate!

LOWEST PRICE: $55.95

VOLKSMODEM 1200 BAUD
- 300-1200 baud smartmodem
- Hayes compatible
- auto-dial/auto-answer
- 5 year warranty

NOW ONLY: $179.95

FREE!!
To First 100 Buyers of the Month!!
$50.00 WORTH of Time on NEWSNET!!
(Must purchase by the 20th day of the month)

SIGNALMAN "EXPRESS"
This modem is an EXACT to the Hayes 1200. PLUS . . . has storage ability of 10 number of 48 digits. Anchor Automations "top-of-the-line" . . . for those who will only have a Hayes!
TODAY ONLY: $299.95

SIGNALMAN MARK 7
A 300 baud auto-dial/auto-answer modem which comes with RS-232 cable, RJ-11 cable and power supply.

DISCOUNTED: $79.95

VOLKSMODEM CABLE
Needed for Volksmodem 1200 and 300. Available for most computers . . . price may vary with computer type.

MOST ONLY: $9.95

CASE R1224
(2400 baud now in stock!!!)
This top-of-the-line modem meets the new 2400 baud CCITT V.22 standard and is Asynchronous/synchronous 2400/1200/300 bps. We have these in stock ready to deliver upon request!!
2400 BAUD TO DELIVER TODAY: $599.95

Call Now Toll Free: 1-800-922-2329
TECHNICAL SUPPORT: (316) 264-5589

OMNICOM

440 MAPLE • WICHITA, KS • 67213

We Service What We Sell!!!

VISA MASTERCARD

SPECIALIZING IN DATA COMMUNICATIONS

We Will Match Any Price!!!

TERMS: C.O.D. acceptable, add $1.90. Personal checks require two weeks for clearance. Visa and MasterCard accepted. All prices reflect 3% cash discount. Add 3% for shipping and handling. Kansas residents add 3% sales tax. All orders returned or refused are subject to a 10% restocking fee. Prices subject to change without notice.
WHY I WROTE A DEBUGGER

(continued from page 5)

tain CPU operations. The numbers following the AF and A'F' register names show their contents.

Note that the A register does indeed contain 5, the result of adding 2 and 3. But the 44 in the F' register cannot be interpreted so quickly. Therefore, the letters (or dashes) to the right of the numeric values are used to show the status of the bits (flags) in the F register. Performing an exclusive OR on the A register (now the A' register, with the registers exchanged) created a zero result; hence, bit 6, the zero (Z) flag, was set to 1.

Also, because the exclusive OR resulted in an even number of bits (zero) in the accumulator, the parity (P) flag was set. Other flags indicate Sign, Half-carry, Negative, and Carry. The "x" in bits 5 and 3 means that those bits are not used by the CPU; a dash means that the flag for that bit is not set (is 0).

If you're just getting started in assembly language programming, some of this explanation may border on gobbledygook. (Maybe more than border.)

Fortunately, there are two books that spell everything out in detail and that I highly recommend: "The Z80 Microcomputer Handbook" by William Barden, Jr. (Howard W. Sams & Co., Inc.), and "Z80 and 8080 Assembly Language Programming" by Kathe Spracklen (Hayden Book Company, Inc.). The Spracklen book is especially helpful to CP/M users because it presents 8080 code and Z80 code side by side, and even if you do have a Z80 system you will inevitably run up against 8080 code somewhere along the line.

Back To Figure 2

The clump of numbers toward the bottom of the display shows the contents of a 128-byte "page" of memory (and what's on display here is the demo program, starting at 0100H). Below the memory display is a line showing the breakpoints that are set; the last line is used for command entry and error message reporting. There are a number of things you can do with DEBUG! when the display is active:

- Page memory display forward or back
- Continue from the current breakpoint
- Single-step through the program under test
- Clear all breakpoints or a specified breakpoint
- Set a breakpoint
- Display a specified area of memory
- Perform hexadecimal arithmetic
- Jump to a specified address
- Fill an area of memory with a specified byte
- Modify the contents of a specified address
- Change the value in any register pair
- Search memory for a string of bytes
- Move the contents of one area of memory to another.

These functions let you see exactly the condition of the CPU and memory at any time and make changes for "what if" testing. The single-stepper is a particularly powerful tool when you're not sure what's happening in a program.

A debugger is an extremely valuable instructional aid. When I first began working with assembly programming, I found it quite abstract in comparison to high-level language programming. I read and reread explanations of what the various instructions did, but for a long time I was left with a "so what?" feeling. Sure, I could understand what happened with any instruction, but I had a tough time relating that to what I wanted to do.

My brother, who has dabbled with assembly language, dropped by while I was working on the debugger. When I finished explaining to him what I was doing, he remarked, "Gee, I wish I'd had this when I was fiddling with the assembler. It sure makes things clear."

How Does A Debugger Work?

So what's the big mystery about debuggers? Actually, there isn't much of a mystery, or I wouldn't have been able to write one.

Implementing DEBUG! involved one trick, one standard procedure, and one item of caution throughout. The trick is to make the program "self-calling," a misnomer in itself. While there are several ways a debugger could be invoked initially, there has to be a way for it to call itself for breakpoints or single-stepping (which is an internally set breakpoint at each executable instruction).

The Z80 provides eight one-byte reset (RST) instructions, which CP/M has chosen not to implement. The reset instructions direct the CPU to preset addresses, with the additional characteristic of pushing the address of the next program instruction onto the stack, just as a call does.

RST 38H, the one I chose to use, transfers program execution to address 38H, the same address DDT uses. The initialization portion of DEBUG! overwrites the jump-to-DDT instruction at 38H with a jump to DEBUG!. In this way, the RST 38H (0FFH) coded into the program invokes DEBUG!.

After that, any time a breakpoint is set, the program code byte at the breakpoint address is replaced by FFH, and when that instruction is encountered, DEBUG! becomes active (and the substituted code byte is restored). Thus, DEBUG! is not really self-calling: the user program code is altered to create the call. The standard procedure is to "preserve the user's environment."

Protecting The User

What this means is that the absolute first thing DEBUG! does is save the user's SP, PC, and contents of the PC (the return address pushed onto the stack by the RST instruction) and push all the user's register values onto the DEBUG! stack. This is exactly the same thing that's done in any interrupt-driven program. Once the user's environment is preserved, DEBUG! can skitter around and do pretty much whatever it wants, as long as it doesn't mess up the user's program or the system.

The caution throughout is to be very sure that the stack is restored upon completion of any action. The last thing DEBUG! does before re-

(continued on page 9)
The Turbo 640 K of memory (the maximum contiguous memory that the PC can address—and they're all 1 megabyte in size). A keyboard that is very similar to the keyboard that used to be in the Apple IIe. A 160-page manual printer port and a character set of 299 symbols (you can't learn that). That battery-powered clock (keeps track of the time and date while the computer is off). A monochrome graphics card that gives you crisp, home screen texture and lets you see graphics on a monochrome monitor displayed as sixteen shades of gray (amber). An amber monitor (very cheap). Two 200 disk drives (AKA Half Height Double-sided Double-density). Two front panel LEDs (one to indicate that you left the computer on from the night before and that it hasn't turned it). The other to tell you if the processor is in the high-speed mode of just going IBM speed. Eight expansion slots (two are used by the multifunction card and monitor card). Four full-sized and two short slots are available). Five available ROM sockets and two bus connection points (load programs stored on your own ROMs). USB wait-power supply (ready for upgrading to a hard disk). User's book and complete technical information (the user's part is easy - the tech part is very technical).

A computer without software is basically a solid state boat anchor and a cattle prod, one at a time. When you buy the Turbo 640 I want you to be able to start using it with my computer. The Turbo 640 runs on the Turbo PS/2 computer. It has been pointed out that the IBM PC is not quite as compatible as a network running the IBM/PC and it works faster than the IBM PS/2. I have heard of people who have put together the IBM PS/2 Turbo 640 and it worked. Right now I have three PS/2 Turbo 640s linked as a network running the IBM/PC and it worked. It is worth while to have a computer that is as compatible as a network running the IBM/PC and it works faster than the IBM PS/2. An oral surgeon friend saw my network and asked me to build one up for him. One of his friends has an IBM/PC and it is his operating room. It runs a program I wrote in Turbo Pascal to help him analyze records—on a single glitch. (The 640 seems totally immune to the effects of Nitrous Oxide.)

More than compatible. Though I have a private pilot's license and quite a few hours flying high performance aircraft, I've never taken the time to learn to use the Flight Simulator program from Microsoft. I have some friends of mine who are flies beautifully on the 640. The phrase "IBM compatibility" is an understatement with the computer: the PS Turbo 640 is just flat out a better machine.

Peter Norton publishes a program that rates hardware and compatibility with the IBM PC. In the normal month, Norton rates the PS Turbo 640 at 5.1 which indicates a 1 to 1 equivalence. In the high speed mode (selected with two keys which are not in the keyboard) the 640 scores a 1.4, that's 40% faster than an IBM PC.

For those of you that mistook me for Bill Gates from Microsoft, let me clarify things. I'm Derek L. Carlin, a private pilot who flies out of Rafael Road Airport in Palm Springs, California. The differences are pretty basic. Bill reads the Wall Street Journal, while I read USA Today. He's super soft on tech and builds great hardware.

Now that you know who I am, or at least who I'm not, let's get to the point.

Problem. Since I got into computers two years ago they have been a constant source of fascination and frustration. Fascination due to the immense power and control that the personal computer has afforded me in my work and personal activities; frustration due to the hundreds of hours I've spent making hardware and software cooperate with one another.

No doubt about it, once you get all the pieces put together correctly you have at your command an incredible tool. There is absolutely no limit to the number and type of tasks that you can accomplish with greater speed, better accuracy and stronger impact; all with far less effort. The trade off however has been the time, energy and uncertainty involved in finding the best buy on the right equipment, then getting all the parts to function as one.

Because I have personally gone through hours of long distance phone calls to hardware manufacturers and waited days for calls to be returned. I've learned that never happened. Because I have found myself so many times in front of a computer with the wrong or combinations of bloody characters at 2 AM unable to get the computer to print a simple business letter. Because I have found myself not with any way of doing anything about it. This is why I have put together the PS Turbo 640.

Solution. What happened was a few months ago I discovered a source of computer mother boards that run IBM type software faster than the IBM/PC. They're a genuine product of Taiwan and they are beautiful. I bought cases, keyboards, cables and disk drives; built the Turbo 640 and put it to work. Right now I have three PS Turbo 640s linked as a network running the IBM/PC and it worked. It is worth while to have a computer that is as compatible as a network running the IBM/PC and it works faster than the IBM PS/2. An oral surgeon friend saw my network and asked me to build one up for him. One of his friends has an IBM/PC and it is his operating room. It runs a program I wrote in Turbo Pascal to help him analyze records—on a single glitch. (The 640 seems totally immune to the effects of Nitrous Oxide.)

More than compatible. Though I have a private pilot's license and quite a few hours flying high performance aircraft, I've never taken the time to learn to use the Flight Simulator program from Microsoft. I have some friends of mine who are flies beautifully on the 640. The phrase "IBM compatibility" is an understatement with the computer: the PS Turbo 640 is just flat out a better machine.

Peter Norton publishes a program that rates hardware and compatibility with the IBM PC. In the normal month, Norton rates the PS Turbo 640 at 5.1 which indicates a 1 to 1 equivalence. In the high speed mode (selected with two keys which are not in the keyboard) the 640 scores a 1.4, that's 40% faster than an IBM PC.
turning control to the user program is restore the user’s environment to the registers and the stack.

Otherwise, virtually everything DE­BUG! does is just various arrange­ments of memory moves or standard algorithms. Single-stepping is inter­esting in that it is a software emula­tion of CPU action. DE­BUG! looks ahead one step to see what the CPU’s going to do next, then sets a break­point at the appropriate address. A bit tedious to code, perhaps, but hardly magical or mysterious.

DE­BUG!, which takes up 3042 bytes, is completely relocatable above address ODA6H. This means that if I am working on a program that will reside at the bottom of memory, DE­BUG! can sit at the top of memory (I usually keep it just below E000H, the top of the TPA in the 4-84). If I’m working on a program that runs in high memory, I can move DE­BUG! down to a spot where it will be out of the way.

For example, I used Micro C’s DUMP484 to make the printed display used in Figure 2. DUMP484 resides at the top of memory, and the demo program is at the very bottom. So I relocated DE­BUG! to end at 5000H, well below DUMP484 and well above the demo program. I also made the maximum possible use of standard CP/M calls in writing DE­BUG! so it would run on any CP/M system with an addressable cursor. It is custom­ized for the Kaypro to the extent of a few cursor locations and the cursor positioning routine — elements that can easily be changed.

DE­BUG! was written for CROW­ECPM, which makes use of only standard Z80 instructions, so a macro is not required to assemble it. Yes, DE­BUG! could contain many more features and functions. A debugger seems to be one of those programs onto which you could add bells and whistles until there was no memory left for a program to use it. What I’ve got now suits my (current) needs — and the price was absolutely right.

Editor’s note: DE­BUG! (along with other goodies) will be on Micro C User Disk K37.

---

**S100**

**BARE BOARDS**

ANY COMBINATION OF 3 BARE BOARDS

**$150**

<table>
<thead>
<tr>
<th>B/B</th>
<th>KIT</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

256K DYNAMIC MEMORY

64K TO 256K TO 1 MEGABYTE CAPACITY

$75 256K/510

64K/360

FDC-1 FLOPPY DISK CONTROLLER 8" & 5¼" DRIVES

$70 $245 N/A

Z80 CPU

4 MHZ/2 MHZ SELECTABLE

$65 $245 N/A

I.O. BOARD - 4 SERIAL PORT USES 8251 US ART

$65 N/A N/A

S100 CLOCK/CALENDAR WITH ON BOARD BATTERY BACKUP

$50 $99 $135

SOURCE CODE MONITOR ON STANDARD 8" DISK USED WITH CP/M 2.2

$25

MONITOR IN PROM - 2716

$25

BUY ANY COMBINATION OF 4 BOARDS & RECEIVE FREE MONITOR PROM

---

**THE WORLD’S FASTEST MOST POWERFUL 8080 RELocating MACRO ASSEMBLER**

**SLRMAC**

ONLY $49.95

AT THIS PRICE, SHOULD YOU BE WASTING YOUR TIME USING SOMETHING ELSE?

This is what they said about Z80ASM, our Z80 assembler. Now the same features and performance are available in our Intel Mnemonic product. SLRMAC is compatible with M80 in .8080 mode, with many extensions. Too many features to list here.

To order or to find out more about our complete family of development tools, call or write:

---

**SLR Systems**

1622 N. Main St., Butler, PA 16001

(800) 833-3061, (412) 282-0884

Telex 559215 SLR SYS

Add $3 shipping.

Z80 CP/M compatibility required.

---

Micro Cornucopia, Number 25, August-September 1985
IC BLOWOUT!

CPU/SUPPORT

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC68000-8 CPU</td>
<td>10</td>
<td>$16.00</td>
</tr>
<tr>
<td>Z80 CPU</td>
<td>1</td>
<td>$1.00</td>
</tr>
<tr>
<td>Z80 PIO</td>
<td>3</td>
<td>$3.50</td>
</tr>
<tr>
<td>Z80 SID</td>
<td>1</td>
<td>$7.50</td>
</tr>
<tr>
<td>Z80A CPU</td>
<td>1</td>
<td>$2.50</td>
</tr>
<tr>
<td>Z80A PIO</td>
<td>1</td>
<td>$3.50</td>
</tr>
<tr>
<td>Z80A SID</td>
<td>1</td>
<td>$10.00</td>
</tr>
<tr>
<td>Z80A DMA</td>
<td>1</td>
<td>$10.00</td>
</tr>
<tr>
<td>8088</td>
<td>1</td>
<td>$19.00</td>
</tr>
<tr>
<td>D8742 CPU/EPROM 12 MHz</td>
<td>1</td>
<td>$20.00</td>
</tr>
<tr>
<td>D8288</td>
<td>1</td>
<td>$22.00</td>
</tr>
<tr>
<td>4164-15</td>
<td>1</td>
<td>$1.50</td>
</tr>
</tbody>
</table>

MODEM/DIALER

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMS 99532 (Modem)</td>
<td>1</td>
<td>$12.00</td>
</tr>
<tr>
<td>TMS 99531 (Dialer)</td>
<td>1</td>
<td>$8.00</td>
</tr>
</tbody>
</table>

FLOPPY CONTROLLER

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD 1791</td>
<td>1</td>
<td>$9.00</td>
</tr>
<tr>
<td>WD 1793</td>
<td>1</td>
<td>$9.00</td>
</tr>
<tr>
<td>WE 1797</td>
<td>1</td>
<td>$16.00</td>
</tr>
</tbody>
</table>

BAUD GENERATOR

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD 8116</td>
<td>1</td>
<td>$8.00</td>
</tr>
</tbody>
</table>

MISCELLANEOUS

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD9216 Sync - Sep</td>
<td>1</td>
<td>$7.50</td>
</tr>
<tr>
<td>MM58167A Clock Chip</td>
<td>1</td>
<td>$8.50</td>
</tr>
<tr>
<td>CA301 Op Amp</td>
<td>1</td>
<td>$0.35</td>
</tr>
<tr>
<td>CA083 Op Amp</td>
<td>1</td>
<td>$0.45</td>
</tr>
<tr>
<td>2N3055</td>
<td>1</td>
<td>$1.00</td>
</tr>
<tr>
<td>Voltage Regulators</td>
<td>1</td>
<td>$0.50</td>
</tr>
<tr>
<td>LM323K &amp; 5V - 4A</td>
<td>1</td>
<td>$6.50</td>
</tr>
<tr>
<td>LM7812 &amp; 12V - 5A</td>
<td>1</td>
<td>$5.50</td>
</tr>
<tr>
<td>6116 2x8 Static Ram</td>
<td>1</td>
<td>$3.00</td>
</tr>
</tbody>
</table>

VIDEO CONTROLLER

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>6545</td>
<td>1</td>
<td>$7.00</td>
</tr>
<tr>
<td>6545A-1</td>
<td>1</td>
<td>$8.00</td>
</tr>
<tr>
<td>6845</td>
<td>1</td>
<td>$6.00</td>
</tr>
<tr>
<td>2716</td>
<td>1</td>
<td>$3.50</td>
</tr>
<tr>
<td>2732</td>
<td>1</td>
<td>$3.75</td>
</tr>
<tr>
<td>2764</td>
<td>1</td>
<td>$4.00</td>
</tr>
<tr>
<td>2764-25</td>
<td>1</td>
<td>$5.00</td>
</tr>
</tbody>
</table>

74LS SERIES

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS 125</td>
<td>1</td>
<td>$0.40</td>
</tr>
<tr>
<td>LS 375</td>
<td>1</td>
<td>$0.95</td>
</tr>
<tr>
<td>LS 375</td>
<td>1</td>
<td>$0.60</td>
</tr>
<tr>
<td>4073</td>
<td>1</td>
<td>$1.50</td>
</tr>
</tbody>
</table>

BATTERIES

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nicad 4 Pack AA</td>
<td>1</td>
<td>$3.25</td>
</tr>
<tr>
<td>Lead Acid D Cell 2V</td>
<td>1</td>
<td>$1.50</td>
</tr>
</tbody>
</table>

CONNECTORS

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>S100 - 100 Pin Board</td>
<td>1</td>
<td>$3.75</td>
</tr>
<tr>
<td>S100 - Extension Board</td>
<td>1</td>
<td>$2.50</td>
</tr>
<tr>
<td>S100 - Terminator Board</td>
<td>1</td>
<td>$3.30</td>
</tr>
</tbody>
</table>

PRINTERS

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centronics 703</td>
<td>1</td>
<td>$395.00</td>
</tr>
<tr>
<td>Centronics 588</td>
<td>1</td>
<td>$225.00</td>
</tr>
<tr>
<td>Centronics 1</td>
<td>1</td>
<td>$105.00</td>
</tr>
<tr>
<td>Teletype 43KSR</td>
<td>1</td>
<td>$300.00</td>
</tr>
<tr>
<td>NEC 2000 with Single Bin Cut Sheet Feeder</td>
<td>1</td>
<td>$1500.00</td>
</tr>
<tr>
<td>Transat 315 Color</td>
<td>1</td>
<td>$375.00</td>
</tr>
</tbody>
</table>

KAYPRO EQUIPMENT

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>9&quot; Green Monitor</td>
<td>1</td>
<td>$35.00</td>
</tr>
<tr>
<td>81 Series ROM's</td>
<td>1</td>
<td>$7.00</td>
</tr>
<tr>
<td>Keyboard (new)</td>
<td>1</td>
<td>$65.00</td>
</tr>
</tbody>
</table>

KAYPRO MAINBOARDS

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>K2 With Sockets</td>
<td>1</td>
<td>$20.00</td>
</tr>
<tr>
<td>K2 Populated - Tested</td>
<td>1</td>
<td>$175.00</td>
</tr>
<tr>
<td>K2 with PRO8-2 Mod</td>
<td>1</td>
<td>$299.00</td>
</tr>
</tbody>
</table>

Miscellaneous

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>9&quot; Hard Disk - 20 Pin Header</td>
<td>$2.00</td>
</tr>
<tr>
<td>14&quot; Dual Inline 16 Pin Male (2)</td>
<td>$2.00</td>
</tr>
<tr>
<td>I/O Connector for Keyboard</td>
<td>$1.25</td>
</tr>
</tbody>
</table>

FLOPPY DISK CABLE

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>19&quot; 2x5/4&quot; to 34 Pin Header</td>
<td>$5.00</td>
</tr>
<tr>
<td>RS232 Cable</td>
<td>$15.00</td>
</tr>
</tbody>
</table>

LOW PROFILE IC SOCKETS

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Pin 50 per Tube</td>
<td>$2.50</td>
</tr>
<tr>
<td>14 Pin 25 per Tube</td>
<td>$1.50</td>
</tr>
<tr>
<td>16 Pin 25 per Tube</td>
<td>$2.25</td>
</tr>
<tr>
<td>18 Pin 25 per Tube</td>
<td>$2.50</td>
</tr>
<tr>
<td>20 Pin 25 per Tube</td>
<td>$2.00</td>
</tr>
<tr>
<td>24 Pin 20 per Tube</td>
<td>$2.50</td>
</tr>
<tr>
<td>40 Pin 10 per Tube</td>
<td>$1.60</td>
</tr>
</tbody>
</table>

POWER SUPPLIES

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.5VDC - .5A Regulated</td>
<td>$4.25</td>
</tr>
<tr>
<td>12VDC - 3A</td>
<td>$2.15</td>
</tr>
<tr>
<td>6VAC - .95A</td>
<td>$8.00</td>
</tr>
</tbody>
</table>

LINEAR

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condor Model CP731 DC Power Supply</td>
<td>$135.00</td>
</tr>
<tr>
<td>Condor Model CP573 DC Power Supply</td>
<td>$125.00</td>
</tr>
<tr>
<td>14KV DC/DC TV Power Supply</td>
<td>$4.50</td>
</tr>
<tr>
<td>5V-6A, ± 25V-3A</td>
<td>$7.50</td>
</tr>
<tr>
<td>2422 FDC &amp; 8&quot; Disk</td>
<td>$99.00</td>
</tr>
<tr>
<td>5V, 10A</td>
<td>$20.00</td>
</tr>
<tr>
<td>28V - 1.3A</td>
<td>$25.00</td>
</tr>
</tbody>
</table>

SWITCHERS

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Band 24VDC Power</td>
<td>$20.00</td>
</tr>
</tbody>
</table>

FLOPPY DISK DRIVES

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mitsubishi M4851 DSDD</td>
<td>$399.00</td>
</tr>
<tr>
<td>Mitsubishi M4853 DS Quad</td>
<td>$198.00</td>
</tr>
<tr>
<td>Qume Trak 142 5'&quot; DSD</td>
<td>$125.00</td>
</tr>
<tr>
<td>Tandon TM848-1 8&quot; SSD</td>
<td>$175.00</td>
</tr>
<tr>
<td>Tandon TM848-2 8&quot; SSD</td>
<td>$275.00</td>
</tr>
</tbody>
</table>

TERMINALS

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter Tube 5 Smart</td>
<td>$225.00</td>
</tr>
<tr>
<td>InteColor 3602 Smart</td>
<td>$395.00</td>
</tr>
</tbody>
</table>

X-Y PLOTTERS

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 x 17 from $300.00</td>
<td></td>
</tr>
</tbody>
</table>

MISCELLANEOUS

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soldering Irons - 30W</td>
<td>$5.95</td>
</tr>
<tr>
<td>5 Blade Muffin Fans</td>
<td>$7.50</td>
</tr>
<tr>
<td>Joystick 4 Switches 1&quot; Knob</td>
<td>$5.50</td>
</tr>
<tr>
<td>Part Boxes - 6 Compartments</td>
<td>$2.00</td>
</tr>
<tr>
<td>NLS 4/5 Digital Panel Meter LCD</td>
<td>$425.00</td>
</tr>
<tr>
<td>5VDC Power</td>
<td>$25.00</td>
</tr>
<tr>
<td>Varactor Tuner</td>
<td>$425.00</td>
</tr>
<tr>
<td>All Band 24VDC Power</td>
<td>$10.00</td>
</tr>
<tr>
<td>Handheld 3½ Digit DMM LCD</td>
<td>$50.00</td>
</tr>
<tr>
<td>ZMF-120 Uninterruptable Power Supply</td>
<td>$25.00</td>
</tr>
<tr>
<td>Naughty Boy</td>
<td>$40.00</td>
</tr>
</tbody>
</table>
The S-100 Bus

By Dave Hardy

As I mentioned last month, multi-processing is probably the greatest advantage of the S-100 (IEEE-696) bus. The S-100 bus performs its multi-processing magic by using something called Temporary Master Access (TMA).

An integral part of the IEEE-696 standard, TMA is the process that lets another processor take over the system to perform data transfers between itself and other devices attached to the S-100 bus.

How The Magic Works

The S-100 bus is designed to support up to 16 processors, and it selects between them by using a simple Request/Acknowledge/Transfer protocol. The whole process works like this:

First, the processor that wants to take temporary control of the bus issues a request by asserting the S-100 bus's HOLD* line. It then waits for the permanent bus master to finish its current bus cycle and respond with the Hold Acknowledge signal, called pHILDA.

Once it sees the pHILDA signal, the requesting processor turns off the permanent master's bus drivers, and turns on its own drivers to take control of the address, data, control, and status busses. After it has done all that, it is in total control of the S-100 bus, and can function exactly as if it were the only processor on the bus.

At this point, the requesting processor is called the Temporary Master because it controls the entire S-100 bus, but it must return control to the Permanent Master when it has finished using the bus.

The point of all of this (in case you missed the June/July issue), is that TMA makes it possible for several separate computers to operate in the same S-100 box at the same time, and share all of the S-100 frame's peripherals, including hard disks, printers, modems, and just about anything else that can be plugged into the S-100 bus, without any hardware conflicts.

Several manufacturers make temporary master processor boards designed to operate in the S-100 TMA environment. These complete single-board computer systems are called slave boards, because of their relationship to the S-100 master processor.

Virtually all of these slave boards are designed to run a CP/M-like operating system called TurboDos that allows each of the slave processors to appear to its operator as a single-user CP/M machine. Because of the low cost of these boards (typically $300-$600), a powerful multi-user system can be built in the S-100 bus at a fraction of the cost of any other method.

Some Alternatives

If you don't like multi-processing, the S-100 bus can also be set up as part of a network. Unlike multi-processing, where several CPUs live in the same S-100 box, networking requires just one processor, and lets your S-100 machine share files and data with other computers that may be miles away.

There are several networking schemes that can be used on an S-100 machine. Probably the most versatile is EtherNet, a high-speed coaxial network used on many large computers. Although it is more expensive than multi-processing, EtherNet has the advantage of letting your machine talk to other computers that are not plugged into the S-100 bus.

S-100 Toolbox

Like all bus-oriented, plug-in systems, S-100 boards and frames have connectors that sometimes get dirty. In fact, about 50 percent of all of the S-100 problems I have had in the last 10 years have been because of dirty edge connectors on S-100 boards, or damaged sockets in an S-100 frame.

There's not much you can do about a bad socket (except replace it), but edge connectors can be cleaned with some isopropyl alcohol and a rag. Because the pads on most edge connectors are plated with a few mils of gold, it is not a good idea to simply rub off the dirt and tarnish with an eraser.

Having been the victim of bad edge connectors that had been erased by well meaning users, even the idea of using a pencil eraser on a gold-plated connector used to make me cringe, not only because of the damage to the gold plating, but also because of the oily residue left behind by the eraser.

But about a year ago, I discovered that one eraser, the Mars Staedtler Rasor 52730, not only cleans the connector without damage to the gold plating, but also leaves no residue.

Since that time, I have used the Mars eraser with excellent results, and have decided to make it a permanent resident in my S-100 toolbox.

Handy Circuits

Readers have sent me a few examples of simple circuits that can be used to troubleshoot sick S-100 machines. Future columns will be devoted to these suggestions, but the following circuit is mentioned here because it is so simple that I have overlooked it in the past (just assumed that everybody has one, I guess).

Figure 1 shows a super-simple "pulse catcher" that you can use to see a pulse on any TTL line. To catch a negative-going pulse, just connect an inverter in front. I have logic probes and analyzers all over the place here, and have just never thought of mentioning what is probably the simplest piece of test equipment around.

In spite of its simplicity, this circuit (or any pulse catcher/logic probe) is often invaluable.

Next Time

Next time, we're going to get into the technical aspects of multiple processors on the S-100 bus, including the IEEE-696 bus arbitration procedure, and a circuit that will let older S-100 boards use bus arbitration; plus, a simple battery backup circuit for an S-100 machine, and some reader feedback.
Borland Introduces the Laws of TURBO DYNAMICS™

Laws That Work Like Magic. Whether considering technological excellence, or innovation in areas such as pricing, not copy-protection, licensing agreements, site licenses, 60 day money-back guarantee—Borland is clearly recognized as the software industry leader. The following three laws of “Turbo Dynamics”™ exemplify our pledge for excellence.

1ST LAW: SPEED, POWER AND PRICE.
Borland products are known to be fast, powerful and to deliver an incredible price-performance ratio. We only believe in absolutely superb software at rock bottom prices.

2ND LAW: NOT COPY-PROTECTED SOFTWARE AND REASONABLE LICENSING AGREEMENTS.
We will always offer not copy-protected versions of our software. Also, our licensing agreement is now so simple that even a child can understand it.

3RD LAW: 60 DAY MONEY-BACK GUARANTEE.
This third law is actually a first in the industry! We are so sure that you will love our software that all of our products now come backed with a 60 day money-back guarantee. No questions asked.

Turbo Dynamics Applies to Turbo Pascal.
Borland's Pascal family of products is growing by leaps and bounds. You can now join hundreds of thousands of users and enter the world of Turbo Pascal programming. And remember, all three laws of Turbo Dynamics apply to all Borland products.

TURBO PASCAL™ $69.95
The industry standard. With more than 500,000 users worldwide Turbo Pascal is the industry's de facto standard. Turbo Pascal is praised by more engineers, hobbyists, students and professional programmers than any other development environment in the history of microcomputing. And yet, Turbo Pascal is simple and fun to use. Free spreadsheet included on every Turbo disk with ready-to-compile source code. Options: We offer the exciting Binary Coded Decimal (BCD) option for your business applications as well as an 8087 option for your number-crunching applications at a very low charge. Please refer to the coupon.

PORTABILITY, Turbo Pascal is available today for most computers running PC-DOS, MS-DOS, CP/M-80 or CP/M-86. Jeff Duntemann, PC Magazine: "In its simplicity it achieves an elegance that no other language compiler has ever displayed."

TURBO DATABASE TOOLBOX™ $54.95
The Turbo Database Toolbox is the perfect complement to Turbo Pascal. It contains a complete library of Pascal procedures that allows you to sort and search your data and build powerful applications. It's another Borland set of tools that will give the beginning programmer the expert's edge. Get started right away: free database included on every Toolbox disk is the source code to a working database which demonstrates how powerful and easy to use our search system. Turbo-Access, really is. Modify it to suit your individual needs or just compile it and run. Remember, no royalties!

TURBO GRAPHIX TOOLBOX™ $54.95
High resolution monochrome graphics and window management for the IBM PC. The Turbo Graphix Toolbox will give even the beginning programmer the expert's edge. It's a complete library of Pascal procedures and functions. Tools that will allow you to draw and hatch pie charts, bar charts, circles, rectangles and a full range of geometric shapes. Procedures that will save and restore graphic images to and from disk. And much, much, more. You may incorporate part or all of these tools in your programs and yet we won't charge you any royalties. Best of all, these functions and procedures come complete with commented source code on disk ready to compile.

TURBO TUTOR™ $34.95
From start to finish in 300 pages. Turbo Tutor is for everyone from novice to expert. Even if you've never programmed before Turbo Tutor will get you started right away. A must. You'll find the source code for all the examples in this book on the accompanying disk ready to compile. Turbo Tutor might be the only reference on Pascal and programming you'll ever need.

BORLAND INTERNATIONAL
4585 Scotts Valley Drive, Scotts Valley CA 95066
Phone (408) 438-8400  Telex 172573

Copyright 1985 Borland International BI-101
Turbo Pascal, Turbo Graphix, Turbo Toolbox, Turbo Graphics, Turbo Tutor and Turbo Dynamics are trademarks of Borland International, Inc.
In The Public Domain

By Stephen M. Leon

Following Sol Libes’ footsteps is hard enough. Writing a column in an area of computing he created is doubly difficult. But following Sol and disagreeing with him rather vehemently — that’s really tough!

By Way Of Introduction...

I came into the world of computing with the TRS-80 model one. I never had an Altair or an Imsai, so I have to use a special calculator to add or subtract in hex. My TRS-80 is still sitting in the den with its Omikron mapper, eight inch drives, Holmes speedup, and 58K of memory. That 58K modification was made very early in the game so I could learn Pascal.

Next to the TRS-80 is a CompuPro with 512K of memory, two eight inch and two five inch drives, and the ability to run CP/M 80, CP/M 86 and MS-DOS. A Wyse 100 terminal sits on the desk for the CompuPro, and on the other side of the desk is an IBM PC with 640K, a 20 meg hard disk, two floppies, an Everex color board, and a black and white monitor. A fourth monitor in the den seemed out of place, so a Sears color TV-monitor combination gives the PC its color.

In mid-1982, I volunteered to help Hank Kee edit the SIG/M library. We were then doing volume 65. Today we are up to volume 230. That comes out to about 40 megs of software since I started the job! I try to test every program we release, and when I can’t get one to work, I indicate that in the documentation. If you consider that we release only 25 percent of what is submitted, one might call me a non-professional professional software tester.

What Comes In And Goes Out

Not all of our programs come directly from the authors. Some of them are sent to us by sysops from almost every country.

How do we decide what goes into the library? First, the program has to work, or be an interesting enough attempt to justify its release so perhaps someone else can get it to work. It doesn’t have to be perfect. Nor does it have to meet a test of universality (or we never would have released an eight volume Yale Bright Star catalog, SIG/M 31-38). It should be, at least, semi-professional. It can be copyright protected and released to the public domain, but we do not issue anything that has a copyright without the author’s release.

A distinguished writer, Herbert A. Simon, defined a management philosophy that I consider the goal of SIG/M software. We aim not to be perfect, but to satisfy. Much of our software is excellent, but some of it could certainly use improvement! We try to provide source code with the program so users can upgrade. More in another column on those who carry upgrading to the edge of sanity.

I disagree with Sol and Hank with respect to distribution of freeware. A lot of people put considerable time and effort into SIG/M. We are not paid. In fact, it costs us both time and money to do this job. If someone wants to engage my services in a commercial venture, I expect to be paid for it. The same holds true for SIG/M. Therefore, SIG/M does not knowingly release any software for which the author seeks a donation or any kind of payment for registration or a manual, etc. We suggest to the author that he go somewhere else for free distribution. More on this in subsequent articles.

Enough background. Let’s get down to serious business.

Our McIntosh Is A REC

The winner of the 1985 Computer Hobbyist of the Year award was Professor Harold V. McIntosh of the Microcomputer Applications Department of the Institute of Science at Puebla University, Puebla, Mexico.

If you use either CP/M 80 or CP/M 86 and do your homework, you’ll know why the Trenton Computer Festival committee honored him. McIntosh developed a tool called REC — Regular Expression Compiler — which can perform mind boggling tasks.

REC is currently spread over eight volumes (SIG/M 164-167, 173 and 213-215). It is a compiler-compiler. In any area of formula or regularity, you can have it write a program to perform an assigned task. One of the samples that Prof. McIntosh provided included a program that translates CP/M 80 ASM code to CP/M 86 A86 code. Another converts ACT assembly language to A86 code. McIntosh rewrote the string find program FIND, gave us a whole series of DUMP and binary compare programs, redid DDT, created a new version of HELP, allowed us to run squeezed or unsqueezed programs from inside a library, etc., etc., etc.

At the Trenton show Rich Conn (an award recipient from another year) offered 50 volumes of software for public domain release. We told him to hold off. Why? Because, while these volumes include everything from word processing on up, they are all in ADA and require a micro computer compiler that’s not available yet.

McIntosh has programs to translate LISP and C to REC and then to 80 or 86 code. With REC, an ADA programmer can translate ADA to any other language. With REC, you can translate dBASEII or III into C or some other language without using the high priced conversion programs that are starting to appear on the market. The possibilities of REC are limited only by the skill of its user.

dBASE Applications As Learning Tools

The SIG/M library contains a wealth of tools for the experienced programmer. It also contains lots of help for the rest of us. dBASEII programs are very popular. Our first dBASE release was a demonstration mailing list program based on Adam Green’s book (SIG/M 110).

In a later volume we released the operational mode of this mailing list program (SIG/M 199). To keep track of disk orders and shipments we wrote a dual data base dBASE program (SIG/M 129). We later released updating modules to the inventory data.

(continued on page 15)
Do it yourself...

A new magazine for publishers, graphic artists, editors, writers and others interested in using personal computers to publish information electronically or on paper.

This is the magazine you've been waiting for. Issues contain tutorials on using page make-up software, techniques for integrating text and graphics, tips on producing more appealing graphics and typography, and helpful information about data communications and electronic publishing.

The bi-monthly magazine covers the latest products for the newly-emerging desktop publishing market, including laser printers, scanning equipment, page make-up software, archival storage devices, and topics of interest to people who do their own electronic or paper publishing.

The cover price is $5, but you can save $11 by ordering a subscription for $24 and paying with your order. You'll get seven issues for the price of six! And there's no risk -- you can cancel after the first issue for a full refund!

We also publish User's Guide to CP/M, devoted entirely to CP/M computer users, and PC USER, a new magazine for users of PC-compatible computers and MS-DOS software. Each magazine is $22 for a six-issue subscription. Send payment now and get seven issues for the price of six!

DESKTOP PUBLISHING: for anyone who publishes information using computers.
base for dBASE version 2.4 (SIG/M 155). The inventory program is an excellent tutorial as well as a working program.

A different inventory program was released on SIG/M 155. On that same disk are checkbook, mailing list, and periodical tracking programs. On volumes 199 and 200 we put the SIG/M Property Manager, a 320K dBASE program for tenanted real estate. It does everything from sending rent bills to preparing financial statements.

SIG/M 223 contains programs to do your genealogy, balance your check needs of my office or of three programs for scholars and others to keep track of reference material (SIG/M 230).

I have written many of the SIG/M dBASE programs to meet specific needs of my office or of SIG/M. But in the code, we try to show users how they can modify the programs for their own applications.

The PC-DOS/MS-DOS Crowd

The dBASE programs run under CP/M 80, CP/M 86 or MS-DOS. Property Manager is available in IBM format (PC/BLUE 75) as are Mail Manager (PC/BLUE 34) and Inventory Manager (PC/BLUE 43).

While I'm on the subject of the PC/BLUE library, let's take a look at volume 73. There are no programs on it, just 68 articles on "Tips, Notes and Techniques for the IBM PC." One day you may be looking for just the information in one of those articles (see Figure 1).

SIG/M and PC/BLUE printed catalogs can be ordered from SIG/M, Box 97, Iselin, NJ 08830. The price is $3 each ($4 foreign). SIG/M disks are $6 each ($9 foreign) from SIG/M. PC/BLUE disks are $7 ($10 foreign) from the New York Amateur Computer Club, Box 100 Church Street Station, New York, NY 10008. Software is also available via the world wide public domain software distribution network.

Next Month

Next issue the topic will be how to generate your own nuclear power — i.e. public utilities. In the meantime, if you have written some software that others may find helpful, useful, or interesting, why not contribute it to the SIG/M or PC/BLUE libraries?

---

**Figure 1 - PC/BLUE Volume 73**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBERT</td>
<td>.NL Personal Editor</td>
</tr>
<tr>
<td>BARRER</td>
<td>.NL Why Join A User Group</td>
</tr>
<tr>
<td>BECKLEY</td>
<td>.NL Processing TAB Characters</td>
</tr>
<tr>
<td>BRIDGES</td>
<td>.NL Interfacing BASIC &amp; Machine Code</td>
</tr>
<tr>
<td>BRIDGES</td>
<td>.NL Defaulting ECHO Off - Batch Files</td>
</tr>
<tr>
<td>CHERTOK</td>
<td>.NL Microprocessor Comparisons</td>
</tr>
<tr>
<td>CHIMENE</td>
<td>.NL A Programming Language Approach</td>
</tr>
<tr>
<td>CONSIGLO</td>
<td>.NL Using The Basic FIELD Statement</td>
</tr>
<tr>
<td>CORTESI</td>
<td>.NL Organizing Pascals Diskettes</td>
</tr>
<tr>
<td>CRIBBS</td>
<td>.NL BASIC Screen Input Routines</td>
</tr>
<tr>
<td>CRL</td>
<td>.COM /</td>
</tr>
<tr>
<td>CRL</td>
<td>.NL Change CURSOR Shape in DOS</td>
</tr>
<tr>
<td>CRRS</td>
<td>.COM /</td>
</tr>
<tr>
<td>CUMMING</td>
<td>.NL Quick Start - DOS 2.0+ - IBM PC</td>
</tr>
<tr>
<td>DAVIS</td>
<td>.NL Running DOS Commands in BASIC</td>
</tr>
<tr>
<td>DIXON</td>
<td>.NL IBM Pascal Compiler v2.00</td>
</tr>
<tr>
<td>EXPNDTAB.ASM</td>
<td>.NL Expand TAB Character Into Blanks</td>
</tr>
<tr>
<td>FOUGGERI</td>
<td>.NL WP in Personal Editor</td>
</tr>
<tr>
<td>FOUGGERZ</td>
<td>.NL Understanding L000</td>
</tr>
<tr>
<td>GASTON</td>
<td>.NL Tracking Time Spent Programming</td>
</tr>
<tr>
<td>GASTEON</td>
<td>.NL Determining BASIC Environment</td>
</tr>
<tr>
<td>GINN</td>
<td>.NL Focus on Word Proof</td>
</tr>
<tr>
<td>HAMMOND</td>
<td>.NL Convenient Timing in BASIC Programs</td>
</tr>
<tr>
<td>HARRINGTON</td>
<td>.NL Using the MORE Filler Command</td>
</tr>
<tr>
<td>HARRIS</td>
<td>.NL Color in Hi Res Graphics</td>
</tr>
<tr>
<td>HERZFELD</td>
<td>.NL Shaping Your Cursor in DOS</td>
</tr>
<tr>
<td>HO</td>
<td>.NL Focus on Word Processing</td>
</tr>
<tr>
<td>HOSKINS</td>
<td>.NL Extended Keyboard Control Device</td>
</tr>
<tr>
<td>JEWELL</td>
<td>.NL Disk Copying on the IBM XT</td>
</tr>
<tr>
<td>JORDAN</td>
<td>.NL Wooden Protocol</td>
</tr>
<tr>
<td>KRUSSGEL</td>
<td>.NL Text Mode Display Compatibility</td>
</tr>
<tr>
<td>LAURINOL</td>
<td>.NL Installing Half Height Drives</td>
</tr>
<tr>
<td>LAVIGNE</td>
<td>.NL Cleaning Disk Drive Heads</td>
</tr>
</tbody>
</table>

**Name**  **Description**

| LEHMANN    | .NL The IBM PC Color Display           |
| MACK       | .NL Using Random Files In BASIC        |
| MARKS      | .NL Recovering Data From Bad Directory |
| METZGER    | .NL Super Color from IBM PC            |
| MINASI     | .NL Dumping Screen GRAPHICS in BASIC   |
| NISSENF    | .NL BASIC COMMON Statement             |
| NISLEY     | .NL Don't Degauss Your Diskettes       |
| PKEY       | .NL BAS referenced by Hoskins.NL       |
| PELTO      | .NL Using The CTY Command              |
| PINTVE     | .NL Installing Memory Chips in PC      |
| REED       | .NL Windowing in BASIC                 |
| RICE       | .NL Storing Numbers in Random Files    |
| RICHTNER   | .NL Word Proofing Review               |
| ROYDE      | .NL COBOL Software Review              |
| ROSE       | .NL Making Life Easier with Macros     |
| SCHIEVE    | .NL Computer Based Instruction        |
| SCHNELL    | .NL Alternate Colors in Medium Res Mode |
| SCHNELL1   | .NL Graphics on the PCjr.              |
| SCREEN     | .NL BAS referenced by Crebs.NL         |
| SCRIPTPC   | .NL Words from the Author              |
| SHARP      | .NL PCjr - First Impressions           |
| TARKOFF    | .NL A View of APL                      |
| TESTTAB    | .NL BAS referenced by Beckley.NL       |
| THOMAS     | .NL POCKing Around on the Fixed Disk   |
| TOBI        | .NL Purpose of User Groups             |
| WATKINS    | .NL DOS 2.0 Buffers                    |
| WEBER      | .NL Review of FFS File                 |
| WEIL       | .NL On Eternal Golden Braid            |
| WESTEND    | .NL BASIC Sort Using A RAM Disk        |
| WHITE      | .NL An Introduction to PC Diskettes    |
| WHITEHEAD  | .NL Disabling Break in a BASIC Program |
| WOLPERT    | .NL The Modem Package                  |
| WORDPROC.MAC| .NL BAS referenced by Rose.NL          |
| YOUNG      | .NL Notes on Decathlon                 |
C'ing Clearly

By Ron Miller

You hear lots of reasons for switching to the C language: program speed, structured programming, portability (mainly of interest to software designers), ease in building libraries, testing, and planning grammar. Doesn't have a chance to keep routines that it's hard for him to keep correspondence and filing so programmer with a crackerjack secrecy.

That's where C comes in. Writing in C is like being an assembly language programmer. Though the assembly language programmer doesn't have a chance to forget what really is happening, he's often so busy rewriting loops, designing tests, and planning elementary routines that it's hard for him to keep track of the bigger picture.

That's where C comes in. Writing in C is like being an assembly language programmer with a crackerjack secretary.

The secretary takes care of the routine correspondence and filing so you can concentrate on the really critical issues.

Such is true, of course, for all compilers and interpreters. C, however, never lets you forget that your secretary is there and that the daily schedule doesn't appear on your desk by spontaneous generation.

Bailing Out

To illustrate the literal-mindedness of C and offer a technique to use for your own programs, let's use C to divert character-in (conin) calls in CP/M.

Suppose you wanted to test every new character from the keyboard for a certain input, and when that character appeared, you made something special happen.

You'll need to test all characters as they come into the CPU, before Microsoft, DRI, Aztec, or whatever you're running gets its mitts on them.

To keep things practical, let's consider a "longjump" routine that lets you bail out of a complex program, when going on means disaster and cold booting means starting over from scratch.

Wouldn't it be nice to be able to leap tall nested loops in a single stroke, landing at some safe point in the program?

You could test all characters read from the keyboard; if one is ASCII 27 (ESC), reset the stack pointer and the CPU and — voila! — you're back again at your safe point. It would be like beaming from here to there in Star Trek.

Pulling off this feat involves two separate projects: designing the long-jump routines and crafting the diversionary routine. Let's do the diversion first.

Diverting Conin

At the head of our program we will do a little pointer work that depends upon two characteristics of CP/M systems known to all hackers: first, that the BIOS routines in high memory are headed by a series of three-byte jump statements (called a jump table). The second jump is to the conin (get a character) routine.

Second, we also know that the address of the warm boot jump is stored at address 01H.

Since this is C, we can treat the BIOS routines like functions we have written ourselves. We can use pointers to alter the parameters in the functions and even use a function pointer to alter the flow of a BIOS call. As any C programmer knows, a pointer is

---

"C/80... the best software buy in America!"

Now available in MS-DOS

In reviews published worldwide the amazing $49.95 C/80 from The Software Toolworks has consistently scored at or near the top — even when compared with compilers costing ten times as much!

The optional C/80 MATHPAK adds 32-bit floats and longs to the C/80 3.0 compiler. Includes I/O and transcendent function library all for only $29.95!

C/80 is only one of 41 great programs each under sixty bucks. Includes: LISP, Ratfor, assemblers and over 30 other CP/M® and MSDOS programs.

For your free catalog contact:

The Software Toolworks
15233 Ventura Blvd., Suite 1118
Sherman Oaks, CA 91403 or call 818/986-4885 today!

CP/M is a registered trademark of Digital Research.
just a shorthand way to have some-body else — the compiler — access
bytes in the memory.
We're going to change the conin
jump address so a jump to conin will
actually jump to our own little charac-
ter checking routine. Then our routine
will call the real conin and it will
return the character to our little check
routine. We, of course, check to see if
conin sent us the character we want.
If so, we do whatever devilment we've
planned. Otherwise, we simply return
the character to the routine which
called conin in the first place.

It's only in this character check
function (called "swerve" below) that
a bit of assembly language must be
used, since CP/M expects console
characters in the A register. Notice
how casually the patch is made.

We're going to change the conin
body else — the compiler - access
characters in the A register. Notice
called conin in the first place.
how casually the patch is made.

It's only in this character check
function (called "swerve" below) that
a bit of assembly language must be
used, since CP/M expects console
characters in the A register. Notice
how casually the patch is made.

We're going to change the conin
body else — the compiler - access
characters in the A register. Notice
how casually the patch is made.

It's only in this character check
function (called "swerve" below) that
a bit of assembly language must be
used, since CP/M expects console
characters in the A register. Notice
how casually the patch is made.

We're going to change the conin
body else — the compiler - access
characters in the A register. Notice
how casually the patch is made.

It's only in this character check
function (called "swerve" below) that
a bit of assembly language must be
used, since CP/M expects console
characters in the A register. Notice
how casually the patch is made.

We're going to change the conin
body else — the compiler - access
characters in the A register. Notice
how casually the patch is made.

It's only in this character check
function (called "swerve" below) that
a bit of assembly language must be
used, since CP/M expects console
characters in the A register. Notice
how casually the patch is made.

We're going to change the conin

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

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

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

- Ultra-fast compilation, linkage and
  execution that produce directly
  executable 8080/280 CP/M command
  files.
- A comprehensive debugger that
  traces program execution and
  interactively displays both local and
  external variables by name and
  proper type.
- Dynamic overlays that allow for run-
time segmentation of programs too
  large to fit into memory.
- A 120-function library written in both
  C and assembly language with full
  source code.

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

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

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

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

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

BYTE Magazine placed BDS
C ahead of all other 8080/280 C
compilers tested for fastest
object-code execution with all
available speed-up options in use.
In addition, BDS C's speed of
compilation was almost twice as
fast as its closest competitor
(benchmark for this test was the
Sieve of Eratosthenes).

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

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

BDS C is designed for use with CP/M-80
operating systems, version 2.2, or higher. It is
not currently available for CP/M-86 or MS-
DOS.
work because playing with C has taught me that HL is the return register in Z80 implementations of C.)

Booting Yourself Back In Line

Tinkering with the warm boot jump statement is just good manners. We need to restore the conin jump statement after the program is finished, or else we'd continue our checking right into the next program.

To do this I simply change the warm boot jump address so it really jumps to a routine that patches the conin jump back to its original state. So any jump to 00H restores the table before you actually get the warm boot. Figure 1 shows the details.

Notice that a global variable "conback" is just an address for the assembler and can be treated as such in an assembly language routine. Automatic variables (variables within functions) can't be seen by other functions.

The Longjump Routines

The functions which alter the jump table are written in assembly language. Since we are operating in C, including assembly language is pretty easy.

Figure 2 illustrates the process for Software Toolworks C/80. C/80 creates 8080 code and passes variables by pushing them in reverse order onto the stack. (Slight alterations may be needed to load the DE and HL registers in other versions of C.) Please note that the compiler takes care of the addresses, the origin, etc.

What I did was devise a function ("setjmp") that stores the present stack and CPU data in a six-byte array. A call to setjmp can be placed at the point or points in the program to which we wish to beat our hasty retreat.

Another function ("longjmp") reads the array and restores the stack and CPU to the prior condition. Since I'm not evangelizing on assembly language, I'll just offer the routines. My apologies for not writing in Z80 mnemonics, but C/80 can't digest them so I fool it with "DB" declarations. If you aren't into assembly language you'll have to take this section on faith.

If Software Toolworks wonders whether I resent having to use 8080 code — I do, I do, I do.

Applications

The program is pretty trivial, designed merely to demonstrate the jump routines. This type of diversion, however, has numerous applications.

A public domain screen dump program from Micro C uses an assembly language variant of this technique. I assume that Borland's Sidekick does, too. CP/M BDOS does it with certain control characters.

A C program can easily be designed to run in high memory to act as a filter for any piece of commercial software. I've patched Perfect Writer and Perfect Calc to display directories.

---

Figure 1 - Patching the BIOS Jump Routine Back to its Original State

```c
#define OOPS 27 /*or whatever you wish to use*/
#define CLNCR 25 /*clear screen: Kaypro this time*/
#define WBOOT 1
#include <printf.c> /*<stdio.h> in other implementations*/
#include <scant.c>

unsigned wbadd,*utilptr,*wbptr;
char conback,jbuf[6],swerve(),*funct),restore();

main()
{
    char line[80];
    utilptr = WBOOT; /*get addr of warm boot jump statement*/
    wbptr = +utilptr; /*get the pointer to the warm boot routine in bios*/
    wbadd = +wbptr; /*squirrel away pointer for restoration later*/
    *wbptr = restore; /*ensure that all boots go through restore()*/
    utilptr = +utilptr; /*get the pointer to conin routine in bios*/
    funct = +utilptr; /*redirect the function pointer to conin routine*/
    *utilptr = swerve; /*ensure that all console calls pass through swerve()*/
    setjmp(buf); /*store CPU state, stack pointer, and top of stack*/
    printf("Enter Jump and return"); /*trivial: here's where the real*/
    printf("Enter something\n"); /*program belongs instead*/
    while(1) scanf("%s",line); /*scanf calls bdos, which calls bios*/
    program loops until "C*/
}

jumbotron() /*test the conin characters for "OOPS"*/
{
    (*funct)(); /*call (not jump to) conin and it returns through here*/
    *asm
    LDA conback /*since dos expects the character in A*/
    *endasm
    ST X conback /*return by conin is in register AX*/
    *endasm
    if((conback=OOPS) longjmp(buf); /*if OOPS, then longjump*/
    else{
        *asm
        LDA conback /*since dos expects the character in A*/
        *endasm
    }
}

jumbotron() /*restore bios jump table before warm boot*/
{
    *wbptr = wbadd; /*restore warm boot pointer*/
    *utilptr = funct; /*restore conin pointer*/
    funct = +funct; /*redirect function pointer to warm boot routine*/
    *funct); /*call--i.e., jump to--warm boot*/
    so
    setjmp(buf); /*setjmp(buf)*/
}
```
Figure 2 - Assembly Language retreat for Software Toolworks' C/80

Of course, there's nothing special using a diversion.

about conin. With a bit of ingenuity, practically anything can be redirected almost anywhere else. BDOS doesn't have to know that it's not getting those characters from the file it called.

The point is that the C programmer can tinker with the BIOS about as easily as he can write a standard program. It's the physically-oriented nature of C that encourages him to remember that there is a machine under there.

I remember about two years ago when, with fear and trembling, I first started looking into Basic. I played with PRINT for at least three weeks without thinking much about the difference between real numbers and integers. But no one could use C for 13 seconds without being aware of the difference.

Of course, there's nothing special using a diversion.

about conin. With a bit of ingenuity, practically anything can be redirected almost anywhere else. BDOS doesn't have to know that it's not getting those characters from the file it called.

The point is that the C programmer can tinker with the BIOS about as easily as he can write a standard program. It's the physically-oriented nature of C that encourages him to remember that there is a machine under there.

I remember about two years ago when, with fear and trembling, I first started looking into Basic. I played with PRINT for at least three weeks without thinking much about the difference between real numbers and integers. But no one could use C for 13 seconds without being aware of the difference.
**FLOPPY CONTROLLERS, DISK EMULATORS, SOFTWARE**

WE HAVE THEM FOR YOUR XEROX 820, BBI, OR KAYPRO.

EMERALD MICROWARE BRINGS YOU THE SUPPORT THAT YOU NEED TO GET THE BEST VERSATILITY AND EFFICIENCY FROM YOUR SINGLE BOARD MICROCOMPUTER.

### X120 DOUBLE DENSITY CONTROLLER BOARD

Unsurpassed double density operation, designed for easy mounting on the Xerox 820-1 (or BBI). Uses the 2793 double density disk controller.

- Runs any combination of 5 ¼" or 8" disk drives, up to four at a time.
- Standard 34 and 50 pin Shugart compatible disk drive connectors on board, as well as returning signals to connector on main board.
- On board circuitry handles clock and data separator switching from drive select.
- Composite video adapter on X120 board.
- 3 ½" x 4 ½" high quality, solder masked, double sided board with manual and schematic.

Bare board ........................................ $26.00
Board assembled and tested .................. $120.00
Board with full kit ............................... $95.00
WD2793 controller chip ........................... $24.00
Hard to find parts set ........................... $12.00
See below for board with rom packages.

### XPRO ROM SET by Micro Cornucopia

Turns your Xerox 820-1 into a Kaypro compatible with superior speed and versatility.

- Runs 63K system for more available TPA
- Fast video and parallel keyboard drivers
- Auto-boot capability, boots from 8" or 5 ¼"
- Rom monitor functions with menu of options
- Centronics and serial printer drivers included, selectable using IOBYTE
- 98% software compatible with both Xerox 820 and Kaypro
- Automatically selects disk formats for:
  - Kaypro II - 191K, SSDD, 48 TPI, 5 ¼"n
  - Kaypro 4 - 390K, SSDD, 48 TPI, 5 ¼"
  - Kaypro 8 - 784K, DSDD, 96 TPI, 5 ¼"
  - Xerox 820-1, 62K, SSDD, 48 TPI, 5 ¼"
  - IBM 3740 - 241K, SSDD, 8" (standard Xerox 820)
  - IBM 5370 - 596K (same as BB II)
  - 8” DSDD - 1.212M
- True double sided operation
- Can be operated with a terminal for console device
- Includes disk format program, sysgens, CBIOS source, easy install files, manual, and 2716 roms
- Works with X120 board for double density

XPRO Rom Set ...................................... $49.95

### PLUS2 ROM SET by MICROcode Consulting

Upgrades your Xerox 820-1 to 820-II compatibility.

- Adds inverse and low intensity video capability
- Logical to physical drive mapping
- Enables real time clock
- Has Xerox rom monitor commands
- Works with the X120 board for double density

PLUS2 Rom Set ..................................... $49.95

### OP/M by MICROcode Consulting

At last an inexpensive replacement for CP/M, with full CP/M compatibility, and many enhancements. Includes 5 new system calls, and 9 new transient commands including time/date stamping of files, backup utility, and a menu driven configure program. Runs on Z80 systems, in the same space as CP/M 2.2.

Available as bootable disk (does not require CP/M) for the Xerox 820-1, 820-II, or the Xerox 820-1 with XPRO rom set.

- OP/M for the Xerox with XPRO roms .................. $64.95
- OP/M for the Xerox 820-1 or 820-II ................. $80.00
- OP/M without BIOS (not bootable) .................. $60.00

### SEMIDISK 2 MEGAGBYTE DISK EMULATOR by SemiDisk Systems

Tired of waiting? The SemiDisk ram based disk emulator board is the single, most significant speed improvement that you can make to your system. And it’s BIG enough to get both your program and your working files on the SemiDisk at the same time, no waiting for one or the other to be accessed from a disk drive. Print buffer software included. Low power consumption - picks up its power directly from your system, with and optional battery backup package available. Works on Xerox 820, Kaypro, or any other Z80 based system.

<table>
<thead>
<tr>
<th>Product Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>2MB SemiDisk Board-assembled</td>
<td>$995.00</td>
</tr>
<tr>
<td>Optional battery backup unit</td>
<td>$150.00</td>
</tr>
</tbody>
</table>

### HALF HEIGHT DISK DRIVES

The finest new half height, 5 ¼" disk drives.

<table>
<thead>
<tr>
<th>Product Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panasonic JA561 - DSDD, 48 TPI</td>
<td>$99.00</td>
</tr>
<tr>
<td>Panasonic JA561 - DSDD, 96 TPI</td>
<td>$114.00</td>
</tr>
</tbody>
</table>

### XEROX CABINETS AND ACCESSORIES

Our stock changes frequently due to quantities and availability of these items, please call to check stock and get shipping charges before ordering.

<table>
<thead>
<tr>
<th>Product Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xerox standard computer cabinet w/monitor frame</td>
<td>$95.00</td>
</tr>
<tr>
<td>Xerox 5 ¼&quot; disk drive cabinet</td>
<td>$18.00</td>
</tr>
<tr>
<td>Xerox 5 ¼&quot; disk drive cable</td>
<td>$12.00</td>
</tr>
<tr>
<td>Power connectors for 820 board or PS</td>
<td>$2.50</td>
</tr>
<tr>
<td>Switching power supply for 8&quot; drives</td>
<td>$69.00</td>
</tr>
<tr>
<td>Xerox 820-1 boards, working</td>
<td>$95.00</td>
</tr>
<tr>
<td>Parallel ASCII keyboard (not Xerox) made by Keytronic, typewriter keys only</td>
<td>$25.00</td>
</tr>
</tbody>
</table>

### SHIPPING AND HANDLING (orders under 2 pounds)

<table>
<thead>
<tr>
<th>Type of Order</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shipping and handling</td>
<td>$3.00</td>
</tr>
<tr>
<td>C.O.D. orders</td>
<td>$6.00</td>
</tr>
<tr>
<td>Bankcards (include S&amp;H)</td>
<td>add 3%</td>
</tr>
</tbody>
</table>

### QUANTITY DISCOUNTS AVAILABLE

UNIFORM by Micro Solutions
The solution to the diskette incompatibility problem. This program allows files to be run or transferred back and forth between close to one hundred different computer formats, including 96 TPI-5 ¼", and 8" diskettes. Lists computers by name, with no special technical knowledge required. Transfers between CP/M, MS-DOS, PC-DOS, and TRS-DOS formats.

Versions available from stock for Kaypro, Xerox 820-1, and Xerox 820-1 with XPRO rom set; others available Uniform ........................................ $64.95
Hold the presses! Xerox has resumed production of the 16/8 computer. The old reliable 820-II with the 8086 card has been re-introduced as Xerox's low-end word processor (Spellbinder is the word processor included).

The computer is an 820-II with 16-bit capability, and Spellbinder is a pretty decent WP, so this "low-end WP" is adequate for most people. This means that the 820-II will be around for awhile.

The "new" system now has a disk expansion module (which finally adds support for 5.25" rigid disks) and a low-profile keyboard. Unfortunately, it looks like a bundled system: you can't buy just the 820-II 8-bit system. I don't have prices; contact Xerox for more information.

Spare Parts
It looks like sources for 820 and 820-II boards are drying up fast. Since 820 production has stopped, that supply is expected to diminish. However, 16/8s (upgraded 820-IIs made in-house) should keep that supply going, right? Wrong! The story I've been hearing is that the Xerox manufacturing outlet (XMO) will be closed before year's end. With the XMO doing such a booming business, the question is "why are they closing?"

Xerox has a policy of destroying key parts of a system, so even if you had all the spare pieces, you still couldn't build an identical system. (The only 820 item I've never been able to purchase from XMO is the 8" disk drive case.)

Evidently, word has reached the upper echelon that complete 820s (among other products) can be assembled from the surplus pieces at a big savings. Judging from the XMO's large volume, obviously there are still many people eager to buy. An 820-II now costs about $2800, but you can buy parts for a complete system for under $400. So is it any wonder why the manufacturing outlet is doing a booming business?

Real-Time Clocks
I've had several inquiries about real-time clocks, so I'll deviate from my intended column for a moment to discuss two of them I recently purchased: the Ztime-I and the Optronics MC-1.

Ztime-I
Ztime-I: By Kenmore Computer Technologies (KCT), this real-time clock is based on the National Semiconductor MM58167A. The chip is directly interfaceable to microprocessors and offers timing down to 1000th of a second, (it is certainly not accurate at that level since the system uses a standard 32,768 Hz crystal).

There's no year register in the clock, but with a bit of fancy programming using a couple of the 8 latches available, you can get around this problem.

KCT offers a neat little board for $69 in kit form and $99 assembled. The easy-to-assemble kit comes with 4 chips and a small collection of resistors, capacitors, and diodes. I recommend the kit over the assembled version.

The assembled board plugs into the Z80 socket, and the Z80 plugs into the Ztime-I board. With no trimming capacitor, it is easy to gain/lose a few seconds a day. If you build the kit, get a trimmer — you will not regret it.

Operation of the Ztime-I is simple: all necessary programs, including source, are included for reading and setting the clock. An assembly language programmer will have little trouble, as all clock registers have their own ports (as well as some other control) which sit across 32 Z80 I/O ports.

Optronics MC-1
Optronics MC-1: The MC-1 is a real-time clock based on the Oki MSM5832. The MC-1 comes assembled and is plugged into the parallel port inside the 820. $69 buys you the works. Although it lacks the manual for the chip (KCT includes one), the disk is full of software slanted towards 820 and 820-II users.

Among the date and time programs for the 820 series there is one which acts like an alarm clock on the 820-II (or 16/8). It is cute, although I have yet to find a use for it.

One feature I do like is the write-protect jumper on the board; one cannot accidentally (through an errant program or system crash) change the time in the clock. Just install the jumper when setting the clock and remove it when you are done.

Summing Up
Both boards do the job, and will keep doing it for about three years with the lithium battery. The Ztime-I is easier to install, does not require power supply cables, and does not eat up the parallel port (important if you already have a parallel printer there). However, the MC-1 costs less, has a write-protect jumper, and does not eat up 32 I/O ports. If you have or plan any expansions to your Z80 bus, 32 lost ports could hurt.

QP/M users will also find that Ztime-I requires conversion from BCD to binary, whereas the MC-1 does not. If you pinned my shoulders to the mat and asked me which one I would buy again, I'd say the Ztime-I. Although my pocketbook favors MC-1, that parallel port is already taken on my last non-real-time clock system.

629A Vrs. 644A
820 owners should be aware that there are two versions of the 820 board floating around: etch 1 and etch 2. Look at the part number on the edge of the board: if the last three digits are 629, then you have an etch 1. If the digits are 644, then you have an etch 2.

Etch 1 is the earlier board which has funny video to support Ball monitors and a faulty parallel port (they forgot to ground the odd-numbered pins — something you will have to do before you plug it in a printer). Etch 2 added true double-sided support for the 5.25" drives, changed the floppy disk data/clock circuitry, fixed the video and parallel port layout, and cleaned up power distribution.

(continued on page 23)
According to leading publications, the PC market is in the midst of a full-blown shakeout. IBM, KAYPRO, OSBORNE and most other PC manufacturers are initiating cutbacks and layoffs due to slumping sales. Their increasing advertising dollar brings less & less results. At the same time sales for the ZORBA portable computer, the computer that is only advertised in a few choice periodicals and relies heavily on satisfied user testimonials, are showing a steady increase.

It's no secret, the public is learning what industrial users have known for years. That is that for daily operation of word-processors, spreadsheets and databases, the trendy options are rarely used and the flashy machines that house them are really no faster, no more reliable and much more expensive.

THE PUBLICS LEARNING WHY THE ZORBA IS STILL AND ALWAYS HAS BEEN THE EXPERTS CHOICE.

ZORBA PORTABLE COMPUTER

FEATURES:
- 9" GREEN OR AMBER CRT
- 19 INDEPENDENT, 55 PROGRAMMABLE FUNCTION KEYS
- TWO 400K DSDD DRIVES
- 64K BYTES 150 NS RAM
- C BASIC COMPILER
- IEEE 488 BUS MASTER PORT
- 24.6 LBS
- CPM 2.2 OPERATING SYSTEM
- M80 (L80, LIB80, CRE80) 
- SOURCE CODE OF THE BIOS
- PLUS UTILITIES
- DATA COMMUNICATIONS SETUP PACKAGE
- SERIAL & PARALLEL PRINTER PORT
- DATA COMMUNICATION PORT

$849.00
W/O Bundle

DEALER INQUIRIES INVITED

$1049.00
With Bundle

General Specifications
ZORBA is the lowest cost full featured portable computer. This light weight computer is ruggedly packaged in a convenient carrying case. The case surrounds a strong inner chassis which further protects the Z80A based computer with its double sided double density disk drives, large easy to read 9" display screen and well designed detachable keyboard.

The ZORBA uses CPM, the industry standard operating system, which means that a wide range of existing software is readily available to the user.

Keyboard
Keyboard communicates serially with CPU Detachable with 2 foot coiled cord
36 keys in standard QWERTY format
13 Key Numeric pad
Independent Caps Lock and Shift Lock
55 Software programmable function keys
All keys auto-repeat after 1 second delay
All Standard cursor and terminal control keys

Display
Display Tubing: 9" diagonal, Green or Amber
High resolution display circuitry
60 Hz refresh rate

Specifications
General Mechanical and Electrical
Width -17.5 inches (44.45 cm)
Height - 9.0 inches (22.86 cm)
Depth -16.0 inches (40.64 cm)
Weight - 24.6 pounds (11.1 Kg)
Power - 80-130 VAC or 190-245 VAC
50/60 Hz
170 watts max

CPU Board
Z80A CPU running at 4 Mhz with no wait states
64K bytes of 150 ns RAM (56k after CPM loaded)
16K bytes of EPROM (2732) can be switched in and out by software
128K available for user EPROMS
8275 CRT controller, DMA driven
1783 floppy disk controller, SMC data separator
Bipolar proms configure 10 addresses
Fully structured interrupts prioritized by bipolar proms

Interfaces
- Full asynchronous RS232 port with modem control. Baud rates and data translation and protocol programmable
- Full asynchronous full duplex RS232 port with hardware handshake (for printers). Baud rates and protocol programmable(Serial Printer Port)
- One 8 Bit parallel port with independent strobe and ready lines. Supports Centronics interface with an available adapter cable.
- IEEE 488 Bus Master Port (ie: General Purpose Instrumentation Bus) not Software Supported.
- 21 Standard Software Programmable Baud Rates: 45.5 to 19,200 BPS

The ZORBA PORTABLE COMPUTER is sold by

RENNINI ELECTRONICS, INC.
130 Baywood Avenue, Longwood, Florida 32750
305-830-8886  800-327-7162
When buying 820 parts, be conscious of possible “etch version” specific hardware. The Xerox 820 Software Technical Manual contains both sets of schematics and can be helpful in determining if there are any conflicts.

Thanks to Dan Costello for suggesting this insert and including a beautiful set of “difference” schematics. Although I have schematics for both versions, the hardware differences are not always clear. I have included a description of the differences in a file called 820DIFF.12 on the RQP/M.

RQP/M

What? Did he say RQP/M? Yes, folks. After a long delay due to moving (and destroying) some computer hardware, my RQP/M is finally on-line. Disk space is still limited (the 20-Mbyte system was one indirect casualty), but there is a disk rotation order until full storage capability is restored.

Besides 820 stuff, there is space for Kaypro, Big Board, Little Board, and QP/M users. The number to call is (213) 320-9309; it will answer around-the-clock except when being used by yours truly. I hope many will call with questions, answers, and CONTRIBUTIONS. Of particular interest are sources for 820/820-11/16-8 parts and prices; a file called 820COST.LST contains the latest prices and places that users have submitted on the BBS or I have found in magazines.

Interrupts

Sorry, but I’ve run out of space this time. Next time I’ll make sure to interrupt (sic) any tidbits until after interrupts are discussed.

POWER THAT GOES ANYWHERE!

Single Board Computer

FAST — 6MHz Z80B CPU
POWERFUL — 64K to 256K RAM, 2K to 64K ROM
— 5½" and 8" Floppy Controller, SASI
— 2 RS-232, Centronics Port
FLEXIBLE — 50-pin I/O Expansion Bus.
SMALL — 5½" x 10"

DAVIDGE CORPORATION
292 East Highway 246
P.O. Box 1869
Buellton, CA 93427
(805) 688-9598

Micro Cornucopia, Number 25, August-September 1985
AUTOMATIC TIME/EITE STAMPING OF FILES WITH

QPIM
by MICROCode

QP/M FEATURES
• 100% compatible with CP/M 2.2 and resides in same space
• Uses your computer's hardware/software clock
• 10-15% faster disk read/write
• User-selectable default drive/user area
• Completely replaces BDOS and CCP
• Efficient backup utility
• Requires QBIOS or CP/M 2.x
• 9 new transient commands, 5 new system calls

ATTENTION XEROX AND BIGBOARD OWNERS--NO NEED TO BUY QPIM!

***Soon Available for Kaypro***

MICROCode offers a custom BIOS for Xerox/BigBoard owners (including SWP Dual Density users), on a bootable QP/M disk.

Written exclusively for the Xerox QBIOS:

- KEYMAP: Not just another keyboard configuration program!
- QPIM PACKAGE
- QP/M Features
- SMARTROMS: Add 820-II features to your 820-I/BigBoard-I.
- Provides real-time clock, screen dump, screen/program pause, reset with CTRL/ESC, CRT blanking after 10 minutes of idle time, 7 or 8-bit keyboard mode, serial/parallel printer routines, 820-II (4.0x) compatible vectors, and MUCH MORE! Replaces your socketed ROMS (no soldering!).

COMPLETE QPIM PACKAGE Includes

D
QBackup
QInstalls
QPip
QStat
QSub
TDCNF
QBios:

- Sorted Direcory program. Displays time/date, system files, and LBR directories.
- Copies and verifies only updated source files.
- Installs QP/M on system tracks.
- All Pip functions, plus copies files with current or existing time/date.
- All Stat functions, plus shows or changes file date(s), archive bit, and MORE.
- Replaces SUBMIT and XSUB. Has nesting capability, internal command set (including conditionals), embedded XSUB, and more.
- Creates time/date initialization module (supports software clock, Z-Time or Optronics clock).

plus COMPLETE documentation.

QBios: Available for the Xerox 820-I, Xerox 820-II, BigBoard-I and SWP Dual Density (820-I/820-II). QP/M bootable disk will be shipped if QBios is ordered with QP/M.

***NEW PRODUCTS NOW AVAILABLE***

KEYMAP: Not another keyboard configuration program! Written exclusively for the Xerox 820/820-II. Full on-SCREEN keyboard image while editing. Up to 8 custom keyboards can be created, executed and toggled.

SMARTROMS: Add 820-II features to your 820-I/BigBoard-I. Provides real-time clock, screen dump, screen/program pause, reset with CTRL/ESC, CRT blanking after 10 minutes of idle time, 7 or 8-bit keyboard mode, serial/parallel printer routines, 820-II (4.0x) compatible vectors, and MUCH MORE! Replaces your socketed ROMs (no soldering!).

PRICES

| QP/M | $60.00 |
| QBIOS FOR QP/M | $20.00 |
| KEYMAP | $20.00 |
| SMARTROM SET | $30.00 |

When ordering, please specify your computer system and media preference (8" IBM 3740 or 5.25" Xerox/Kaypro format).

QIPIM HOME OFFICE 90508-9001
(213) 212-5877 (24 hour recorder)

HPLot

A PLOTTER EMULATION PROGRAM FOR YOUR OKIDATA, PROWRIGHT, GEMINI, OR EPSON PRINTER.

- Powerful HP-GL PLOTTER SYNTAX: SCALING, LINES, WINDOWS, ETC.
- FASTI GRAPHS IN 4 MINUTES.
- HI-RES MODE: UP TO 136x144 DPI.
- PLOT SIZES 11"x14" TO 7"x48".
- 100 PAGE ILLUSTRATED MANUAL.
- SOURCE CODE IN C FOR PROGRAMS THAT USE HPLOT TO MAKE PIE CHARTS, GRAPHS, ETC.
- Requires 54K 280 CP/M 2.2.

ORDINATE SOLUTIONS
MAIN P.O. BOX 0308, OBERLIN, OH 44074

Micro Cornucopia, Number 25, August-September 1985
Introducing **GRAF 3.0**

the complete BUSINESS and SCIENTIFIC printer graphics program

**SALES by DIVISION**

<table>
<thead>
<tr>
<th>Year</th>
<th>Division A</th>
<th>Division B</th>
<th>Division C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1982</td>
<td>30</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>1983</td>
<td>40</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>1984</td>
<td>50</td>
<td>30</td>
<td>20</td>
</tr>
</tbody>
</table>

**BUSINESS APPLICATIONS**

* display floating point data directly from spreadsheets, data bases, and word processors (or the keyboard) in a wide variety of bar, pie, line, and scatter plots
* plot and group up to 6 different variables on a single graph, distinguished by up to 14 different "fill-in" patterns and 8 different point-plotting symbols
* menu driven operation supporting automatic graph scaling, labeling, and legend creation
* program default values may be set once and for all

Fourteen Different "Fill-in" Patterns

**SCIENTIFIC APPLICATIONS**

* simple interface allows plotting floating-point data obtained from all popular programming languages
* plot any number of curves (e.g. experimental data vs. theoretical values) on the same graph, choosing from 8 different plotting symbols.
* automatically created legends distinguish variables
* add up to 5 different-density grid lines, and choose from a wide variety of numerical labeling options
* high/low graphs are supported directly

**Sample High/Low Graph**

**TERMS:** We ship via first class mail. The above prices include $5.00 for s/h. (Orders outside USA require additional $5.00 for postage. NJ residents add 6% tax.) When ordering you MUST state your computer and printer make and model. We support MS-DOS (PC-DOS) version 2.0 or later on computers with at least 192k RAM, and CP/M-80 version 2.2 or later on 286 computers (other than Apple) supporting a TPA of at least 54k (requires 64k of RAM). Most soft-sector disk formats are available. (If you can read several formats, please send us a list.) GRAF 3.0 works with any printer fully compatible with one of the following: Epson FX, RX, LX, MX (with GRAFTRAX), or LQ-1500; C. Itoh Prowriter; NEC 823, Star Micronics Gemini 10X, 15X, SG-10, SG-15, IBM Graphics Printer, Okidata 192, and earlier Okidata models equipped with the "IBM Plug 'n' Play" chips. (If you have an Okidata printer, other than the 192, the Plug 'n' Play chips are required.)

**GRAF 2.0 Update Policy:** Returning your original GRAF 2.0 disk to MSC entitles you to $20.00 off the above prices.

**MSC Microcomputer Systems Consultants**

301 North Harrison Street CN5279, Suite 228 Princeton, New Jersey 08540
I would never want to take away from the great accomplishment of either the original or the Xerox version of the PFM-80 monitor. When I think back to my initial exposure to PFM-80 (reading through Micro C even before I got my 820 board up and running!) I thought it was indeed Pretty F&%$# Magical, and I'm still amazed at its abilities.

However, there were a few unexplained gremlins that bothered me. After I had hammered out all the hardware bugs, the gremlins were still there. So, it was time to attack the software. After many hours of studying PFM-80 I've discovered where some of the gremlins hide out.

Figure 1 - Bit Changes

<table>
<thead>
<tr>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>00= A or 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01= B or 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive Select: 10= C or 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11= D or 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk Access: 1 = de-select</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 = select</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port B: Keyboard Port is 'Ready'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB: U.O. Unidentified Output?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X-820: 8/Not 5 Drives:1=8&quot; 0=5-1/4&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB: Not Used</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X-820: 400/460 single or double sided 5-1/4&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB: Disk AC: 1=off 0=on</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X-820: Alternate Character Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bank Select: 1=ROM Bank 0=RAM Bank</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2 - Autoboot Routine

```
AUTOB 
EQ U TRUE 

SIGNON: EI  
CALL PNEXT 
DB '... System monitor 3.66 ..' ;@  
DB CR,LF,0 ;@  
IF AUTOBT ;if you power up your system all together 
;it is best to wait a couple of seconds @@ 
MVX A,2 ;set up for a 2 second delay @@ 
STA MOTOR ;use motor countdown timer with OTC3 
DLAY: CALL PAUSE ;as the 1 second clock, motors are off 
LDA MOTOR ;at this point so we won't hurt anything @@ 
ANX OFFH ;set flags 
JNZ DLAY! 
CALL BOOT ;do an autoboot for the sophisticated user 
CALL PNEXT ;print an error message if boot fails @@ 
DB CR,LF,0 
DB 'Disk in A? Enter B <CR> to boot CP/M' ;@ 
DB CR,LF,0 ; and then we enter the monitorhé 
ENDIF 
JMP WARM ;GO ENTER MONITOR 
```
Gremlin #2

I found another gremlin in the disk drive PIO. Have you ever been computing along and suddenly your computer does a warm boot? Where did that come from?

Line 233:
DEFB 00011000B
;MAKE BITS 4 AND 3 BE INPUTS

This looks like a simple enough instruction. But the PIO was expecting an interrupt vector when it got this command, and poof, an interrupt vector to FF18H is born. (The number above is an 18H and FF00H is added to this number on an interrupt request.) The address FF18 belongs to SYSVEC which is not set to anything since we didn’t expect to use it. It is still zero from the zeroing of the scratch RAM (line 50).

So on an interrupt request from the PIO, the processor gets its new Program Counter from FF18H (0000H) and this happens to be the Jump to Warm Boot address of CP/M. This sequence would never start if the interrupts were disabled but the command to disable interrupts is 0110 0111B. This section of code should read:

Line 234: DEFB 4,BITCTL
Line 233: DEFB 11001111B
;SET SYSPIO PORT A TO
;BIT MODE=> MODE 3
New line: DEFB SYSVEC
;INTERRUPT VECTOR
Line 233: DEFB 00011000B
;MAKE BITS 4 AND 3 BE INPUTS
Line 234: DEFB 01100111B
;INTERRUPTS DISABLED

I don’t know why an interrupt was generated other than to say they weren’t disabled as we thought they were and Murphy’s Law says, “If interrupts aren’t disabled, you’ll probably get a few to point out the oversight.” (I really don’t buy this and I am still looking for the reason.)

Autoboot

I find that 99% of my power-ups are destined for CP/M and only 1% are to the monitor. So I’ve implemented the routine shown in Figure 2 to do an autoboot after waiting 2 seconds (to give slow power supplies time to crank up).

If there is a boot error, you’ll get an error message and automatic access to the monitor. I’ve surrounded the autoboot region of the monitor with an IF statement. Thus, an AUTOBT EQU FALSE will cancel this feature.

P.S. I must say “hats off” to a hom­towner and his product—the EZ-PROM from Optronics Technology. It is just fantastic. It helped make all these ROM changes possible.

820 - 1

820 reset switch .................................... $ 3.00
820 composite video adapter. Generates true
RS-170 compatible video for your 820. Outputs
to RCA phono jack and plugs directly onto the
820 video connector. Assembled and tested
820 video cable. 10 pin AMP to CRT edge card
w/power conn & provisions for brightness pot
820 8" disk cable. 37 pin ‘D’ to dual 50 pin
edge card ........................................... 45.00
820 5.25" disk cable, w/power conn ............. 20.00
820 9 pin power connector w/pins ............... 4.00
820 9 pin PC mount power connector .......... 4.00
8" disk DC power connector w/pins ............. 1.50
8" disk AC power connector w/pins ............. 0.75
5.25" disk DC power connector w/pins .......... 1.10
820 connector plate w/mth hardware ........... 7.50
820 “C” size schematics, set of 6, very legible 18.00
820 CB IOS source, ROM source, BOOT, and
formatter, all in source code. M80 compatible,
conditional assembly flags for 8" or 5.25"
disk. 8" SSSD ......................................... 35.00
820 compatible keyboard w/cable for 820,
complete w/schematics & source code for the
encoder ROM ........................................ 55.00
256 K RAM expansion kit for the 820-1 or the
BIGBOARD-1. Complete with schematics,
table of operation, RAM DISK software,
installation and test instructions. Clean instala-
tion with no cables or jumpers. Good beyond
5 Mhz.
KIT W/O RAM .................................. 125.00
KIT WITH RAM .................................. 195.00

J.B. FERGUSON, INC.
817-840-0207
P.O. BOX 300055 ARLINGTON, TEXAS 76010
TERMS: MasterCard/VISA add 3%, money order, cer-
tified check, UPS cash COD. Allow 2 weeks for personal or
company checks to clear. Shipping/handling extra. Texas order
add 5.125% tax.

Micro Cornucopia, Number 25, August-September 1985
The "Better Board" Gets BETTER!

For the past 4 years, Colonial Data has been supplying thousands of its original SB80 "Better Board" computers to the O.E.M. market. Now, the "Better Board" is even BETTER with the introduction of the SB80-II. More Standard Features at a NEW LOWER PRICE!

- **4MHz Z80A CPU WITH NO WAIT STATES**
  Enhanced BIOS uses mode 2 interrupts.

- **64K MEMORY STANDARD (EXPAND TO 128K)**
  Parity checked RAM utilizes 4164
  Up to 8K of EPROM (4K is standard)
  Accepts Pin-compatible EPROMS from 2716 thru 2764

- **NUMEROUS FLOPPY DISK STORAGE OPTIONS**
  Uses the advanced NEC765A controller chip
  Supports 8" and 5½" drives simultaneously
  Automatic Density Selection (Single/Double/Quad)

- **4 SERIAL I/O PORTS (2 STANDARD)**
  Software selectable baud rates to 19,200
  Uses Z80 S10/0 and SMC 8116 baud rate generator.

**SB80-II SYSTEM OPTIONS:**

<table>
<thead>
<tr>
<th>Option</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDITIONAL 2 SERIAL PORT (INSTALLED)</td>
<td>$ 60.00</td>
</tr>
<tr>
<td>ADDITIONAL 64K MEMORY (INSTALLED)</td>
<td>$ 60.00</td>
</tr>
<tr>
<td>SASI INTERFACE OPTION (INSTALLED)</td>
<td>$ 35.00</td>
</tr>
<tr>
<td>XEBEC HARD DISK CONTROLLER</td>
<td>$290.00</td>
</tr>
<tr>
<td>CP/M 2.2 OPERATING SYS W/MANUAL</td>
<td>$ 75.00</td>
</tr>
<tr>
<td>CP/M 3.0 OPERATING SYS W/MANUAL</td>
<td>$275.00</td>
</tr>
</tbody>
</table>

**Z80A PIO PARALLEL PRINTER INTERFACE**
Centronics compatible printer interface is Standard!

**SASI HARD DISK INTERFACE CIRCUITRY ON-BOARD!**
SASI interface provides access to hard disk.
Compatible with XEBEC and other controllers.
Just populate with TTL and add connector.

**4 CHANNEL COUNTER TIMER-STANDARD!**
Allows custom software applications.
User accessible 125Hz interrupt.
Provides system date/time clock.
4th Channel used for the NEC 765 interrupt.

**CP/M 2.2 BIOS ENHANCEMENTS**
Allows flexible device assignments, baud rate selection, automatic density detection, a system date and time clock and hard disk support.

**OPTIONAL CP/M 3.0 (CP/M PLUS)**
Extensive disk buffering speeds system throughout.

Z80™ — ZILOG  CP/M™ — DIGITAL RESEARCH (Calif.)

**HOW TO ORDER:**
Call or Write Colonial Data—Orders paid with bank card or cashier's check are shipped within 3 working days. Allow 3 weeks for Personal Checks.
Shipping: Add $5.00 for UPS Shipping (Brown) Within the United States.

80 Pickett District Road,
New Milford, Conn. 06776
Telephone (203) 355-3178
### Colonial Data SB-80-II

**FULL 90 DAY PARTS & LABOR WARRANTY**

- **Dimensions**: 13½"W x 5¼"H x 17" D

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIA - 2/380K DSDD DRIVES</td>
<td></td>
<td>$895.00</td>
</tr>
<tr>
<td>IIB - 2/780K DSOD DRIVES</td>
<td></td>
<td>$995.00</td>
</tr>
<tr>
<td>IIC - 10 Meg H/Disk w/380K Floppy</td>
<td></td>
<td>$1895.00</td>
</tr>
<tr>
<td>IID - 10 Meg H/Disk w/780K Floppy</td>
<td></td>
<td>$1995.00</td>
</tr>
</tbody>
</table>

* SB80-II (Board Only) Wired and Tested $298.00

<table>
<thead>
<tr>
<th>SB80-II SYSTEM OPTIONS</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Models:</td>
<td></td>
</tr>
<tr>
<td>Additional 2 Serial Ports (installed)</td>
<td>$60.00</td>
</tr>
<tr>
<td>Additional 64K Memory (installed)</td>
<td>$60.00</td>
</tr>
<tr>
<td>CP/M 2.2 Operating System w/Manual</td>
<td>$75.00</td>
</tr>
<tr>
<td>CP/M 3.0 Operating System w/Manual</td>
<td>$275.00</td>
</tr>
<tr>
<td>Models IIA and IIB only</td>
<td></td>
</tr>
<tr>
<td>SASI Interface Option (installed)</td>
<td>$35.00</td>
</tr>
<tr>
<td>XEBEC Hard Disk Controller</td>
<td>$290.00</td>
</tr>
</tbody>
</table>

### Shugart 5¼” Disk Drives

- **MODEL 455**
  - DSDD Floppy
  - Over 380K Capacity
  - 2 For $370.00

- **MODEL 465**
  - DSOD Floppy
  - Over 780K Capacity
  - 2 For $430.00

### Colonial Data Cabinet Kits

- **MODEL CKP1**
  - Metal Cabinet will accept Single Boards up to 12"x16"
  - complete with fan, RFI filter with power switch, front panel and mounting for two 5¼” half-height floppy. Pre-punched for 4 DB25 and 1-50 pin.
  - Includes 60 watt power supply with ±12V, +5V.
  - $225.00

- **MODEL CKP2**
  - Same as Model CKP1, but includes 90 Watt Power Supply to power half-height Winchester 5¼” hard disk and single 5¼” half-height floppy.
  - $425.00

- **MODEL CKX**
  - Cabinet only with fan, RFI filter, power switch, prepunched connector holes. No Power Supply.
  - $139.00

### Colonial Data Services Corp.

80 Pickett District Road,
New Milford, Conn. 06776
Telephone (203) 355-3178
Direct BDOS Calls In CP/M

By Thomas Geldner

Discussion among three programmers at the local watering hole:

Joe: Well, Sam, what's new?

Sam: Oh, the usual. I'm porting a modified Phoneyx Op-Sys to the Grumblefratch 0286 using conversational interrupt drivers and an I/O slave driver. Should be finished in a day or two. How 'bout you?

Joe: Yeah, well, I'm doing a co-flipper based on the IEEE-487 and now I can beep the console bell! Various parts of were pretty theoretical (at least the standard for slave driver. This month, BDOS will be done in a variety of ways immediately apparent). This month, BDOS will be done in a variety of ways including your first (and probably last) assembly language example. And yes, you're going to learn how to "beep the console bell." (Gosh, Maude, this sounds exciting!)

Review

Programs use the BDOS and, occasionally, the BIOS to perform Input/Output tasks. The BDOS handles all disk drive related functions and most console activity (keyboard input, printer output, CRT output, etc.). The BIOS does specific hardware related tasks and occasionally handles console activity, but is used primarily by the BDOS as its window to the world.

When you write a program in a high-level language such as Turbo, access to CP/M is provided for you in the form of statements such as Read and Write. (The equivalents for BASIC include PRINT, INPUT, GET, and PUT statements.) When you tell Turbo to display a character on the screen, Turbo actually tells the BDOS to do it. BDOS in turn tells the BIOS to do it, and then the BIOS does it.

Accessing The BDOS

Access to the BDOS takes the form of function calls. (This is true both in assembler and in high-level languages.) There are 38 CP/M 80 BDOS functions available, doing such things as resetting disk drives, deleting files, and printing. To have the BDOS perform one of its functions, you put information in your 8080 or Z80's registers (information that BDOS needs) and then call BDOS.

Don't panic if you know nothing about registers. Doing a BDOS call is sort of like baking a cake. If you just follow the recipe, everything will come out OK (as long as you don't open the oven door at the wrong time).

The Two Parts Of A BDOS Call

There are two main ingredients in a BDOS call. The first is the number of the function being called. Each of the 38 available functions has been assigned a number from 00H to 2FH (37 decimal). For example, the SYSTEM RESET function is 00, CONSOLE INPUT is 01, CONSOLE OUTPUT is 02, etc. The function number is always placed in the C register.

The second ingredient is optional and is called an Entry Parameter (or parameters). The Entry Parameter can be a character to be sent to the screen or printer, a memory address, or some other value that BDOS will use. (There is also an Exit Parameter which we will deal with later. This is a result, though, and not an ingredient.)

Assuming we wish to use the BDOS to send a character to the console, we would put the character to be sent in Register E and put the function number 02 (CONSOLE OUTPUT) in register C.

Now, once the registers are loaded your program can CALL (or JUMP) to location 0005 (the BDOS Entry Vector). At location 0005 is a JUMP to the address specified at locations 0006 and 0007 (the location of BDOS).

Once the JUMP to BDOS is completed, function 2 will be executed. When function 2 is finished, the BDOS will do a RETURN, thus sending execution back to where it started.

Now, let's see how we get the ingredients to the right places at the right times.

Assembly Language Programming 1A

As promised, Figure 1 shows our first assembly language program "beep the console bell" using BDOS function 2, CONSOLE OUTPUT.

To use, type Figure 1 exactly as shown using Perfect Writer, WordStar, or any other word processor and save as a file called BEEP.ASM. Use TABS to separate the columns, and in WordStar, use the non-document mode. (You don't have to enter the semicolons and the comments with them.) Assemble using ASM.COM, then load using LOAD.COM as shown in Figure 2.

Figure 1 - Assembly Language program to 'beep the console bell'

```
; BEEP the Console bell
ORG 100H ;start where programs start

BDOS EQU 0005H ;BDOS entry vector
CONOUT EQU 02H ;BDOS CONSOLE OUTPUT function
BELL EQU 07H ;ASCII bell character

MVI C,CONOUT ;load Register C with function number
MVI E,BELL ;load register E with bell character
JMP BDOS ;go to address 0005
JMP 0 ;Jump back to op/m
```

Figure 2 - LOAD.COM

A>ASM BEEP <== creates file BEEP.HEX on A: Drive
A>LOAD BEEP <== creates file BEEP.COM on A: Drive

Phoneyx Op-Sys
Joe: Well, what's new?

Sam: What's up with you, Bill?

Bill: Gee, guys, I just read Tom Geldner's latest article in Micro Cornucopia, Number 25, August-September 1985
Now, you can run BEEP.COM just as you would anything else. Simply enter “BEEP” and hit RETURN. You should hear the console bell ring once, and then you will be returned to the A>. (See, you really do have mastery over the little beeper.)

I don’t intend to conduct a tutorial on how to program in assembler (I wouldn’t know how, anyway), but here’s a brief explanation of what’s happening in this program.

First, the assembler is told where the program is to ORIGINATE using the statement “ORG 100H”. Under CP/M, a program that will operate by itself will normally originate at 100H (where CP/M places control after it loads a .com file).

Next, we establish some EQUATES which make the program easier to read. In this case, the assembler is told that the CONSTANT BELL has the value 07, BDOS is 0005, and CONOUT is 02. (Note that ASM doesn’t care whether we enter a number as 5, 05, or 0005. Programmers do this for clarity’s sake. Memory locations are usually entered with four numbers such as 0005 or 0F7EF (note the leading 0 if the first character is a letter), byte values are entered as two digits such as 01 or 7F.)

Now we get to the meat of the program. The things we’ve done previously have been instructions to the assembler (ASM.COM). The next instructions produce the actual Z80/8080 code. The first instruction, MVI C,CONOUT, means to MOVE IMMEDIATE the value of CONOUT into register C of the CPU. Since the assembler has been told that CONOUT is always 5, this has the effect of loading register C with a 5. The next instruction loads register E with the value 2. You could have loaded the numbers directly instead of giving them names first, but the program would have been harder to understand.

The next instruction activates the BDOS by JUMPing to it. BASIC programmers can think of this as a GOSUB where the BDOS itself is the subroutine. After the BDOS is through executing function 2, it will RETURN program execution to wherever it came from, in this case, our program.

The final instruction, JMP 0, sends control back to CP/M.

Now let’s look at our program using DDT.

A> DDT BEEP.COM(cr> <= you type
DST YERS 2,2 <= DST responds
NEXT PC
0180 0100
-0100,106 <= you type

This will display the first 8 byte values in HEX and ASCII. You should find the following:

100 0E 02 1E 07 03 00 00 ........

The first 6 hex bytes are the actual machine language instructions that resulted from our original assembler source file! Now let’s disassemble BEEP by using DDT’s List command. Just type an L and hit RETURN. You should see:

0100 MVI $,02
0102 MVI $.07
0104 JMP 0005
0107 NOP <= means NO OPERATION
(all the zeros)
0108 NOP

Recognize the guts of our assembly listing?

Turbo, SBASIC and dBASEII BEEPers

Again, since this is not an assembly language column, let’s go back to the higher world and see how direct BDOS calls are made.

Here’s what BEEP looks like in Turbo Pascal:

begin
Bdos($02,$07);
end;

Since Borland has conveniently provided us with a Bdos Procedure, specifying addresses and registers is unnecessary. (There is also a BdosHL Function and a Bdos Function for those BDOS calls requiring access to the HL registers or A register. More on this later.)

(continued on page 33)
Make your KAYPRO computer IBM-PC compatible for $29.95!

READ, WRITE and FORMAT more than 25 different types of disks (including IBM PC-DOS/MS-DOS) with Media Master™

Are you tired of trying to find your favorite software package in KAYPRO format? Would you like to use your KAYPRO generated Wordstar files, dBase II data and spreadsheet files on the IBM-PC at work (and vice-versa)? Do you want to trade public domain software with a friend who owns an Osborne?

MEDIA MASTER gives your KAYPRO instant access to program and data files in over 25 disk formats including:

- Osborne SD & DD
- IBM PC-DOS 1.0 & up
- IBM PC-DOS 2.0 & up
- IBM CP/M-86
- Morrow MDZ
- Systel II
- LNW-80
- TRS-80 with Omikron CP/M
- TRS-80 III w/Memory Merchant
- TRS-80 IV w/CP/M+
- Heath Z100
- Heath w/Magnolia CP/M
- Cromemco w/Int'l Term
- Cromemco CDOS SSD
- Cromemco CDOS SSD
- TI Professional CP/M-86
- Actrix
- Lobo Max-80
- Xerox 820 I SD
- Xerox 820 II DD
- Zenith Z90
- DEC VT100
- NEC PC-8001A
- Kaypro II

Now available for KAYPRO 2 (and II): All other models soon!

ORDERING INFORMATION: Include $3 per order for postage/handling. Overseas airmail add $10. California residents add 6% tax (LA County, add 6.5%). To place COD or credit card orders,

Call TOLL FREE 24 hours: 800-824-7888 (Ask for Operator 409)
Alaska, Hawaii: 800-824-7919 (Ask for Operator 409)
Technical questions: call (818) 716-1655

For more information, a free brochure (including sample printouts and reviews), or to order, contact:

dg/systems technologies, inc.
22458 Ventura Blvd., Suite E
Woodland Hills, CA 91364

All programs also available for OSBORNE and DEC RAINBOW computers. Dealer inquiries invited.
In dBASEII things are even less obvious. From the dot prompt enter:

```
poke 52000,14,2,30,7,195,5,0
set call to 52000
call
```

First, we select an area of memory not normally used by dBASEII and POKE the decimal numeric equivalent of our assembler produced BEEP.COM program into it. This statement has the form:

```
POKE address,
<data byte>[,<data byte>...]
```

Then we give dBASEII the address to be CALLED with the SET CALL TO command. Finally, we do a CALL that tells dBASE to go to the address specified previously and execute the instructions there.

Now For Something Useful!

Obviously, this hasn't been the most constructive application for direct BDOS calls. Heck, PRINT CHR(7) or Write(G) will work just fine, thank you. So what IS the most constructive application? The answer is: anything your high-level language doesn't allow you to do directly. And to answer the question "What doesn't my high-level language allow me to do?" you'll need to know what the BDOS functions can do.

There are two ways of finding out what the BDOS functions are. The first is to attempt to read the Digital Research CP/M manual that came with your computer. Good luck! This little gem ranks right up there with the SBSAIC manual for the all time worst word collection.

The recommended way is to get one of the many books available on CP/M. My personal favorite is Thom Hogan's "CP/M User Guide," copyright 1982, Osborne/McGraw-Hill. It's an excellent guide for the non-assembly programmer. There are many others, but most are either too simple or else intended for the assembly language programmer. (Editor's note: "Inside CP/M," available from B. Dalton or Micro C, is also a good book for this use.)

Now, as promised, here's something practical (at least I think it's practical). For lack of anything better, let's call this the "What is the currently logged disk drive?" procedure.

We all know that most high-level languages, acting through CP/M, will automatically save a file (if you don't specify the drive) to the currently logged drive. There may be an occasion, however, when we ALWAYS want to save a file or look for a file on the disk that was logged the first time the program was run.

For example, suppose you have an optional help file that your program can access. The assumption is made that this help file will always be found on the same disk as the main program. Once the main program is started, if we subsequently allow the user to change the logged disk drive, the program may be unable to find the help file again. One way to get around this is to force the user to always run the program from the A: drive. Not a very classy solution.

A better way is to figure out which drive was logged at the time the program was originally run, and then use that drive throughout to specify the help file. We do this with BDOS Function 19 (hex), RETURN CURRENT DISK. (The Editor again: Libraries would use this function a lot if they started loaning floppies.) Here's a Turbo program that does this:

```pascal
var DiskDrive: Char;
begin
  DiskDrive := Char(Bdos($19)+$41);
  Writeln(DiskDrive);
  WriteLn(DiskDrive);
end;
```

Note that in this example, the BDOS Function is used. This is similar to, but not the same as, the BDOS Procedure. The BDOS Function in Turbo is used without an entry parameter, and the result returned is the value found in the A: register of the Z80 or 8080. In this case, the result will be 0 for the A: drive, 1 for the B: drive, and so on. Here, we convert the value to a drive letter (+$41). One usage might be:

```pascal
FileName :=
  CurrentDrive + ':' + FileName;
```

This will allow us to search for a file on the currently logged drive, and then get back to the file even if the logged drive has been changed.

End Of Pedantic Lecture #3

There are certainly other ways of accomplishing the things I've illustrated in this article. Your comments and suggestions are always welcome. In fact, good questions and/or problems will form the basis for future articles.

---

### NEW... FLOPPY DISK CONTROLLER BOARD

**FOR THE XEROX 820-II**

**DIRECT REPLACEMENT FOR**

**THE STANDARD 820-II FLOPPY DAUGHTER BOARD. USES THE**

**FDC1793/8229B FOR RELIABLE**

**OPERATION. PERFECT FOR**

**SURPLUS 820-II BOARDS.**

- **ASSEMBLED/TESTED** $109.95
- **COMPLETE KIT** $79.95
- **BARE BOARD w/9229B** $59.95
- **BARE BOARD w/DOC** $39.95
- **INCLUDE $3.00 SHIPPING**
- **VISA/MC COD ADD $1.50**

---

**REPAIR SERVICE**

XEROX 820 II/III BIG BOARD I

CALL OUR NEW BULLETIN BOARD

**312-892-3260**

---

**THE DATA MILL**

**12 CRESCENT COURT**

**MONTGOMERY, IL 60538**

**312-892-8573**
WARNING:
Do NOT read this flowchart:

Unless you have time to spare... in extreme cases a few people have found they had thirty years to spare.

You read the flowchart anyway. Why? Because flowcharts are a powerful graphic way of communicating ideas. The big problem is producing them: they take a lot of time to do well and are difficult to revise or correct.

The flowchart above was produced using EasyFlow, a computer aided flowchart generation tool. You decide how the flowchart is to be laid out and describe the flowchart to EasyFlow using a simple command language. EasyFlow then does the hard part of actually producing and printing the flowchart. EasyFlow automatically centers text inside shapes and routes lines; changes and corrections are easy since EasyFlow re-centers text and re-routes lines as necessary!

EasyFlow is a well designed, thoroughly tested and comprehensively documented package.

- **FAST**: Produces a typical flowchart in 12 seconds.
- **EASY**: The command language is straightforward and easy to learn.
- **POWERFUL**: Automatic text centering and line routing.
- **PRINTERS**: Works with all printers.
- **SIZE**: Flowcharts up to 5 shapes across by 11 shapes down.
- **SHAPES**: 18 standard shapes. User defined shapes easily added.
- **LINES**: User selectable line-drawing characters.
- **MANUAL**: Complete, comprehensible and over 100 pages long. Also included is a reference card and ten demo flowcharts.

**EasyFlow**: $49.95 Minimum memory: MS-DOS/PC-DOS 96K; CP/M-80 48K

**EasyFlow-PLUS** is an advanced version of EasyFlow which contains all the features and capabilities of EasyFlow plus the following enhancements:

- Huge flowcharts; up to 16 by 16 shapes in size.
- Wide charts can be printed in strips; big charts from small printers.
- Text blocks can be placed anywhere in chart; comment your chart.
- Arbitrary lines can be drawn anywhere in the chart.
- Can also be used to produce organization charts.
- Chart can be previewed on graphics screen.1

Subject to available memory.

1 Requires IBM/PC or compatible with color/graphics adapter.

**EasyFlow-PLUS**: $89.95 Minimum memory: MS-DOS/PC-DOS 128K; CP/M-80 64K

Available for MS-DOS/PC-DOS machines on IBM/PC format 5" diskettes; for 286 CP/M-80 machines on 8" SSD and most soft sectored 5" formats. Check, money order, VISA or company PO.

---

**BOOKS**

- **Inside CP/M...** by David E. Cortesi
  - $27.95 (US, Can, Mex)
  - $37.95 (Other foreign)
  - This is one of the best books on CP/M for users and programmers. It's divided into two sections — a tutorial and a reference guide — to satisfy novice and expert alike. We've been very pleased with Inside CP/M at Micro C.

- **Your Fortune in the Microcomputer Business** by Victor Wild
  - $26.45 (US, Can, Mex)
  - $36.45 (Other foreign)
  - This is the best, most complete collection of "working for yourself" information I've found (and I've heard nothing but good comments from those who've received it). This two volume set is perfect for those times when you need a break from monitor watching.

- **The Small C Handbook...** by James E. Hendrix
  - $17.95 (US, Can, Mex)
  - $22.95 (Other foreign)
  - If you really want to solve the mysteries of the Small C compiler, you'll want the Small C Handbook. Includes chapters on 6808 assembly language programming, program translation, code generation, and the Small C source.

---

**MICRO CORNUCOPIA**

P.O. Box 223 • Bend, Oregon 97709
Order No. (503) 382-5060

Micro Cornucopia, Number 25, August-September 1985
If you are like most folks, you aren't comfortable with an RS-232 interface. Actually, many feel that if God had intended man to understand a serial interface he wouldn't have let a committee design it (and there is no question that it was designed by committee).

To be precise, RS-232C is an interface defined by the EIA (Electronics Industry Association) for low speed serial data communication.

The RS-232C standard is a piece of paper issued by that committee which defines such things as voltage levels, loading characteristics, timing relationships, and signal responsibilities.

How do you tell if you have a serial interface that needs understanding? Usually you'll see a female 25-pin connector mounted on the back of your computer (a DB-26S). The connector will probably be marked "printer" or "modem."

DTEs And DCEs

There are two main classes of RS-232C devices — DTE (Data Terminal Equipment) and DCE (Data Communications Equipment). Terminals, computers, and printers are usually DTEs; modems are DCEs. The point of all this is that the esteemed EIA committee assumed you'd want to connect a DCE to a DTE. They set it up so that you could usually connect the two together simply by connecting pin 1 to pin 1, pin 2 to pin 2, and so on, up to pin 25.

You can also interface DCE to DCE (or DTE to DTE), but you must build a special cable in which some of the signal and control lines are crossed. A null modem cable which allows you to directly connect two computers is an example of such a special cable.

RS-232C is intended for short (50 feet or less), low-speed serial communication. Low speed normally means 19,200 baud (bits per second) or less although 38,400 baud and higher is becoming more and more common. Serial communication means that all the information is moving one bit at a time down a single signal line. (It's like blowing peas through a straw.)

Most parallel interfaces send data 8 bits at a time.

Is It Simplex?

RS-232C defines simplex, half-duplex, and full-duplex communication.

A simplex channel is one way (for example: from a computer to a printer). In half-duplex, data may travel either direction but only one direction at a time. In a full-duplex channel, data may travel in both directions simultaneously.

Standard 300 and 1200 baud modems are full-duplex, so you can see characters on your own screen as you enter them on your keyboard. The characters are being echoed back from the remote system.

The Electrical Definitions

The RS-232 interface is made up of 25 signals or lines. (A line is defined as a continuous wire from one device to another.) In the simplest case, an interface may be implemented with as few as two wires.

Pin 4: RTS — Request To Send

RTS is the signal that indicates the DTE wishes to send data to the DCE. When it is high (true), the DCE knows that the DTE is ready to send.

Pin 5: CTS — Clear To Send

CTS is a signal from the DCE that indicates it is ready to receive data from the DTE. Normally this line will be low (off). When the DTE raises RTS the DCE will raise CTS allowing the DTE to send data.

Pin 6: DSR — Data Set Ready

This signal informs the DTE that the DCE is alive and well. It's normally kept high by the DCE.

Pin 7: SG — Signal Ground

SG is the neutral or return line for all the other signals. It must always be present.

Pin 8: DCD — Data Carrier Detect

This signal from the DCE (modem) tells the DTE (computer) that the modem sees a carrier (is the phone off the hook?).

Pin 15: TC — Transmit Clock

TC provides the clock for the trans­mitter section of a synchronous DTE. It may or may not be running at the same rate as the receiver clock, and must be present on synchronous interfaces. (Very few systems are set up for synchronous communication, so pins 15 and 17 are seldom used.)

Pin 17: RC — Receiver Clock

RC provides the clock for the receiv­er section of a synchronous DTE, and may or may not be running at the same rate as the transmitter clock. It must be present on synchronous interfaces. Both RC and TC are provided by the DCE.

Pin 20: DTR — Data Terminal Ready

This circuit provides the signal that informs the DCE that the DTE is OK. It's pulled high by the DTE at power­up and left there. Note that a typical
DCE must have an incoming DTR before it will function normally.

Figure 1 shows the signal names of the entire RS-232 interface.

Types Of Data Communications

There are two ways to send data serially: asynchronously and synchronously.

Asynchronous Data

Asynchronous serial data is sent character by character. The bits within a character are sent with very precise timing. Once the character is sent there is a pause or signal of some sort to tell the system to get ready for the next character.

Normally, during an inter-character gap the line is at a logic 1 state. When the character starts, a start bit (logic 0) has occurred, the other data bits follow, least significant bit first.

There are normally 7 or 8 data bits in a character are sent with very precise timing. The bits with­in a character will be set or reset to make an odd number of ones. (For this reason, parity has largely been replaced by CRCs and check sums.)

The parity check won't catch the error. There are normally 7 or 8 data bits sometimes followed by a parity bit. When running with even parity the parity bit will be set or reset to make an even number of ones. In odd parity, the bit will be set to make an odd number of ones.

In theory, parity should catch most bit errors because things wouldn't come out even (or odd) if one bit gets crunched by the phone line. However, if two bits get crunched in the same character, there is a good chance that the parity check won't catch the error. (For this reason, parity has largely been replaced by CRCs and check sums.)

Synchronous Data

Synchronous data transfer means that the sending and receiving units are synchronized by a constant clock signal. This clock makes sure that the two units won't get out of step so that strings of characters can be sent without stop bits or start bits. The

<table>
<thead>
<tr>
<th>PIN</th>
<th>NAME</th>
<th>EIA</th>
<th>CCITT</th>
<th>DTE</th>
<th>DCE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CG</td>
<td>AA</td>
<td>101</td>
<td>&gt;&gt;</td>
<td>&lt;&lt;</td>
<td>Chassis Ground</td>
</tr>
<tr>
<td>2</td>
<td>TD</td>
<td>BA</td>
<td>103</td>
<td>&gt;&gt;</td>
<td>&lt;&lt;</td>
<td>Transmit Data</td>
</tr>
<tr>
<td>3</td>
<td>RD</td>
<td>BB</td>
<td>104</td>
<td></td>
<td>&lt;&lt;</td>
<td>Receive Data</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>CA</td>
<td>105</td>
<td></td>
<td>&lt;&lt;</td>
<td>Request To Send</td>
</tr>
<tr>
<td>5</td>
<td>CTS</td>
<td>CB</td>
<td>106</td>
<td></td>
<td>&lt;&lt;</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>CC</td>
<td>107</td>
<td></td>
<td>&lt;&lt;</td>
<td>Data Set Ready</td>
</tr>
<tr>
<td>7</td>
<td>SQ</td>
<td>AB</td>
<td>102</td>
<td></td>
<td>&lt;&lt;</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>8</td>
<td>DCD</td>
<td>CF</td>
<td>109</td>
<td></td>
<td>&lt;&lt;</td>
<td>Data Carrier Detect</td>
</tr>
<tr>
<td>9*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;&lt;</td>
<td>Pos Test Voltage</td>
</tr>
<tr>
<td>10*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;&lt;</td>
<td>Neg Test Voltage</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Not Used (usually)</td>
</tr>
<tr>
<td>12+</td>
<td>SCDC</td>
<td>SCF</td>
<td>122</td>
<td></td>
<td>&lt;&lt;</td>
<td>Secondary DCD</td>
</tr>
<tr>
<td>13+</td>
<td>SCTS</td>
<td>SCB</td>
<td>121</td>
<td></td>
<td>&lt;&lt;</td>
<td>Secondary CTS</td>
</tr>
<tr>
<td>14+</td>
<td>STD</td>
<td>SBA</td>
<td>118</td>
<td></td>
<td>&lt;&lt;</td>
<td>Secondary TD</td>
</tr>
<tr>
<td>15#</td>
<td>TC</td>
<td>DS</td>
<td>114</td>
<td></td>
<td>&lt;&lt;</td>
<td>Secondary ED</td>
</tr>
<tr>
<td>16#</td>
<td>SRD</td>
<td>SBB</td>
<td>119</td>
<td></td>
<td>&lt;&lt;</td>
<td>Receive Clock</td>
</tr>
<tr>
<td>17#</td>
<td>RC</td>
<td>DD</td>
<td>115</td>
<td></td>
<td></td>
<td>Not used (usually)</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Secondary RTS</td>
</tr>
<tr>
<td>19+</td>
<td>SRTS</td>
<td>SCA</td>
<td>120</td>
<td></td>
<td>&lt;&lt;</td>
<td>Data Terminal Ready</td>
</tr>
<tr>
<td>20</td>
<td>DTR</td>
<td>CD</td>
<td>108.2</td>
<td></td>
<td>&lt;&lt;</td>
<td>Signal Quality</td>
</tr>
<tr>
<td>21#</td>
<td>SQ</td>
<td>CG</td>
<td>110</td>
<td></td>
<td>&lt;&lt;</td>
<td>Ring Indicator</td>
</tr>
<tr>
<td>22</td>
<td>RI</td>
<td>CE</td>
<td>125</td>
<td></td>
<td>&lt;&lt;</td>
<td>Data Rate Selector</td>
</tr>
<tr>
<td>23#</td>
<td>CB</td>
<td>CI</td>
<td>111</td>
<td></td>
<td>&lt;&lt;</td>
<td>Data Rate Selector</td>
</tr>
<tr>
<td>24#</td>
<td>XTC</td>
<td>DA</td>
<td>113</td>
<td></td>
<td>&lt;&lt;</td>
<td>Ext. TC</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Busy</td>
</tr>
</tbody>
</table>

* Denotes rarely used
+ Denotes only if secondary channel implemented
# Denotes only on synchronous interfaces
>> Denotes from DTE to DCE
<< Denotes from DCE to DTE
== Denotes Ground line
CCITT = International Telegraph and Telephone Consultative Committee

Figure 2 - Transferring an 'A'

|     |     |     |     |     |
|-----|-----|-----|-----|
| High| Strt| 0   | 0   | 0   | 0   |
|     |     | 1   | 0   | stop|

Figure 3 - Three ways to hook up DTE to DTE or DCE to DCE devices. Note pin reversals, and that pin 20 may be used in a full-handshake system.

<table>
<thead>
<tr>
<th>NULL MODEM</th>
<th>NO HANDSHAKE</th>
<th>FULL HANDSHAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>1-20</td>
<td>1-20</td>
</tr>
<tr>
<td>3-2</td>
<td>3-2</td>
<td>3-2</td>
</tr>
<tr>
<td>4-1</td>
<td>4-1</td>
<td>4-1</td>
</tr>
<tr>
<td>5-1</td>
<td>5-1</td>
<td>5-1</td>
</tr>
<tr>
<td>6-1</td>
<td>6-1</td>
<td>6-1</td>
</tr>
<tr>
<td>7-1</td>
<td>7-1</td>
<td>7-1</td>
</tr>
<tr>
<td>8-1</td>
<td>8-1</td>
<td>8-1</td>
</tr>
<tr>
<td>20-1</td>
<td>20-1</td>
<td>20-1</td>
</tr>
</tbody>
</table>

NOTE: Pin 1 is optional for Chassis Ground between Terminals.
strings (often called blocks) of data usually have special beginning and ending characters which identify the string and contain check sums.

Synchronous data transmissions are normally more intelligent than asynchronous (smarter operators). Packet networks such as X.25 often transfer data synchronously.

Final Details

The data transfer speed (baud rate) is usually determined by the serial clock. The number of bits per character is usually handled by the UART or SIO chip. (Most data communications are done at 8 bits/character, 1 stop bit, no parity.)

To send an ASCII "A" (41 hex), the SIO would output 0100 0011 (binary) with the right-hand 1 being the least significant bit, and the left most 0 being the most significant bit. Figure 2 shows what you'd see on the data output pin of the RS-232 connector. Bit flow is from left to right, and note again that the least significant bit is sent first.

Logic 1 is low as shown.

Non-Standard Wiring Connections

There are several ways of hooking together two devices of the same type (DTE — DTE, or DCE — DCE).

If only one-way communication is required, you'll need at least two wires, one data line, and pin 7 ground. All handshaking conditions can be satisfied by looping back the appropriate signals on both ends of the cable (see Figure 3 — Null Modem diagram). The three typical connections are null modem, no handshake, and full handshake.

Figure 3 shows the necessary wiring. Note that the manufacturer may wire the device either as a DTE or DCE, so you need to read the manual.

(I know, it's a last resort, but...)

I hope this info saves you some time; I've spent many hours uncovering it the hard way.

Final Details

The data transfer speed (baud rate) is usually determined by the serial clock. The number of bits per character is usually handled by the UART or SIO chip. (Most data communications are done at 8 bits/character, 1 stop bit, no parity.)

To send an ASCII "A" (41 hex), the SIO would output 0100 0011 (binary) with the right-hand 1 being the least significant bit, and the left most 0 being the most significant bit. Figure 2 shows what you'd see on the data output pin of the RS-232 connector. Bit flow is from left to right, and note again that the least significant bit is sent first.

Logic 1 is low as shown.

Non-Standard Wiring Connections

There are several ways of hooking together two devices of the same type (DTE — DTE, or DCE — DCE).

If only one-way communication is required, you'll need at least two wires, one data line, and pin 7 ground. All handshaking conditions can be satisfied by looping back the appropriate signals on both ends of the cable (see Figure 3 — Null Modem diagram). The three typical connections are null modem, no handshake, and full handshake.

Figure 3 shows the necessary wiring. Note that the manufacturer may wire the device either as a DTE or DCE, so you need to read the manual.

(I know, it's a last resort, but...)

I hope this info saves you some time; I've spent many hours uncovering it the hard way.

DriveLine

Check Floppy Drive Alignment
No tools or interfacing needed
SSSD Runs on All CP/M* 22 or 3.1
8" Systems
Digital Diagnostic Disk Included
$65 Check or MO Ppd
Other formats special order
Chandler Software
273 West Shore Dr.
Marblehead, MA 01945
(617) 631-4985

* TM Digital Research Inc
&TM Olsen Corporation

Windowing for CP/M!
$49.95 Basic Compiler

MTBASIC

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

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

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

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

ORDERING: Specify format when ordering. We accept Visa, MC, checks and COD. Send $49.95 plus $2.50 shipping and handling ($10 overseas) to:

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

Still Searching For Files Without EUREKA!

??

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

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

People who try EUREKA! love it....

"Just started cataloging with comments - Great Idea " GR-NJ
"Great time saver in locating material on disks." WB-NY
"Your manual is the best written I have ever seen. " MT-NS
"We finally chose EUREKA!...largely because it has the ability to read comments directly from a file,...EUREKA! is easy to learn and use, has more access and report choices, finds files by many different ways, and has an attractive price."

T. Bove & C. Rhodes, USER'S GUIDE No. 11

EUREKA!, the popular CP/M disk cataloger

Still only $50

Mendocino Software Company, Inc.
Dept. MC-1
P.O. Box 1564
Willits, CA 95490
(707) 459-9130

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

Suffolk Software, Inc.
P.O. Box 2412 Columbia, MD 21045-1412
301/792-8096
Hey, Wait! DON'T TURN THAT PAGE! It's still me, only the name has been changed to protect the incompetent. That's right, this is the latest incarnation of The Slicer Column. We've changed the name because many new readers have no idea what a Slicer is ("some kinda tater masher, huh, Joe Bob?" "Naw, Frank Chuck, I think it's the name of that new slash murderer down in San Francisco.").

Along with being more descriptive of the column's contents, the new title will give me more latitude in my discussions (not that I have had a latitude problem in the past. An attitude problem perhaps...).

The '86 part means that I will be covering anything happening with machines that use the 8088, 8086, 80188, 80186, 80286 (80386, 80486 ???) processors. I wouldn't mind covering the 68000 as well, but the line must be drawn somewhere. Maybe later — hint, hint. (Editor's note: Hold on, there. Leave something for the rest of us!)

By the way, when I say "anything" I don't mean reviews of new spreadsheets and databases. I will restrict my software opinions to programs that can be considered "system" software or programmer's tools (i.e., compilers and editors, but not word processors).

Of course, none of this means I'll be ignoring the Slicer. Since the 86 system I have is a Slicer, the content of the column will necessarily be heavily Slicer-oriented. There will probably be very little change from the past — just a broadened perspective. This broadening will take awhile though, since I have to get the information before I can report on it.

Turkey Talk (Fowl Play)

"World" may seem a bit ambitious to use in the title of a mere magazine column, but this truly will be a "world class" column. After months of weighing different career choices, I have finally chosen not to make a choice. Instead, I'll be leaving the U.S. in mid-August for a two year stay in Ankara, Turkey, where I'll be working for The Development Foundation of Turkey.

This will put a crimp in my telephone habits, but I hope I'll be able to use Telex and the mail to keep on top of new developments (anybody have a spare satellite channel I can borrow?). Maybe someone could come up with an inexpensive world-wide network system based on shortwave radios (gotta be some dollars in that one).

DFT helps farmers in rural areas modernize their operations. I'll attempt to put computer technology into use in as many places as are technically and financially feasible. If anyone out there has been involved in anything related to this, I would really love to hear from you. This is going to be the biggest challenge I've ever faced, and aid will not go unnoticed.

Eventually, I'll be able to let you know my address in Ankara. In the meantime, just send any letters to Micro C, and they'll be forwarded.

New Products

Slicer intends to release at least two new products at the SOG this year. The first is something they introduced at last year's SOG, but still needed time to mature. Yep, that's right, you can finally buy Concurrent CP/M for the Slicer. And it's not just plain old vanilla CCP/M — it can window on both the PC video board and any serial terminal with cursor positioning commands. You can even connect multiple terminals and have windows on all of them (if you're feeling really schizophrenic).

Performance seems to be extremely good — one person in Australia reportedly has more than six terminals on his system, and it still performs acceptably. They're working on the PC-DOS emulation module, too; it may not be ready in time for the SOG, but very soon you'll be able to run CP/M and MS-DOS programs concurrently.

The other product is really a variation of an existing product. Slicer will now be selling a 1 megabyte expansion board as well as the original 256K version. It has the four serial ports, the printer port, and the clock just like the 256K unit, but it uses 256K DRAMs to fit an entire megabyte of RAM onto a single board.

Since the 80186 can directly access only 1 megabyte at a time, and more than 256K of that is already taken up by the memory on the main Slicer board, the RAM is split up into four chunks of 256K. The first 256K resides immediately above the Slicer's 256K; the other three banks of 256K can be switched in one at a time into the first 256K of memory space (one of these banks is mapped slightly differently to allow access to video memory at the same time). All of the bank switching and mapping is done with PALs, so you should be able to easily modify the mapping for custom applications.

Having this much memory opens the door for some incredible multi-track disk buffering schemes. It also may be possible to put parts of the operating system into a bank that's switched out most of the time, leaving more memory open for applications.

Project X

Also to be announced at the SOG is PC Tech's "Project X," now called "The Scepter," I understand. This is an 80186 board that magically fits into the same space as an IBM motherboard. This allows you poor unfortunate with aging PCs to upgrade your system without giving up that precious white cabinet and silver emblem.

The Scepter can run at either 8MHz or 10MHz. It has a socket for a floating point chip (the 8087, I assume), a SASI port for connecting a Winchester drive, and identical floppy hardware to the IBM. There is no onboard video circuitry, but there are five PC-type card slots so you can plug in any video card you like (a Hercules hi-res, maybe?). I haven't seen one yet (nobody has), but it appears that the only thing not PC compatible on the board is the clock speed (and the performance). More on this one after I
have seen it in the flesh.
Ampro will be officially releasing its 80186 Little Board at the SOG, too (it has its SCSI/Plus multi-master bus and runs at 8 MHz).

New User Disks
Since everyone else is releasing new products, I've decided to release some of my own. I've put together two new CP/M 86 user disks, #7 and #8.

Disk #7 contains: VFILER, a screen oriented SWEEP-type utility, AR-CHIVE for automatically backing up all files modified since the last back-up, expansion board clock routines written in Turbo Pascal, and several other programs. Aside from the clock programs, everything on this disk should work on any CP/M 86 system.

Disk #8 has the latest Slicer BIOS, a new STRANGE program that allows the Slicer to read both Morrow and Kaypro format 5 inch disks, and lots of other great stuff. The BIOS now supports CP/M's IOBYTE feature as well as printing the current time and date during cold boot if the expansion board clock is installed. Double-sided 5 inch IBM format is supported in the BIOS, so it's no longer necessary to use STRANGE to read IBM disks.

We'd like to assemble more CP/M 86 disks as well as MS-DOS disks, so send anything useful to Micro C and get a free user disk of your choice.

By the way, I discovered the other day that MODEM7 from disk #1 would not run on my Slicer with the PC video board installed. Following the advice given in the video board manual, I made the stack larger and reassembled; now it works fine.

Half Tracked Format
To allow formatting 40 track formats on 80 track drives on the Slicer, edit SLIFORM.A86 and change the drive-type bytes of the 40 track formats to indicate a "half-track" format, then reassemble. It's better to format 40 track disks on a 40 track

(continued on page 41)
The Slicer

REAL 16-BIT POWER ON A SINGLE BOARD
Featuring the Intel 80186

- Complete 8 MHz 16-bit microprocessor on 6"x12" board
- 256K RAM, plus up to 64K EPROM
- SASI port for hard disk controller
- Two full function RS232C serial ports with individually programmed transmission rates 50 to 38.4K baud
- Software compatibility with the 8086 and 8088
- 8K of EPROM contains drivers for peripherals, commands for hardware checkout and software testing
- Software supports most types and sizes of disk drives
- Source for monitor included on disk
- BioS supports Xebec 1410 and Western Digital WD 1002 SHD controller for hard disks

Fully assembled and tested only $995

The New 1-Megabyte System Expansion Board!

FOR EXPANDED MEMORY, ADDITIONAL PORTS, AND REAL TIME CLOCK

- 1-MB additional dynamic RAM
- 2 RS232C asynchronous ports with baud rates to 38.4K for serial communication
- 2 additional serial ports for asynchronous RS232C or synchronous communication (Zilog 8530 SCC)
- Real Time Clock (with battery backup) for continuous timekeeping
- Centronics type parallel printer port

Fully assembled and tested only $800

The Slicer

PC Expansion Board

GIVES YOUR SLICER HIGH PERFORMANCE VIDEO CAPABILITY

- IBM compatible monochrome video
- Video memory provides 4 pages of text or special graphics capability
- 2 IBM type card slots for color video, I/O expansion, etc.
- IBM type keyboard port

Fully assembled and tested only $600

SLICER COMPUTERS INC.
2543 Marshall Street N.E.
Minneapolis, MN 55418
(612) 788-9481

CONCURRENT CP/M — with manuals $250
without manuals $200

The SLICER DOS Kit $295
includes Bare Board, CPU and choice of MS-DOS or CP/M-86

The SLICER Enclosure System
alone $125
with 135W Power Supply $245
with 135W Power Supply and 2-80 track 5-1/4" Disk Drives $695

PC Compatible Keyboard $150

PC Compatible Amber Monitor $175

10MB (formatted) 5-1/4" Half-Height Hard Disk $700

Western Digital 1002-SHD Hard Disk Controller $200
assembled and tested

8087 Math Co-Processor Board $300
bare kit form

Check the SLICER Bulletin Board (300/1200 baud) at 612/788-5909 for complete set of manuals and access to Slicer's design team.

All products are available in several kit forms:
Operating Systems are:
CP/M-86 by Digital Research, Inc. $85
MS-DOS by Microsoft Corporation $175

MasterCard, Visa, Check, Money Order or C.O.D.
Allow four weeks for delivery. Prices subject to change without notice.
drive, but when you're like me and don't have a 40 track drive, there's no alternative.

The drive type byte is the very first byte in each table of disk info in SLIFORM. To indicate a half-track format, just change bit 2 from 0 to 1. For example: 1011B becomes 1111B.

AT Drives

Also included on User Disk #8 is the new code that must be patched into the Slicer ROMs to allow proper operation of the new drives used on the IBM AT. These drives can look identical to either an 8 inch or a 5 inch drive. When in 8 inch mode, you can use any of the Slicer 8 inch formats (i.e., you get 1.3 megs on a 5 inch double-sided floppy!). These drives are still more expensive than the quad density 5 inch drives, but I have heard of prices in the $200 range.

Both software and hardware changes are necessary to properly use the full capability of these drives. Without changing the software, they could be used in 8 inch mode, but that wouldn't be nearly as nice as automatically selecting 5 or 8 inch modes.

The following description of the hardware and software changes needed to add this capability to the Slicer is a direct transcription (with some editorial remarks added) of information sent to me by Brian Schieferstein & Richard Herr of Gilford Instrument Laboratories in Oberlin, Ohio.

Thanks, guys. Hope you don't mind if I plagiarize a little...

Schieferstein & Herr's Auto Select

IBM PC/AT compatible 1.6M byte 5-1/4 inch (Shugart SA475 et. al.) modifications for the Slicer —

To implement SA475 type drives on the Slicer, two signals must be added to J6 (5-1/4 inch floppy connector). To add these signals, make the following modifications to the Slicer board:

1. Open your Hymnals to page 3 of the Slicer schematics. Say a short prayer to the God of Cold Solder Joints.

2. Lift U31 pin 12 out of the socket. This removes the default ready signal when in MINI mode.

3. Connect J7 pin 22 to J6 pin 34. Now we can get a READY signal from drives on J6.

4. Connect the MINI signal from U31 pin 13 to U35 pins 1 & 2. (Note that we are using a NAND gate as an inverter here. The newly disconnected

(continued next page)
inverter on U31 could be used instead, but that would require cutting a trace.) (Remember: THINK first, then do.)

5. Connect U35 pin 3 to J6 pin 2. This is the speed control signal.

These modifications will not function properly in the 5-1/4 inch mode with the current software due to the way the READY signal is changed (but really isn’t) after the drive is selected. This can be corrected with the patches to the version 1.3 monitor EPROMS in Figure 1. These changes will automatically select any Slicer compatible format (8 inch or 5 inch) in the two speed drive. To put these changes into the EPROM code, type them into a file and use an INCLUDE statement to include the new code in SLICER.A86 just after the INCLUDE PATCHES statement, then reassemble and burn the new EPROMS.

**Video Board Bugs (And Fixes)**

While we’re on the subject of hardware mods, there is a slight timing problem on some of the earlier Slicer video boards. This mistiming causes the “underline” attribute to work intermittently (i.e., flashing), if at all. Slicer has corrected the problem by changing one of the PALs, but you can fix it yourself with just a single 10pF capacitor.

Connect the capacitor between U36 pin 8 and ground. This slows the leading edge of the BLANK signal slightly and clears up the problem.

There is also a bug (actually an undocumented feature) in the earlier video board software. The video attribute setting commands were mistakenly put in the ROM as ESC-a and ESC-b instead of IBM’s ESC-b and ESC-c. This was changed in later versions of the ROM. If your manual doesn’t list these commands at all then you probably have the version with incorrect commands; the problem was discovered and corrected when the new commands were being added to the manual. If you have the older ROMs, call Slicer about an update.

---

**Figure 1 - Patches to the Version 1.3 Monitor EPROMS**

```
;  5-20-85. implements drive select for dual speed
;  5 1/4'' drives such as the Shugart SA475.
;  FLOPSEL and SELFLOP are modified.

PATCHNEW1:
  ORG OFFSET FLOPSEL
  ;Select floppy disk.
  NEWFLOPSEL:
    MOV DX, SYSPORT+12 ;Min1?
    MOV AL, DTYPE[DI]
    OUT DX, AL
    MOV DX, SYSPORT+14 ;Double density?
    SHR AL, 1
    XOR AL, 1
    OUT DX, AL
    MOV AH, SELLOW ;Motor on for up
    MOV AH, DOSTIME ;Set deselect time
    ADD AH, AL
    MOV DESELECT, AH
    ADD AL, DTIME
    MOV FDSTOP, AL
    TEST AL, COUNT
    JNZ NEWFLOP6
    CALL DURING ;Turn drive on.
    MOV AL, COUNT
    ADD AL, FDelay ;to get up to speed.
    NEWFLOP5:
    CMP AL, COUNT
    JNZ NEWFLOP5

NEWFLOP6:
  MOV CX, 4
  MOV AL, DISK[BX] ;Select.
  MOV DX, SYSPORT+10 ;Drive 0.
  MOV AH, DL ;Compute lower part of
  SUB AH, AL ;address for selected
  ADD AH, AL ;drive
  NEWFLOPO:
    MOV AL, 0 ;Deselect all drives
    CMP AH, DL ;except the one that
    JNZ NEWFLOP4
    INC AL ;matches AH

NEWFLOP4:
  OUT DX, AL
  SUB DL, 2
  LOOP NEWFLOPO
  RET

NEWSELFLOP:
  ORG OFFSET SELFLOP
  PUSH BX
  MOV BX, DSKPTR ;Set xlate to do nothing
  MOV WORD PTR [BX], OFFSET NOMEM
  POP BX
  AND DTYPE[DI], FLOPPY-MINI-DIGITAL
  CALL FLOPSEL

NEWFD7: PUSH WORD PTR STEERATE[DI] ;remove verify
  AND STEERATE[DI], NOT 1
  MOV LSTTRK[BX]+1, 1
  MOV TRACK[BX], 2 ;calibrate then seek track 2
  MOV SIDE[BX], 24H ;read address
  PUSH DSKSEG ;set disk memory
  POP DSKMEM ;to select buffer
  MOV DSKSEG, SEG SELBUF
  MOV SELLEN[DI], 6
```

Micro Cornucopia, Number 25, August-September 1985
I discovered both of these problems on my own and was thinking, as I dialed the phone, of how fun it was finding new bugs and being the first to tell the developers about it. I told Earl (Hinrichs at Slicer) about the underline problem, and he said “Okay, we found that a few days ago. The fix is...” Then he told me about the attribute bug before I even had a chance to mention it. Geez, come on, let me have a little fun, huh? Even I, Laine Stump, need a little ego boost now and then.

### The Wini Works

Speaking of things being fixed (don’t ever say that around your male cat — you may never see him again), my aging 506 Winchester drive is again functioning. The problem was just a blown chip on the Slicer board, probably from plugging in the first Western Digital controller I received (it was the wrong model). It has now worked flawlessly for two months.

### Next Please

Brian Schieferstein and Richard Herr also sent me instructions for using 256K DRAMs on an expansion board to get 512K on one board. I’ll be trying to use the same technique to put 512K on the main Slicer board. I may have space to take a quick look at assembly language programming and I/O redirection under MS-DOS, too. I/O redirection capabilities simplify many assembly programs greatly, since there is no need to explicitly open and close files, etc.

While I’m on the subject of things for the future: Dave and I (and now Earl, I hear) have been looking for information on the MIDI (Musical Instrument Digital Interface) standard. I’ve had no luck, and neither had Dave last time I spoke with him. Anyone who has information on this (or any other computer/musical type things), drop us a line.

P.S. No Commodores, please.

---

Micro Cornucopia, Number 25, August-September 1985

43
SUPERSPEED FOR YOUR KAYPRO

Electronic RAM disk and Printer Buffer Combination—the ultimate add-on for Kaypro II, 4, 2-84, 4-84 and 2X Computers.

Incredible speed and efficiency are offered using MicroSphere’s dual operation Electronic RAM disk and printer buffer. No longer will you need to wait for your Kaypro to slowly finish mundane tasks such as running a printer or waiting for floppy drives to turn when you have better things to do.

RAM disk size  Configuration price
Unstuffed board—any size; fully tested $295.00
256K 4K or 32K printer buffer $395.00
512K 4K, 32K or 64K printer buffer $495.00
768K 4K, 32K or 64K printer buffer $595.00
1mb 4K, 32K or 64K printer buffer $695.00

U.S. add $5.00 shipping International add $15.00 shipping

When ordering, please include the model of Kaypro to be used. Boards can be upgraded should your needs change.

Typical speed increases you can expect to see using MicroSphere’s RAM disk:

- 4MHz Kaypro
  - Floppy Disk: Perfect Calc 9.31, 25 11:17, 78
  - RAM Disk: 9.38* 2.12*

- 20k file, 11 pages, 2586 words, using PIP to the LST device
  - Time in seconds

Comes complete with cabinet, cables, software and connectors.

TLC Logo for Kaypro Computers. Easy and exciting language for all ages.

TLC LOGO is an exceptionally complete logo with vectors, multiple turtles, full floating point decimals and extremely fast program execution.

TWO versions offered:
- STANDARD version using only internal graphics of Kaypro 2-84, 4-84, 2X and 10; no hardware additions or modifications are required; 16,000 pixel resolution.
- DELUXE version for all Kaypro Computers that have the MicroSphere Color Graphics board. Features 16 colors, 32 sprites, 49,000 pixel resolution and utilities such as screen dump to printer or disk.

Time comparison of 3 common Logo programs currently offered:

<table>
<thead>
<tr>
<th>DR LOGO</th>
<th>TLC LOGO</th>
<th>Apple LOGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>320k IBM PC</td>
<td>64k Z80</td>
<td>64k Apple Ile</td>
</tr>
<tr>
<td>Circle test</td>
<td>10 seconds</td>
<td>3 seconds</td>
</tr>
<tr>
<td>PolySpiral1</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>PolySpiral2</td>
<td>out of stack</td>
<td>7</td>
</tr>
<tr>
<td>Square Test</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>Four Bugs</td>
<td>78</td>
<td>N/A</td>
</tr>
<tr>
<td>(req. 4 turtles)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Times provided by The Lisp Company ... (note: out of stack indicates inadequate implementation of “tail recursion”) DR LOGO is copyright Digital Research Company, Apple Logo is copyright Apple Computer Company, and TLC Logo is copyright the Lisp Company.

- Standard Version of TLC Logo ........................................ 99.95
- Deluxe Version of TLC Logo ......................................... 129.95
- Special: Color Board and Deluxe TLC Logo ....................... 199.95

A NEW DIMENSION FOR KAYPRO COMPUTERS: Color Graphics Board. Features 16 colors, 32 sprites, 256x192 bit mapped graphics. 16K of RAM on the color board itself allows creation of graphics without losing internal memory of the Kaypro.

Software includes 3 editors, drivers and routines to access graphic system. Utilities include screen dump to disk and printer. Dual screen operation features internal Kaypro screen for text and commands, external graphics screen for results. A TV set can be used with addition of RF Modulator.

- Color board/Kaypro II, 4, 2-84, 2X, 4-84, 10 & Robie.................. 145.00

INSTANT GRAPHER 2.1 (For use with Color Graphics Board) Creates bar charts, stacked bar charts, hi/low, line graphs from keyboard, Perfect Calc, CalcStar or text files. Single and Double size dumbscreen to printer, each color prints a different pattern on a standard dot matrix printer .................. $40.00

SUPPLY STORE SPECIALS

Mag Media Disks:
- SSDD ........................................... $20.00 box of 10
- DSDD ......................................... $24.00 box of 10
- 100 SSDD Bulk .................................. $160.00

These are great disks! You will love them! We have run the SSD on DSQD without a problem.

64k Dynamic RAM Chips:
- 150NS or 200NS .................................. $.99 ea.
- TMS 9918 Color Chip & Crystal ................ 29.95

Infocon ADVENTURE Games
- Kaypro or CP/M 8" Available
  - 2 Zork 1 ..................................... $34.95
  - 3 Zork 2 ..................................... 39.95
  - 3 Zork 3 ..................................... 39.95
  - 4 Deadline .................................... 44.95
  - 4 Starcross .................................... 44.95
  - 4 Suspended ................................... 44.95
  - 2 Witness ..................................... 44.95
  - 2 Planetfall ................................... 44.95
  - 2 Enchanter ................................... 44.95
  - 3 Infidel ..................................... 39.95
  - 3 Sorcerer ..................................... 39.95
  - 1 Seastalker ................................... 34.95
  - 2 Cutthroats ................................... 34.95
  - 2 Hitchhikers ................................... 34.95
  - 3 Suspect ..................................... 39.95

LEVELS: 1=Jr., 2=Standard, 3=Advanced, 4=Expert

Invisicubes ........................................... $7.95 ea.

6MHZ Z80 Support Chips
- Z80B CPU ....................................... $12.00
- Z80B PIO ....................................... 12.00
- Z80B SIO/O ..................................... 20.00
- Z80H CPU (8MHZ Z80) .......................... 20.00
- RF Modulator for Color Board and TV operation ..................... $25.00

MicroSphere’s Composite video generators allow connection of a STANDARD external monitor to the Kaypro Computer. Custom monitors or modifications are not required.

Composite video generator:
- Kaypro II and 4 ................................ $49.95
- Kaypro 2-84, 4-84, 2X and 10 .................. $89.95
- IBM-TTL Compatible Monitor Adapter, Kaypro 2-84, 4-84, 10 and 2X .................. $49.95
- Quadram Amberchrome IBM-TTL Monitor .............................................................. $195.00

Zenith Monitors
- ZVM 122 Composite Amber ..................... $119.00
- ZVM 123 Composite Green ...................... $114.00
- ZVM 135 Composite Color/RBG/Green Hi Res. ........................................... $500.00

NEW

JL

MicroSphere, Inc.
P.O. Box 1221
Bend, Oregon 97709
503-388-1194
9-5 Pacific Time

Dealer inquiries invited.
I spend a lot of time on an IBM-PC (my mainstay at work) and a Kaypro II. Until recently, I didn't need to transfer files between these machines. Now I do—mostly Turbo Pascal source files, numerical data files, and text files.

There are only a handful of transfer programs on the market. For (my mainstay at work) and a Kaypro II. Review there's Vertex Systems' Xeno Copy week.

1. ...in any of the 25 formats.
2. ...arguments to list single-sided formats.
3. ...or wash. The single-sided limitation is also a problem.)
4. ...and 24 other programs like pip, type, or wash. The single-sided limitation is also a problem.)
5. ...you make the proper choice, voila, you get the disk's directory.
6. ...or by daily exposure to other software. Occasionally, my fingers succumb to a vagrant compulsion to type an 'L' to log in a disk, an 'F' to format, etc. To Media Master's credit, nothing happens until I type a '4', the correct command. Media Master doesn't permit syntax errors but waits instead for you to get it right.

I have used Media Master after running Micro C's dump24 (a screen dump program). It turns out that erasing a file under Media Master causes a screen dump to the printer.

Overall, I think Media Master does everything it claims to do.

NOTE: Since this article was typeset, DG/Systems has changed its name to Spectre Technologies.
Kaypro Column

By David Thompson

A lot of people have called asking if the 8" interface for the Kaypro really exists. Yes, it does (in the flesh or a facsimile), and it's available.

You get the Pro-8 version 2 features (if you have an 85 Kaypro), or the Pro-8 MAX features (if you have an 84 Kaypro). With either you can run up to four drives on your system. A: and B: can be any type of 5" drive; C: and D: can be any type of 5" or 8" drive. Double-sided 8" drives will read and write single-sided, single density 241K disks (like the SIG/M disks) as well as single- and double-sided double density disks (600K and 1.2 meg). That's 16 sectors of 512 bytes per sector, 77 tracks.

If there's room in your system, the monitor will also read and write 1024 bytes per sector in double density. This gives you 670K per single-sided disk and 1.34 meg per double-sided disk. You can make room by moving CP/M down 1K.

The price for the 8" board is $190. You need to specify whether your keyboard is '83 or '84. If you have a Kaypro II-83 (the monitor ROM's paper reads 81-149), then you'll need to do the II to 4 upgrade. See issue #15 or #24 for upgrade information.

It's easy to install the board. Just unplug the 1793 floppy disk controller chip, plug in our board, and then plug in the 5" and 8" drive cables.

You supply the 8" drive(s), cabinet, power supply, and cables. There are many sources for drives, and many different types of 8" drives. Siemens drives are the standard on the surplus market. They aren't the best because the stepper motors die if you use them a lot, but they're cheap ($100 - $130 single-sided and about twice that for double-sided).

Cascade Electronics usually has a variety of slightly used Shugarts. Check with them; their drives and service are consistently good, and they also may have power supplies and cabinets. Their number is 507-645-7997.

If you want to go fancy, you might consider the Mitsubishi 8" half-height double-sided units. They look good, and they don't require 110V AC, so you can run them in portables.

Some say it's a contradiction to talk about portable systems with 8" drives, but you can ignore those people.

286i

Most of the systems in our office haven't seen their boxes or their tops since the day they arrived (unless they're systems we built, which means they never had tops). However, our second 286i sits forlornly on the floor in its unopened box.

It's not that we don't have great plans for the system. We do. In fact, we've spent a good deal of time trying to add a hard drive to the 286i. We haven't even gotten it to select the Winnie. The system supposedly supports a number of different hard disks (they are numbered between 1 and 14), but we don't have a comprehensive list yet. The 20 meg Tulin TL200 Series, the 10 meg Shugarts, and the 20 meg Seagates work (at least in Portland and Solana Beach). Rodime, High Technology, and the Seagate ST-506 have had problems. Xenix requires type 2 drives to work, and PC-DOS is limited to 30 meg per drive. (Whew.)

The Rodime drive we tried was never selected. We checked the drive select lines, and none corresponded with the hard disk select light on the 286i's front panel. After we returned our first system to Kaypro, they called and said that it interfaced properly with a #2 drive (type #2 in the AT Diagnostics). If you're adding a Winchester, you have to purchase the AT's advanced diagnostics from IBM ($270) in order to format the Winnie.

Kaypro doesn't supply MS-DOS 3 with the system; you have to purchase that from an IBM dealer (it'll cost you $60).

When Kaypro chose to ship their $4500 system without a Winchester, they said it was to "allow" users to install drives of their choice. You'd think they could at least write a disk formatter and supply a $60 copy of the operating system. Kaypro says it's negotiating with Microsoft for a multi-user version of MS-DOS to include with the 286i. We have heard rumors, however, that one of the bugs still residing in the 80286 makes it impossible to support multi-processing or multi-user. As for the formatter, I think they need to write one.

I tried formatting some ordinary floppies as 1.2 meg in the 286i. Our Mag Media disks that format perfectly as quad density (784K) work very poorly at the higher density.

1.2 meg format is 96 tpi (80 tracks per side) like the quad density, but the disks spin at 360 rpm instead of 300 rpm — and the data is written at 500K bits/second (like 8" double density) instead of 250K bits/second. So the number of tracks is the same as quad density, but there's almost twice as much data crammed into each track.

The double density Mag Media disks format to about 800K usable space after formatting. At least MS-DOS automatically traps the bad sectors, but with a disk that marginal, I wonder how long it will take before errors show up.

So I have to figure on spending $4 each for the high density disks. Fortunately, the system can read 360K disks, and if you need to write data onto these standard format disks, you can exchange a double-sided 48 tpi drive for one of the 1.2 meg drives.

Three 286is

The 286i is now available in three versions — a stripped down model with one 1.2 med drive, a rigid/floppy controller board, 512K RAM, and GWBasic for $2995; a spiffier model with two 1.2 meg floppies, controller, 512K RAM, graphics/parallel card, parallel/series card. GWBasic and MicroPro software for $4795.

The internal design of all models is very AT-like, except Kaypro is using 256K DRAMs instead of 64K chips piggybacked (a la AT).
Kaypro 10 Hard Disk Failure

Jeff Brown, Kaypro's district manager in Oregon, says he's been seeing a lot of hard disk failures on the Kaypro 10's that are directly related to heat (usually a clogged fan) or inadequate power protection. The first is easily remedied.

For the second, he suggests you use a dedicated power line (don't run your coffee pot or air conditioner on the same connection), and make sure that you have good solid +12V (+ or - .35V) to the disk and the controller. It's best to use an uninterruptable 110V source (UPS) for triple A insurance (some service companies won't sell extended warranty contracts on hard disk systems that don't have UPS because of hard disk failures from power-related accidents).

SMART-PIK
NFL & USFL Football Handicapping Program

PURPOSE: Do-it-yourself Program. For Kaypro users interested in predicting the outcome of professional football games.

HOW IT WORKS: Each week, enter the stats required by the SMART-PIK formula. SMART-PIK projects the number of points that each team is likely to score (and allow) against its next opponent.

RESULTS: A winning record, against the pointspread, in each of the last 4 Pro Football seasons. (1983 and 1984, NFL and USFL. More than 500 selections.)

QUICK & EASY: Required stats are obtainable from any newspaper. I/O takes about 2 hours per week. (Also, data maintenance via mail or modem is an available option.)

VARIABILITY FEATURES: You can easily vary the SMART-PIK selection determinants. Further experimentation may produce still better results.

Developed by Marty Mendelsohn and Jim Thompson of Las Vegas Kaypro Users Group.

$49.00 includes postage and handling. Phone or Write for Free Brochure

Please specify Disk format
CP/M 8" IBM format, KAYPRO II, XEROX 820, OSBORNE I, others.

Please specify method of shipment, UPS or Postal Service.

California residents add 6% Sales Tax. Dealer Inquiries invited.

Micro Cornucopia, Number 25, August-September 1985
If's like having a friend to help:

Customize WordStar® for
- Faster screen response
- Optimal printer output
- Custom patches

Get started with Communications
- Transfer files across the country or across the room
- Communication Software on disk with Examples

Connect Equipment
- RS-232 & Handshaking explained
- Cable Connection Examples for over 70 printers

FastFacts Command Reference
- Quick access to commands for common programs

Learn CP/M® Shortcuts
- Commands with examples
- Turn your Kaypro into an electronic typewriter with PIP
- Modify programs with DDT

Understand the great Public Domain Software included on disk
- Catalog and Organize disks
- Time and Date stamp files without a clock
- Writers & Disk Utilities

Programmers Reference
- Software & Hardware Interface points

SLIPCASED KAYPRO COMPANION BOOK AND DISK PACKAGE $35.00

PLUS WORDSTAR® RECOVERY PROGRAM TO SAVE FILES WHEN DISK IS FULL OR WHEN OTHER SAVE PROBLEMS OCCUR WSFIX.COM a $30 value

AN APPLICATION RESOURCE
- It's like having a friend to help:

Customize WordStar® for
- Faster screen response
- Optimal printer output
- Custom patches

Get started with Communications
- Transfer files across the country or across the room
- Communication Software on disk with Examples

Connect Equipment
- RS-232 & Handshaking explained
- Cable Connection Examples for over 70 printers

FastFacts Command Reference
- Quick access to commands for common programs

Learn CP/M® Shortcuts
- Commands with examples
- Turn your Kaypro into an electronic typewriter with PIP
- Modify programs with DDT

Understand the great Public Domain Software included on disk
- Catalog and Organize disks
- Time and Date stamp files without a clock
- Writers & Disk Utilities

Programmers Reference
- Software & Hardware Interface points

SLIPCASED KAYPRO COMPANION BOOK AND DISK PACKAGE $35.00

PLUS WORDSTAR® RECOVERY PROGRAM TO SAVE FILES WHEN DISK IS FULL OR WHEN OTHER SAVE PROBLEMS OCCUR WSFIX.COM a $30 value

DIABLO 620 (H-P 2602)
RS-232 STORE DEMO LETTER QUALITY
25 CPS WITH SERVICE MANUAL $449.00

DIABLO 630 45 CPS
RS-232 STORE DEMO
45 CPS LETTER QUALITY $900.00

Z-125 9X9 DOT MATRIX 10, 12, 13.2, 16.5 CPI
RS-232 150 CPS WITH BLOCK GRAPHICS
95 ASCII CHARACTERS STORE DEMO $400.00

KEYBOARD 113 KEYS 8 BIT ASCII
PARALLEL OUTPUT WITH CASE
AND COMPLETE DOCUMENTATION $40.00

SA800-2 DISK DRIVE CLEAN - USED $65.00
SA850 DISK DRIVE CLEAN - USED $119.00
STANDARD MOUNTING - GUARANTEED 60 DAYS

CABINET FOR ABOVE HOLDS TWO DRIVES
POWER WIRING COMPLETED (AC AND DC)
FAN AND INPUT AC WIRING DONE
20 X 9 1/2 X 17 DEEP $60.00

POWER SUPPLY FOR TWO DRIVES (SWITCHING)
WILL POWER DRIVES AND SBC 6 X 8 X 2 1/2
5VDC @ 8 AMPs, +12VDC, -12VDC, -5VDC @ .5 AMPs EA AND 24VDC @ 2 AMPs $60.00

SHUGART DISK DRIVE PARTS
SA800 HEAD $15.00
SA800 STEPPER $10.00
SA800-2 LOGIC BRD NON-WORKING $10.00
SA800-2 LOGIC BRD WORKING $35.00
SA850 HEAD AND STEPPER ASSY (BI OR TRI) $50.00
SA850 LOGIC BRD NON-WORKING (BI OR TRI) $30.00
SA850 LOGIC BRD WORKING (BI OR TRI) $60.00
SA800 OR SA850 SPINDLE BEARINGS (SET) $ 3.00
SA800 OR SA850 INDEX LED (SET) $ 3.00
OTHER PARTS ON REQUEST $20.00

SA800 ALIGNMENT AND CLEANING $20.00
SA850 ALIGNMENT AND CLEANING $30.00

Z80A PARTS CPU, CTC, SIO/1, 1771, 2716 6845 CRT CONTROLLER $ 2.00
6 FOR $15.00

DIABLO PRINT WHEELS
COURIER 10, ELITE 12, COURIER 72 $ 3.00
6 FOR $15.00

CASCADE ELECTRONICS
ROUTE 1 BOX 8 FREE
RANDOLPH MN 55065 SHIPING
507-645-7997 ON OVER $40.00

COD ADD $2 CREDIT CARD ADD 5% MN ADD 6%
LIMITED TO STOCK ON HAND EVENING CALL OK
Color Speedometer For The Kaypro

By Mike Sweeney

The 5MHz speed up on the older Kaypro IIs is one of the truly great mods. But remembering your speed setting can be another matter.

I've formatted disks at 2.5MHz, and then jumped into another application without switching back to 5MHz. By the time I realized things had really slowed down, I was in too deep to back out for a speed change. And if I'm running too fast (for a formatter, for instance) I've really got problems.

Even if I do remember to check which speed I'm at, I still don't like reaching behind the computer to feel the switch position or push the reset button.

Fix

I decided to experiment with a tri-color LED (light emitting diode) that changed colors when the dc polarity was reversed. LEDs glow yellow on AC voltage and either red or green on DC voltage, depending on polarity.

I substituted the existing red DC power LED on the front of a Kaypro with a tri-color LED. Then I replaced the single-pole double-throw speed switch with a 3-pole double-throw switch, using one pole for the speed select and the other two poles for reversing the DC power to the LED.

Moving The Switch

Relocating the switch and the reset button to the front panel ends all the hopping up and down to reach the back of the Kaypro, and makes the wiring short and neat. The switch and reset button can be left in back, however, if you don't want to drill the front panel, or if you need the exercise.

How To Wire (Figure 1)

Remove the old power indicator LED by pulling off the collar inside. Then push the LED in from the front and cut the wires close to LED.

Solder some jumper wires on the new LED. Then install and connect it to the center studs of two of the poles on the new switch. Connect the 2 wires that previously went to the power LED, to the top studs of the same 2 poles.

Connect jumpers (making an X) between the top stud of each pole and the bottom stud of the opposite pole. Move the speed select wires to the new switch using the third pole set. Reverse LED wires and/or speed select wires as necessary to get the speed and color of your choice.

Results

Now at 2.5MHz, my power light glows red, just like a normal Kaypro, but when I switch to 5MHz the power light glows green. Plus the switch and reset button are right at my finger tips.

Tip

When drilling the front panel, stick a piece of tape on the inside of the new hole to catch drill chips. Silver duct tape works well. If you really want to keep things sanitary, turn on your vacuum cleaner and hold the hose up next to the drill bit.

Parts

Both the tri-color LED #276-035 and the 3PDT switch #275-661 are available at Radio Shack.

Out “keys” the competition!

Thinking about buying a “key” program? (You know, the names all start with words like “smart”, “magic”, “pro”, etc.) Looking for a faster, easier way to calc? Process words or databases? Yes? Then you owe it to yourself (and your computer) to check out XtraKey. Quite frankly, we think it’s the best. At any price! (And wait ‘till you see our price.)

Just like those other “keys”, XtraKey lets you redefine your regular keyboard keys to be anything you want. A word, a paragraph, a series of commands... whatever you hate to type over and over again! Change or make up new definitions anytime. Even while running a favorite program like WordStar or dBASE III! Unlike other “keys”, there’s no limit on definition length.* Plus our advanced XShift feature lets individual keys have up to 16 meanings.

XtraKey can also talk to your printer or video display. Change from pica to elite while working on a spreadsheet. Address an envelope while in a document. Or call up your own custom help or menu screens (almost like having windows!)

There’s more! Built-in screen dump**, keypad redefinition**, clear screen, printer on/off, definition chaining, program chaining, input pause and batch processing. All in one, neat little package that uses less memory AND disk space than the leading program.

Now, for $39.95, aren’t you ready for real key power?

* to available CP/M TPA ** XtraKey Custom versions for Kaypro (all models) & Gnat 10 only

XtraKey will run on CP/M 2.2 based computers. To order, send check or money order for $39.95 ($49.95 for XtraKey Custom for Kaypro or Gnat 10) plus $3.00 shipping (U.S. & Canada). California residents MUST add 6% sales tax. Specify make and model of computer and disk format. VISA and MasterCard accepted.

CP/M, dBASE II, WordStar and Kaypro are trademarks of Digital Research, Ashton-Tate, Micropro and Kaypro respectively.

Micro Cornucopia, Number 25, August-September 1985
The 32-Bit Super Chips

By Trevor Marshall, George Sclaro,
Dave Rand, Tom King, and Vince Williams

Most designs come about because of an unfilled need. That was certainly the case when we sat down to design a 32032 co-processor system for PCs and PC-clones. But first a little background.

During 1983 and 1984, a group of us at Definicon developed some advanced signal processing software.

One of our most successful algorithms used a time to frequency-domain conversion technique which required many floating point calculations, but it gave better resolution than a standard fast Fourier transform.

Benchmarks Vrs. Fortran

The program was written in FORTRAN and initially we ran it on a VAX 11-780 minicomputer, but we soon moved the program over to an HP9000. A typical analysis took about 15 minutes on the HP.

But the cost of the HP9000, about $60,000, seemed a bit high. After all, who wanted to pay that much just to run a piece of software? So during 1985 we converted our programs to run on IBM XT's. We ran a number of benchmarks on an Eagle Turbo (an XT clone) and on a real XT, and concluded that the Eagle was about twice as fast as the IBM. (The Eagle runs an 8086 processor and 8087 math co-processor, while the XT runs an 8088 and 8087.)

In fact, the benchmarks indicated that the Eagle's performance should nearly equal the HP9000's. But after we finished transferring over the FORTRAN source, we found that the Eagle took seven times as long as we had predicted.

I couldn't see any reason why a minicomputer like the HP9000 should run that much faster than a micro, especially considering the benchmark times. So I investigated! (Of course, a benchmark is just a small fragment of code, and it's obvious that benchmark results do not necessarily reflect performance on a larger, more complex program.)

Wherein Lies The Discrepancy

If we examine the data for the Sieve of Eratosthenes (the Prime Number Sieve) in Table 1, we see that for n8191 the difference in execution speed between the XT and the VAX11/780 is about 10 to 1. However, by increasing the size of the searched array to 40,000 numbers (rather than 8191), the difference widens to 55 to 1.

The program flow has not changed, but the larger size of the problem shows some problems with the Intel processors. (If we want to identify the primes in 80,000 numbers, then compilers running on the PC-XT or PC-AT won't work. They place a 64K limit on the number of elements of an array.)

Floating Point Benchmarks

We devised the FLT benchmark to exercise the floating point processors. The benchmark consists of a few arithmetic operations inside a loop. The FLOAT benchmark also exercises array handling.

The 8087 and 80287 arithmetic processors ran about 1/20 MFLOP (1 MFLOP = 1 million Floating Point Operations per second). In fact, the 4MHz 80287 in the PC-AT ran slightly slower than the 4.77MHz 8087 in the PC-XT!

The VAX11/780 clocked in at about 1 MFLOP, or 20 times faster. The major problem with the 8086/80286/80386 family is not shown in these benchmarks, however.

64K Addressing

If we had been able to take the sieve above 64K elements we would have found that the Intel processors slow down even farther. Whenever any data, code, or stack structure exceeds 64K, you must check the segment you're in before doing an access. Even business programs (such as 123 and dBASE) are slowed substantially by segment checking.

The 68000 and 32000 series processors are not affected by this overhead, since they can usually access many megabytes of address without segmentation.

Consequently, late in 1984, we reluctantly accepted that 8-bit (8088) or 16-bit (8086,80186,80286) processors had significant limitations. We also accepted the fact that 32-bit microprocessors have something significant to offer.

32-Bit World

There really is only one operational 32-bit microprocessor—the National Semiconductor NS32032. I will not argue this point, but merely state that the 80386 is currently just a pipe dream, and the bug lists I have seen for samples of the 68020 still cover several pages. At the time I wrote this article, even the Intel 80868 was not fully functional in all modes.

Bugs In The 68020 Set

1. The barrel shifter does not work.
2. The co-processor interface does not work.
3. The data sheet AC characteristics cannot be met.
4. In addition, the floating point co-processor works only at unacceptably slow clock speeds and does not yet talk properly to the CPU. (If ya' want a number, take a number, any number!)

Motorola has never demonstrated a complete 68020/68881 system (not even at the trade shows) and has continued to use the 12.5MHz 68020 demo I first saw at NCC in mid-1984. This demo, incidentally, gives vastly different benchmark times depending on whether you select Single Run or Continuous mode. Try it at the next trade show, and see if you can get an explanation for this irregularity!

In fairness, I have seen a 12.5MHz 68020 running non-virtual UNIX in 68000 emulation mode.

Motorola does not admit having bugs any longer. (The last official bug list was distributed in 1984.) Still, time will be the ultimate arbiter.

To put this all into perspective, National made its first NS32016 in September 1980. In December 1984, it was sufficiently operational to release to production.
ON the 32-bit processors, it’s easy to know which instructions allow which addressing mode. You can use any addressing mode with any instruction.

For example,

```
ADDD 100000, LAST
```

adds immediate 100,000 to the 32-bit memory location LAST.

```
ADDB 8, LAST
```

treats LAST as a byte memory location and adds 8 to it.

```
ADDF 1.31978, LAST
```

adds immediate the floating point number 1.31978 to LAST (treating it as a 4-byte floating point number).

```
DIVL FIRST, LAST
```

divides the two memory locations FIRST by LAST, treating them as double precision 8-byte floating point numbers, and places the result in LAST.

```
EXTS 5(81,80,4,11)
```

EXTracts the Short bit field from the Double precision 32-bit word at the address in R1 plus 6, puts it into R0 (right justified), starting at bit 4 for 11 bits.

Making Them Work

Two problems with producing these chips should be immediately apparent. First, it is not a trivial matter to get these instructions to work the first time. Second, it is quite difficult to adequately test all the instructions after the chip has been manufactured.

Conversely, these sophisticated chips make it much easier to write efficient software. In general, anything you want to do, they can do.

In addition, compilers can do much more optimization, and produce much more compact code.
Once we realized the need for a real 32-bit system and had checked out the options (32032, 68020, and 80386) -- see the previous article by Trevor, et. al., in this issue -- we started to design the system.

After pricing hardware for a complete standalone system, we decided to plug our 32-bit microcomputer (the DSI-32) into an IBM PC clone. Not only does the PC offer all the I/O peripherals we needed (we even have a 1/2" magnetic tape drive on one of our development machines), but it also has a processor which can handle files for the 32032.

In addition, MS-DOS (or PC-DOS) is not unlike CP/M. It has sufficient additional features (such as subdirectories and almost all of the enhancements given by programs such as ZCPR3) to make it useful for applications development.

We were also pleased with Concurrent DOS (from Digital Research). In addition to all the features of MS-DOS, it offers multiple consoles with up to three background tasks. This environment makes software development a pleasure!

The DSI-32 uses a 10MHz 32032 CPU running at about 1.5 MIPS (million instructions per second) with a 32081 Floating Point unit (about 1/3 MFLOPS) and optional 32082 memory management unit for a true virtual software interface.

Two 38.4 KBAud RS-232 serial ports handle high speed I/O totally independently of the PC. In addition, direct port I/O and memory access for the 8086 environment is provided to the 32032 through the software interface.

The host 8086 (or 8088) can access all the 16-Mbyte logical address space of the 32032 in addition to the added serial ports. Data is transferred from the DSI-32 using Direct Memory Access, so the 32032 usually only misses one or two 100 nsec T states for each byte transferred to the 8086.

Writing A Program

To write a program for the DSI-32, you first edit the source file (in C, Pascal, FORTRAN, FORTH, Tiny...
BASIC, or Assembler) using your favorite MS-DOS editor (we use WordStar and Vedit). Then you run the appropriate Definicon compiler, assembler, and linker to produce an executable 32032 program.

For example, let's develop a program called HELLO.C which says "Hello World." First we edit the source (in C) using WordStar and compile it using

```c
cc hello.c
```

assemble the resulting source code with

```as
as hello
```

and link the object modules with

```link
link ind=hello
```

Finally it is run with the MS-DOS command

```load
load hello
```

This invokes the 32000 loader, which reads the 32000 executable HELLO.E32 file off the disk, loads it into the DSI-32 memory, tells the 32000 to commence execution, and then stays resident to service I/O requests from the 32000 program. When execution is complete the loader exits and passes control back to DOS.

We took a shortcut with the linking. The command IND=HELLO means that the linker takes its constructional data from an indirect file called HELLO.IND. See Figure 1 for the file.

Because of the complex information needed by the linker, it is always easier to prepare this indirect file when you write the application, then invoke it automatically as you compile. This also allows a BATCH job to run the complete compilation, assembly, and linkage process automatically.

Note that 32000 code is totally position independent, and the linker does not perform relocation. It builds

(continued next page)
a series of MODULE TABLES (one for each code module) that contain data for the CPU's static base and control registers. These registers are loaded from the table each time the table's corresponding module is executed.

Definicon's Monitor

Finally, Definicon has a MONITOR program for debugging. It allows in-line disassembly (including full floating point operands), run to breakpoints, execution trace, display, fill and substitute memory, display of CPU control registers, display of CPU data registers, display of the Floating Point Unit's registers, and display of the Memory Management Unit's registers. Its commands are similar to DDT and DEBUG.

Availability

The DSI-32 is available both in kit form and fully assembled. Call for information about assembled and tested boards. The kits are a special deal to Micro C readers and are available from Definicon at the address shown at the top of this article.

Assembling The Kit

If you have never assembled a microcomputer kit, contact your user's group, or enlist an expert's help before you attempt construction. And be sure to observe static safety procedures when you handle ICs. Definicon assumes no responsibility for replacement of components which have been mishandled or incorrectly inserted into their sockets. Definicon's liability is limited solely to the replacement of those components which were faulty at the time of shipment.

Starter Kit

32032 CPU — 6MHz clock rate
32081 FPU — 6MHz clock rate
256K Bytes of RAM (32, 64K chips)
Wave soldered, partially tested, printed circuit board, fully socketed

Full set of ICs and instructions for assembly
Diagnostic software disk
Simplified NSX compatible assembler/linker/loader
MS-DOS interface software, advanced debug monitor
Public Domain software disk (supplied upon request)
Cost: $995.00

Advanced Kit

32032 CPU — Full 10MHz, no wait states
32081 FPU — Full 10MHz, no wait states
1 Megabyte of RAM (32, 256K chips)
Wave soldered, partially tested, printed circuit board, fully socketed
Full set of ICs and instructions for assembly
Diagnostic software disk
Simplified NSX compatible assembler/linker/loader
MS-DOS interface software, ad-
Advanced debug monitor

Public Domain software disk (supplied upon request)
Cost: $1495.00
(Note: The DSI-32 draws up to 15 watts (5V at 3 amps) from the PC's power supply, so make sure you have this much spare power capability before ordering.)

Public domain compilers/interpreters will be available for FORTH, Small C, Pascal, and Tiny BASIC. The disk containing these programs will be included free (but only if you ask for it).

The DSI-32 is suitable for use with any IBM PC clone (but your money back is the only guarantee of compatibility we can offer).

Commercial Software
2. Green Hills Pascal Compiler: Full Berkeley 4.2 UNIX compatible plus many extensions.
5. Definicon/Computer Systems Design NS32000 Library Manager and Programmer's utilities: LIB32 program to form and examine libraries of object modules, assembly and high level language examples for direct (OEM) interface to the Definicon MS/PC-DOS interface.
NOTE: The C and Pascal compilers will run in 256K in a limited way. The FORTRAN compiler will not run in 256K.

Prices of Commercial Software
Library manager/Programmer's utilities $49; Assembler/Linker purchased separately $149; One compiler (your choice), including assembler/linker $299; Two compilers including assembler/linker (one purchase) $649; Any compiler, purchased alone, (needs assembler above) $249.

Note: Green Hills Software is making these compilers available at prices well below those of the identical compilers for their original UNIX environment. Prices in the above table are specially discounted for Micro C readers.

A hard disk is almost essential if you want to run the Green Hills compilers (which range up to 250K of code).

The only support that Definicon can offer to purchasers of this software is a guarantee to check out written bug reports promptly. We assume that Micro C readers are proficient in the basic programming syntax of a language, and the documentation provided reflects this assumption.

Figure 1 - File HELLO.IND

```
EXEC=HELLO ;Call the executable program 'HELLO'
MODULE=CLIB/START=2000,NOCOPY ;Use static base for CLIB
CODE=4000 ;Program load address
RAM=0..FFFFF ;We have 1 megabyte of RAM to use up
STACK=+4000 ;Allocate 16K of stack somewhere
MOD=80 ;Set the module table at 80H
[[ DT ]]
```

GENERAL UTILITIES PACKAGE

Rolland Management Systems Utilities Package offers maximum access to files for CP/M® users

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort System</td>
<td>Organize and reorganize any file</td>
<td>$39.95</td>
</tr>
<tr>
<td></td>
<td>* fixed or variable length records</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* unrestricted sort keys</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* skip or include records or blocks</td>
<td></td>
</tr>
<tr>
<td>Menu System</td>
<td>No need to &quot;chain&quot; back to menu</td>
<td>$29.95</td>
</tr>
<tr>
<td></td>
<td>* unrestricted construction of menus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* restrict user modification of menus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* enhanced batch processor under the menu</td>
<td></td>
</tr>
<tr>
<td>Batch List Utility</td>
<td>List many files with one command</td>
<td>$19.95</td>
</tr>
<tr>
<td></td>
<td>* optional page heading, page number or date</td>
<td></td>
</tr>
<tr>
<td>Dump Utility</td>
<td>Decode and display entire file contents</td>
<td>$19.95</td>
</tr>
<tr>
<td></td>
<td>* show hexadecimal, integer or RAD50 value</td>
<td></td>
</tr>
<tr>
<td>Mastercard or Visa call</td>
<td>(919)276-6070</td>
<td></td>
</tr>
</tbody>
</table>

Rolland Management Systems, Inc.
Rt 5, Box 135
Laurinburg, N.C. 28352

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

Save hours of time on your Kaypro with this powerful utility!

- Format, copy and verify 96 tpi disks in under 4 minutes (exact speed depends on type of drives and year of Kaypro)
- Format, copy and verify Kaypro double and single-sided disks in approximately 1 minute and 2 minutes respectively
- Format, copy and verify any soft-sector, single or double-sided 5¼" disk format on your Kaypro at comparable speeds (e.g. Osborne SSD in under a minute). Large library of formats supplied, with capability of adding others.
- Use any combination of Drives A, B, C or D as source or destination drives

(Not currently available for the Kaypro 10)

Please send MULTICOPY to:
Name ____________________________
Address ____________________________

Enclosed is $39 plus $3 shipping
(California residents add 6% sales tax)
Check or money order □ VISA □ MasterCard □
Number ____________________________ Exp. __________
Signature ____________________________

($5 shipping charge for orders outside N. America)

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

Plu'Perfect Systems
BOX 1494 • IDYLWILD, CA 92349 • (714) 659-4432

IF YOU HAVE TWO OR MORE KAYPROS*
THEY SHOULD TALK!
(* & MORROWs OR XEROX 820-1ts ALL IN THE SAME OFFICE)

CONNECT THEM WITH THE LAN/ROVER™
The ADEVCO LAN/Rover is a full-featured Local Area Network system which enables Kaypro CP/M computers to work together. Information from disk drives (including hard disks) can be exchanged with other computers (as many as 60 computers on a network.)

- TRANSFER FILES TO OR FROM REMOTE DISK DRIVES AND USE OTHER PRINTERS OVER 3,000 FEET AWAY.
- FULL SECURITY AVAILABLE WITH ENCODED PASSWORDS AND USER PRIVILEGES.
- FIELD-PROVEN TECHNOLOGY IN OVER 1000 INSTALLATIONS.

Each system includes hardware, software, cabling, full documentation and network management tips. The LAN/Rover provides a low cost, finely engineered and reliable network for your KAYPRO and other computers. Phone or write for details.

ADEVCO, INC.
3790 El Camino Real. Palo Alto, CA 94306, 415/493-7466
Kaypro, MasterCard, & Xerox are registered trademarks of the ADEVCO Corp., MORROW DESIGN & XEROX.

CP/M EPROM PROGRAMMING SYSTEM

2708  2516
2758  2732A
2716  2712B
2732  27256
2764  27CXX

- STAND ALONE BOARD - ELECTRONIC SWITCHING OF EPROM TYPES
- USES 48 VOLT XPMR FOR POWER - ALL SUPPLIES/TIMING ON BOARD
- DESIGNED WITH EASY TO GET PARTS - COMPREHENSIVE 36 PG MANUAL
- WORKS WITH ANY CP/M SYSTEM - SOFTWARE INSTALL UTILITY
- * INTERFACE TWO WAYS *
- 1 - ONE 8 BIT INPUT PORT AND ONE 8 BIT OUTPUT PORT - 16 WIRBS AND A GROUND - NO HANDSHAKE LINES REQUIRED
- 2 - CENTRONICS PRINTER PORT - 8 OUTPUT DATA BITS AND ONE INPUT DATA BIT (BUSY LINE) - 9 WIRBS AND A GROUND
- ** CONTROL PROGRAM COMMANDS **

- PROGRAM EEPROM(S) FROM DISK
- READ DISK FILE INTO RAM
- VERIFY EPROM IS BRASURED
- DISPLAY/CHANGE RAM
- MONITOR MODE WITH 16 SUB COMMANDS
- FULL-DUMP-XFER-EXAMINE-MODIFY-BAD-PROGRAM-VERIFY, ETC.

BAR'S P.C. BOARD WITH COMPLETE DOCUMENTATION
AND SOFTWARE ON 8 SINGLE DENSITY DISKETS

$69

(ABOVE WITH COMPLETE PARTS KIT - $169 & 4 T - $189)
SOFTWARE AVAILABLE FOR OSBORNE, KAYPRO AND OTHER 3 1/4 FM'S.

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

CALL OR WRITE FOR MORE INFORMATION - ADD $3.00 FOR SHIPPING
OHIO RES ADD 5.5% TAX - VISA/MC ACCEPTED - $0.00 FOR COD

56
FINALLY, THE LIGHT DAWNS FOR CP/M USERS.

With the MRS/OS Source Code, you can see the light.

If you own a CP/M compatible operating system, you've had to put up with the mistakes and quirks of someone else's programming. Until now. Now you can see the light with MRS/OS. In fact, MRS is a full operating system designed to replace CP/M 2.2 or CDOS and it comes with complete source code. MRS is designed for Z80 processors, runs CP/M software, and can interface directly to a CP/M BIOS, saving you a lot of sysgen time.

With MRS, you get more than what you pay for. For under forty dollars you receive fully commented source code for standard and extended BDOS functions, a sample BIOS, our all-in-one utility package and a 130 page manual.

So if you're tired of being in the dark with some other guy's program, here's the answer to your prayers.

$39.95 (includes shipping & handling in N. America; overseas add $12) Mass. orders include 5% sales tax

Order by phone 24 hours a day!
(617) 478-3102

Tech. inquiries: 7:30pm-10:30pm EST (Wed. ONLY)
(Tech. line closed August 7 & 14)

SPECIAL FREE OFFER TO 8" DISK USERS!

ACT NOW and receive our ultra-fast, flexible disk format program, FREE! FORMAT can format a 77 track 8" disk in just 25 seconds and provides you with four of the most common disk formats used with CP/M based systems:

- 8" SSSD 128 byte sectors, 26 sectors/track, 76 tracks
- 8" SSDD 256 byte sectors, 26 sectors/track, 76 tracks
- 5.25" SSDD 256 byte sectors, 18 sectors/track, 40 tracks
- 5.25" SSDD 512 byte sectors, 10 sectors/track, 40 tracks

YES! Please send me ____ MRS/OS Source Code(s) for just $39.95 each on □ 8" SSSD disk or □ 5¼" Kaypro format disk. If I act right now, I'll receive a FREE FORMAT program included on the disk (8" disk users ONLY).

□ VISA/Mastercard # _______ exp. date ________ (circle one)
□ Check
□ Please send more information

Name ____________________________
Address __________________________
City ______ State ______ Zip ______
Phone ____________________________

Send to: OCCO, Inc.
28 Clifton St., Milford, MA 01757

CP/M is a registered trademark of Digital Research Corp.
CDOS is a registered trademark of Comemco Corp.
It looks like National Semiconductor has a winner this time, so we’ll skip the jokes. (Jokes have been a National Pastime for years.) The 32032 has eight 32-bit general purpose registers and eight dedicated registers. The contents of any general purpose register can be used as base pointer, index value, or data. Data can be either 8-, 16-, or 32-bit. Take a look at Figure 1 for details on the dedicated registers.

Integer & Floating Point Instructions

The 32032 instruction set is comprehensive enough to keep even Niklaus Wirth busy. Integer instructions operate on 8-, 16-, and 32-bit elements —

- Add(ADD)
- Subtract(SUB)
- Negate(NEG)
- Absolute Value(ABS)
- Multiply(MUL)
- Divide(DIV)
- Modulus(MOD)
- Quotient(QUD)
- Remainder(REM)
- Move(NOV)
- Compare(CMP).

by adding a length suffix: B, W, or D to the instruction —

- ADDB, ADDW, ADDD
- MOVEB, MOVEW, MOVD

The complete syntax for an ADD is —

- ADDi source, destination

"i" represents integer length character.

Floating point arithmetic can be either single or double precision by adding a suffix: "f" (float) or "l" (long float) to the general form of the instruction. For example, a Negate instruction would have the syntax —

- NEGF source, destination

The floating point instructions are —

- Add(ADD)
- Subtract(SUB)
- Multiply(MUL)
- Divide(DIV)
- Negate(NEG)
- Absolute Value(ABS)
- Compare(CMP)
- Move(NOV)

Floating point instructions are —

- Addf(ADD)
- Subtracf(SUB)
- Multiplyf(MUL)
- Dividef(DIV)
- Negaf(NEG)
- Absolutef(ABS)
- Compareref(CMP)
- Movef(NOV)


either single or double precision by

The bit field instructions are —

- Extract Field(EXT)
- Insert Field(INS)
- Insert Field Short(INTS)

The string instructions are —

- Move String(NVS)
- Compare String(CMPS)
- Skip String(SEPS)

Array instructions are —

- Bounds Check(CHECK)
- Calculate Index(INDEX)

Logical, Bit, Bit Field, and Array instructions all accept the same length suffixes. Add B, W, or D depending on whether you're accessing a byte, word, or double word.

Processor Control And Service

The remainder of the 32032 instruction set is made up of branches, local procedure calls/returns, external procedure calls/returns, explicit trap instructions, trap/interrupt returns, effective address, context instructions, register/stack manipulation, and memory management. We won't list all of them; instead, we’ll try to give you the flavor of a couple of the more interesting. For example, let's say you're calling a procedure in Pascal.

Programmeng The 32032: Setting Out

By Gary Entsminger and Bruce Berryhill Micro C Staff

Micro Cornucopia, Number 25, August-September 1985
Four instructions handle subroutine calls and returns (hope they’re happy ones). They are BSR, RETURN, ENTER, and EXIT.

BSR and RETURN handle the simple minded subroutine call where you push onto the stack the parameters you’re passing and then branch (BSR) to the subroutine. The RETURN instruction discards the passed parameters from the top of the stack when you’re done.

ENTER and EXIT are additional instructions you use when entering and exiting a subroutine. ENTER allocates space on the stack for the subroutine’s local variables and saves any or all of the values in the processor’s registers. EXIT does the reverse, restoring the registers and discarding the local variable space.

Frame Pointing

A frame is a region of the stack which contains the parameters passed to the procedure, the procedure’s local variables, and its saved registers. The Frame Pointer (a dedicated register) points just below the passed parameters to the first local variable (See Figure 2). The Stack Pointer points just beyond the end of the frame.

Invoking A Procedure

The three instructions beginning with MOV D(-8)(FP),TOS make up the procedure calling sequence (see Figure 3).

A procedure’s parameters are accessed by adding a positive displacement to the address in the Frame Pointer. Local variables are accessed with the same addressing mode but use negative displacements.

In our example, we’re pushing copies of the local variables (a and b) onto the stack. These will become parameters when the called procedure executes. See Figure 2.

External Procedures

The 32032 supports program modules (one of its selling points) which are accessed via a module table. External procedures are procedures outside the current module. To call one —

CIP ; call external procedure
or, CIPD ; call external procedure with descriptor

Then, RBF ; return from external procedure

(continued next page)
Multi-tasking

In multi-tasking operations the efficient use of memory requires that a procedure be locatable anywhere. (Unlike single-user systems, there is no guarantee that a specific location like 100H will be available.) This use of “virtual memory” frees the programmer from the restraints of physical memory. You can think linearly even though memory is actually being partitioned. (Partitioning is used to divide up memory so multiple processes can reside there at the same time. It is not 8086-like segmenting.)

For example, let’s say you’ve got 1000 bytes of free memory (let’s say addresses 1000-2000), but your program is larger than 1000 bytes. You start out by loading 1000 bytes of your program into memory and running it.

A JUMP to an address not currently in memory generates a page fault. (A page fault means we need more of the program.) So a JUMP to program address 1500 (would be to memory address 2500 since we’re starting from 1000) forces the system to go to the disk and overlay the next 1000 bytes of the program into the 1000-2000 portion of memory. Then execution continues.

Supervisors And Users

The 32032 was designed to be a 32-bit COMPUTING SYSTEM. All system-level features are designed in. Currently the MMU (Memory Management Unit) and the FPU (Floating Point Unit) are separate chips, but National Semiconductor intends one day to have the entire system on one super chip.

Definicon’s 32032-based board (DSI-32) is current microcomputer state-of-the-art (or thereabouts), and plugs into a PC clone using the PC’s microprocessor (8088 or 8086) for I/O.

By using Concurrent DOS (which allows background tasking using MS-DOS software), you can edit a file with the 8088 for instance, while the DSI-32 is executing another program. A lot of MS-DOS software will run under Concurrent DOS, so this really opens up the possibilities.

Supervisor requests allow the 32032 and the 8088 (8086) to talk to each other.
If you need to write a character, open a file, or write to a file, for examples, send the 8088 the appropriate supervisor request by loading the request number and other particulars into 2 or 3 registers.

Finally

The 32032 has some powerful features, so don't be surprised if the chips are very popular, if the chips are infiltrating PCs world-wide.

Co-processors that can communicate with MS-DOS or UNIX will be very popular, if the chips are available. Insiders claim National Semiconductor finally has it together and can get the chips out (and we're not talking potatoes). Let's cross our fingers.

---

**$29.95 SpellSys**

Are you signing your name with an X because spelling doesn't come easily?

Then you need SpellSys!

With this full-feature package, you can write prose with the pros. SpellSys features a 42,000 word dictionary and all the bells and whistles of those expensive checkers—including rhyming, crossword search, letter unscrambling, etc.

SpellSys is made up of a group of individual programs which you can use together or separately. With SpellSys you can setup and maintain your own custom dictionary (in addition to the main dictionary). These are real dictionaries, not hash tables, so you edit or remove words from your own dictionary at will.

---

**1000K RAMdisk Plus**

FOR BB-I and XEROX 820: 2-piece system using 2-80 daughter board + 8.5'' x 13.75'' expansion board for on-the-bus memory and I/O at 2.5 or 4 + MHz.

**BANK-SWITCHED MEMORY:** Add 4 rows of 64K or 256K dynamic RAMs + 32K (four 8K x 8) EPROMs or CMOS battery-backed static RAMs. FAST "HOLE" memory mapping. High-performance "delay-line" memory timing.

**I/O PORTS:** 1 SIO, 1 CTC, 2 PIO, MM58167 real-time clock, ADC0808 8-bit, 8-channel ADC, two DAC0800 8-bit DACs, AY-3-8910 sound chip.

**EASY TO BUILD AND INSTALL:** Easy to get parts. Complete documentation + full schematics are included. Piggyback or side-by-side mounting; no modifications needed for BB-I or 620 installation. Versatile connection scheme allows for further expansion.

**SOFTWARE INCLUDED:** RAMdisk software, 250K and 436K RAMdisks for SWP DD & standard CP/M also patches in the real-time clock. Utility programs included.

**OPTIONAL SOFTWARE:** [1] Upgraded RAMdisk driver supporting 244K through 1000K drives, with built-in ZCPR 1.1 and keyboard translator that uses static RAMs for alternate character or string storage. RAMdisk is drive A: Uses SWP DD or standard CP/M. [2] Public domain computer music play/compose program uses DAC for output; great sound quality and fun to use! Includes printed installation guide.

- Bare daughter and main boards (silk-screened and solder-masked); full documentation, 8" SSSD software disk
- Delay line (required for RAMdisk)
- New RAMdisk software with ZCPR + keyboard translator
- Computer music program: program + songs (2 disks)

Shipping add $5.00 (U.S.A.), $8.00 (Canada); no other exports. Check or MO only; no CODs. CA residents add 6% tax. Delivery: stock to 8 weeks. NOTE: All proprietary software comes with fully commented source code. Please enquire about compatibility with BB-I and other computers. Software furnished on 8" SSSD; 5¼" (Xerox) for extra charge. User's group discounts available.

See review in MicroCornucopia Issue #22 (Feb 1985).

---

**MicroCornucopia**

Micro Cornucopia, Number 25, August-September 1985
**NEW**

AMPRO "Little Board" MAIN/FRAME

- Assembled & Tested
- Heavy Duty All Metal Cabinet
- Fan & Dust Filter
- Heavy +5 & +12 Regulated Supply
  Runs Floppy Drives & Computer Boards
- Accepts 5½ Drives, Full or Half
- 2 ea DB25, 1 ea Centronics Connector Cutouts
- Front Panel Reset & Power Switch
- Power Harness from Supply to Drives & Board
- Line Fuse, EMI Filter, Detachable Line Cord
- I/O & Drive Data Cables Available Separately
- Cabinet & Power Supply Available Separately

AMPRO & Little Board are TM AMPRO computers.

$150 (1 Piece) *

2800 5¼ Drive & Little Board Enclosure
(includes power supply & fan)
* Call for quantity pricing
(Disk Drives and Little Board not included)

---

Single Board Computer Chassis & Power Supply

- Assembled and Tested
- Heavy Duty All Metal Cabinet
- Fan & Dust Filter
- Heavy +5, +12, +24 Regulated Supply, Runs drives & computer board
- Accepts 2 ea 8" Floppy Drives, Half of Full
- 2 ea DB25, 1 ea Centronics, 1 ea 50 pin cutout on back
- Front panel reset switch
- Power Harness for Drives and Board
- Flexible Board Mounting
- Line Fuse, Filter, Line Cord, Power Switch
- Cabinet & Power Supply Available Separately

$275 (1 Piece) *

2901 SBC Chassis & Power Supply
* Call for quantity pricing.
Disk Drives not included.

---

Write or call for our brochure which includes our application note:
"Making micros, better than any of box computer."

INTEGRAND RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291  209/651-1203
We accept BankAmericard/Visa and MasterCard

---

Micro Cornucopia, Number 25, August-September 1985
A FORTH Engine can be loosely defined as any single board computer that has FORTH built into it. There are three main approaches to building a FORTH Engine: FORTH in ROM, Microprogrammed FORTH, and the FORTH Processor. I’ll cover all three in this column.

FORTH In ROM

A general purpose microprocessor such as the Z80 can be used in conjunction with ROM memory to emulate a FORTH computer. For example, the ROM FORTH or Idaho FORTH ROM sets that were sold by Micro C enabled the Big Board I to execute FORTH upon reset. This ROM technique is not limited to FORTH. The IBM PC, Commodore 64, and other personal computers have BASIC in ROM. Even monitor ROMs fit into this category, since they accept keyboard commands.

The primary advantage of FORTH in ROM is the ease of implementation, since any general-purpose micro can be programmed to execute FORTH. But in all cases, the ROM eats into the available program space. Since few micros have the register set or addressing modes for efficient FORTH, ROM implementations tend to be slower than the other two methods.

Microprogrammed FORTH

Think of a microprogrammable processor as one with a limited internal instruction set that uses short subroutines to handle the more complex instructions. In effect, such a processor is both a processor and a ROM as described above. The microcode resides on a separate internal bus and controls the processor. In fact, from the outside you only see the microcode, not the processor.

The 68000, 8086, 80186, and 80286 are examples of microcoded processors. Each clock cycle of the 68000, for instance, consists of several internal microcycles.

Most 16-bit micros use microcode since it’s easier to develop microcode than to do the same complex instructions in hardware. The difference between the 8086 and the 80286 is that more functions on the 80286 are implemented in hardware, using fewer microcycles for each instruction.

You can buy microprogrammable processors, normally called bit-slice processors. Given that such CPUs can emulate any instruction set, it makes sense to microcode FORTH. The H4TH engine is a good example, where a 4-bit processor executes FORTH instructions. A 16-bit addition needs at least four microcycles since only 4 bits of the result can be obtained at once. For this reason, the H4TH is not the fastest FORTH engine even at its internal 14MHz clock rate. A 16-bit chip such as the AMD29100, or a 32-bit chip such as the NCR/32 might be better choices.

Here again, FORTH is not the only language that can be implemented in microcode. The Pascal engine of the late 1970s and the Modula II engine are other examples. The LSI-11 and MicroVAX are microcoded processors.

The FORTH Processor

Even the bit-slice processors use an underlying general purpose chip to emulate FORTH. The best design would be a custom chip with special circuitry to implement FORTH instructions directly. NEXT best would be a single instruction instead of several microcycles. Obviously, this is the fastest and most efficient method, requiring only a single chip with no emulation. Unfortunately, it is also the hardest approach. Someone has to design such a chip and convince a semiconductor company to invest huge sums of money to create the mask and produce the chip.

Charles Moore (the inventor of FORTH) started such a project several years ago, and has recently succeeded. The March 21, 1985 issue of Electronics Design introduced the Novix microprocessor — a single chip FORTH engine. The first chip to be released, the NC4000A, will run at a clock rate of 8MHz and is capable of executing the Sieve benchmark in 0.4 seconds, about the speed of optimized 68000 assembly language. A 16-bit processor, the NC4000A can use 64Kb of memory for program space and can store data in a 128Kb region.

The Novix chip is the most novel micro to be released in years. Its speed is such that dedicated controllers can be programmed in a high-level language instead of assembly language. Look for the Novix in many products in the future.

Comparing The Three

Each of the three approaches listed above has strong points. While the slowest and most memory wasteful, FORTH in ROM can use available co-processors such as the 8087, as well as family members such as the 280 DMA.

The microprogrammed FORTH uses conventional techniques, with arithmetic accelerator chips available to speed operations. For example, the NCR/32 has an accelerator that can perform floating-point operations in two clock cycles.

The FORTH processor has the most potential in speed and compactness, but may be more expensive, does not have co-processors, and has limited memory space.

Figure 1 shows benchmarks for the three reviewed systems as well as two common computers. Read my earlier columns on benchmarking for information about the individual tests. The H4TH compares favorably with the VAX, though the latter is a 32-bit implementation.

For further FORTH engine references, see The Journal of FORTH Application and Research (v.2 n.1 1984) and A Bibliography of FORTH References (1984), both available from the Institute for Applied FORTH Research, Inc., 70 Elmwood Ave., Rochester, NY 14611.

Of all the high-level languages currently in vogue, FORTH appears to be the one targeted towards hardware implementation. It behooves you, therefore, to get familiar with the language before your boss tosses a...
chip on your desk with the comment, “Have this project ready for me tomorrow.”

The Rockwell Chip

Rockwell introduced the 65F11 microcontroller chip in 1983. The chip includes a serial port (up to 7200 baud at 1MHz clock and 9600 baud for the 2MHz chip), two 8-bit parallel ports, counter/timers, 3K bytes of masked ROM, and 192 bytes of RAM. It can address up to 16K bytes of external memory directly, or one of the parallel ports can be used to bank select into a much larger memory space.

The unique feature of this controller chip is the tiny FORTH that resides in the ROM. Its 133 words can be used as-is, with a primitive micromonitor or “outer interpreter” that understands numbers and can execute words. The 65F12 is functionally identical, with three additional 8-bit parallel ports.

You can implement a working 65F11 system with just the chip, a TTL latch, and up to 1K of EPROM. However, the more normal development configuration includes Rockwell’s 8K byte development ROM, providing 200 additional words includ-

<table>
<thead>
<tr>
<th>Figure 1 - Benchmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
</tr>
<tr>
<td>T1 (loop)</td>
</tr>
<tr>
<td>T2 (II)</td>
</tr>
<tr>
<td>T3 (I+)</td>
</tr>
<tr>
<td>T4 (E+)</td>
</tr>
<tr>
<td>T5 (C+)</td>
</tr>
<tr>
<td>T6 (D+)</td>
</tr>
<tr>
<td>T7 (fI+,fix)</td>
</tr>
<tr>
<td>T8 (F)</td>
</tr>
<tr>
<td>T9 (F/)</td>
</tr>
<tr>
<td>T10 (F+)</td>
</tr>
<tr>
<td>T12 (sqr)</td>
</tr>
<tr>
<td>T13 (log)</td>
</tr>
<tr>
<td>T14 (exp)</td>
</tr>
<tr>
<td>T15 (atan)</td>
</tr>
<tr>
<td>T16 (sin)</td>
</tr>
<tr>
<td>T18 (move)</td>
</tr>
<tr>
<td>Sieve</td>
</tr>
</tbody>
</table>

FORTHwords
(continued from page 63)
ing a line editor, an assembler, a complete Fig-compatible FORTH system, and EPROM programming capability. The CPU/ROM set is easily interfaced to a 1793 floppy disk controller for mass storage.

NMI has incorporated the 65F11 into a tiny (100 millimeter square) board of impressive power. In addition to the 65F11 and the development ROM, there are three RAM/ROM sockets, RS-232 interface circuitry, and parallel port buffers. Only a single 5V power supply is required. One of the sockets can program 2764s directly. The documentation includes the 200+ page Rockwell FORTH manual, data sheets, and a 33-page hardware manual.

Rockwell has provided a useful set of words in the development ROM. You can target compile: headers can be compiled separately from word bodies. Only the bodies need be burned for dedicated applications. Autolexecution of a word during startup is possible by placing an A55A (hex) pattern on a 1K boundary; the startup software then assumes the code field address of a word is in the next two bytes and executes that word directly. The ROM has all the normal disk commands as well as the ability to format a disk.

Both low- and high-level interrupts are supported. A page zero location contains an interrupt status word. One bit is used for enable/disable, and 1 bit indicates a pending interrupt. The actual interrupt is processed by a low-level word which sets the pending bit. When NEXT is executed, it checks the request bit and vectors to a user-defined high-level processing routine. Any one of 16 different interrupt sources can be serviced, including serial in/out, counter/timers, etc.

The memory sockets are mirrored so they appear in more than one place in the memory space. This permits developing code at one location, burning the PROM, and then relocating it at a different address either by physically moving the EPROM or by changing a jumper.

How The System Rates
I like the BASIC system. In a small space you have enough interface to the outside world to perform many dedicated control tasks. The built-in kernel provides run-time support, so you can pack considerable code into the external memory. Therefore, as a controller the card has potential.

As a development system, however, it is greatly lacking. It doesn't have a good line editor, much less a video editor. You would have to purchase the $340 development board to have

(continued on page 67)
At last! here is a chance to purchase a hard disk for your system at an affordable price. We recently participated in a buyout of Shugart’s finished goods inventory of these drives. Brand new in the original factory packaging, and guaranteed by us for 60 days. These 5 megabyte drives are considered by some to be the most reliable in the industry. These units directly replace a ST506 or similar drive.

When installing your system’s first hard drive, a hard disk controller card and appropriate software are required. For an IBM-PC with a 60 or 60W power supply, you will need a 130W supply. (SEE BELOW)

Performance Specifications

Formatted (33 sectors track)
Performance Specifications

Per Drive 5.40 Megabytes
Per Track 1.25 Megabytes
Per Sector 2860 bytes
Cylinders 160
Tracks 640
Index 1

Transfer Rate 5.0 Mbits sec
DC Voltage Requirements: +12Vdc ± 5% 1.6 A typical
Average 99 msec
Maximun 216 msec
Average Latency 8.33

Start Up Time (Tvp) 12 sec

ASK about quantity pricing.

**DISK DRIVES:**

**NEW**

- TEAC SD510 Half height 10 megabyte requires 130W power supply for IBM PC $40.00
- Microsine H612 1/2 ht. 10 megabyte $465.00
- Microsine H725 1/2 ht. 20 megabyte $955.00
- Tulln TD200 20 megabyte $630.00
- SUPERS Tandon TM 55-2, 5.2”, double sided, 48TP40 track, 6 minsec. track to track* $90.00
- SUPERS Shugart SA455, 5.5”, half height, 48TP4, 40 track, 6 minsec. track to track/dbl.sided $90.00
- NEW Remex 5.5” height with full height face plate, 48TP1, 40 track, doublesided. May be used in place of TM 100-2* $80.00
- NEW Shugart SA455 / Phoenix JA511/2N 5.5”, half height, 48TP1, 40 track, 6 minsec., doublesided. **$124.00**

**POWER SUPPLIES:**

- 130W power supply, IBM PC compatible $140.00...

I.C.’s:

- Available for immediate delivery –
  - 76151 1024 x 8 EPROM 4M only 2.50
  - 8427 64K RAM 2.50
  - 8425 16K RAM 2.50
  - 8196 32K RAM 5.00
  - 8232 64K RAM 5.00
  - 8327 256K RAM 10.00
  - 8242 512K RAM 15.00
  - 8256 1M RAM 25.00
  - 2114 8086 Assembly 25.00
  - 2115 8088 Assembly 30.00
  - 2125 80386 Assembly 30.00
  - 2126 80486 Assembly 40.00
  - 2127 80487 Assembly 40.00
  - 2128 80488 Assembly 40.00

**MISC. ITEMS**

- TELEVISION 806 Boards - for Startype Television Network or stand alone CP/M system. As is $25.00
- Working without LS1 parts $125.00
- Working with all parts $295.00
- Users manual $10.00 Service manual $35.00

- MOTORS:
  - TRW 403A117-3, 24 V, 5600 RPM $7.50
  - Canon EN3S-T1012A, 12VDC, 3400 RPM, 12 g/cm, 110 ma $1.00

- STEPPER MOTORS:
  - Copier Electric SH-65, 40 60, 12 V, 1.8° $2.00
  - Head positioning motors for Shugart 8" Drives $7.50

- AC ADAPTORs:
  - Viewisnics VSDAP-20, 9 VAC, 150 ma $2.00
  - Basler Electric BE2420, 24V, 20A $2.00

- LCD DISPLAYS:
  - Toshiba LT8026-35, 16 Line X 46 Column $7.50

- DOCUMENT CARRIERS:
  - 2-pocket leatherette 8 1/2" x11 $1.95

- SPEAKERS:
  - Matsushita ES45155A, 8n, 1-5/8" 15/4.100

- TERMINALS, Wyse 100 $195.00

- MONITORS, Osborne 5" Green Screen $35.00

- CAPACITORS, 1.0 muF monolthic caps. 100/° 8.00
  - 1,000/$60.00

- A.C. ADAPTORS:
  - 9VAC 150 ma $2.00
  - 24VAC 800 ma $2.00

- 4" SOLAR CELLS, 1.5 Amp minimum at .45V Closed Circuit $3.95

- MAGICUL PARTS:
  - Upgrade of B line to C line unit $200.00
  - 32K RAM Capsules 30.00
  - 64K RAM Capsules (no plastic) 75.00
  - Mostek T1 1.00

- Call for availability of 3/8” and 5/8” drive expansion units.

---

**DATASPEC "XP Series" DATA SWITCHES**

- 25 PIN RS232
  - All pins switchable. Can be used with IBM PC parallel port.
  - DB 25-S(female) connectors with gold plated contacts
  - AB-25 $57.00 ABC-25 $93.75

- 36 PIN CENTRONICS PARALLEL
  - All 36 pins switchable
  - Female centronics connectors with gold plated contacts
  - AB-36 $95.00 ABC-36 $142.50

---

**COMPUTER COMPUTER COMPUTER COMPUTER COMPUTER COMPUTER COMPUTER COMPUTER COMPUTER**

**Typical Applications**

**PRINTED**

**MODEM**

**PRINTED**
disk access. The 8K development ROM leaves only half of the 16K byte external memory for your program. The 65F11 is BASICally a 6502 processor, and at 1MHz it's quite slow.

I'd suggest using a host computer to perform development and editing, and then downloading to the 65F11 to test. Note that the built-in micromonitor provides debug capability at the customer's site with only the addition of a serial terminal. Without case, ROM, or power supply the system can be purchased in quantity for $150 or less, providing high-level control for a custom project at low cost.

'100 squared'
New Micros, Inc.
808 Dalworth
Grand Prairie TX 75051
214-642-5494

Bare board: $38
BASIC system: $250 (65F11) $290 (65F12)
Development system: $340 (65F11)

The Vesta SBC-88
The SBC-88 is an example of a small single-board computer that has sufficient EPROM space to incorporate an extensive FORTH, BASIC, or other ROM-based high-level language.

Its 5.5x6.5" footprint includes: a 4MHz 8088, 8 or 16Kb ROM, four additional memory sockets for up to 32Kb RAM-ROM, an RS-232 serial port at up to 9600 baud, two 8-bit parallel ports, seven individually addressable inputs and seven outputs, and an 8-channel A/D (8 or 10-bit). One of the RAM sockets includes provision for programming. The sockets support 2Kx8 or 8Kx8 RAM/EPROM devices. The A/D is a National ADC0804, and has a maximum conversion speed of about 200 microseconds. The optional power supply includes a reset button, EPROM programming voltage, BSR transmitter, and a 120Hz clock signal.

The board comes with 10 pages of hardware description and a schematic. I really like the BASIC board, and think it's one of the best bargains in the controller market. However, I have several complaints about the hardware.

The I/O and serial cables use DIP connectors instead of the usual dual in-line headers. In addition, the DIP connectors aren't on the edge of the board, but instead form the second row of sockets. The EPROM socket isn't Zero Insertion Force (ZIF). Power is always present at the programming socket, so you can't copy an EPROM to RAM, remove it, and program a duplicate (you usually destroy the program). Ground is not available on the parallel port DIPs. Instead, you must run a separate lead (why use a socket at all if this kind of kludge is necessary?).

Two different ROM operating systems were available at the time I wrote this: an 8K BASIC and an 8K FORTH.

BASIC Vrs. FORTH
Vesta has spent more time on the BASIC ROM. It includes a primitive download capability as well as Intel Hex file support, printer toggle, timer wait, autostart, line editor, EPROM programming, period measurement from an input port transition, and buffered A/D collection. The A/D data can be plotted using crude character graphics on the console or printer. The documentation is 32 pages of function description plus a partial listing of the BIOS.

The FORTH ROM is quite primitive, supporting the Fig model with a full line editor, EPROM programming capability, and A/D input words. Its documentation consists of the Fig-FORTH vocabulary list, a few pages of supplemental words, and a partial listing of the BIOS.

We used the FORTH to develop an automatic power line monitor for a customer and were frustrated at every turn. There are several bugs, such as the lack of printer toggle, ordering of timer words not following their double-precision integer convention, incorrect rotation of the eight 128-byte block buffers, and documented words that were missing from the ROM. The Fig line editor is clumsy when you are used to a video editor. The 'C' command will insert unwanted nulls into blocks.

After a few weeks, we quit using the SBC88 as a development board. Instead, the source code was developed, edited, and tested on a host computer before downloading to the SBC88. This method worked well enough to get the project out the door. The serial line is preset at 4800 baud at power-up, but when using FORTH we had to slow the baud rate down to 1200 to prevent character loss.

Deciding
If I had to choose between the BASIC and FORTH ROMs I would take the BASIC because of the additional functions. Its major deficiency is the usual slowness of BASIC, which can be important in controller applications. However, I like the board, and the price is right.

I've made arrangements with Vesta to distribute a specially configured UNIFORTH in place of their FORTH ROM. The SBC88 UNIFORTH is FORTH-83 Standard and includes a video editor, assembler, downloader, autoloader, printer control, time/date stamping, floating point, buffered A/D, and primitive graphics (amazing what you can fit into 16Kb when you try!). We've also included a 200-page manual. Check with Vesta for availability and pricing.

SBC-88
Vesta Technology, Inc.
7100 W. 44th Avenue, Suite 101
Wheatridge CO 80033
303-422-8088

Socketed board: $79.95
Full board: $169.95
Board w/operating system: $279.00
Power supply: $79.95

H4TH By Hartronix
As you might expect from the price differences shown below, you are in a different ballpark with the H4TH
The basic 6.6x11.0" board includes the microprocessor, 32Kb RAM, calendar/clock, two RS-232 ports (to 38.4K baud), a floppy disk controller (supports two 800Kb 5" floppy's), and a 500K baud high speed serial port. The optional case/power supply/floppy combination includes three expansion slots for more memory, serial channels, or CRT controller.

The system is well built though definitely utilitarian. The box won't win any designer awards, but everything fits well. A 2" ring binder contains the hardware and software documentation. My system came with preliminary documentation that was not user-friendly, but was almost usable if you were a proficient FORTH programmer. They have recently hired a technical writer and promise better documentation Real Soon Now.

The H4TH engine is a 4-bit processor that executes FORTH as its native (microprogrammed) instruction set. It is fast, as shown in the benchmarks, executing FORTH as fast as a VAX 11/750. It can run the Sieve benchmark as fast as the IBM PC can run optimized assembly language. Since the processor is microprogrammed, they have included a virtual computer that looks like an expanded PDP-11 (if you wish to use assembly language instead of FORTH). They also provide an assembler to create new microprograms for the processor (to add new instructions).

The major hardware limitation is that you are locked into their proprietary bus. They will provide interface information so you can wirewrap your own cards, but I doubt any second-source manufacturers will enhance Hartronix's selection of peripherals.

Software

The software is impressive in some areas, and deficient in others. You get a full multi-user system, with memory management, a breakpoint monitor, tree-structured file system, local labels, and a line editor. Files are time/date stamped, and have version numbers (though a new version overwrites the current file).

However, the formatting program supports only one format (presumably, so does sector r/w), so transporting software becomes more difficult.

No user protection is provided on the formatter: no "are you sure" type messages, and no indication of how the formatting is progressing. An optional floating point package is available, but it needs work.

---

**Perfect for the Beginner...**

**Invaluable Teaching Aid...**

**Quick Reference Guide for the Pro...**

**AXXESS KEYBOARD TEMPLATES...**

- Improved Angle Reduces Eye and Neck Strain.
- Windows Visually Organize Commands for Quick Reference
- Larger, Easier to Read Type
- Most Frequently Used Commands Listed First
- Created for Users by Users
- Easy to Understand English, Not "Computerese"
- No special symbols or Abbreviations to learn
- Examples Given in Key Situations
- Immediately Provides Full Use of Your Software
- Eliminates "MANUAL LABOR"
- Reduces Errors, Eliminates Frustration, While Building Confidence

**AXXESS KEYBOARD TEMPLATES...**

- Wordstar/Mailmerge/Spellstar
- Perfect Writer/Filer/Speller
- Perfect Calc
- dBase II
- M Basic
- CP/M-80

$14.95

The right command every time!

YES! Save COD charges by enclosing your check or money order, or use your Diners Club or Carte Blanche Credit Card! 0 M.O 0 C.O.D.

---

Name __________________________ Address __________________________
City __________________________ St. ______ Zip ______
SIGNATURE ____________________
CARD# __________________________ EXP __________________________
QTY DESCRIPTION __________________________

Subtotal ______
6% CA Tax ______
C.O.D. ______
Shipping ______
Total ______

AXXESS SOFTWARE SUPPORT SYSTEMS
P. O. Box 40169
San Diego, CA 92104
(619) AXXESS-1

---

Micro Cornucopia, Number 25, August-September 1985
The TAN/ARCTAN function has bugs, the word set is not consistent with the proposed floating point standard, numeric entry involves the use of a separate exponent, and the IEEE numeric format is not used. The Fig line editor is usable, but a full screen editor should be provided (all other vendors do, and this is billed as a development system). I never used the virtual computer or the target compiler since no documentation was provided for these features.

Hartronix supports the FORTH-83 Standard for the most part. Their division is floored — the first vendor I’ve seen that really supports this aspect of the Standard. Divide by zero is trapped, but overflow is not. Stack underflow is trapped, but words like ROT that do not change the stack depth will work even if insufficient parameters exist. I was hoping a microprogrammed system could implement proper stack error checking.

My major complaint with the software is that Hartronix has redefined many standard words: CONSTANT becomes CON, VARIABLE becomes VAR, etc. They use word addressing instead of byte addressing. This means you get 128Kb address space using a 16-bit FORTH, but must kludge string and other byte-addressed operations.

I converted a word-addressed FORTH on a PDP-11 to a byte-addressed system and don’t recommend the experience. You run across many circumstances (such as ‘I + C@’ and ‘1 +LOOP’) where you continually get burned in transporting software unless you fully understand both the source and destination systems, as well as what the application does.

If someone needed a high-speed FORTH development system, or had some application that needed the multi-user features, I wouldn’t hesitate recommending this system — when the documentation is finished. You may need to port your development tools (such as the video editor) from your current system before much programming is accomplished. I’m harsh on the system because it has great potential but hasn’t reached full maturity yet. Give this system another six months and it should have excellent software and documentation to complement the hardware.

H4TH Engine
Hartronix, Inc.
1201 N. Stadem Drive
Tempe AZ 85281
602-966-7215

Single board computer: $2000
Dual floppy system: $4200
Winchester system: $8400

Summing Up
I’ve examined FORTH engines in general and three implementations in particular. For compactness and low cost, the ROM FORTH single-board computers are good choices for dedicated controller and other OEM applications.

For situations where high performance is required, the microcoded FORTH engines or the single-chip Novix are wise choices. Consider these products in your next custom application. The ease of programming in a high-level language will shorten the development cycle and usually result in a higher quality product.
Pascal Procedures

By John P. Jones

In this column I'll give you a few pointers on pointers. Actually, it'll be more than a few pointers—we're going to wade right into this non-trivial subject. As you'll see, pointers can be a very powerful tool, but a lot of people don't use them because they don't understand them. We'll begin by looking at allocation of variables, and then we'll see what that has to do with pointers.

Static & Dynamic Variables

Variables declared in the PROGRAM section and in non-recursive procedures are static variables. This means their storage is allocated and fixed at compile time. Two mechanisms are built into Pascal for creation and use of variables at run time—recursive procedure calls and dynamic variables.

Procedures compiled with recursion enabled will, when called, generate new copies of their internal variables on the recursion stack. On return from the procedure, their space is released for other use. This is all transparent to the programmer and application.

Pointers

The other mechanism for run-time variable allocation involves pointer variables.

A pointer is a variable which contains the address of a variable instead of the data itself. Pointers are declared by prefixing the pointer character ('*') to a variable type.

```
type
  strarray = array[0..5] of string[16];
  intptr = 'integer;
  aryptr = 'strarray;

var
  dynint : intptr;
```

Pointers can be assigned and compared in much the same way as other variables. The primary function of pointers, however, is to allow for the creation and use of dynamic variables. Free memory is assigned to the heap, and this area is available for newly created dynamic variables. The statement:

```
new(dynint);
```

will create a variable of type INTEGER on the heap, assign the integer's address to dynint, and reduce the size of the heap by one integer's storage space.

```
dynint:---> | dynint^ |
```

Dynamic variables are accessed by using the pointer character as a suffix to the pointer name. So the above says that "dynint" is the address while "dynint^" is the value contained at the address.

```
dynint := 3;
otherintvar := dynint;
```

Two mutually exclusive mechanisms are used for de-allocating heap space. Your compiler will have one or both of them.

```
dispose(dynint);
```

Dispose is the complement of New and will do what its name implies—dispose of the storage for dynint.

The other method for reclaiming heap space is called "Mark" and it lets you mark a variable so you can reclaim it and everything beyond it.

```
Mark(dynint);
```

This places a marker on the heap at dynint's position. Later, when the statement: RELEASE(dynint) is executed, all space allocated from dynint onward will be de-allocated. If your compiler provides both methods, results will be unpredictable if you use both in the same program.

If this were as far as the story went, dynamic variables would not be much more useful than statics. Since pointers can point to records and can also be fields of records, uses for dynamic variables are limited only by the imagination.

Linked Lists

Linked lists are tail in, first out devices. Your compiler provides both methods, recursive and non-recursive, for creating them.

```
type
  lstptr = 'listitem;
  listitem = record
    itemdata : datatype;
    nxt_item : lstptr;
  end;

var
  root, item : lstptr;
```

we can construct a simple linked list. The pointer root is our anchor on the world, and it points to the start of the list. The list is initialized with:

```
root := nil; { nil is the pre-defined value for a null pointer }
and items are added in this way:
```

and items are added in this way:

```
new(item); { create new var }

item.itemdata := data;

item.nxt_item := root;

root := item; { reestablish anchor }
```

Singly linked lists as shown in Figure 1 can be useful, but they have disadvantages. First, the components can only be accessed serially from the previous item. Another disadvantage is that insertions and deletions can be cumbersome. Also, if you use a list like this, all procedures that access it must be sure to check for an empty list.

The First Shall Remain So

By using an additional reference pointer that always points to the oldest item, a structure called a queue is formed. See Figure 2.

In this example, items are added using the pointer LAST, in a way similar to the above for a simple list. The additional factor to consider is that the initial item should be added with the pointer FIRST. Removal of items from this queue can be done as shown in Figure 3.

A queue is thus a first in, first out structure, while the earlier simple linked list is a last in, first out device.

```
type
  lstptr = 'listitem;
  listitem = record
    itemdata : datatype;
    nxt_item : lstptr;
  end;

var
  root, item : lstptr;
```

```
Double Trouble

As mentioned above, insertions and deletions in these singly linked lists are cumbersome. At the cost of a little more storage, doubly linked lists have a much cleaner implementation of these functions.

type
dblptr = "dbl_item;

dbl_item = record
  data: datatype;
  before, after: dblptr;
end;

var
  root, crnt, nxt: dblptr;

With these declarations, each list item has two linkages to the other list items. BEFORE, the backward pointer, points to the preceding item, while AFTER, the forward pointer, points to the succeeding item. If the forward

(continued on page 73)
MONITORS

12" - hi res - 18Mhz
80 x 24 chars.
composite video
RCA video cable included
completely refurbished
w/new CRT
mfgd. by BMC
video thru port
$65.00 green phos (P-31)
$55.00 B&W (P-4)

9" - hi res. - 18Mhz
80 x 24 chars.
composite video
RCA video cable included
completely refurbished
w/new CRT
steel case
video thru port
$45.00 B&W (P-4)

MODEMS

Rixon T212A

300/1200 baud direct connect
Bell 202 comp - RS-232
modular connect to phoneline
auto ans/orig. - self test
$199.00

POWER SUPPLY

w/case-fan-AC cable
on/off switch
9L x 4 1/2W x 3 1/2H
single board
5 VDC 5amp
-5 VDC .5amp
12 VDC 2.5amp
-12 VDC .5amp
$40.00

I.C.'s

Z-80 Support System

1- Z80A CPU
1- Z80A PIO
1- Z80A SIO/1
1- Z80A CTC
1- 2764 Eprom
$15.00 per set

Eproms

2708 (pulls)  2/$1.00
2716 (pulls)  1.00
2732 (new) Hitachi 3.50
2764 (pulls)  2.50
2584 (new) T.I.  4.00

Memory

4164 - Mostek - 150ns
$30.00/Tube (27 pcs.)
4164 - Mls. - 150ns
$40.00/Tube (25 pcs.)
4164 - Imloss - 100ns
$35.00/Tube (25 pcs.)

CABLES

IDS20/IDS20 - 3' $4.00
IDS26/IDS26 - 5' 6.00
IDS34/IDS34 - 2' 5.25
IDS50/IDS50 - 5' 10.00
IDS26/IDE26 - 6' 7.75
IDS26/DB-25 - 10.00
DB-25(m/m) 25' std. aerial
pins 1-8,15,17,20,22,24
$10.00 ea.  3/$28.00

parallel printer cable
for IBM & compats. - 6'
$20.00

18 pin LP sockets 10/$1.00
28 pin LP sockets 16/$2.00

8" DISK DRIVES

Shugart 800-2 SS/DD
refurb & tested
$75.00

MPI BR88A-A DS/DD
Shugart 800 series comp.
refurb & tested
$125.00

SELECTRONICS

Dealers in new & refurb
electronic equipment

1229 S. Napa St.
Phila.,Pa. 19146
215-468-4645
Visa & Mastercard accepted
all prices f.o.b. our plant
pointer of the last item points to the first item, and the backward pointer of the first item points to the last item, you get a doubly linked ring. An empty ring can be defined as a single dummy item with both forward and backward pointers pointing to itself. See Figure 4 for a doubly linked ring.

To insert nxt after crnt:

```pascal
nxt\.after := crnt\.after;
{nxt's forward := crnt's forward}
nxt\.before := crnt;
{nxt's backward := crnt}
crnt\.after\.before := nxt;
{crnt's successor points to nxt}
crnt\.after := nxt;
{crnt's forward points to nxt}
```

To insert nxt before crnt:

```pascal
nxt\.after := crnt;
{what comments would you make?}
nxt\.before := crnt\.before;
{crnt's before := nxt}
crnt\.before\.after := nxt;
{field "after" in record pointed to by "before" in record pointed to by "crnt" is assigned nxt}
crnt\.before := nxt;
```

Deletions are even simpler. To delete crnt:

```pascal
crnt\.after\.before := crnt\.before;
{crnt's before := crnt before}
crnt\.before\.after := crnt\.after;
```

Since a ring has no ends, when you traverse a ring, you have to remember where you start.

```pascal
nxt := crnt\.after;
while nxt \neq crnt do
begin
process_item;
nxt := nxt\.after
end;
```

**Trees**

When a record contains multiple pointers to other records of the same type, a structure called a tree can be formed. If the tree is constructed (grown? no, groan) so that each node can point to two other nodes, left and right, the result is a binary tree.

```
type
    nodeptr = "node;

    node = record
       data : datatype;
       left, right : nodeptr;
    end;
```

If the insert procedure is written so that the new item traverses the left branch if it is less than the current node’s data or the right branch if greater than, a binary search tree is formed. One important characteristic of a binary search tree is that a traversal that always visits the left subtree, then the node followed by the right subtree will yield an ordered (sorted) sequence of data.

**Caution:** If the data is entered into a binary search tree in ordered sequence, the result is a linear list. For optimum results, the data should come in randomly.

**Homework Assignment**

Pull out your favorite Pascal text and use it to expand on this brief look at dynamic data structures — they are one of Pascal’s greatest strengths. Especially when working with lists and trees, it is very helpful to draw pictures and diagrams. They will often make the difference between "I don’t understand" and "Gee, that’s clever."

**Turbo Power**

Shortly after sending in the column for issue 24, I received a copy of Turbo Power! for review. This utility is written in Turbo Pascal and can be either run standalone, or from within Turbo with the eXecute command. TP provides facilities to compare, copy, rename, delete, and type files or groups of files. If you are running standard CP/M 80, TP is well worth the $25 price. On the other hand, if you have ZCPR2 or ZCPR3, most of the facilities are available as transients which can be run individually with the eXecute command.

Installation is simple, and necessary, since the program is highly screen oriented in its operation. For the version I received, Turbo version 2 was required to install the program, since my modified Big Board terminal software did not match any of the terminals in the standard installation. Once installed, TP will run under Turbo version 3.

The program works as stated in the ads and documentation, and I found only one problem with it. If the overlay file, TP.000, is not on the currently logged disk, the program will abort when some functions are attempted. It could have been written to check for the overlay file’s absence and lock out the functions that need it.

The seven-page manual is sufficient, since on-screen instructions are clear and most of the commands are either obvious or logical. The few places where additional detail is needed are covered in the documentation. I would suggest some expansion of the manual for novices to CP/M, especially on the subject of wild card characters.

The program is well thought out, and provides most of the ‘bells and whistles’ that previously would have required leaving Turbo. Recommended.

**Pascal Power!** is available from:

Pascal Power!
5666 La Jolla Blvd., Ste. 136
La Jolla CA 92037
Turbo Memory Assignments

By James R. Shiflett

In the April 1985 issue of Micro C, Tom Geldner mistakenly stated that programs written in Turbo Pascal for CP/M 80 cannot dynamically adjust their memory usage to match the computer's available memory.

While it's understandable that Tom would believe this, especially in light of the fact that even Borland's technical support staff know of no solution, it is nonetheless quite false. The Turbo reference manual explains that the memory of a compiled program is normally organized as shown in Figure 1.

The memory addresses assigned for all variables are assigned at compile time and they are placed right under BDOS. If the compiling computer has more room than the computer it will be running on, then really bad things happen.

The Turbo Pascal Reference Manual suggests you lower the END ADDRESS during the compilation to restrict the highest address the program can use. This works, but has one major disadvantage: the memory above the specified END ADDRESS is unavailable to the program (unless you know a trick).

A more reasonable solution is to rearrange the structure of the memory assignments to avoid putting the fixed location items at the top of memory. My solution forces Turbo to put the variables just slightly above the object code. Then the HEAPPTR and STACKPTRs can be dynamically adjusted to utilize full available memory. Figure 2 illustrates this.

You can implement this arrangement easily, and it makes all the machine's memory available to the program where it needs it most — in the Heap and Stacks.

Simple Solution

The commands shown in Figure 3 are placed in the program so that they're the first commands executed.

That's all it takes. The specific values shown are from a program I'm preparing for release this summer called Le-Menu. It's a user friendly operating shell for CP/M that's written in Turbo Pascal (with a few assembly language subroutines for efficient CP/M interfacing). I'm writing it specifically to demonstrate a number of the advanced techniques I've developed in two years of intensive Turbo Pascal programming. (What good are neat algorithms if they're never shared?)

During the development and debugging phase of a program I simply let Turbo use its default END ADDRESS. When I'm ready to release a version to the public I run Turbo in the COM mode to create a disk file image. This gives the following report:

```
Code: 19487 bytes (1FC9-6B88)
Free: 24136 bytes (CA32-E03A)
Data: 5640 bytes (CA32-D03A)
```

I can now add the size of the DATA group to the CODE group's top address and calculate the desired END ADDRESS to give to the compiler. For the above example I used 6BE8h + (E03Ah-CA32h). To play it safe, I add a little and round up to the next page. Therefore I set the HEAPPTR to $840F (16 bytes above the END ADDRESS). I set the Turbo compiler option END ADDRESS to 8400h and recompile the program (no real time lost since Turbo is so fast).
The report:

Code: 19487 bytes (1FC9-6BE8)
Free: 526 bytes (6BE9-6DF7)
Data: 5640 bytes (6DF8-8400)

The size of the disk program is unchanged, but the result is a program that can execute properly on nearly all CP/M systems while retaining optimum memory.

Turbo Pascal is such a powerful implementation that it can be viewed as much a revolution in language utility as it is a revolution in quality software marketing and pricing. I hope others will apply the “nothing’s impossible” attitude to Turbo and help raise the general level of public expertise.

---

Attention Okidata 92/93 owners... Announcing

FEDIT

a font editor for your Okidata printer.

FEDIT is a complete system for designing and defining characters for the downloadable character graphic generator (DGCG) in Okidata 92/93 printers. The designed fonts are kept in individual files on disk available for editing and/or sending to the printer by FEDIT utilities.

FEDIT features include:

- full-screen control of the character guide
- using single-stroke control codes like the popular word processor
- complete, transparent font file handling, including backup files
- includes a predefined file of cursive and scientific characters for immediate use
- complete documentation including design tips and a sample layout sheet
- separate Send utility for finished fonts
- complete. Salus type available also.

FEDIT is currently available on 8” 5500 CP/M v2.2 for Big Board I using the on-board video and Xerox 820. FEDIT is also available on 5.25” CP/M for Keypro, 620, 12, 14, 16. Cost is 619.95 US $ 625.95 CAN. For information and orders contact:

Brooke Computer Systems Ltd
4th Street West
Owen Sound, Ontario
CANADA N4K 3K8

Shipping and Handling please add 2.50 US and CAN.
Check or money order only, no COD’s. Ontario residents add 7% sales tax. CP/M is a trademark of Digital Research Inc.

---

Pascal Power!

5666 La Jolla Blvd., Suite 136
La Jolla, California 92037
(619) 587-0857

*** Press Release ***

The most valuable computing resource is your time. Pascal Power, a company dedicated to providing Pascal tools to enhance the use of popular microcomputer Pascal compilers, proudly announces its product, TurboPower.

TurboPower provides an integrated set of software tools to supplement microcomputer program development using Borland International’s Turbo Pascal system.

While Borland International’s Turbo system is an excellent product, it fails to provide mechanisms to accomplish many of the complex and tedious tasks associated with software development. TurboPower supports the programmer in these tasks, allowing him to find differences in files, look for patterns in files, erase files, rename files, backup diskettes, print files, copy files and more. All operations are performed on a single file or on a list of files. If a change in the program being developed causes it to break, TurboPower will tell the user what was changed. TurboPower helps the user organize all of the pieces of a software project.

With the ability to operate on an arbitrary subset of files, the TurboPower user can quickly initiate compound tasks, such as backing up those files (and only those files) that have been changed since the previous backup. The users’ guide illustrates how TurboPower can be used to quickly accomplish many complex tasks.

TurboPower is highly graphic and simple to use with a single keystroke, the user operates on objects that TurboPower places on the screen.

TurboPower is currently available by sending $24.95 check or money order to Pascal Power, 5666 La Jolla Blvd., Suite 136, La Jolla, California 92037.

TurboPower is available for all CPM/80 versions of Turbo Pascal in 5¼” disk formats.
Excerpts From A Talk By Niklaus Wirth

The Two-Part Module
There is a definition part and an implementation part of the module. In the definition part you declare types and procedures which are going to be seen from the outside. Those objects are exported. In the implementation part you declare all the things that aren't exported.

That's precisely what we needed for system design. In a large group, you first design a system's definition modules, and the group must agree on these modules. Then individuals can go and work on their respective implementations. It sounds a bit too ideal — certainly an abbreviated sketch of what it really is.

The language covers only the syntactic aspects. You know which procedures are exported, and how many parameters.

The semantics are not checked. There you need the trustworthiness of your colleagues. For instance, you must be able to trust that if someone writes a mathematical routine he will not name it 'sine' and compute the cosine. That would not be checked by the compiler.

Modula satisfies most of the aspects of system design. I mean all these things that ADA offers a solution for.

Modula Vrs. ADA
First of all, a very significant advantage of Modula over ADA is that it is here. We at the institute have worked on it now for six years, and it's proven to be useful.

Furthermore, Modula II compilers are effective and efficient.

And last, the language can be wanted to get into multi-programming, explore methodologies for designing programs that are safe, pure, correct. And instead of setting up rules, we ended up formulating these rules in terms of a language.

Modula I had the very important concept of a module, but apart from that it was really an elementary language. Obviously not good enough if you wanted to have a system with a single language.

What Makes Modula II Interesting?
It is in the tradition of the structured languages, Algol and Pascal. In addition, it has those things that Pascal lacks for writing larger systems — primarily the module structure.

A module is hard to explain in a few sentences, but crudely said, it is a collection of procedures and data structures around which you build an intransparent fence. Then you can poke holes in the fence. Then you can say these and these objects are visible from the outside, and conversely you can say these and these objects outside are visible from the inside. They are unidirectional holes (innies or outies) called import and export.

Separate Compilation
What makes this concept particularly useful is the combination of separate compilation — by no means a minor feat.

We have a strongly typed language (every variable must be declared). By that we mean that the program has quite a bit of redundancy which the compiler can utilize to do a consistency check on types.

Think of FORTRAN which has separate compilation. If you call a subroutine which you bring in from somewhere else, and you happened to forget one of the subroutine's parameters or the parameter's type, you don't know this until you execute the program.

When you have structures, all these things are based on the concept of type, so it's essential that you don't lose type checking when you cross module bounds.

We found that the effort it took us was drastically reduced because we had a highly structured language with separate compilation, which makes it possible to work in teams.

Why? Because the separately compiled modules are first divided into two parts, a concept that was developed at Xerox Park. And I merged that concept into the module concept of Modula.
defined in a manual of not more than 30 pages. In a way, I am proud to say it is only 30 pages, but in a way I'm ashamed it's so much. Because when you do significant system design you just can't afford to have a manual that is stacked with stuff.

I designed a new version of the Modula II compiler in the last two years in my spare time.

Compiler Size

This Modula is about 5,000 lines of source code. That compares extremely well with Mesa at 50,000 lines or ADA with 100,000. And I must tell you very frankly that 5,000 lines is about the limit which I can honestly say I understand. I got back to the compiler a few weeks ago, and boy o' boy, certain things I have to study again until I understand exactly why I did this or that.

If someone offers you a compiler and says it is written in its own language, ask him how long it takes to compile itself. Mine is now less than two minutes on the Lilith computer. The first Modula compiler took about half an hour on the same machine.

Let me not extend my monologue too much. Some of you aren't very much interested in these details. May I open the floor for questions and you direct me.

(Throughout the rest of the article, the bold-faced lines are questions from other attendees.)

You have no feel for Turbo Pascal or Turbo Modula II?

Not about Turbo Modula, but Turbo Pascal I know is a very compact system, very handy. I think it is a very good system. But don't make the mistake of trying to use it for large programs, because it is slow running.

As far as I know it is a system well suited for introductory programming courses and fast compilation with the immediate integration of the editor. Good for debugging.

I mean that is the main reason that I did Modula, because all the compilers that were becoming available in Pascal were fine for toy programs, fine for introductory courses, but I would like to show that structured programming languages are not just for the school. Their real value comes when you do big systems. For that you need efficient compilation.

Are you using Turbo Pascal for teaching?

No, we are not. For the introductory courses we use Apple IIs. (Loud chuckling from the group.)

You were here for a sabbatical (8 years ago) and the result of that was Modula II. Now you're here again. What have you got coming?

I came here because it is a chance to learn something new. And as a teacher in a field which expands so rapidly, this is an absolute necessity. I came this time to learn something about LSI design.

I did a revision of Algorithms and Data Structures that's coming out in August. It's based on Modula.

We do have a new work station under development based on the National 32000 processor. One of the extra benefits of my compiler was that I understood it completely so could target it easily to other machines. So I did for the 32000. I will also design an operating system for that machine.

I got converted to Turbo Pascal. Now I understand Borland is coming out with Modula II. Have you heard about this?

I have heard indirectly about this, but I'm not sure.

Does the 32032 bother you because of the size of the instruction set? In other words, if you can't document it in 30 pages, it's too big?

It bothers me as a compiler writer, but as a Modula programmer, the compiler protects you from all that. I can compare the two compilers because the first prototype of the work station is now operational.

The 32016 system takes 1.6 times as long to compile, and the same ratio holds for code length. The same program in 1.6 times longer compiled for the 32000. It turns out that most of the additional code length is due to the fact that the shortest address length is 8 bits whereas on the Lilith it is 4 bits. If you measure code you see that over 50 percent of addresses are offsets between 0 and 15.

What's the fundamental difference between modules in Modula, and functions in C.

First of all, in a module you can have a bunch of procedures, so the module can have several entry points. Furthermore, you can have types and variables that you export.

For instance, you can define a file system and a block that hides from you all the administration that takes care of allocating storage on the disk and so forth. You explain in the definition module only an abstraction of the file system. Someone who is using that file system will not have access to the disk. Instead, he must go through this module, so you can guarantee that certain things are satisfied. Data integrity is guaranteed.

You could write a module that defines a type 'complex number,' and the procedures you provide are the basic operators. You can guarantee that certain things are guaranteed.

When you compile an implementation module it generates code. When you compile a definition module you don't generate code, just a symbol table.

Later, when you compile a module that imports from (modules) A and B and C and so on, the compiler goes to fetch the symbol files that were
generated when compiling those definition modules.

When I designed this latest compiler I found I had to make a major change to the data structure. It would absolutely unthinkable to do that if I had coded in assembly language, because it would be guaranteed to overlook some consequences. Here the compiler tells you.

You can import a standard module called SYSTEM, and in there you have defined a type address that’s compatible with all kinds of pointer types, so it works against (doesn’t do) type checking. So you better import things from system when you are willing to take all the responsibility.

For instance, it is possible to program the storage allocation, garbage collection. It’s also possible to program device drivers because it is possible to access specific machine registers.

People who have written extended Pascal have added the same things, but here it is not indecent to do it because the programmer is encouraged to introduce these low level features in only one or two modules. For instance, a device driver module. So in all the big part of his system he has the full type safety of Modula and still has access to the low level facilities.

What are the problems you face that are unique to Europe?

The language is less of a problem than you might think. A lot of people, especially those in computation centers, develop their own Yiddish — a very strange combination of German, Swiss-German, and English. It even goes into their publications.

For our students, that isn’t a particularly big problem. I recommend that they read English journals. You must teach in German — in high German which is a foreign language for us.

Certainly the big impetus in our field is from the U.S., and that gap continues to grow, especially in hardware. In the U.S., you do a job quickly, and if it doesn’t work you buy more memory. It’s so easy to do that.

What should we be watching for from Europe?

It’s a bit presumptuous to say that Europeans think out a problem more thoroughly, but at least I’ve found that when I develop a system it pays off that I think very thoroughly about the system before I start coding.

With all these personal computers at every desk, people forget this. I remember 20 years ago when we carried around card decks — you surely thought carefully before you put your cards in the deck. If there was an error it cost you half a day. I’m not saying I’d like to go back, but resources, when they become cheap they are wasted.

I would hope that if anything comes out of Europe it is in the area of systematic methodologies. I would point out the work of Dijkstra and Hoare which is first rate.

If I have any recommendation for teaching programming it would be to introduce more of that style programming. I’m afraid this is very unpopular because it means that you need mathematics and mathematical thinking. We have, in my opinion, badly neglected the use of mathematics in computer science. We are always using it the other way around. We solve mathematical problems with the aid of the computer, but we should start doing it the other way around — solving computing problems with mathematical methods.

I have never written a large program where I follow Dijkstra/Hoare in minute detail everywhere, but I must say that by acquiring this way of thinking about programs, the places where the algorithm becomes sophisticated I will write down the variants. Making them explicit helps me tremendously.

On the one side what makes programming safer, more effective, are tools for programming in the large: the module, type checking. On the other side, for programming in the small are mathematical methods: rigorous methods: clean, clear thinking; and I wish people would use these earlier.

What is your opinion of artificial intelligence?

I must say that my opinion of artificial intelligence is very, very low.

Artificially low?

When I ask what is artificial intelligence I never get a clear answer. If I ask what has artificial intelligence produced in solid results I never get a clear answer. They have been living for the last 25 years on big promises and big money from the Department of Defense which thinks that sometime the break will happen.

Of course, they are in an unlucky position because if they really produce a result, it’s not artificial intelligence anymore. But by and large, it is a euphemism for extensive searching methods.

What are your comments about personal computing in the Soviet Union?

I know very little about it. But I’m not deaf. The fact that you hear little about it probably says something. I was in Hungary 1 1/2 years ago and they are very eager to learn. Even in the research centers they struggle with tools that we had 10 years ago. And to get an 8080, that is remarkable.

How do the Japanese fit into the software field?

The Japanese are beyond the period of just being good at copying our things. They have recognized that software is very important. They have become very active, and I have had several visitors from Japan who are interested in Modula.

If you look at the market now in software there isn’t much there, but I would expect to see things happening in the next 10 years.

On the other hand, I think this great fear by Americans that they will be rolled over by the Japanese is vastly exaggerated. It’s a business trick. And mind you, it’s the artificial intelligence people who make a big
I understand that you originally wrote Pascal for teaching and didn’t intend it for the business environment. Is that true?

I hear that all the time. I don’t know. Wouldn’t someone be a fool if he designed something just to use in the classroom and afterwards it was no good? That would be cheating the students.

It is true that I designed it with the classroom in mind. At the time, we had Algol compilers, which were lousy, and a FORTRAN compiler, and I was looking for something that people couldn’t say was lousy.

Also, I was in the business of writing compilers and systems software. I had developed Algol W in a working group, and I wasn’t happy with what came out of there. It was a product that had the burden of being developed by 30 people. With Pascal I was freed from that, and I definitely wanted a language that was good for systems engineering.

Commercial use is a good test of a language, and I want Modula to be used commercially.

Dijkstra is now a professor at the University of Texas at Austin. There is a whole group at the U of Texas, so I wouldn’t be surprised if something good came out of there. Also Professor Hoare is going there on a sabbatical next year. They are massing.

How do we get increases in software productivity to match our increases in hardware power?

These are two different things. Surely we have made big advances in producing software, but they are not as big as in the area of hardware. If you want to be fair you have to compare how much we have learned to develop software and hardware quicker. Not quicker hardware. As far as I have heard, developing the 68000 processor cost more than developing the 7090 computer which filled this whole room. So hardware advances are not all that much different than software.

Designing a complex system is an intensive intellectual activity. It takes time and brain, be it software or hardware. As a matter of fact, it’s unfair to say that hardware has made more progress than software — chips cost 100 times less than 10 years ago while software costs go up, and while we have not learned to design hardware cheaply, we have learned to duplicate it cheaply. Copying a piece of hardware is almost as cheap as duplicating a magnetic tape. That would be the fair comparison.

The design process is always expensive, even with artificial intelligence in the offing.

I understand that Dijkstra has been working on a method of proving the correctness of software. Have you heard anything about that?

He is working on a method of developing a program with the proof inherent. That’s a significant difference.

He is working on a book which should come out late this year. The book which best exposes his theories has the somewhat presumptuous title “The Science of Programming.” I think it is well written, but it is too heavy. It consists of three parts, of which the middle one is the smallest but most important.

The first part is a lengthy introduction into mathematical logic and predicate calculus. The middle part is the exposition of Dijkstra programming, and the third part tries to apply these methods. I think the second part is well worthwhile.

Dijkstra is a professor at the University of Texas at Austin. There is a whole group at the U of Texas, so I wouldn’t be surprised if something good came out of there. Also Professor Hoare is going there on a sabbatical next year. They are massing.

Accuracy, speed and security are three key reasons to use a PERCON® E-Z-READER™. It significantly reduces the chance of human error while providing much faster data entry usually WITHOUT SOFTWARE MODIFICATION. Call us today to see just how easy it is to make your life—and your clients’—better with bar code.

$595 Quantity One
IBM PC/XT Keyboard Compatible
IBM AT Keyboard Compatible
Custom orders welcome
• Reads dot matrix, standard and high density bar code labels accurately
• Supports Code 39, UPC A/E, I 2 of 5 and Codabar/ABC
MS DOS bar code printing software available

Substantial discounts are available for quantity orders and qualifying Value Added Resellers, OEMs and Distributors.

PERCON® E-Z-READER are trademarks of Peripheral Connections

Sidekick for CP/M!
Poor Person Software brings you

Write-Hand Man
Desk Accessories for CP/M

Suspend CP/M applications such as WordStar, dBase, and SuperCalc, with a single keystroke and look up phone numbers, edit a notepad, make appointments, view files and directories, communicate with other computers. Return to undisturbed application! All made possible by Write-Hand Man. Ready to run after a simple terminal configuration! No installation. Don’t be put down by 16 bit computer owners. Now any CP/M 2.2 machine can have the power of Sidekick.

Bonus! User extendable! Add your own applications to Write-Hand-Man. All you need is M68000 or RMAO.

$49.95 plus tax (California residents), shipping included! Volume discounts.

Available on IBM 8 inch and Northstar 5 inch disks. Other 5 inch formats available with a $5.00 handling charge. CP/M 2.2 required.

COD or checks ok, no credit cards or invoices

Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
tel 415-493-3735

Write-Hand-Man trademark of Poor Person Software, CP/M and RMAO trademarks of Digital Research, Sidekick trademark of Ballard International, dBase trademark of Ashton-Tate, WordStar trademark of Micropro, SuperCalc a trademark of Sorcim, M68000 trademark of Intel.

Micro Cornucopia, Number 25, August-September 1985
No Lower Price Anywhere

We have the lowest prices for Sanyo SOLVEX MARKETING

203-928-3654 PHONE ORDERS ONLY

$799 !

TeleVideo Portable Computers

TPC-1

# TELEWRITE - Word Processing Software
# TELECALC - Spreadsheet
# TELECHART - Integrated Business Graphics
# CP/M-80 v. 2.2
# 2-368 K / Drives Formatted
# 64 K RAM, Expandable to 128 K
# 32 K Graphics RAM
# 640 x 240 Bit Mapped Display
# 9" Yellow Non-Glare Screen
# Serial Port, Printer Port, Mouse Port
# IBM Style Keyboard
# Easily Connected to Televideo Network Systems
# Read Dozens of CP/M Disk Formats

with Mycroft Lab's COMPAT. Only $49.00

PR5000 $299

# Letter-Quality Printer
# Accepts Paper up to 13" Wide
# 14 cps Bi-Directional Printing
# 10, 12 or 15 cpi Print Pitch
# 2 ips Paper Feed
# Centronics Interface
# Tractor Feed Available ($125)

SANYO MBC 3000 $599

# WORDSTAR, SPELLSTAR,
MAILMERGE, CALCSTAR,
INFOSTAR, BASIC,
CP/M-80 v. 2.2
# DUAL 1 MB 8" DRIVES
# READS IBM 3740 DISKS
# 2 SERIAL RS-232 PORTS
# PRINTER PORT
# INTEL MULTIBUS SLOT

64K PRINT BUFFER - $89.

★ CASH PRICE ONLY ★

Check in advance. Add 3% for VISA/MC. No Returns.
Shipping & Handling charges will be added to each order.

For complete technical literature send $1 to:
P.O. Box 689
Putnam, CT 06260

PHONE ORDERS ONLY:
203-928-3654

203-928-6555

111 ALSO IN STOCK 111

# DUAL 8" 1 MB FLOPY: $399.
8 MB HARD DISK: $999.
(For MBC 1100 Series)

# CRX 1100 TERMINAL -
8 PF KEYS, MENU SETUP,
4 DIFFERENT EMULATIONS
PRINTER PORT -- $299. (3 or more)
Old TVs used to generate their 20,000V picture tube anode voltage directly from 110VAC. The transformer they used was huge (and expensive). Now TVs use a high frequency oscillator (the horizontal) running off 12VDC to do the same thing, quite inexpensively.

With the advent of inexpensive DC to DC converter chips (high frequency oscillators), you too can take one voltage (5V) and generate another voltage (25V, for instance). The input voltage can be whatever you have, usually 5V or 12V — and depending on device, the output can be anywhere between -25V and +25V. You probably wouldn’t run an 8” drive off one of these circuits — most are only good for 20 to 500 milliamperes (500 mA is 1/2 of an amp), but they are perfect if you need 25V for programming ROMs, or + and -12V for an RS-232 interface.

The DC to DC converter described below uses +5V to provide the +25V needed to program most proms. It employs a TL497 integrated circuit and a few other commonly available components, fits on a few square inches of board, and costs less than $10 to build. The output voltage is variable and regulated.

The rest of the article talks about how to customize the circuit, but if you only want +25V, you can stop reading here. Just turn to the schematic (Figure 1), and start building.

Designing
I found the TL497 specs in a TI voltage regulator handbook. The section on the 497 was labeled “Bulletin No. DL-S 112422, June 1976 — revised September 1977,” but you shouldn’t need the tech data. I’ve included the pinout diagram (Figure 2), the configuration I used (Figure 1), the matching design equations (Figure 3), and the timing capacitor/on-time chart (Figure 4).

You can use different regulator configurations which produce negative voltage, higher current, or a voltage lower than the supply voltage. But you’re on your own for these.

Performing
The circuit (with my values) puts out 35 or 40mA at 25V and varies less than 0.1V from no load to a 35mA load. PROM data sheets list the programming voltage/current requirements as 25V plus or minus 1V at 30mA max for 2716s, and 25.5V plus or minus 1V at 35mA max for 2732s (except for Intel 2732As which require 21V plus or minus 0.5V at 30mA max). When tested at an output of 35mA continuously, the TL497 was barely warm (it’s rated at 500mA, but that sounds awful high).

Getting Started
Using the design equation (see Figure 3), \( V_{subI} \) (the input voltage)= 5 V ; \( V_{sub0} \) (the output voltage)=25V; \( I_{sub} \) LOAD max=35mA.

Let \( L=100 \) uH (more on this in the construction section). Let ripple voltage (\( V_{subRIPPLE} \))=0.05V.

Solving the equations, we get \( I_{subPK}=0.36 \); \( T_{subON} \) (the time the chip is on)=7.2 microseconds.

Now, exercise some engineering judgment and scoot \( T_{subON} \) up to 11

(continued next page)
microseconds, then select a timing capacitor value of 100pF from the table (Figure 4). (I prefer picking a value from the table to pushing a chip to its limit.)

Next, solve the R1 and RsubCL equations. R1=24.8K. The current limiting resistor (RsubCL)=2.8 Ohms. I found I needed to lower RsubCL to 1 Ohm to keep the output voltage from drooping at a 35mA load.

Finally, CsubF 23mF. Use whatever you have with a voltage rating above 30V or so. I used a 500mF electrolyte. The input capacitor across the +5v line is just for insurance (good design?) to prevent load changes at the TL497 input from being fed back down the line as voltage changes to other logic devices. Now that we've laboriously designed this dude, let's talk components.

**Components**

L, the inductor (see Figure 3), must have a low internal resistance (less than an Ohm, I'd guess). I used a Radio Shack #273-102A (about $1), 100 uHenry at 2 Amps. The TL497 came from Jameco ($3.25 or so).

The resistor combination R1-R2-R3 determines the output voltage. If these resistors heat up and change resistance, the output voltage may vary, so you might want to use precision resistors with low temperature coefficients for R1 and R2. But it shouldn't be necessary.

R3 can be a multturn trimpot, or may be replaced with a fixed resistor. As shown, R3 will vary the output voltage from 21V to 30V.

CsubT isn't critical. I used a disk ceramic.

For CsubF, a tantalum capacitor works best, but a 200 or 300mF electrolyte should work fine. The transient response of a tantalum is superior, and it'll last longer.

The current limiting resistor is a garden variety carbon. (Watch out for those cute little tomatoes in your resistor patch.)

**Constructing And Checking**

I wired wrapped mine, gluing the larger components to the perfboard with RTV, and wire-wrapping directly onto the leads. Gross, but effective. And cheap.

When you complete the power supply, test it by hooking up a load resistor which will draw 35mA (680 Ohms at 1 Watt) across the output. Leave it on for a few minutes and then check for smoke. Monitor the output voltage while connecting and disconnecting the load to ensure that it doesn't vary appreciably.

Everything worked the first time for me (much to my amazement), so I don't have any troubleshooting advice to offer. If you need help, call or write me.

**A Gotcha**

Pin 14 of the TL497 is B+ (+5 in this example), but the ground pin is pin 5, not pin 7 as we've come to expect.

**Credits**

Don Black, an especially adept Air Force hardware engineer who kept the Airborne Laser Laboratory's instrumentation working for several years, suggested this circuit. Thanks, Don.

Jameco Electronics
1355 Shoreway Rd.
Belmont CA 94002
415-592-8097

Micro Cornucopia, Number 25, August-September 1985
Goodies From Micro Cornucopia

BB I, BB II, and XEROX 820 USERS DISKS

The following are full 8” disks of software. Each program has a .DOC (documentation) file and come with source.

USERS DISK #1 - Two fast disk copy programs, both with source

USERS DISK #2 - One single disk drive copy programs, both with source

USERS DISK #3 - EPROM burning software for BB I

USERS DISK #4 - CBOS, custom boot for Tender drives

USERS DISK #5 - REZ, 8080/280 assembler, TDL mnemonics

USERS DISK #6 - PRINTPRN, print Crowe listings

USERS DISK #7 - PLINK66, line reader & programmer

USERS DISK #8 - BB I Double Density New BB I Monitor BIOS, character ROM, Winchester interface, Z80, and Z80 from Trevor Marshall. See BB I expansion article in issue #11.

USERS DISK #9 - BB I Double Density New BB I Monitor BIOS, character ROM, Winchester interface, Z80, and Z80 from Trevor Marshall. See BB I expansion article in issue #11.

PRINTPRN: Print routine for CROWEASM, PRN files.

LIBRARY: Utilities which let you combine many files into one, copy, paste, or erase any file within the larger system.

USERS DISK #21 - Winchester Utilities BACKUP: Helps you back-up the Winchester onto multiple floppy's. Creates a catalog of the files on each disk and includes the date of the latter backup. Will not back-up an unchanged file more than once. Plus many more super features.

MULTICOPY: Uses this like BIP but it prompts you to choose disks. Accepts ambiguous file names.

MDIR: Displays files in all user areas on selected drive. Many features.

MAKE, MOVE, PIP-like utilities that make it easy to move files between user areas.

SWEEP: The famous disk cleanup and transfer routine that does just about everything you can do with TYP, ERA, DIR, and PIP.

UNSD: This is the latest disk utility unqueser. Enter UNSQ, * and it will check every file on the disk. All squeled files will be unqueued.

USERS DISK #22 - Pascal Compiler This is a real Pascal compiler. It supports only a subset of the language (no records, pointers, booleans, reals or complex) but it generates a real .COM file. Everything is on this disk: the compiler, its source, example programs and documentation.

USERS DISK #23 - Xerox Utilities This disk contains Xerox specific utilities including a scanner dump from Wayne Blunk's documentation and software for implementing the SWP package including ZCP, a new monitor, and a clock/calendar from Mitch Milnar and Jim Mayhugh's new monitor (see issue #19). A very special disk for Xeroxers.

USERS DISK #24 - Prowriter Graphics This is a complete Prowriter graphics package written by the same Micro C subscriber who wrote the MX-80 graphics package. Plot points, lines, circles, boxes, and more. Examples, documentation.

USERS DISK #25 - Z80 Macro Assembler This is a real Z80 macro assembler Syntax closely follows RMAC and MAC. Also includes pseudo-ops to support conditional assembly etc. No phase or relocatable code.

USERS DISK #26 - BB1 CP-M/3.0 Banded BIOS/Winchester Support CP-M/3.0 Banded BIOS implementation for the BB I. Roy Epperson's software to support the Acorn ACB-4000 SCSI and the Rodime R204 5” Winchester on the BB I (see issue #19). Plus more Winchester programs.

USERS DISK #27 - BYE Remote CP/M System BYE programs to run your BB, BBII, or XEROX 820-1a remote CP/M system using a Hayes Smartmodem compatible modems. Includes programs to allow restricted access.

USERS DISK #28 - VFILER and Extended Single Density VFILER is a screen-oriented file manipulation utility, similar to SWEEP, CLEAN, and COMBO, Larry Blank's documentation and software for implementing extended single density (334K) on eight inch disks.

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709
EPIC SALES WINCHESTER SUB SYSTEMS

Featuring Drives by Seagate

- Supports CPM 2.2 operating system
- Host board plugs into 80-80 socket
- WD 1002 controller board external in cabinet
- Boot CPM from 8" or 5 1/4" floppy
- Easy installation

Complete documentation package is available for $35.00.
This package includes all manuals and software on a diskette (if required).
The $35.00 is refundable with the purchase of a Winchester sub-system from EPIC SALES.

<table>
<thead>
<tr>
<th>Seagate ST-212</th>
<th>CMI-5619</th>
<th>Seagate ST-225</th>
<th>Ervin 110</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2 High (Low Power)</td>
<td>15 MEG Internal</td>
<td>15 MEG External</td>
<td></td>
</tr>
<tr>
<td>IBM-PC &amp; COMPATIBLES</td>
<td>$749</td>
<td>$949</td>
<td></td>
</tr>
<tr>
<td>TI PC &amp; COLUMBIA</td>
<td>$799</td>
<td>$999</td>
<td></td>
</tr>
<tr>
<td>APPLE</td>
<td>N/A</td>
<td>$999</td>
<td></td>
</tr>
<tr>
<td>XEROX 820A KAYPRO</td>
<td>$1049</td>
<td>$1249</td>
<td></td>
</tr>
<tr>
<td>S-100</td>
<td>$1099</td>
<td>$1299</td>
<td></td>
</tr>
</tbody>
</table>

**SATISFACTION GUARANTEED:**
1-YEAR FULL WARRANTY
Every unit is thoroughly tested before shipping and our warranty includes both parts and labor.

**TECH SERVICE HOTLINE**
If you have a question or a problem call our service technicians.

**8" FLOPPY DRIVES**
- Siemens FD200-8
- DSDD 8" Floppy Drives
(Shugart 850 Compatible)
- $229 NEW

**5 1/4" SEAGATE WINCHESTER DRIVES**
- ST-212 10 MEG 1/2 HIGH
- ST-225 20 MEG 1/2 HIGH
- 5 MEG FULL HIGH
- $549
- $949
- $179

**WINCHESTER CONTROLLER & HOST BOARDS**
- DTC 5150BX IBM Winchester Controller
- $279
- ZOBEX TI PC Winchester Controller
- $350
- XEBEC 1410 Controller
- $295
- Apple Host (Xebec)
- $125
- S-100 Host (Xebec)
- $150
- Multibus Host (Xebec)
- $195
- Software
- $35
- Xerox & Bigbord Host (Epic)
- $95
- Western Digital 1002-05HDC
- $295

**POWER SUPPLIES**
- IBM-PC Replacement 135 Watt
- $149
- Bosch XLS1-563OR
- 81 Watts Maximum
- $39

**5 1/4" HALF HIGH DRIVES**
- For IBM PC, TI PC, Xerox 820, and others
- TEC or Panasonic Drives
- (8A-455 Compatible)
- $119 NEW
- Case & P/S for (2) 5 1/4" Half High Drives.

**8X DRIVE SUB-SYSTEM**
- Includes: (2) FDD - 200-8
- DSDD Drives, Vertical
- Case and Power Supply
- $629 NEW

**XEROX 820-I & II**
- Disk Drives Cables 5 1/4"............$20
- Disk Drive Cable 8"..................$30
- 8/5 Multiplexer Box..................$149
- (Run 8" & 5 1/4" drives simultaneously)

**TERMS:** Master Card, Visa, American Express add 3% service charge. Allow 10 days for checks, $4.00 Minimum UPS charge, $1.85 for COD, Texas Residents 6 1/8% Sales Tax.

84 Micro Cornucopia, Number 25, August-September 1985
I need to port software from micro to mainframe and that leaves out Turbo Pascal, unless Mr. Kahn has forthcoming versions of IBM 370, CDC, UNIVAC, and VAX. FORTRAN 77, however, is available for everything but the Z80. Many F77 features are available in RATFOR, a FORTRAN preprocessor.

RATFOR, Where Art Thou?
Where does one get RATFOR (not to mention FORTRAN) for the Z80? It turns out that The Software Toolworks has RATFOR (rational FORTRAN) for a mere $50 (a real buy), and Microsoft has a FORTRAN for the Z80 (though it's not $50). In the following piece, I'll cover a few ideas for extending this RATFOR to work with FORTRAN 80.

RATFOR, What For?
Nothing can be done about the lack of CHARACTER as a native datatype—but you can get IF-THEN-ELSE, DO-ENDDO (ala VAX), and, with a minor mod or two, regular DO loops (the ST RATFOR package specifically states that normal DO loops are verboten).

The Software Toolworks package includes a nice reference guide, but the definitive work on RATFOR is “Software Tools” by Kernighan and Plauger. It is a classic.

Editor’s note: There are two versions of “Software Tools.” The original one has examples written in RATFOR. In the later one the examples were transliterated into Pascal. For those of you who aren’t familiar with RATFOR, a couple of comments. At first glance, RATFOR code looks a lot like C. In fact, it is somewhat a cross between C and Pascal and is quite easy to learn if you are familiar with either language.

If I had to write code for a FORTRAN machine, I would definitely write in RATFOR so I could see what my code was doing. Then I’d let my system translate the RATFOR source into FORTRAN. The one disadvantage of RATFOR environment is that the translator’s output must still be compiled before you can test a program.

Following is Figure 1. The resulting output will look extremely funny, but it will work.

```
"Conventional" RATFOR uses {} (like C) to delimit things like
DO I=1, N;{stuff more stuff}

Adding define(ENDDO,}) to the list of standard "defines" (primitive MACRO'S) allows us to write
DO I=1, N;{stuff more stuff ENDDO

Just like a VAX (except for ;, which one globally deletes on port up)

Adding
define(THEN,{} define(ELSE,} else I)
define(ENDIF,})

makes
IF (condition) THEN stuff ELSE more stuff ENDFIF

Which is just like standard F77.

The only fly left is the SW RAT package to insist that one NOT use
DO 100 J=1, 199 stuff
100 CONTINUE

To allow this, we must modify the RATFOR preprocessor itself; since the source is provided (the original authors placed it in PD, modified versions may not be). The offending subroutine is called "docode" if it is made to read
```

---

Figure 1 - DO Loop modification

```
# decode - generate code for beginning of do
# MODIFIED to allow standard FORTRAN do if desired

subroutine docode(lab)
  integer labgen
  integer lab
  character lexstr(MAXTOK)
  character dostr(4)
data dost/LETD,LETO,BLANK,EOS/
call outtab  # space to start of line
call outstr(dostr)  # write "do"
call outch(BLANK)  # =
lab = labgen(2)  # make some labels
junk = gettok(lexstr,MAXTOK)  # for normal do logic
call outch(lexstr)  # ratfor functions
character junk  # temp store
# string dostr "do"
character dostr(4)
call outstr(dostr)  # write "do"
call outch(BLANK)  # =
lab = labgen(2)  # make some labels
junk = gettok(lexstr,MAXTOK)  # at next token
call pstr(lexstr)  # ... put it back
  # if normal a FORTRAN do
  if (type(lexstr(1)) = LETTER) {
    call eatup
    call outdon  # just put it out
  }
call outnum(lab)  #
call outch(BLANK)  # regular logic
call eatup  #
call outdon
return
end
```
On Your Own
By David Thompson

I intended the following to be part of the editorial, so it's mostly opinion. But it is also too important to languish in the editorial (in my opinion anyway).

This is a discussion of freeware or "user supported software," and I have moved it into this column. It's a very significant option for those of you who are trying to market software.

Freeware?

There is lots of controversy about freeware. First of all, what is it? Most people define it as publicly distributed, copyrighted software which contains a plea (or pitch) for money.

In other words, national public domain organizations, local user groups, friends, and bulletin boards are the distributors. Many people are exposed to the programs, and those who find them valuable are encouraged to become licensed (paid-up) owners.

In one sense, there isn't too much difference between freeware and commercial software. People pay up front for commercial packages and then distribute them (despite threats of dire consequences) to their friends. Eventually, many users of this kind of "freeware" pay up, so they have licensed copies. (This usually happens when a newer, fancier version hits the streets and they buy a copy to share with their original benefactor.)

In the legitimate freeware system the original copies are free, but end users are encouraged to buy a license. Purchase of a license usually entitles the user to one or more of the following:

1. A free upgrade to a more powerful version.
2. A more complete or printed manual.
3. Additional utilities or sample files.
4. Source.
5. Notification of new versions or fixes.
6. Support via phone, mail, or the author's bulletin board.

Crippled Software

Commercial software houses have been releasing crippled versions of their programs to get people hooked on their products. Often the only difference between these cheap or free copies and their expensive brethren is a bit of code that limits you to: 200 entries (an accounting package), or 300 baud (a modem package), or a 5K source file (an assembler or compiler). You pay the full license fee to get the full-blown package (or to learn how to turn off the stifle).

Now we're seeing the same trips from freeware houses. They are releasing crippled copies to the public domain, and once people have invested considerable time learning how to use the program, or entering data, bam! Pay up or else!

PC-Calc

Some of the freeware houses are releasing partial documentation to encourage users to buy a manual. One that comes to mind is PC-Calc. The documentation with their spreadsheet tells you just enough to get started. It even tells you where information has been left out.

Then they threaten an aggressive suit (much worse than double breasted wool) against anyone who adds the missing pieces to their documentation.

Actually, they would probably have trouble winning a case against someone who sat down and through trial and error figured out all the commands. As long as he hadn't seen their printed manual (and didn't use their words or style), they wouldn't have copyright protection.

They came down with a really heavy trip (sounded a lot like a chain letter) that tried to make me feel like a scum for not sending them $75 even though I haven't used their spreadsheet (and don't plan to). They even threatened me with unknown (but terrible) things if I removed their threats.

What a bummer. I was sorry I saw the package.

Another package took a much friendlier approach. Not only did it put me on my honor (and of course I'm as honorable as anyone I know), but they told me I could send whatever I wished. I might be willing to try that package and would send money if I found it valuable.

What I'm finding is that freeware is not much different than commercial software. There are friendly folks, and there are heavy-handed ripoffs.

Mixed Feelings

I have mixed feelings about freeware. In one sense people are taking advantage of the public domain distribution channels for their own profit. A lot of dedicated people are volunteering their time to put together quality public domain disks (including writing and modifying software). (A significant percentage of the disks we ship are in exchange for software donations and articles.)

Steve Leon, head of SIG/M and columnist in Micro C, refuses to put freeware on SIG/M disks. He doesn't want to donate his time (and his organization's resources) to making profits for commercial software writers. This is no doubt one reason why I haven't seen any freeware aimed at CP/M 80. (See Steve's column in this issue.)

Meanwhile, PC-Blue, the MS-DOS public domain distributor that is closely associated with SIG/M, has lots of freeware on its disks. It appears that freeware makes up more than half the new public domain disks for the PC.

On The Other Hand

In most cases, the sources of freeware are individual hackers trying to support themselves writing commercial (or almost commercial) quality software. There is no other way, that I know of, for these people to get their software into the market place.

The top promotional agencies say it takes over $1,000,000 to launch a new software product. This is just for the advertising and the parties. (I'm not sure who the parties are for -- maybe for the promotional agencies.)

A few of the freeware programs have generated over $100,000 in li-
license fees, which would make any author very happy. However, a large software house couldn't live on that kind of return if it had spent a million big ones for starters.

So where do I stand? I'm still dancing around a bit (and I'm playing the tune by ear), but from the authors I will not accept:

Heavy Trips
This means no chain letters, no threats of dire consequences, no basting. We all should be able to feel good about ourselves. If we receive something that was released for copying and passing around, then we shouldn't feel guilty about using it.

Crippled Software
This is a little bit difficult. I have no problem with someone coming out with a better, fancier version to encourage people to upgrade from the free version, but the free version had better be complete and usable. Forever.

Now For The Users:
You have some responsibility in this, too. There is a lot of freeware out there that is feature-for-feature competitive with commercial software costing 10 or 20 times as much. If you and I support these folks, there'll be a steady stream of inexpensive software out there (like Borland is doing in the commercial sector).

We're helping ourselves when we support these writers because over the last 12 months a lot of software houses have died or been swallowed up by the large ones. In one year, the number of software publishers that are actively advertising their products has dropped from 4200 to 400.

We could wind up with a few large software houses dominating the commercial scene and charging whatever they wish. Freeware could keep that from happening by making it easy for individuals to market their software without having to sell their souls (or their ideas) to venture capitalists.

Quite a few people have asked me how they can sell their programs inexpensively. There are a number of reasons why they can't do that in the commercial market (unless they are writing something that will sell millions of copies like Borland is doing).

1. Advertising is expensive. Unless you have a very special audience that you can reach cheaply, you will pay a lot for space and production to reach a lot of people who aren't interested in your product.

2. Dealers and distributors would rather make 60 percent of $400 than 60 percent of $40. And with display space at a premium they are not likely to carry inexpensive packages.

3. Software support is very time consuming. Even if your package is perfect (unlike Thorne) you'll spend a good deal of time coaching new users. ("You sent me a defective program. It won't boot on my system.")

Conclusion
So, for all these reasons, I think freeware is worthwhile. Steve Leon has a valid point. A lot of people donate their time to gather and distribute public domain software.

On the other hand, they are contributing to a good cause when they help the public get even better software and they help programmers reap a real benefit from their efforts.

One last thought. Freeware should meet the same standards that we expect from commercial software, from the point of view of usefulness, freedom from bugs, and documentation (on disk). Anything less constitutes abuse of those who distribute the software and those who are asked to pay for it.

[TO BE CONTINUED]
Kaypro Power Connectors & Co-Power

I called you after I installed a Pro-8 and found that my disk drives began turning themselves on at strange times. You were right that neither the Pro-8 nor the decoder board was the problem.

As you suggested, I tried replacing the PIOs, 74S04s, power supply, and virtually every other replaceable chip. Nothing helped. However, I accidentally discovered that the +5V connector on the main board (from the power supply) was dirty, causing a slow drift from 4.9V to 3.7V. At 3.7V my Mitsubishi quads got the message to start spinning. By that time everything was so flaky that I would have to power down for 10 to 20 minutes before I could run at all. All I had to do to fix the problem was clean the connections.

My Kaypro II is one of the originals (serial # in the 1700s). Despite the new higher rated power supply, the screen still twitches. I wonder if the connectors... Also, I have a tip for Co-Power users. If SmartKey, QuickKey, Fkey, etc. don't run when you have the Co-Power RAM configured as the A: drive, try the following:

1. Enter "RAMDISK A,Y" (to bring up the RAM disk as A: and initialize the directory).
2. Press the reset button.
3. Call up Smartkey, QuickKey, or whatever.
4. Enter "RAMDISK A,N" so the RAM disk locates itself below the Smartkey, etc. and both run without problems. (My 13-year-old son figured this out.)

Ron Rock
P.O. Box 211
Flossmoor IL 60422

Kaypro Speed-up Compatibility

Since I bought my trusty Kaypro II in January 1983, I have maintained a "hands-out" attitude ("If it ain’t broke, don’t fix it'’), but I am fascinated with the thought that it might run faster. So, I’m going to try a speedup, as well as your improved character set.

My main concern is that PluPerfect Writer, Backgrounder, and CP/M 2.2E may not operate properly at the new speed. I use them heavily, and have read conflicting reports on the success of the speed-up with them. Can you offer the definitive answer?

Also, please try not to ignore the older Kaypro owners with "Perfect" software and older hardware—we’re still out here!

Jim Chernoff
1351 Royal Way #19
San Luis Obispo CA 93401

Editor’s note:

I know you are out there. In fact, the older IIs and 4s are the most dependable systems that Kaypro has manufactured (occasionally the drives, data separators, and power supplies give trouble).

There is no reason that the speedup should affect the programs you mentioned. Your disk formatter/copier will probably not work at 5MHz, and games get too fast for mortals (though not too fast for 10-year-olds), so you should definitely add the 2.5 to 5MHz switch.

Smart Answering Machine

I have considered buying an answering machine, but many of my telephone counterparts, like my Dad, would rather hang up than talk to a recorder (can’t say I blame them). On top of that, a growing number of people who would like to talk to me have computers and modems. So how about an electronic mail answering machine? (Great idea for some ambitious entrepreneur.)

Rig the system to sense the number that is calling before answering. If it’s Dad on the other line, the system doesn’t answer.

Upon answering (it’s not Dad), the system then announces that I’m not home and allows the caller to leave a message either by voice or by a RBBS style hookup. If you want to get classy, digitize the voice message and store it on the disk, removing the requirement for a cassette.

This has to be a small, fairly simple machine that my Aunt Mary can hook up and operate. Seems to me the technology exists, but just needs to be adapted to the application. If you fully integrate it with the phone, as some companies are starting to do with dumb terminals, you might get a hot consumer item.

Rex A. Buddenberg
Naval Postgraduate School
SMC #1309
Monterey CA 93943

Editor’s note:

Hi Rex, interesting idea. I’m not sure how the answering device could know the number of the calling phone (although the FBI does it on TV two or three times a week). If that were possible, they wouldn’t need call back procedures to provide security for the big mainframes.

If the caller had a special tone generator (blue box, anyone?) into which he could enter a code, then the system could decide how to act based on the code received. However, before the receiving system could hear the code it would have to "pick up the phone." Any other ideas, anyone?

Expanding RAM On The SWP

I have a question about expanding the RAM on the SWP co-processor board. My board, installed in a Kaypro 4-84, is the 256K version, and I’d like to expand it to 512K or higher, if possible. I’ve read about the expanded version offered by SWP, and replacing my present board is...
probably the best way to go. It’s also probably the most expensive considering our present US/CAN exchange rate. Instead, I would like to buy the parts up here and install the board myself, thus getting my feet wet in making modifications to my Kaypro.

I’ve heard you can replace RAM chips with ones of higher value. A friend told me he did this to an IBM and had to install another DIP switch. A second friend told me an additional ROM chip was needed.

Obviously, there’s a difference of opinion which I hope the readers of Micro C will straighten out for me. If there’s anyone out there who would like to help a novice computertist get started, I’d sure appreciate the help.

Jim Dalgliesh
79 Athlone Drive
Winnipeg Manitoba Canada R3J 3K9

Editor’s note:
Contact SWP and see if they would sell you the non-memory parts and software you’d need. Otherwise, if you are just using your SWP board as a RAM drive you might want to check into MicroSphere’s unpopulated RAM disk. It goes to 1 Meg if you fully stuff it.

Simple System Patch
Could you publish a simple operating system patch so the delete key would duplicate the function of the backspace key? This has got to be one of the most frustrating problems for beginners, but I guess most of us tend to forget about it after a while.

Seems to me it should only involve a single hex digit, installed with the DDT S command. All we need to figure out is where that hex digit, and where it should go. Installation could use SYSGEN and a temporary file like the submit files on your K-22 ZCPR disk.

Leon H. Braskamp
1459 Via Cortez
Placentia CA 92670

Programmers’ Blues
I work for a consumer software publisher, and in case you’re not aware, consumer software publishing and distribution is in shambles. Inventories have remained high, drying up new orders for the publishers. Add to this the fact that now over 2000 software companies are competing for about 100 spots on the retail shelves, and you have quite an interesting brew. This has resulted in the falling prices that we’ve all seen, and the incredible failure rate of software publishers.

I wouldn’t feel too bad for these guys, though. They don’t really treat us programmers all that well. We’re blessed with royalties of 5 to 15 percent payable whenever cash flow permits, usually at least six months after a completed program is delivered (it can take that long just to sell into and collect from distributors and retailers). And we pay for any marketing, management, or personnel problems with lost sales and lost royalties.

Therefore, I’ve decided to start a programmer-owned software publishing company, and I’d like to invite any interested Micro C readers to contact me. We’ll publish our own software and distribute direct. Each of us will have the buying power and advertising clout of a publisher, but will retain control of our products. This type of arrangement means that a quality program can be sold at a very attractive price, while still providing a reasonable income for the programmer. We might even have a little fun.

Michael Fitzgerald
1924 N. 6th Street
Concord CA 94519

Editor’s note:
Good idea, Michael. I can see some problems such as documentation, support (phone type), software quality control (so a few marginal products don’t ruin the company for the good products), sharing advertising costs, etc. However, it could still work.

Rah, Rah, Rivendell
I recently purchased an expansion board from Rivendell Audiocomp in Poyway, CA and am now in the process of building it.

The Theory of Operations Manual is excellent. The Assembly Manual is not quite of Heathkit quality, but is more than adequate to build and debug the board. Rivendell is to be complimented on an excellent job.

Robert B. Base
7600 France Avenue South
Edina MN 55435

HSC Co-processor Compatibility
My Kaypro II has recently become a Kaypro 8 thanks to your Pro-8 ROM. I used TEAC 55s and traded in my old Tandons. I also added an HSC CO16-1668 co-processor, and both modifications are compatible. I thought that readers might like to know this because apparently the SWP 8088 co-processor will not operate without at least drive A: being an original Kaypro part.

Also, the review of the HSC 1668 in Issue 20 must have been of an older model because HSC is now shipping cross compilers and sources as part of the original package.

Now for the questions. I’ve begun using the Waltz Lisp interpreter and would like to hear from anyone out there who has worked with it or another Z80 Lisp. I am interested in programs (for writing courses) that ask questions, save the answers, and then allow the student to edit and rearrange those answers.

Second, I have an old Timex (Z80 based) computer with 16K of RAM which I am sure can, somehow, be turned into a print spooler for the Kaypro, but I have no idea how to do it. If anyone has any suggestions about what to do with an old Timex, save using it as a high tech door stop, I would be glad to hear about it.

Finally, now that I have this HSC processor, I figure there is some way its ROM can be fooled into becoming bank switched memory for the Z80. You have run articles in the past about bank switching on single board computers, but I have seen none for the Kaypro.

Robert Royar
1628 South Third Street
Louisville KY 40208
Enabling Shift Keys

I wanted a spare keyboard with lots of keys, so I ordered the $40 job advertised by Cascade Electronics. It is a used Micro Switch board in a good quality case with Hall-effect switches and should last forever. There is just one problem with it. You select upper or lower case letters with a jumper requiring removal of the motherboard. I cobbled up the circuit shown below to enable UC/LC with the shift keys.

This keyboard requires nearly one amp of +5 volts, whereas keyboards with mechanical switches need only 100 milliamps or so.

Turner F. Caldwell
PO Box 116
Burgess VA 22432

Bringing Up Xerox To 4MHz

One of the Letters To The Editor, Issue 3, Dec. 1981, tells how to bring a BB up to 4MHz. I've now modified two Xerox 820-1s, and they both worked with no difficulty the first time I powered them up. Following is my little contribution to your fine publication. It is merely a Xerox numbering of the original article, but I hope it will help someone as much as that letter helped me.

1. Change the crystal from 20MHz to 16MHz.
2. Remove and discard U12.
3. Change U9 from 74LS04 to 7404.
5. Cut the trace from U11 pins 4 and 5.
6. Connect U11 pin 3 to the trace from U11 pin 4.
7. Connect U11 pin 4 to the trace from U11 pin 5.
8. Run a jumper from U10 pin 5 to U12 pin 8.
9. Run a jumper from U9 pin 3 to U10 pin 5.
10. Run a jumper from U10 pin 4 to U12 pin 9.

Incidentally, the best automatic disk drive shutoff unit I've run into (for drives with 110V motors) is CMR's ASU-2A. It does its job well (auto-turnoff of drives), can also control external 110 volt appliances, and can be controlled from an external device, too. Write to CMR, P.O. Box 7022, Alexandria VA 22307, for more info.

Bill Bretzius
1902 Ancilla Court
Alexandria VA 22307

Jumpering Toshiba Drives

I recently installed two new drives in my Kaypro. I chose Toshiba ND04D Model FDD8401-EK01, which are double-sided 48 tpi units. I read in issue #21 about a mod which would make the drive lights work the way they were always supposed to.

This mod was part of the 4-drive decoder circuit which I was not particularly interested in at that time, so I tried something on my own.

The Toshiba drives have a LED jumper option in addition to the HM/ HS head load option. The options are LI, which gives you a drive light with an in use signal, and LD, which gives you a light with a drive select signal. With the LI option jumpered, I got no light at all. With the LD option I got a drive light on the logged-in drive just as always. A third option was to jumper both the LI and the LD option, which would give a light whenever both a drive select signal and an in use signal were present.

The Kaypro was not set up with an in use signal, so I did the signal by jumpering from U81 pin 2 (motor on) to J6 pin 4 (in use). Actually you can trace U81 pin 2 to a solder pad right next to J6. That did it. Figure 1 shows the jumper configuration and the LI, LD, HM Table.

Ron Lokey
1020 East 12th
Colorado City TX 79512

Figure 1 - Toshiba drive jumpering options

| LI | LED comes on when In Use is set to 1. If LD is shorted also, LED comes on only when the drive is selected. |
| LD | LED comes on when Drive Select is set to 1. If LI is shorted also, LED comes on only when In Use is logic 1. |
| HM | Head loading is controlled with the Motor On signal. |
| HD | Head loading is controlled with the Drive Select signal. |

90

Micro Cornucopia, Number 25, August-September 1985
Tips On Articles In Issue 22
The Z80 to S-100 interface in Issue 22 was very interesting. I believe you were right about there being a mistake in the RD/ logic. The circuit as shown gates the data from the S-100 bus into the Z80 at the proper time, but the single board system RD/ is not disabled. I believe the circuit should be as shown below.

![Diagram of RD pin connections](image)

The statement about no modifications to the S-100 board is also misleading. The clock circuit must be disabled. If the S-100 card generates any signals from a higher frequency than the Z80 clock and divides down to the clock, it may be rather difficult to generate all the required signals. If you choose the S-100 board carefully you should be able to simply pull out the clock generation chip.

The sound generator for the STD bus was also a good article. It mentioned the WAIT/line and the conflict with the onboard wait state generator. The problem is caused because this line is supposed to be an open collector line, but the Caltex mod (it may now be a regular OR 97006). The problem shows up at track 2CH and high tracks where bit density is high. Disks formatted with the new value will be clean, but disks formatted with the old value resistor will continue to give trouble.

The circuit is very much like the one in Micro C Issue 11, page 5. Users of this system with a similar problem might try setting the pot to 2.2 or 2.7k and reformat their disks before trying to use them.

Joseph Malik
221 Bradley Dr.
Wilmington NC 28403

Cold Solder Joints
My Kaypro 2-83 crashed and wouldn't read disks when warm. It did OK with the top off, most of the time, but a fan didn't help. The problem turned out to be a cold solder joint on the power supply board where the plug for the power cable is located. The power supply had several of these bad joints, causing the drives to occasionally get 2 volts instead of 5. Not an obvious problem, but one that's easy to check out and fix!

Charles H. Kuttner, M.D.
213 Water Ave. NW, Suite 300
Albany OR 97321

MITE Software Hints
I own a Kaypro 2-84 and was impressed with the communications package I received from Mycroft Labs to upgrade the MITE software.

There are two installation options (F and G on the Portable Computer installation menu) available for the 2-84. One requires that either U35 (a Z80A PIO) be installed or pins 22 and 24 of U35's socket be jumpered. The other option uses the Z80's interrupt system to service the incoming data stream. The other option requires no modifications, but uses a polling scheme and suffers from problems in synchronizing the simultaneous display and printing of incoming data.

If one elects the option of installing U35 (necessary if the Real Time Clock is to be added) then the CP/M (version 2.2G) supplied with the 2-84 needs to be modified to initialize the PIO. Examination of the operating system indicates that a functionally useless subroutine at F87F through F88B is called during a cold boot. Replacing the code at these locations with:

```
3E 17 D3 22 D3 23 ED 00 00 00 00 09
```

will initialize the PIO when a cold boot occurs and allow MITE to work correctly.

Additionally, the code at F6BA through F6D0 initializes the Z80A SIOs that handle the printer and modem serial ports. All initial values are stored in a table extending from F647 to F660. The boot occurs and allow MITE to work correctly.

```
C9
```

Changes to any of the above locations can be made by using the following procedure:

First with a system disk in drive A save the system image by entering:

```
SYSGEN
A
<return>
<return>
```

```
SAVE 40 CPMD.COM
```

To edit the CP/M image enter:

```
DDT CPMD.COM
```

Make the required changes using DDT commands — CP/M starts at 0980 (i.e. E000=0980)

```
Enter: "C"
```

To install the edited image enter:

```
SYSGEN
A
<return>
<return>
```

Jim Hays
2015 43rd Ave. SW
Seattle WA 98116
FREE
CATALOG

MICRO CORNUCOPIA'S

KAYPRO DISKS
GUIDE & CATALOG

KeyPro Disk K1
Modem software.

KeyPro Disk K2
Utilities
Really clothes of spiffy little (end big) programs to help you get full use of your KayPro.

ASSEMBLER FOR 8080 AND A.

DASM: Really oodles of spiffy

UNERA.COM: Simply enter

ADVANCED SCREEN/PRINT UTILITIES:

UNERA.COM: Simply enter

ADVANCED SCREEN/PRINT UTILITIES:

DON'T MISS THIS DISK SERIES!

DISK 1

KeyPro Disk K13
Source of fig-FORTH

KeyPro Disk K14
Smartmodem Program

KeyPro Disk K15
Hard Disk Utilities

KeyPro Disk K16
Pascal Compiler

KeyPro Disk K17
Z80 Tools

KeyPro Disk K18
System Diagnosis

Just as we finished editing the routines on this disk, we received a copy of KeyPro's diagnostic disk. The memory test and drive exercise routines on this disk are more powerful than KeyPro's versions. (Plus, it's only $12) Setup for KeyPro II & 4.

KeyPro Disk K19
Prowriter Graphics

KeyPro Disk K20
Color Graphics Routines

KeyPro Disk K21
SBASIC Routines & Screen Dump

SBASIC: Finally a disk of SBASIC software. There are some good examples of structured programming on this disk (including one program written both ways so you can see the difference).

SCREEN DUMP: This is a screen dump for all KeyPro's new and old. You can buy a similar package elsewhere for $60.

KeyPro Disk K22
ZCPR (Again)

This disk is filled with ZCPR files. You get ZCPR for the KeyPro II, KeyPro 4, and the KeyPro 10. This version is fixed so that you can pass control characters from the keyboard to the printer, and you can choose to have it recognize the semi-colon for drive select (as well as the colon). So you can enter "B" or "C" to select drive B. Super neat!

ZCPR, for those of you who don't know, makes CP/M a lot friendlier. It searches drive A for any COM file it doesn't find on the current drive, the TYPE command scans text 24 lines at a time, and a new LIST command outputs a file to the printer.

KeyPro Disk K23
Fast Terminal Software & New BYE

KeyPro Disk K24
MBASIC Games & Keyboard Translator

We sifted through many, many games before coming up with these gems. All will work on any KeyPro and all come in MBASIC source.

USOPEN shows you the fairway on the screen. You select the club and direction for each stroke. After you reach the green the display shifts to show details of the green and flag. For one to four players.

DUCK is an offshoot of aliens (parody the pun). Hunter tries to shot down the ducks while ducks try to bomb the hunter. (Much fairer than real life)

CASTLE is an adventure in which you select your attributes (strength, dexterity, and Intelligence), and you get to purchase arms and protection. Great documentation and a very interesting game.

KSTROKES is a keyboard translator similar to Smartkey. Bill Forbes did an excellent job creating this program. You can create and save translation files on disk. The program even includes a table which generates WordStar commands from the KeyPro's keypad. You can define 8 keystrokes at up to 63 characters each.

KeyPro Disk K25
Z80 Macro Assembler

KeyPro Disk K26
EPROM Programmer & Character Editor

KeyPro Disk K27
Typing Tutor

A complete typing tutor for beginners and experts. Written In Australia, it comes complete with source. This was customized for KeyPro II, 4 and 10 and by Barry Cole of WLAKUG.

The documentation says you can learn to touch type in 8 hours (probably a little longer for mortals).

KeyPro Disk K28
Modem 730

KeyPro Disk K29
Turbo Pascal Games 1 With Source

KeyPro Disk K30
Turbo Pascal Games 2 With Source

KeyPro Disk K31
Turbo Bulletin Board Complete Bulletin Board Package for only $12

KeyPro Disk K32
Forty-83 Much Faster Fort

KeyPro Disk K33
Super Utilities disk containing:

NewSweep, NumLock, Superzap, & VDO-XP.

KeyPro Disk K34
Five games plus source (mostly Turbo); includes improved Othello and Blitz.

KeyPro Disk K35
Small C Compiler & Source - Ver 2.1;
(Requires Microsoft's M80)

KeyPro Disk K36
Small C Library of 100 Functions.

Write or call for a free catalog
[also included with each order from]

MICRO CORNUCOPIA
P.O. Box 223
Bend, OR 97709
(503) 382-5060
9-5 PST Monday-Friday

DISK $12.00 ea. postage paid

KAYPRO USERS DISKS
for Keypro II, 4 and 10
UPGRADES

Schematic Packages

Finally, a complete schematic for your portable Kaypro, logically laid out on a single 24" by 36" sheet, plus a very complete illustrated Theory of Operation that’s keyed to the schematic. You’ll find detailed information on your processor board that’s available nowhere else.

For instance, all packages contain serial and parallel port details and programming examples as well as complete coverage of the processor, clock, I/O, and disk controller information that is not even available in Kaypro’s own Dealer Service Manual.

Kaypro Schematic Packages

Kaypro II & 4 (pre-84) .................. 120
Kaypro 10 (pre-84) .................. 120
Kaypro 84 series (II & IV) .......... 120

All prices include postage.

ROM's from Micro Comucopia

Pro-8 Version 2 for Kaypro

Guess what we've just upgraded! The venerable Pro-8 now has a version 2 and it's really neat! The new features include:

1. Select slow or fast stepping rate for each drive.
2. Select dump with selectable dump character.
3. ROM that automatically figures out what type of drive you are using.
4. No more delays when using both single and double sided disks.
5. Of course, you get all the original Pro-8 features such as: selectable cursor filling, or ignore, ignores null and your choice of 64 drives of the 121B, 390K, and 784K variety. (Use of 3 or 4 drives requires drive decoder.)

Installation requires no cuts or jumpers. The ROM simple plugs into a Kaypro II with the original I/O processor board then you must do the II Kaypro IV for II-83 with a Kaypro 4 processor board. See these 822 for details. Your Kaypro I or has the original I/O board if the monitor ROM is 120K if the paper stock to its tip)

If you already have a Pro-8, you can upgrade to version 2 for half price.

Pro-884 Monitor

The long awaited PRO-ROM for the Kaypro 2-84 and 4-84 is ready! You have not saved your nickels and dimes in vain!

The PRO-ROM does everything your old Kaypro ROM does plus:

1. Gives you 784K bytes of storage per disk (with quad-density 80-
   track double-sided drives)
2. Includes any combination of Kaypro II, IV, or quad-density drives.
3. Comes with complete printed instructions on ROM installation and
debugging.
4. Runs up to four drives with the Plus-4 Decoder Board.

The ROM that does everything but wash the dishes is here! There's so much stuff in this ROM, we didn't think it would fit!

The PRO-884 ROM does everything the PRO-884 does PLUS:

1. Emulates a DEC VT-52 or Heath H-19 video terminal (both the
display and keyboard).
2. Does FASTER screen output.
3. Has ZCPR in ROM, eliminating the need for a system disk in drive
   A, during a warm boot.
4. Sends the screen to your printer at the touch of a single (user
definable) key.
5. Displays the program you're running.
6. Displays the time on the Kaypro's status line (Kaypro 4-84 only).
7. Inserts the time and date into text while you're running your

Prices:

- Pro-8 Version 2 Pkg. .................. 49.95
- Pro-884 Pkg. .......................... 99.95
- Pro-884 Max Pkg. .................. 79.95

SPECIAL PRO-884 NOTE:

The PRO-884s are sensitive to the version of CP/M you are running.

1. Neither the PRO-884 nor the PRO-884 Max will run on
   CP/M 2.2U. However, if you have a CP/M 2.2F or
   2.2G system disk (your dealer should have a copy) you
   should be able to run our 884 monitors. (Don't try to boot
   F or G before you change monitors.)

2. There are two distinct versions of CP/M 2.2G. Only the
   Pro-884 Max is sensitive to the version of 2.2G you have
   - it's the ZCPR in ROM that's the problem. (If you have CP/M 2.2F
   then you have a Normal CP/M.) So, before ordering the Max,
   boot your original system disk and read the sign-on. If it's
   CP/M 2.2G then we need to know whether it is the high (normal)
   version or the low (minus) version.

   To determine your G version (you'll become a G Whiz!):
   A. DDT or
   B. LS or
   C. 16K drivelogic

   The first line of the response will be a JMP D000 or a JMP D000. The
   JMP D000 means that you have a low (minus) version, and the JMP D000
   means that it's a normal version. When you order your Pro-884 Max, be
   sure to specify whether you want the normal Max or the minus Max. Other-
   wise, we'll just guess that you need the normal Max.

PLUS-4 Decoder Board

With this nifty little plug-in board, your Pro-8 ROM can access up to four 5¼

You just plug a four-drive 34-pin cable into this board and you can add up to two

Now you can run any mix of 121B, 390K, and

You can run your original drives as A and B then add 380K or

You can even run your full-wides inside your original Kaypro,

The Plus-4 Decoder Board for only 9.50

Watch for 4-84 and 10-84 compatible ROMs coming soon.

UPGRADE YOUR KAYPRO!
The Perfect software packages have always had their share of problems. For instance, they got no awards for speed, and creating a new printer interface led more than a few to swear off hard copy altogether. In addition, Perfect’s software support group got about as many calls as their counterparts at Ashton-Tate. (Just a guess—but I wouldn’t be surprised if the original Perfect writer and formatter weren’t copies of Mince and Scribble, an editor and formatter written in BDS C and sold for many years within the BDS C users group.)

When the Perfect package was dropped by Kaypro (they got tired of all the calls, too), the Perfect people just about disappeared (how many perfect people do you know?). The software and the name were bought up by Thorne, which rewrote some of the code and then rereleased the programs, aiming their ads at current Perfect users.

But Thorne added copy protection, some of the packages ran slower than the originals, and the new Perfect filer couldn’t read files created by the older version. Pow, pow, pow! They did a really bad job with the copy protection. We got some calls from folks who had found that the program wouldn’t run with some systems (including ones which contained the Pro-8). They went back to their original ROMs to make the packages run, but after finding the other problems, they were reinstalling the Pro-8s and returning the new Perfect packages. Pow!

If you’re going to do copy protection, you do your darndest to make sure the software is going to run when it’s supposed to. The worst thing you can do is create a program that isn’t robust enough to work dependably. Pow!

Anyway, I just got word that Profiles will no longer be carrying advertising for Thorne. The staff at the magazine is getting tired of complaints about the new Perfect. Kabooom!!

It would really be unfortunate if these blunders meant that the early Perfect series would no longer be available. A lot of people wouldn’t trade their original Perfect writer for three copies of WordStar and a Mickey Mantle baseball card. Heck, if Thorne sold the original writer, calc, and filer for $50 a set (without support) they’d probably do just fine.

Kaypro 1

Speaking of the original Perfect package, I understand that Kaypro is again shipping it with the Kaypro 1. Yep, the Kaypro 1 is a Kaypro 4-83 with two double-sided drives and the original Perfect package for $995. It’s amazing what happens when a company has a warehouse full of spare parts.

On the other hand, I’d rather see them ship systems out the front door at cheap prices than see them dump parts to the tin recyclers.

Using Canons From Scratch

We just got a report that a few of the 2/3 height Canon drives are eating disks (for brunch). According to one caller, a quick head load/unload cycle can scratch a track so badly that the disk can’t be reused.

He also mentioned that BG Micro has had only 40 drives returned out of 8000 units (not a bad average at all).

Keeping the head loaded all the time and then turning the drive motor off to minimize disk wear (what Tandon has been doing all along) sounds like the safest bet.

SOG

If you’ve been following Micro C for any length of time (over five minutes), you’ve become accustomed to hearing about SOG. Well, I don’t know what to say about it this time.

You’ll see this issue after the SOG (or during the SOG if you show up in Bend), but I haven’t the slightest idea how it turned out, since today is May 13.

I could, however, do some anticipatory journalism.

Anticipatory Journalism

Anticipatory journalists announced that Dewey had lost to Truman. They also announced: the Z800 (and they are still announcing it), MicroSoft’s Windows (but we can see right through that one), the Japanese computer invasion (actually, the Japanese are scared to death that Commodore will start marketing in Japan), and the demise of CP/M.

With that kind of hysterical precedence, there is no question that I must come up with some definitive anticipatory copy about the SOG. So here it is:

For those of you who missed the SOG, I’m happy to announce that the event went off without a hitch. The weather was clear with highs around 85 (unless you prefer it a little cooler) and lows in the mid 40s (that’s as high as our lows go).

Over 25,000 happy folks rafted down the McKenzie River. We set a new record when we stuffed 814 freaked programmers into a single raft (or was it 814 single programmers into a freaked raft). I understand that it was the first time a 6-ton load of rank beginners shot that stretch of river without a guide (he just didn’t fit in). Next year we’ll be rafting in the Queen Mary (another first).

Philippe Kahn announced that Borland’s secretly working on a BASIC interpreter for the Apple II. He said that after turning Z80s and 8088s into speed freaks he had a burning desire to see how slow a computer could go. Plus, he’s been fascinated by BASIC ever since he discovered mazes.

Integrand announced the hacker’s cabinet. This ordinary looking cabinet not only holds your system, drives, and power supply, but it also holds hundreds of tiny antennas. When you press the interference button on the front panel, the top ejects, implanting itself in your ceiling and employing the antennas, all of them connected to your system clock. No one within a radius of one mile can receive a usable FM, AM, or TV signal.

This little gem really shuts down Johnny Carson when you’re into more important things (like computing). Note: this cabinet has not yet received FCC approval.

The Ampro folks announced a big change for the little (board) company. It’s the Jumbo Board, and it’s a really
hush-hush operation. Funded by the Department of Defense, this system should be completely secure because it is constructed entirely out of (very) discrete components.

They were unable to display the system at the SOG because the main board was too large to fit on a flatcar. Assembly is supposed to be finished by 1990. Debugging will be expensive because of the long-distance phone charges between engineers working at opposite ends.

We had to cut SOG IV a bit short because of reports of a worldwide technology crisis. It appears that having that many hackers away from their terminals that long has been a major setback for mankind (and womankind).

Finally, if you believe all this, you’ve been reading the editorial too long.

Culture Corner Interrupts Serious Meeting

The first indication anyone had that George Morrow (you know, Morrow Computers) had arrived at the Sacramento Micro Users Group (SMUG) 10th anniversary celebration was a loud guffaw at the back of the hall.

While waiting to speak, he had made the mistake of picking up a copy of Micro C #23 and reading the Culture Corner (“Confusing At Catap­trophy Manor”).

As he started his talk, George held up the copy of Micro C and strongly suggested that everyone read page 82. He said the poke at Jerry Pournelle was “right on.”

Bill Godbout (of CompuPro) also read that Culture Corner, and during his talk added: “I’ll tell you how Jerry Pournelle got Ezekiel II.”

“Several years ago we sent a computer to Byte for review. It was a loaner, and they had it for 30 or 60 days and then sent it to Jerry Pournelle.

“One day, when the evaluation period had just about expired, someone called and said he had an evaluation machine from us and he didn’t want to return it, so how much would it cost to keep it.”

“Linda Barlow got the call, and it was one of her good days, so instead of telling him ten times the list price, she looked up her retail prices, told him, and bang, 10 days later here was a check in the mail. From Jerry Pournelle.

“Pournelle also asked for a review machine from Morrow, but no one in the Morrow organization recognized the name, so they said to hell with him.

“One time in Atlanta or San Francisco, George and I and Jerry Pournelle and a bunch of other people were having dinner together when Jerry asked George when he was going to send him (Pournelle) a Morrow. Jerry is a little hard of hearing, so George’s response was loud enough for the whole restaurant to hear.

“George said, ‘I’m not, and anyone in my organization who sends you one will be fired.’"

At this point, George Morrow spoke up from the audience. “Bill, I figured that if he bought one from you, he could buy one from me.”

Jerry Pournelle Gets Micro C Free

I was sitting there thinking smugly that I hadn’t given Jerry anything (very self-righteously). However, just weeks after I returned from Sacramento, I received a letter from Kevin Rhoads ordering (and paying for) a 1-year subscription to Micro C for Jerry Pournelle.

So now Dr. Pournelle has a free subscription to Micro C.

Contradictions

During his talk in Sacramento, Bill Godbout mentioned some common phrases that are self-contradictory. The formal term is, I believe, “oxymoron,” but the term makes no more sense than the phrases it describes (which is, perhaps, appropriate).

Some oxymorons to think about:

Civil servant
Athletic scholarship
Military intelligence
Fiduciary trust
Artificial intelligence
Postal service
Petty cash
Electrical engineer

If you come up with some of your own — especially computer related — keep them to yourself. We’re confused enough already.

David Thompson
A Sensible Oxymoron

FREE CATALOG AND SIGNAL PROCESSING BOOKLET

<table>
<thead>
<tr>
<th>CIRCUIT ANALYSIS</th>
<th>SIGNAL PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPP</td>
<td>$59.95</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ACNAP/DCNAP</td>
<td></td>
</tr>
<tr>
<td>ACNAP Version</td>
<td>$69.95</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>260</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Any Size Circuit</td>
<td></td>
</tr>
<tr>
<td>Input / Output</td>
<td></td>
</tr>
<tr>
<td>Impedances</td>
<td></td>
</tr>
<tr>
<td>Monte Carlo</td>
<td></td>
</tr>
<tr>
<td>Analysis Transient</td>
<td></td>
</tr>
<tr>
<td>Analysis Manipulation</td>
<td></td>
</tr>
<tr>
<td>Spectra Manipulation</td>
<td></td>
</tr>
<tr>
<td>Transfer Function Manipulation</td>
<td></td>
</tr>
<tr>
<td>Editing and Error Trapping</td>
<td></td>
</tr>
<tr>
<td>Free Format Input</td>
<td></td>
</tr>
<tr>
<td>ASCII and Binary Files</td>
<td></td>
</tr>
<tr>
<td>Fast Machine Code</td>
<td></td>
</tr>
<tr>
<td>Component Power</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GRAPH PRINTING</th>
<th>PLOT PRO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPP</td>
<td>$49.95</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ACNAP/DCNAP</td>
<td></td>
</tr>
<tr>
<td>PLOT PRO</td>
<td>$59.95</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Any Printer</td>
<td></td>
</tr>
<tr>
<td>Vertical/Horizontal</td>
<td></td>
</tr>
<tr>
<td>Screen Graphics</td>
<td></td>
</tr>
<tr>
<td>Plot Resolution</td>
<td></td>
</tr>
</tbody>
</table>

**BV Engineering**

Professional Software

2000 Business Way, Suite 207 • Riverside, CA 92501 • USA (714) 781-0252

Micro Cornucopia, Number 25, August-September 1985
The following folks are reaching you for only 20 cents per word. If you would like to reach the same audience, send your words and 20 cents for each to Micro Cornucopia.

Superb mailing list program stores and manages names and addresses that can be revised at any time. Its size is limited only by the disk storage available. The address labels may be code selected and printed in five different formats on your computer paper or on label rolls. In addition to the name and address fields, there are four more fields in each record for telephone number, date, and two amount fields if desired. At any time the entire roster may be printed out. For CP/M 2.2 based systems with two disk drives and time the entire roster may be printed out. For money order.

IBM PC/XT/AT/XTL 4P6. Only USA $19.95 postpaid check or money order. Offer expires August 31/85.


Public Domain Ug Software Rental: C/P/M Ug Vol 1-92 on 16 " floppy, 45, SIG/M Ug Vol 1-210 on 108 " floppy, $125. PICONET Vol 1-34 on 12 " floppy, $25. KUG (Charlottesville) 39 disks, $25. IBM-PC SIG 1-300 PC-DOS, $325, PC-Blue 1-110 MS-DOS, $125, directory disks $5. PP. Also avails on 90, 5.25 " disks. Rental is for 7 days after receipt with 3 more days grace for return. Credit cards accepted (preferred). Download, Xerox Group Software Automatic Update Service, $7.50 per 2 volume set. PP. 619-727-1015 24 hrs. 619-941-9425 info. 9-5. National Public Domain Software Center, 1633 Avohill, Vista, CA 92083. 8" drive Cleaning Kits - 12 cleaning disks and carrier jacket made by Datafile, regular price, $25.95. Close out price just $6.00 each plus $2.00 postage. Limit 1. P.D. Software, 1633 Avohill, Vista, CA 92083.


Xerox 820-3, complete working board with SWP DD disk controller card & software $170.00/ Rochech power supply & box (not required) & 2 SA801 drives $250.00/SA1403D hard disk controller card, SA1004 8" hard disk upgrade for 820-11 $400.00/PP/DPC Box 1217, Costa Mesa, CA 92626.

Two new universal data systems, 300/1200 model 212A modem. Cost $625. Sell both for $500 with manual & ASCOM, P.D.C., Box 1217, Costa Mesa, CA 92626.

Xerox 820-11, complete. Two 5.25" drives, manuals; 820-1 board with or separate. Gary L. Smith, (703) 759-9160, (703) 442-6312, 1040 Artemel Lane, Great Falls, VA 22066.

Big Board II System. In Tek 8500 series box with power supply, 2 Shugart 8" SSDD drives, keyboard, NO monitor (recommend Amdek 300). Runs good. CP/M 2.2, 4K documentation. Some software extras including user's disk 9, 14, 15, 650. J. Binkley, 1945 SW 131th, Beaverton, OR 90465, (503) 676-7720.

New bulletin board system in Montanal Feature: 300/1200 baud, software downloading (games, utilities, languages, database, others), classified ads, etc. For more information, call Curt at 406-728-1097. To join the Pony Express Bulletin Board, please send $25 annual subscription to Curt Forteux, Structured Data Systems, 211 W. Front, Missoula, MT 59802, or go online 406-728-1088.

$100 equipment, Integral 15 slot motherboard $65, Jade Double Disk controller (factory assembled) with documentation $590, 2SK static RAM board $50. Also Novation Cat 300 baud acoustic modem $60, CDC 9404 8" SSDD drive, 220v model with transformer for 117v $60. H. Ragg, Box 1462, Cynthiana, MD 20650.

CBASIC Version 23.8 original Digital Research disk with manual and 1 backup copy. 8" CP/M 3.740 format. Runs on Big Board 1. $52.00 includes postage. No pens to check please. Chris Beachy, Box 4645, APO, NY, 09755.

20 Megabyte 5.25" hard disk. New, Seagate ST500 $50. Limited quantities. B.W. Systems, P.O. Box 9791, Austin, TX 78766. (512) 255-8350.


Bob Bruner, 10230 N. 39th Lane, Phoenix, Az 85015, (602) 978-4952.

For Sale: Compupro CPU-Z 4mhz $125.00 6mhz $155.00. Cromenoco Tuart w/cables $125.00. PMH1-100 modem works to 600 baud $150.00. Integrang X/5 mainframe $150.00. Any two items less 10%. All tested & working. (219) 665-9945. Al Marshall, 408 Oakwood, Angola, IN 46703.

Xerox 820-1 boards, complete and tested - $95.00. Xerox ASCII keyboard with case and - $55.00. (313) 974-5409. P.O. Box 4268, Ann Arbor, MI 48105.

TBKUG/Data COM Network supports CP/M- 80, CP/M-86, Kaypro, S-100, IBM, and compatible users with public domain software via modem. We "drive" three on-line holding over 40mb of CP/M and PC/MS-DOS programs available 24 hours at either 300 or 1200 baud. We specialize in CP/M disk utilities, 2CRZ & 3, MEX, DOS utilities, Turbo Pascal, word processing, RCP/PM utilities, BBS software, library utilities, catalog utilities, and much more. The Florida systems will soon be merging into one large multi-user system with many incoming lines. The TBKUG has 63 specific user disk volumes for Kaypro owners and over 5,000 files available by mail. We sponsor a monthly magazine that distributes electronically by various national bulletin boards called the $/O Read Only. It has many product reviews, public domain user information, Turbo Pascal column, and other interesting topics for CP/M and DOS users. The TBKUG has been in existence for over two years and has more than 600 members worldwide. We have been imitated. D2R2 Here's numerous Profiles' articles and proudly support the CP/M users of the world with high quality software. An application may be downloaded via modem by calling (613) 337-3068, or send a SASE to: TBKUG/DataCOM Network, 14 Cypress Drive, Palm Harbor, Fl 33703. Annual dues are $30.00.

THE ULTIMATE SOFTWARE IS IN . . .

ORDER YOUR MICRO C T-SHIRT TODAY!

These user-friendly tan shirts are formatted in mahogany border with black enhanced mode design. The fully integrated system is compatible to size S, M, L and XL, for only $6.95 ppp. ($8.50 all foreign).

MICRO CORNUCOPIA

P.O. Box 223
Bend, Oregon 97709

96

Micro Cornucopia, Number 25, August-September 1985
Tidbits

By Gary Entsminger

Thanks for the prompt response to our Turbo Pascal programming contest. It’s going to be a lively competition, judging from the first entries. Code and documentation have been clear as snowmelt. (What a chilling thought.)

No doubt, Z80 romantics are already dreaming up uses for a 1 megabyte RAMdisk (one of the grand prizes). Bruce couldn’t wait (of course he isn’t eligible anyway, being a Micro C staffer). His Kaypro already has one of Microsphere’s 1 meg screamers along with four drives (an 8”, two 5” quads, and a 5” 48tpi) and 5MHz. He connected his RAMdisk to a separate power supply, so it stays loaded around the clock.

If you want to know more about the RAMdisk contact:

MicroSphere, Inc.
P.O. Box 1221
Bend OR 97709
503-388-1194

A Write Hand Person

It’s been a good week for CP/M. Poor Person Software has released a semi-Sidekick, with notepad, calendar, file and directory viewing, phonebook, and communication programs included. Perhaps more importantly, you can write your own programs to be called from Write Hand Man. Your programs need to be in assembly called from Write Hand Man. Your capes and you’re back to the file you were editing.

One flaw - Write Hand Man doesn’t refresh the screen. Your editor will have to take that responsibility. But that’s no big deal in WordStar or Vedit. Costs $49.95.

For more info:

Poor Person Software
3721 Starr King Circle
Palo Alto CA 94306
415-493-3735

Micro C Bulletin Board

Thanks for your support and suggestions. Lots of users have been patient as we work out the kinks in the system.

If you need a fullscreen file debugger, we’ve added EDFILE.COM in the NEW directory. EDFILE has a handy search function and lengthy documentation (in EDFILE.DOC). We don’t have source, but it’s useful as is.

Also Eric N. Skousen has contributed a Micro C index. It’ll be on the bulletin board and available on a user disk for those of you without modems. Thanks, Eric.

Keep calling: 503-382-7643.

Xerox 8” Adaptor

Merit Computer is selling a disk drive adaptor which enables the Xerox 820-II to run 5” or 8” drives.

It’s available in kit for $50, and assembled and tested for $70. Fits on the rear of your 820.

Contact:

Merit Computer
1658 Westport Rd.
Merritt Island FL 32952
305-452-4655

Surplus

Erac Co. in San Diego is advertising Kaypro mainboards, monitors, keyboards, and other miscellaneous parts.

8280 Clairemont Mesa Blvd.,
Suite 117
San Diego CA 92111
619-569-1864

Speed Tidbits

The Cray-2 has the largest internal memory in the world with 2 billion bytes. It’s speed is 1.2 billion FLOPS (floating point operations per second). It’s 6-10 times faster than the Cray-1 (you were worried?), and maybe 50,000 times faster than a Kaypro 484.

In terms of how fast we used to be — what took about a year (and 72 dozen pencils) in the early 1950s takes about a second now. So who needs speed?

Simulators are current bigtime speed users. Scientific and military researchers play the “what if” game and depend on oodles of details for accuracy. Engineers can create lifelike worlds with fast computers capable of high resolution graphics. In theory, our knowledge of the world increases as we add more and more details. To a computer, that means more data.

Two sci-fi novels I read recently address the subjects — simulation and speed — in rather offbeat fashion. If you like “serious” SCIENCE fiction, try Stanislaw Lem’s “The Chain Of Chance” and Philip Dick’s “Ubik.” They’re both short, thought-provoking, and depict highly-plausible futures of advanced human beings. Lem’s simulator uses a computer to piece together the “mystery,” and Dick’s characters, who have computing power of a sort, use themselves to solve the puzzle.

Both are in paperback, and have been around for years, so you should be able to find them in a library or used book store for $1 or so each.

New From Micro C

We’ve been working hard to improve our user disks, and have just finished Kaypro Disks 33, 34, 35, 36 and CP/M 86 Disks 7 and 8 (see Laine’s 86 World for details on these last two).

K33 is a utilities disk containing some very useful programs:

NSP207.COM — an excellent file utility written by Dave Rand.
NULU11.COM — a program for creating, manipulating, and extracting libraries.
SUPERZAP.COM — a full screen debugger.
VDO-KP.COM — a fast, mini-editor that’s small and easy to use.

All are well documented. (What’s this, a change in policy?)

K34 is a games disk (DBlick, Eliza, Germs, Othello, Cribbage). This inverse video version of Othello is the best public domain one I’ve seen. All the games come in Turbo Pascal
source, except Othello (written in FORTRAN), so besides having fun playing, you'll have examples of Turbo programming.

K35 and K36 are Fred Scacchitti's upgrade of the Small C compiler (version 2.1) and C library. He's incorporated Jim Hendrix's additions and added a few of his own tricks. 35 is the compiler and source (in a library), and 36 is a library of around 100 functions (also in a library). Thanks for the good work, Fred.

Forth 83
C.H. Ting's "Inside F83" should be a valuable addition to your FORTH library. It's chocked full of information (288 pages), and definitely written for the knowledgeable.

Ting carefully explains F83's source and new features, using MS-DOS examples.

It's not a light volume (weighs at least two pounds), and Ting has lapses in grammar (does the name tell you something?), but it's FORTH. 825 in paperback.

Available from:
Mountain View Press
P.O. Box 4656
Mountain View CA 94040
415-961-4103

And that's Tidbits!

---

Z SETS you FREE!

Z Operating System, an 8-bit OS that flies! Optimized HD64180/Z80 assembly language code — full software development system with proven linkable libraries of productive subroutines — relocating (ROM and RAM) macro assembler, linker, librarian, cross-reference table generator, debuggers, translators and disassemblers — ready to free you!

High performance and flexibility! Productivity results from dynamically customized OS environments, matching operator, tasks and machine.

Real-time control kernel option allows quick software development for industrial control applications, other tools and utilities for office desk-top personal computing functions, local area networks to Ethernet, AppleTalk, Omnimacs, ArNet, PC-Net (Ethernet) — from micro to mainframe command, control and communications. Distributed processing application programs are easily developed.

- Extreme organizational flexibility, each directory another environment
- Multiple Commands per line
- Aliases (complex series of commands known by simple names) with variable passing
- Named Directories with absolute password security
- Full-screen command line editing with previous command recall and execution
- Shells and Menu Generators, with shell variables
- Command-file search paths, dynamically alterable
- Screen-oriented file manipulation and automatic archiving and backup
- 3/4 megabyte file sizes, 8 gigabyte disks handled
- Auto disk reset when changing floppies
- TCP/IP database handles characteristics of over 50 computers and terminals, more easily added
- Tree-structured online help and documentation subsystem
- Z's syntax-compatible support utilities

Your missing link has been found — ZI Now fly with eagles! Fast response, efficient resource utilization, link to rest of computing world — shop floor to executive suite, micro to corporate mainframe. Call 415/948-3820 for literature.

---

Z80 System Owners
Join The 16/32 Bit
Revolution Through
Evolution

Starting At
$695.00

CO-PROCESSING
The most cost effective way for Z80 system owners to obtain 16/32 bit processing power and software compatibility is via the HSC CO-16 Attached Resource Processor. CO-16 is compatible with any Z80 system running CPM 2.2 or CPM 3. A few examples include:
- KAYPRO 2/14/10 • TRS 2/3/12/16
- AMPRO LITTLE BOARD
- HEATH 89 • SUPERBRAIN
- XEROX 820 • TELEVIDEO 802/803
- MORGAN • EPSON QX-10
- LOBO • OSBORNE 1/EXEC
- CROMEMCO • Plus many more

CO-16
Every CO-16 is delivered with
- 16/32 bit micro processor • 16 bit Operating System • 256 Kilo RAM
- Z80 interface • 16 bit RAM disk driver • CPM80 2.2 RAM disk driver
- CPM 2.2 or CPM 3 compatibility
- Sources with tools • hardware diagrams • board level or case with power supply

CO-1686
The only Z80 16 bit co-processor includes
- INTEL 8086 • 6Mhz no wait states • MSDOS 2.11 • IBM BIOS emulator
- Memory expansion to 768K • 8087 math co-processor
- 3-channel Real Time Clock • Runs many IBM PC applications • Shares hard disk space with CPM80 • PC diskette compatibility on many systems • CPM86 • Concurrent CPM is coming.

CO-1688
The only Z80 16/32 bit co-processor includes
- MOTOROLA 68000 microprocessor • 6 Mhz no wait states • CPM68K • Full “C” compiler with UNIX V7 library and floats
- Memory expansion to 1.25 million bytes • NS16081 math co-processor
- Real Time Clock • Complete software development environment
- 100% file compatible with CPM80
- O/S8/68 UNIX look alike coming in February.

Dealer, Distributor and OEM's invited

Hallock Systems Company, Inc.
267 North Main Street
Harkimer, N.Y. 13350
(315) 866-7125

---
The Pascal Runoff

I'm not going to beat around the bush about this contest (see the Editorial for bush beating). In short, we're having a contest, you're invited (in fact, we're not inviting anyone else), and the prizes are really spiffy.

So Let's Get To The Prizes

The Grand Prize is your choice of a Microsphere 1 Megabyte RAM DISK or Trevor Marshall's 32032 Coprocessor kit. It's the full 1 megabyte 32032 board that plugs into the K16 or any other PC clone.

Each of the Next Five Scorers receive the following:

Choice of two products from Borland (including the Modula 2 Compiler) AND ... choice of $100 worth of products from Micro C.

What To Do To Get A Prize

To enter just write a program in Turbo Pascal and send it to Micro C. Make sure you specify "Turbo Pascal Contest" on the envelopes, so we'll know it's an entry. If possible, include a listing on paper along with your disk.

We're not looking for a magnum opus, just something useful or interesting (or both). It doesn't have to be long - a lot can be said in Turbo in 100 lines.

This contest is intended to encourage concise, clear programming style.

Contest deadline is November 1, and we'll announce the winners in the February-March '86 issue of Micro C.

Scoring

Programs will be judged by Philippe Kahn and the Micro Cornucopia staff on a point system. Total points decide the winner.

0-15 for ALGORITHM
0-15 for READABILITY OF CODE
0-30 for FUNCTIONALITY
(including ease of use)
0-20 for ORIGINALITY
0-20 for DOCUMENTATION

So Start Programming ... An editor, a business application, a game, a utility, something educational - anything that interests you probably interests us. Just make sure the program you submit is original, unpublished, and written by you in Turbo Pascal during 1985.

PS -

Hackers in other languages, stay tuned. Your contest is coming.

Any version of Turbo Pascal is OK — 2.0, 3.0, CP/M, MS-DOS

---

Program
Title: ________________________________
Purpose: ________________________________
_____________________________________
_____________________________________
_____________________________________
_____________________________________
_____________________________________
_____________________________________

NOTE: I hereby release this program to the Public Domain and give Micro Cornucopia the right to print this listing.

Signature ________________________________

Free Pascal Runoff T-shirt
Size: □S □M □L □XL (For the first 100 entrants.)
Please list all people involved in the development of this program ________________________________

Name ________________________________ Ph: ( ) ________________________________
Address ________________________________
City ________________________________ State ________________________________ Zip ________________________________

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709
Order No. (503) 382-5060
Micro C works because it is a central information exchange for the doers in this crazy industry. So we encourage you to share your trials and tribulations. That way we can invent new wheels rather than redoing the old ones over and over.

What kind of exciting adventure (misadventure) are you working on?

What information would you like to see in Micro C?

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
<th>Price Each</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUBSCRIPTION (6 issues per year)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ New □ Renewal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>USERS DISKS – 8” SSSD, CP/M #’s</td>
<td>$15</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>CP/M 86 DISKS – 8” SSSD #’s</td>
<td>$15</td>
<td>$15</td>
</tr>
<tr>
<td></td>
<td>KAYPRO DISKS for the 11, 4, &amp; 10 – 5” SSSD #’s</td>
<td>$12</td>
<td>$12</td>
</tr>
<tr>
<td></td>
<td>KAYPRO SCHEMATIC PACKAGES</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ 11 &amp; 4 (pre-84)</td>
<td>$20</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>□ 2, 4, 10 (84)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ 10 (No Modem)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BACK ISSUES #’s</td>
<td>$3</td>
<td>$3</td>
</tr>
<tr>
<td></td>
<td>T-SHIRT (specify S, M, L, XL)</td>
<td>$6.95</td>
<td>$8.50</td>
</tr>
<tr>
<td></td>
<td>FREE KAYPRO CATALOG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OTHER ITEMS:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Make checks payable to: MICRO CORNUCOPIA ENCLOSED

☐ Check or money order enclosed (US funds only, payable on a US bank)

Prices include postage (Air Mail for Other Foreign)

Card No. ____________________________ Exp. ____________________________
☐ Visa ☐ MasterCard

Are you a current subscriber to Micro Cornucopia? ☐ Yes ☐ No

NAME ____________________________ PHONE (?) ____________________________

ADDRESS ____________________________________________

CITY ____________________________ STATE ___________ ZIP ___________

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

Order Number (503) 382-5060 • Technical Calls 9 - Noon PST Only (503) 382-8048
Micro Cornucopia, Number 25, August-September 1985
There's just enough room left in Micro C this issue for a little logic, so we'll close a few circuits.

Analog Or Digital?
These two systems differ in the way they represent information. In an analog system, data are represented by a continuously varying signal (wavelike). In a digital system, data are represented discretely—in bits that have set values—either 0s or 1s (OFF or ON). Your stereo is an analog system (probably). Your computer is digital (unless it's a very early Heathkit). For information about analog systems read RollingStone or Stereo Review. For digital—stay in tune with Micro Cornucopia.

In A Digital
Circuits open and close to control the flow of information. A gate is a simple circuit which originally (in the 1950s) was made up of individual transistors, resistors, and diodes. The big breakthrough of the early 1960s was discovering how to combine some of these transistors, resistors, and diodes (which made up a gate) on one piece of silicon. This was called Small-Scale Integration (SSI), and it changed the world.

The first integrated circuit consisted of fewer than 12 gates on a square about the thickness of paper and the width of a pencil lead. The gates or circuits were connected to the outside world via tiny wires. By the 1970s, LSI (Large-Scale Integration) was rolling, and as many as 50,000 parts were squeezed onto a single chip.

Buffers
A buffer is the simplest device in your computer. It outputs whatever it gets for an input, either straight or inverted. Buffers are often used when there are lots of inputs listening to the same signal line (say a data line or address line). It takes a lot of output to drive such a line, and most processors, for instance, don't have very strong outputs. So we stick a buffer between the processor and all those inputs.

Buffers can also protect the computer from the outside world. The most common RS-232 line drivers and receivers are simple inverting buffers (1488 drivers and 1489 receivers). All Kaypros, Big Boards, and Xeroxes use these.

The 1488 drivers change the .7V and 4V (off-on) TTL signals to the +12 and -12V (space-mark) RS-232 signals. The 1489 receivers change the +12 and -12V RS-232 signals back to .7V and 4V, the standard off-on signal levels inside the computer. The 1488s and 1489s are also designed to shrug off the static and other electrical garbage that would destroy less handy chips.

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).

Buffers can also protect the computer from the outside world. The most common RS-232 line drivers and receivers are simple inverting buffers (1488 drivers and 1489 receivers). All Kaypros, Big Boards, and Xeroxes use these.

The 1488 drivers change the .7V and 4V (off-on) TTL signals to the +12 and -12V (space-mark) RS-232 signals. The 1489 receivers change the +12 and -12V RS-232 signals back to .7V and 4V, the standard off-on signal levels inside the computer. The 1488s and 1489s are also designed to shrug off the static and other electrical garbage that would destroy less handy chips.

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).

Buffers can also protect the computer from the outside world. The most common RS-232 line drivers and receivers are simple inverting buffers (1488 drivers and 1489 receivers). All Kaypros, Big Boards, and Xeroxes use these.

The 1488 drivers change the .7V and 4V (off-on) TTL signals to the +12 and -12V (space-mark) RS-232 signals. The 1489 receivers change the +12 and -12V RS-232 signals back to .7V and 4V, the standard off-on signal levels inside the computer. The 1488s and 1489s are also designed to shrug off the static and other electrical garbage that would destroy less handy chips.

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).

Buffers can also protect the computer from the outside world. The most common RS-232 line drivers and receivers are simple inverting buffers (1488 drivers and 1489 receivers). All Kaypros, Big Boards, and Xeroxes use these.

The 1488 drivers change the .7V and 4V (off-on) TTL signals to the +12 and -12V (space-mark) RS-232 signals. The 1489 receivers change the +12 and -12V RS-232 signals back to .7V and 4V, the standard off-on signal levels inside the computer. The 1488s and 1489s are also designed to shrug off the static and other electrical garbage that would destroy less handy chips.

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).

Buffers can also protect the computer from the outside world. The most common RS-232 line drivers and receivers are simple inverting buffers (1488 drivers and 1489 receivers). All Kaypros, Big Boards, and Xeroxes use these.

The 1488 drivers change the .7V and 4V (off-on) TTL signals to the +12 and -12V (space-mark) RS-232 signals. The 1489 receivers change the +12 and -12V RS-232 signals back to .7V and 4V, the standard off-on signal levels inside the computer. The 1488s and 1489s are also designed to shrug off the static and other electrical garbage that would destroy less handy chips.

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).

Buffers can also protect the computer from the outside world. The most common RS-232 line drivers and receivers are simple inverting buffers (1488 drivers and 1489 receivers). All Kaypros, Big Boards, and Xeroxes use these.

The 1488 drivers change the .7V and 4V (off-on) TTL signals to the +12 and -12V (space-mark) RS-232 signals. The 1489 receivers change the +12 and -12V RS-232 signals back to .7V and 4V, the standard off-on signal levels inside the computer. The 1488s and 1489s are also designed to shrug off the static and other electrical garbage that would destroy less handy chips.

Build A Buffer
Now let's see how a simple buffer works. (Go ahead and build it, then use a voltmeter or LED to see what's happening.) You can use any garden variety NPN transistor, but the 2N2222A (or equivalent) costs about $1 and is commonly available at Radio Shack and other parts houses.

A transistor has an emitter, base, and collector (E, B, and C). Before you leave the store, make sure you know which lead is which—lead identification may be marked on the transistor or on the packaging, or the dealer may have to look it up.

You can think of your transistor as a sensitive switch. A small amount of current flowing into the base (B) and out the emitter (E) allows a larger amount of current (about 100 times as much) to flow into the collector (C) and out the emitter (E).
DSDBO
FULL SCREEN SYMBOLIC DEBUGGER

"THE SINGLE BEST DEBUGGER FOR CP/M-80. A TRULY AMAZING PRODUCT."

LEOR ZOLMAN
AUTHOR OF BDS C

- Complete upward compatibility with DDT
- Simultaneous instruction, register, stack & memory displays
- Software In-Circuit-Emulator provides write protected memory, execute only code and stack protection.
- Full Z80 support with Intel or Zilog Mnemonics
- Thirty day money back guarantee
- On-line help & 50 page user manual

NOW ONLY $125.

P.O. BOX 49473 AUSTIN, TEXAS 78765 (512) 478-4763
ICs

PROMPT DELIVERY!!!
SAME DAY SHIPPING (USUALLY)

OUTSIDE OKLAHOMA: NO SALES TAX

8087-3 MATH $105.00
8087-2 COPROCESSORS 140.00

DYNAMIC RAM
256K 256Kx1 120 ns $ 4.49
256K 256Kx1 150 ns 3.25
64K 64Kx1 150 ns .99

EPROM
27C256 32Kx8 250 ns $15.99
27256 32Kx8 250 ns 9.10
27128 16Kx8 250 ns 3.47
27C64 8Kx8 200 ns 7.85
2764 8Kx8 250 ns 2.50
2732A 4Kx8 250 ns 2.75

STATIC RAM
6264LP-15 8Kx8 150 ns $4.99
6116LP-3 2Kx8 150 ns 1.95

ADD 840 Byte TO 8272 8 SLOT MOTHERBOARD
CONSUME NO EXPANSION SLOTS: $78.50

OPEN 6½ DAYS: WE CAN SHIP VIA FED-EX ON SAT.

MasterCard/Visa or UPS Cash COD
Factory New, Prime Parts
MICROPROCESSORS UNLIMITED
24,000 S. Peoria Ave., (918) 267-4961
BEGGS, OK. 74421

Prices shown above are for July 2, 1985
Please call for current prices. Prices subject to change. Please expect higher or lower prices
on some parts due to supply & demand and our changing costs. Shipping & insurance extra. Cash
discount prices shown. Orders received by 6 PM CST can usually be delivered to you by the next
morning, via Federal Express Standard Air @ $6.00, or Priority One @ $15.00

LEAP INTO A NEW DIMENSION

WITH AZTEC CII

Full implementation of "C" with standard floating
point, library, and I/O subroutines. UNIX VER 7
compatible. Produces relocatable 8080 (optional 8280)
assemble code. Relocating assembler and linker
supplied with package or use Microsoft M80 and L80,
.SID./ZSID debugger interface. FAST COMPILATION
AND EXECUTION.

AZTEC CII FOR CP/M $199
(Special price for Micro C subscribers $149)

MANX
SOFTWARE SYSTEMS
Box 55, Shrewsbury, N.J. 07701
(201) 780-4004

Also available for Apple DOS, HDOS, CP/M-86, PC-DOS

CALENDAR/CLOCK

ZTIME-I

$69 KIT

NOW WITH FILE
DATE STAMPING!

• Works with any Z-80 based computer.
• Currently being used in Ampro, Kaypro
  2, 4 & 10, Morrow, Northstar, Osborne,
  Xerox, Zorba and many other computers.
• Piggybacks in Z80 socket.
• Uses National MMS8167 clock chip, as
  featured in May '82 Byte.
• Battery backup keeps time with CPU
  power off!
• Optional software is available for file
  date stamping, screen time displays,
  etc.
• Specify computer type when ordering.
• Packages available:
  Fully assembled and tested $99.
  Complete kit $69.
  Bare board and software $29.
  UPS ground shipping $ 3.

MASTERCARD, VISA, PERSONAL CHECKS,
MONEY ORDERS & C.O.D.'S ACCEPTED.
N.Y. STATE RESIDENTS ADD 8% SALES TAX

KENMORE
COMPUTER TECHNOLOGIES
P.O. Box 635, Kenmore, New York 14217 (716) 877-0617

MICRO CORNUCOPIA

P.O. BOX 223
BEND, OREGON 97709

MICRO TECHNICAL JOURNAL