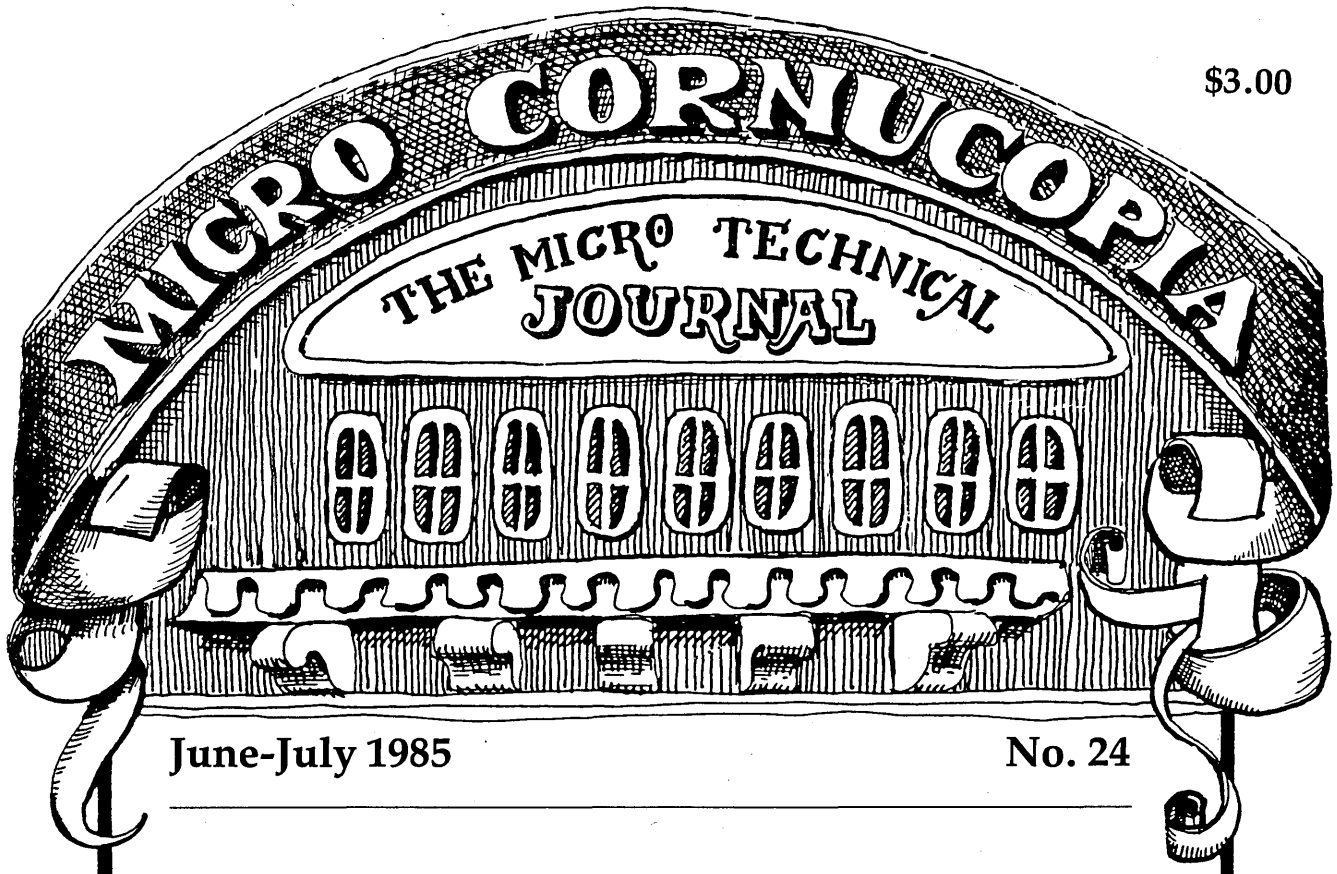
$3.00



June-July 1985 No. 24

TABLE OF CONTENTS

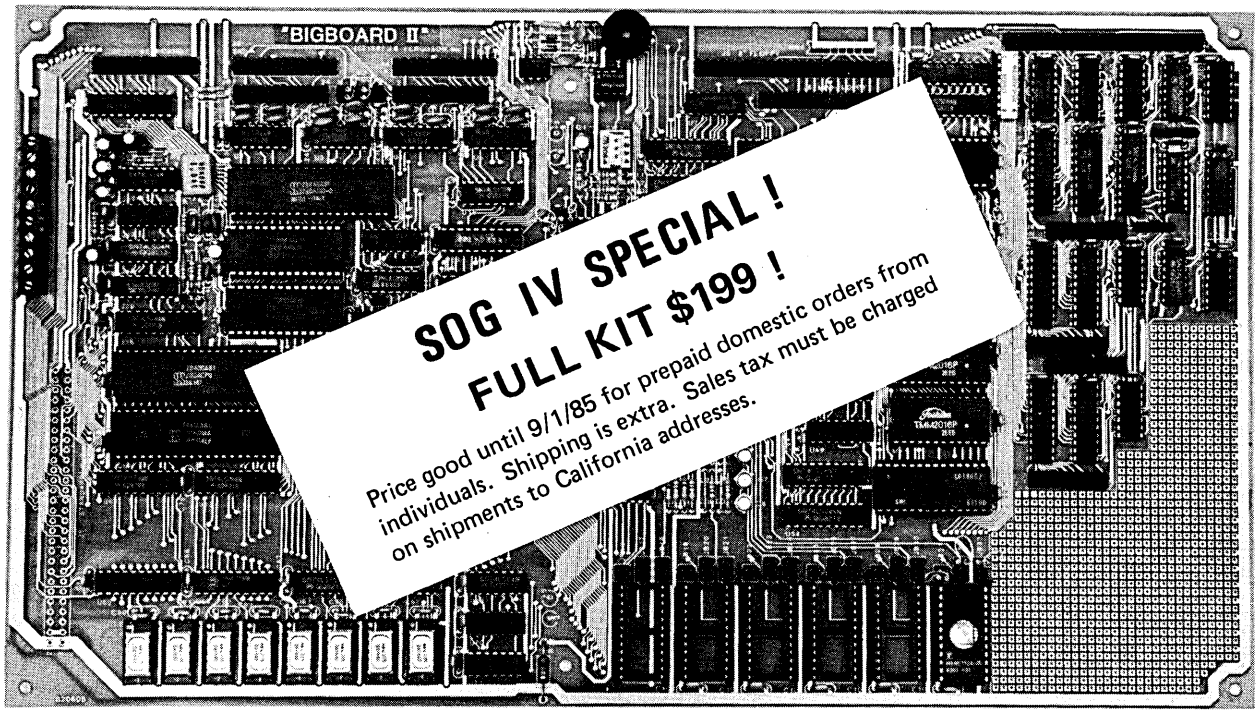
Regular Features

Future Tense

NEW LOWER PRICES! ***NOW IN "UNKIT"* FORM TOO!***

"BIG BOARD II"

4 MHz Z80-A SINGLE BOARD COMPUTER WITH "SASI" HARD-DISK INTERFACE



$795 ASSEMBLED & TESTED **$545 "UNKIT"*** **$245 PC BOARD WITH 16 PARTS**

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(E)PROM 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, an "unkit"*, 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 1793 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 880 Kbits per second with minimal processer overhead. When a hard-disc subsystem is added, the DMA chip makes impressive disk performance possible.

SIZE: 8.75" x 15.5"
POWER: +5V @ 3A, +-12V @ 0.1A

■ "SASI" Interface for Winchester Disks

Our "Big Board II" implements the Host portion of the "Shugart Associates Systems Interface." Adding a Winchester disk drive is no harder than attaching a floppy-disk drive. A user simply 1) runs a fifty-conductor ribbon cable from a header on the board to a Xebec controller that costs only $295 and implements the controller portion of the SASI interface, 2) cables the controller to a Seagate Technology ST-506 hard disk or one compatible with it, and 3) provides power for the controller-card and drive. Since our CBIOS contains code for communicating with hard-disks, that's all a user has to do to add a Winchester to a system!

■ Two Synchronous/Asynchronous Serial Ports

With a Z80-A SIO/O and a Z80-A CTC as a baud-rate generator, the new Ferguson computer has two full RS232-C ports. It autobauds on both.

■ A Parallel Keyboard Port + Four Other Parallel Ports for User I/O

The new Cal-Tex single-board computer has one parallel port for an ASCII keyboard and four others for user-defined I/O.

■ Two Z80-A CTCs = Eight Programmable Counters/Timers

The new Ferguson computer has two Z80-A CTCs. One is used to clock data into and out of the Z80-A SIO/O, while the other is for systems and applications use.

■ PROM Programming Circuitry

The new Cal-Tex SBC has circuitry for programming 2716s, 2732(A)s, or pin-combatible EEPROMs.

■ CP/M 2.2**

CP/M with Russell Smith's CBIOS for the new Cal-Tex computer is available for $150. The CBIOS is available separately for $25.

** The "unkit" is a fully-socketed, wave-soldered "Big Board II". It requires NO soldering. All an "unkit" purchaser must do is carefully insert the prime ICs we supply in the proper sockets and systematically proceed to bring up and test the board.*

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

CAL-TEX COMPUTERS, INC.

780 E. TRIMBLE ROAD #504 • SAN JOSE. CA 95131 • (408) 942-1424

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.

MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709
503-382-5060 Orders Only
503-382-8048 Tech. 9 - Noon

Editor & Publisher
David J. Thompson

Operations Manager
David Pogue

Assistant Editors
Rebecca Ozrelic Gary Entsminger

Accounting
Sandra Thompson

Graphic Design
Craig Lannes Michael Odell

Technical Department
Dana Cotant Eric Roby
Bruce Berryhill Laine Stump

Advertising Director
Alice Holbrow

Staff Assistants
Dorcas Dsenis Tracey Braas
Cary Gatton Carla Miller
Laura Pendley

Typography
Patti Morris & Martin White
Irish Setter

MICRO CORNUCOPIA supports systems programming languages and builders of single board and S-100 systems.

Application to mail at second class postage rates is pending at Bend, OR 97701 and additional entries. Published bimonthly by Micro Cornucopia Inc., 155 NW Hawthorne, Bend, OR 97701. Postmaster: Send address change to Micro Cornucopia Inc., PO Box 223, Bend, OR 97709.

SUBSCRIPTION RATES:

1 yr. (2nd Class)	$16.00
1 yr. (Canada & Mexico)	$22.00
1 yr. (Other foreign)	$30.00

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

ADVERTISING RATES: Available on request (call Alice Holbrow).

CHANGE OF ADDRESS: Please send your old label and new address.

SOFTWARE, HARDWARE, AND BOOK VENDORS: We would very much like to review your software, Single Board, and S-100 products. Send materials to the Review Department, Micro Cornucopia.

ISSN 0747-587X

MICRO CORNUCOPIA

June-July 1985 The Micro Technical Journal No. 24

On Your Mark!

A Contest To Start All Contests

Usually people announce contests to end all contests. This one is a beginning. We've been sitting around trying to decide how best to do it and, meantime, nothing has happened.

We kicked around the idea of leaving it open. You know, anyone can use any language, create any length program, on any subject.

"Too general," was the reaction from all the reactionaries in the office. (What else are they good for?)

We kicked around the idea of specifying the language, the version of the language, the subject, the program length, and the color of the programmer's eyes.

"Too specific," was the reaction (from you-know-who).

It was immediately obvious that we were going to have to compromise on this contest (or get rid of the reactionaries). So we're specifying the language (Turbo Pascal), but not the version. We've sort of narrowed program length, but not the subject. As for eye color, we've chosen "bleery red streaks" so all inveterate hackers will qualify easily.

Check out the contest article in this issue. (Dr. Dobb's just held a contest and had four entries. Let's see what we can do.)

10,000 Subscribers

I remember when Lifelines magazine announced they had 5,000 subscribers. I was green with envy because Lifelines and Micro C started about the same time.

We had around 500 circulation at that time (5,000 seemed an impossible dream), and I was still working at Tektronix. I figured that about the time we hit 1,000 I could quit Tek and cut my workday back to 20 hours.

When we printed labels for the April-May issue this year we had over 9500 subscribers, and we celebrated the 10,000 mark in mid-April. We hit 5,000 almost exactly a year ago. (Now, if you'd all come to SOG IV . . .)

Speaking Of SOG IV

Don't forget to send in your SOG registration form. Let us know if you're coming (or at least maybe coming), even if you're not planning on rafting, dining, staying in the dorms, or feeding the chipmunks. (The dorm has been full since mid April, but you'll find plenty of space in local motels and camp grounds.)

There is an Experimental Aircraft Association meet every year. During August, every home-built, antique, and non-antique private aircraft that isn't cruelly tethered, takes its family and flies to Oshkosh.

The airplanes park on a grass field next to the airstrip, and the families camp for a week under the wings of their craft (everyone wanders around meeting everyone else).

I'd like to make the SOG into a similar event. All I need to do is find an appropriate field and equip it with basic necessities, and we could make SOG a real experience (not that a lot of families don't already camp out when they come to the SOG).

In fact, the EAA brags that it has the largest collection of porta-potties in one place, at one time, in the world. They might even have 5,000 of them. (Of course, 5,000 potties seems like an impossible dream for Micro C, but . . .)

32032 Support

The National 32032 and 32016 chips have been languishing in the shadow of the 8088/8086 and the 68000. Well, Trevor Marshall, well known among the Micro C crowd for his BBI winchester designs, has spent the last year working on a 32032 based co-processor board.

Trevor not only designed the system (with the help of two friends), but he has also written two articles (hardware de-

(continued on page 72)

LETTERS

Rebuttal For Kamas

I write to take friendly but spirited exception to Jack Rodenhi's lukewarm review of Kamas (by Kamasoft, formerly Compusophic Systems). My disagreement is mostly on the degree of usefulness of an outline maker. In the few months I've had Kamas, it's been a huge help in getting me started on writing projects.

To illustrate how I've integrated Kamas into my own writing process, here's my list of steps:

1. Make my outline with Kamas.
2. Enter as much text as I want into text leaves.
3. Rearrange, fix stuff, and do general editing.
4. Output a WordStar compatible file to disk.
5. Call up WordStar for correcting and final editing.
6. Call up The Word + for checking spelling and typos.
7. Switch back to WordStar and print the thing out!

One other thing: Kamas is the only outline making program available right now for the CP/M-80 Kaypro. It's not as if we had a choice. It's either Kamas for my Kaypro II or nothing.

Lucian W. Minor
Box 1101
Wellfleet MA 02667

---

dBASE Advice

I read Issue 22 and wanted to comment on your problems with using dBASE II to print your mailing labels. I have been programming in dBASE for about 2½ years, but the version numbers you mentioned were new to me. Are they Kaypro dBASE version numbers? CP/M-80 dBASE II is currently up to version 2.41. Version 2.3B was full of bugs. 2.40 was a good improvement, and 2.41 added a little more. If you're not using at least 2.40, I'd suggest you get it. It could be your problem. I wouldn't use 2.3B for anything!

I am about 90 percent finished re-writing The Master Check Register program. (Version 2.0 will be called "Turbo Cheques.") I started programming in Turbo Pascal last summer, and saw the advantages of switching over from dBASE to Turbo P. I decided to re-write the entire program in Turbo Pascal. The speed improvement is unbelievable! In addition, there are many things Turbo P has that dBASE II lacks: Procedures and Functions, no 32 field limit, no 64 memory variable limit, etc. I feel free to be more creative in my programming designs.

The 8-bit CP/M Compiler from WordTech is a flop! It is far too slow to compile and link, and the finished application runs SLOWER than in dBASE II. It is worth the effort to learn to program in Turbo Pascal to do 8-bit CP/M programming. The speed improvements over dBASE II are enough by themselves. And remember, Turbo P only adds 8K to your total applications file size! (For the RunTime library.)

Ralph E. Freshour
7 Silver Eagle Road
Rolling Hills Estates CA 90274

Editor's note:

Thanks for the comments, Ralph. The version numbers were errors (mine). I am using 2.3b and 2.40. I know that 2.3b is buggy, but it leaves a little more space for memory variables so some of my programs run under it that won't run under 2.40. For instance, adding a record to a large indexed file is much faster under 2.37 than under 2.40. Also, my copy of 2.40 won't recognize the decimal point in the picture clauses, so the gals prefer entering receipts under 2.37.

We're currently futzing over all our dBASE routines to Turbo. I'll keep everyone posted on how it works out.

Composite Video Update

Last week one of the local dealers sent me to the Kaypro service clinic. You just can't stay on top of things in this biz. Since submitting my "Composite Video" article (see Issue #22, page 50), I have found that you can't use that approach on the 10. Now I've delved into the 2-84/4-84 boards and the new "universal" board and found the same video circuits as the 10. The graphics are nice, but the hi-res method of achieving them nukes my cheap video approach. However, my board does work on Xerox 820s.

Richard Bugg
2703 N.W. 20th
Oklahoma City OK 73107

Kaypro Goes Arabic

This is in response to a question in Issue 22 about Arabic capabilities for the Kaypro. Zadian Research has an Arabic conversion package which runs on the Kaypro 2, 2X, 4, and 10. The package includes software to convert CP/M, WordStar, CalcStar, and other programs to Arabic. It also comes with a set of programs for Arabic text processing and printing and full documentation.

Zadian Research has also developed an arabization kit for the Prowriter dot-matrix printer. It allows the Prowriter to print Arabic without losing any of its English capabilities. Interspersed Arabic and English and enlarged Arabic are also supported.

The bilingual Kaypro models and the arabization kit for the Prowriter printer are available from the International Marketing Department of Kaypro Corporation, 533 Stevens Avenue, Solana Beach, CA 92075.

Zadian Research
1749 Jonathan Avenue
San Jose CA 95125

Selling Prototypes

I'm wondering if there is some way to profit from my ideas without having to do everything myself. Is Micro C, or some company that advertises in Micro C, interested in buying working hardware prototypes to be finished (e.g. packaging, purchasing volume parts, making PC boards, and writing software) in exchange for royalties or something? Frankly, a $15 diskette of software is poor compensation for several days worth of work. Is there some other possibility I haven't thought of?

Phil Hunter
655 S. Fair Oaks #E-317
Sunnyvale CA 94086

Editor's note:

Sure, we'd be glad to look at any projects you are working on. Give me a call and let's discuss what you're doing. Maybe we can do a combination article, circuit board, etc. It could also be a straight product.

Also, the disk is not the only payment for an article. When we receive an article we send out a disk, pronto. Then when we print the article we send out additional goodies: the

LETTERS

special feature for each issue is worth $75, five copies of Micro C, and an author's T-shirt (great for wearing at the SOG). The regular articles are worth $25, three magazines, and the author's T-shirt.

Tri-Flow Triumphs

I am in a group of nine engineers working for Xerox, providing technical hardware and software support to about 1300 technicians in four states for our copier and duplicator lines.

A while back, in your Kaypro Column of Issue 15, you stated that Xerox Service Centers use WD-40 to lubricate disk drives. To bring you and Micro C readers up to date, in early 1984, Xerox cancelled the use of WD-40 and began using Tri-Flow in its products.

There were some good reasons for this decision. Several fires in our copier/duplicators were caused when WD-40 spray came in contact with a high voltage component, created an arc, and ignited. In addition, we felt that although WD-40 did a good job of washing the old lubricant from a bearing, its own lubricating properties were very short lived.

The only good application I can think of for WD-40 in any of our products is to clean nasty, dirty drive chains

I am very much in favor of lubricating the lead screws and slides with Tri-Flow. The standard CP/M-80 that comes with the 820-II, 8" SS, is configured for a head step rate of 15ms, which is too slow.

Any 8" drive in good shape will step reliably at 10, 6, or even 3 ms. I have seen cases where new 8" drives out of the box would not read at 6ms, but after the lead screws were lubricated with Tri-Flow I could change this time to 3ms and boot 99+ percent of the time.

It would be wise, however, to keep a disk in the archives with the step rate left at 15ms just in case the drives get gummy and you can't boot. All of my working disks have been at 3ms since I did the Tri-Flow trick over a year ago.

Another tip: the oilite bearings found in most drives should not be lubricated. This is a porous metal which is impregnated with oil during manufacturing. Adding oil will free the bearing for a short time, but will cause the pores to clog and accelerate the wear of the bearings.

The best way to remove dirt and gum from these bearings is with a clean, dry, lint-free cloth.

If an oilite bearing is properly manufactured and, more importantly, stored correctly prior to drive assembly (paper or other absorbant packaging materials will leach the oil from the bearing), enough oil will remain after the bearing is cleaned.

Doug Felton
1215 Oxley Rd.
Columbus OH 43212

Arizona Kaypro Users Group

As owner of a Kaypro dealership in Phoenix and sponsor of the Arizona Kaypro Users Group, I wrote to tell you what tremendous success I've had with the mods and upgrades for Kaypro suggested in Micro C. I did nearly 100 5MHz upgrades last year, and installed more than 30 disk drives behind Pro-8 ROMs just last month. I find the instructions for the upgrades very complete and quite clear.

During this month's AKUG meeting the club's officers demonstrated their 4-drive Kaypros. Two of our officers (John and Sharon Wertz) have 'his-n-hers' Kaypros, each sporting four Mitsubishi quads.

Our RBBS/RCPM is tagged "Lost Dutchman's Gold Mine #2" and is devoted exclusively to Kaypro support and utilities. The number is (602) 863-1435. The board operates 300/1200 baud, 24 hours a day. There is no fee, and our only request is that users log on with real names, not handles.

Kelvin Paul Giles
13829 North 19th Ave.
Phoenix AZ 85023

TPA Program Fix

In Issue #23 Tom Geldner (in "Running In CP/M's TPA") gave a programming example in SBASIC that was supposed to dynamically size an array according to available memory. His example works, except that the way he did it, the array will always have a maximum size of 32767 even if more room is available.

The problem lies in the way SBASIC uses the signed value of INTEGER variables when dealing with FOR..NEXT loops. (FOR..NEXT uses signed integers rather than unsigned integers like WHILE.) The solution is to use a WHILE..DO structure.

```
x = 0
WHILE x <> max.memory DO BEGIN
        x = x + 1
        sample.array(x) = 0
END
```

Richard Levine
3105 Meadow Grove Dr.
San Diego CA 92110

Packet Power

Maybe you already know this, but the 820 is becoming the standard for packet radio mailbox systems. W0RLI has written an excellent package that includes, among other things, automatic forwarding of mail to other mailboxes based on a table of users for each mailbox. There's also an add-on HDLC chip board to allow using it for a TNC. In our area the packet users have grabbed about 20 of them for various types of experiments. Fun stuff!

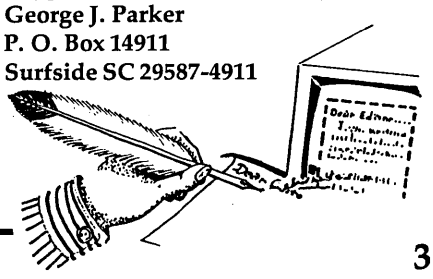
Jon Pearce, WB2MNF
109 Pine Cone Trail
Medford NJ 08055

U.S. Robotics Modems OK

While leafing through your Kaypro Users Catalog, I noticed on page 11 there is a statement written about the U.S. Robotics modems being unreliable for dialing from preloaded directory files.

I have two U.S. Robotics modems, an AUTODIAL 212A and a PASSWORD, and in the past six months I have not experienced any difficulties in dialing from the phone directories of TELPAC, MDM712, KM300, KM1200, NDM730, PHONE.001, etc. This is true for the Kaypro 2-83 and Kaypro 2X.

George J. Parker
P. O. Box 14911
Surfside SC 29587-4911



C'ing Into Turbo Pascal

By Ron Miller

1157 Ellison Drive
Pensacola FL 32503

Like a great number of hackers, I am convinced that for really getting down to manipulating the system, C is the only way to fly.

Not only does C offer the means to attack a problem at the byte level, but C syntax is also more elegant, more straightforward, and more readable the morning after than any of the alternatives.

When Not To C

However, the complex sequence of library searching, compiling, assembling, and linking that gives such hands-on control when writing a utility program in C becomes a genuine pain when crafting an inventory system or typing out a loop to average grades. And reading and writing records to a random-access file is not something one does casually in C straight out of the box.

The answer for me, and apparently for lots of people, is Turbo Pascal.

It's good, it's fast, and it's inexpensive. Debugging can actually be fun when the full screen editor places the cursor right on the trouble spot.

Absolute Addressing & More

I remember opening my package from Borland over a year ago, wondering if I'd been a fool to buy something with such a silly name. Well, for 50 bucks, what can you lose? Your cynicism, for one thing.

My prior Pascal experience with JRT didn't prepare me for the delights to follow. These folks, I immediately recognized, think like C programmers.

There were structured constants (read: initialized variables); free placement of variable declarations, functions, and procedures (read: libraries); absolute addressing (read: pointers to the operating system); interconversion among scalars (read: casts); and bit manipulations (read: packed fields, tagging, and all the rest).

Turbo has never threatened to replace my beloved C/80 for getting down eyeball-to-eyeball with my Z80. Pascal's syntax is just too cumbersome when I get serious about manipulating bytes.

Incrementing A Character Pointer

For example, to increment a character pointer in C after printing the current character, all one writes is:

```
putchar(*charptr++);
```

Whereas in Pascal the best one can do is:

```
write(charptr^);
charptr := PTR(SUCC(ORD(charptr)));
```

But Turbo does give you the transfer functions to do the job.

At times, I'm even willing to put up with ORDs, SUCCs, PREDs, CHRs, ADDRs, and PTRs rather than give up Pascal's set variables, arithmetic-style string operations, interactive debugging, and compilation in the twinkling of an eye.

And there are even lower-level tricks lurking in Turbo Pascal.

Beginners may find these sample routines an encouragement to get a little closer to their operating systems. Experts will undoubtedly see better ways and assure themselves smugly that FORTH or C or assembly language does it better. I can only echo Dr. Johnson's comment on dogs walking on their hind legs: it's amazing not that it's done well, but that it's done at all.

Figure 1

```
PROGRAM directory;
 CONST
    ADDRESS = $b000;
    fcb:array[1..13] of char=#00'Z???????????'#00;   {wildcarding fcblock}
 VAR
    i,j:byte;
    charptr:^char;
 BEGIN
    bdos(26,ADDRESS);          {setting the dma}
    bdos(17,ADDR(fcb));        {"seek first"; structured consts have addresses!}
    charptr := PTR(ADDRESS); {setting the char ptr to beginning of dma}
    FOR i:=0 TO 127 DO
      BEGIN
       IF i mod 16 = 0 THEN writeln;   {rows of 16, ddt-style}
       IF (i mod 32) IN [1..11] THEN write(charptr^,' ') {if in filename}
          ELSE      {write out hex numbers for other chars}
           BEGIN
             j := ORD(charptr^) div 16;  {high nibble of hex byte}
             IF j < 10 THEN
                WRITE(CHR(j+ORD('0'))) ELSE WRITE(CHR(j-10+ORD('A')));
             j := ORD(charptr^) mod 16;  {low nibble}
             IF j < 10 THEN
                WRITE(CHR(j+ORD('0')),' ') ELSE WRITE(CHR(j-10+ORD('A')),' ');
           END;
       charptr := PTR(SUCC(ORD(charptr)));
      END;
 END.
```

Figure 2

```
                    {Contents of ARGREAD.LIB}

 CONST {Don't let "CONST" fool you: these are initialized statics. See below.}
    argv:array[1..4] of string[15]=('','','','');
    argc:byte=1;      {the COM file itself is number 1}

 PROCEDURE argread;
  CONST
      place:byte=$82;       {simply the initial memory address in the dma}
  BEGIN
      WHILE(mem[place] <> 0) DO   {just as in C: loop until a null}
       BEGIN   {since a blank/non-blank pair always begins a new argument}
         IF (mem[PRED(place)] = 32 ) AND (mem[place] IN [33..126]) THEN
               argc  := SUCC(argc);
         IF mem[place] <> 32 THEN
               argv[PRED(argc)] := argv[PRED(argc)] + CHR(mem[place]);
         place := SUCC(place);      {See? These CONSTs act just like VARs.}
       END;
      END;
```

Looking Into File Block Allocations

Let's practice a little Pascal string work by applying it to a common enough low-level operation—the extraction of directory information by using a BDOS "seek" function.

Suppose I want to examine the file block allocations within the first directory sector listing a file beginning with the letter Z. I would read a 128-byte sector into an unused spot in memory and then scan the sector to extract the file names and locations (in hex) from the 32-byte file records. (See Figure 1)

With a little gymnastics, even Pascal can speak hex (better than Post Office, anyway).

The address here is noteworthy. I would ordinarily use the 80H junk area for this sort of work, but Turbo won't let me. It writes over that convenient dumping ground from 80H+20H onward. This can be quite disconcerting if you are using BIOS read and write functions.

Extracting Directory Info

Move the DMA address to some area in free memory and you can extract directory information, and read tracks, sectors, and the like with abandon. Leave the DMA at 80H and you'll either become hopelessly hung up, or find you've just crashed back into the operating system.

To load a COM file, CP/M puts the command line arguments into a string beginning at address 82H in the default memory area. Unlike C, Turbo doesn't offer readymade facilities to extract the string or strings so they can be used by the program. But they can be extracted.

The trick is to trot through Turbo's predefined "mem" array (consisting of all the 64K bytes in memory) from 82H onward until a null is located, reading the non-blank bytes into strings within your applications program. Imagine it's a library file (see Figure 2).

I could pull it off more elegantly in C—but then in C, I wouldn't need to pull it off at all. What we are creating is essentially a standard-issue C program written in Pascal.

Notice I allow only four arguments. I could have allocated them one by one with NEW. But if I'm that crowded for space, I wouldn't be putting up with Turbo's 7K+ of overhead. Besides, if the command line is more than 30 bytes long (see above on Turbo's treatment of the DMA), the CCP's calling card is trimmed along its edges.

Redirecting I/O Unix-style

Unix-style redirection doesn't work unless the runtime package is prepared to channel the standard I/O to and from files and devices. Turbo lets you approach this. Like C, it treats peripherals as files that can be reassigned.

Suppose you occasionally want to divert CRT output to a printer. You could, of course, write a dual set of "write(x)" and "write(lst,x)" routines with an attendant array of "IF" statements. But that can be a pain and often means a larger source file. There's a better way.

Turbo contains a set of predeclared I/O pointers that can be redirected with a simple equals sign.

The example in Figure 3 reads a file indicated by the command line, strips the

(continued on page 7)

Figure 3

```
        BEGIN
          mem[3] := mem[3] AND $3f; {setting IOBYTE to serial}
          ConOutPtr := LstOutPtr;   {CON: file pointer becomes LST: pointer}
        END;
    REPEAT
      READ(dumpfile,c);
      WRITE(UPCASE(CHR(ORD(c) AND $7f)));   {first strip off parity bits}
    UNTIL EOF(dumpfile);                    {so UPCASE works: Wordstar, maybe?}
    CLOSE(dumpfile);
END.

PROGRAM toggle;
{$I ARGREAD.LIB}      {Using the code above as a library file}
 VAR
     c:char;
     dumpfile:TEXT;   {An unstructured ASCII file}

BEGIN
   argread;
   ASSIGN(dumpfile,argv[1]);
   RESET(dumpfile);
   IF(argv[2] = '>LST:') THEN   {possible redirection with second argument}
```

Figure 4

```
PROGRAM disker;
 TYPE
   param=RECORD
       SPT:integer;  {sectors per track}
       BSH:byte;     {block shift--actually, 3+log@-(2)(block size in k's)}
       BLM:byte;
       EXM:byte;
       DSM:integer;     {one less than the block count on disk}
       DRM:integer;
       AL:integer;  {allocation field for directory blocks}
       CKS:integer;
       OFF:integer; {number of reserved tracks}
     END;
  VAR
       dpbptr:^param;
       reserved,disksize,blocksize,allocptr,i:integer;
BEGIN
  allocptr := bdoshl(27,0);
  dpbptr := PTR(bdoshl(31,0));
  WITH dpbptr^ DO
   BEGIN
     blocksize := 1 shl (BSH - 3); {i.e. 2@+(BSH-3)--in k's}
     reserved := 0;
     FOR i:=0 TO 15 DO reserved := reserved + (AL shr i) AND 1;
     disksize := blocksize*(DSM + 1 - reserved);
     write(^M^J'Disk size = ',disksize,'k''s --- ');
     FOR i:=reserved TO DSM DO disksize := disksize -
       blocksize*( ((mem[allocptr + i div 8] shl (i mod 8)) AND $80) shr 7);
   END;
  writeln('Room left = ',disksize,'k''s');
END.
```

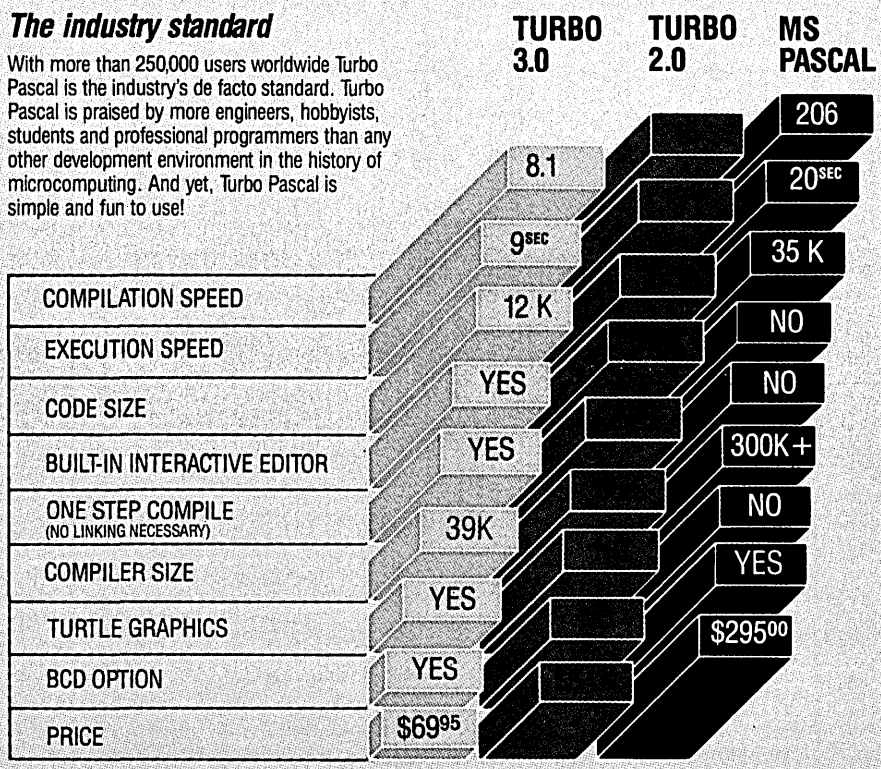
(Listing continued on page 7)

MOST SIGNIFICANT PRODUCT OF THE YEAR - PC WEEK

They said it couldn't be done. Borland Did It. Turbo Pascal 3.0

The industry standard

With more than 250,000 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!



The best just got better: Introducing Turbo Pascal 3.0

We just added a whole range of exciting new features to Turbo Pascal:

- First, the world's fastest Pascal compiler just got faster. Turbo Pascal 3.0 (16 bit version) compiles twice as fast as Turbo Pascal 2.0! No kidding.
- Then, we totally rewrote the file I/O system, and we also now support I/O redirection.
- For the IBM PC versions, we've even added "turtle graphics" and full tree directory support.
- For all 16 Bit versions, we now offer two additional options: 8087 math coprocessor support for intensive calculations and Binary Coded Decimals (BCD) for business applications.
- And much much more.

The Critics' Choice.

Jeff Duntemann, PC Magazine: *"Language deal of the century . . . Turbo Pascal: It introduces a new programming environment and runs like magic."*

Dave Garland, Popular Computing: *"Most Pascal compilers barely fit on a disk, but Turbo Pascal packs an editor, compiler, linker, and run-time library into just 39K bytes of random-access memory."*

Jerry Pournelle, BYTE: *"What I think the computer industry is headed for: well documented, standard, plenty of good features, and a reasonable price."*

(*) **Benchmark** run on an IBM PC using MS Pascal version 3.2 and the DOS linker version 2.6. The 179 line program used is the "Gauss-Seidel" program out of Alan R. Miller's book: *Pascal programs for scientists and engineers* (Sybex, page 128) with a 3 dimensional non-singular matrix and a relaxation coefficient of 1.0.

Portability.

Turbo Pascal is available today for most computers running PC DOS, MS DOS, CP/M 80 or CP/M 86. A XENIX version of Turbo Pascal will soon be announced, and before the end of the year, Turbo Pascal will be running on most 68000 based microcomputers.

An Offer You Can't Refuse.

Until June 1st, 1985, you can get Turbo Pascal 3.0 for only $69.95. Turbo Pascal 3.0, equipped with either the BCD or 8087 options, is available for an additional $39.95 or Turbo Pascal 3.0 with both options for only $124.95. As a matter of fact, if you own a 16-Bit computer and are serious about programming, you might as well get both options right away and save almost $25.

Update policy.

As always, our first commitment is to our customers. You built Borland and we will always honor your support.

So, to make your upgrade to the exciting new version of Turbo Pascal 3.0 easy, we will accept your original Turbo Pascal *disk* (in a bend-proof container) for a trade-in credit of $39.95 and your Turbo87 original *disk* for $59.95. This trade-in credit may only be applied toward the purchase of Turbo Pascal 3.0 and its additional BCD and 8087 options (trade-in offer is only valid directly through Borland and until June 1st, 1985).

NOT COPY-PROTECTED

TURBOPASCAL

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133

Carefully Describe your Computer System!

Mine is: ___ 8 bit ___ 16 bit
I Use: ___ PC-DOS ___ MS-DOS
___ CP/M 80 ___ CP/M 86

My computers' name/model is: _____

The disk size I use is:
☐ 3½" ☐ 5¼" ☐ 8"

Name: _____

Shipping Address: _____

City: _____ Zip: _____

State: _____

Telephone: _____

For update: original Turbo disk must accompany order

YES! I want the Best! Please send: Quantity

Pascal 3.0	$ 69.95	____
Pascal w/8087	$109.90	____
Pascal w/BCD	$109.90	____
Pascal w/8087 & BCD	$124.95 (SAVE $24.90)	____

*These prices include shipping to all U.S. cities. All foreign orders add $10 per product ordered.

Subtotal _____
(CA 6% tax) _____
Trade-in Credit Claimed: _____
Amount Enclosed: _____

Payment: VISA MC BankDraft Check

Credit Card Expir. Date: ____/____

Card #: _____

COD's and Purchase Orders WILL NOT be accepted by Borland. California residents: add 6% sales tax. Outside USA: add $10 and make payment by bank draft, payable in US dollars drawn on a US bank.

BORLAND INTERNATIONAL

Software's Newest Direction
4585 Scotts Valley Drive
Scotts Valley, CA 95066
TELEX 172373

(continued from page 5)

high bits off, and then capitalizes the characters before sending them to the console. Easy redirection to the serial printer.

In this case, the space saved is trivial, but it looks like Unix. Moreover, since this is a text file, we don't even have genuine binary stream I/O. Things will halt with the first ASCII 26. To scan a binary file byte by byte in Turbo, the file must first be read into a buffer sector with BLOCKREAD and then scanned by "mem" or a character pointer. Things just aren't as easy as in C.

Manipulations And The Operating System

Suppose you wanted to test the capacity of a disk—perhaps as a warning to the operator of a database.

The disk information is stored in two places:

1. Disk information is stored in the disk parameter block (the address of which is returned in the hl register after a BDOS 31 call).

2. Disk information is stored in the allocation vector bit field (the address of which is returned by a BDOS 27 call).

The parameter block itself is a string of bytes defined in the "type" listing in Figure 4.

Turbo provides both the system calls and the tools necessary to extract the information. The resident bitfields are scanned by Turbo's shift functions and bitwise-ANDing.

The rather complex left and then right dance while scanning the allocation vector is necessary because the allocation field length is not necessarily a multiple of eight bits. Therefore, we may need to test only the leftmost bits on the last byte.

Note that in the last loop, "i div 8" is the displacement in bytes past the beginning of the array; "i mod 8" is the bit in question on that byte.

Bitfield Operation Solution

Actually, there's an even neater, though not so general, solution for bitfield operations.

Since a set on the stack is nothing more than a 32-byte bitfield, the "IN" operator will scan any bitfield of up to 256 bits if a pointer to a numerical set is directed toward the beginning of that field. Figure 5 shows what the program becomes, using this kludge.

In either case, Pascal syntax hardly gets in the way at all.

The C version of the more general bitfield operation is only minimally cleaner:

```
<< I%8 & 0x80) >> 7);
```

Irritations And Impossibilities

The inelegance forced upon the programmer by Pascal's strong typing remains little more than an irritation as long as low-level work is a minor portion of a program. I find string operations the most consistently frustrating, since I've become accustomed to C's delightful increment and decrement operators and its treatment of characters as short integers.

But Turbo's "byte" variable definition gives you partial relief if you juggle things a bit to subvert Pascal's attempts to protect the programmer from typological confusion. At times I'd give anything to be able to do something useful inside a control structure rather than having to waste time with booleans.

Considerably more limiting is Turbo's way of requiring you to enter assembly language routines in machine code. Now there's entertainment that rivals Howard Cosell.

The other day I was working up a "mail merge" program for Perfect Writer. I needed a routine in high memory to:

1. Load the printer program at the bottom of the TPA.

2. Read addresses byte by byte from a mailing label file.

3. Feed them into a formatted letter that is printed repeatedly by Perfect Printer running at 100H.

A bit of tinkering with raising the origin and lowering the stack of my C/80 runtime package generated the driver program, along with the necessary diversions for Perfect Printer's attempts to call the BIOS and terminate with a warm boot. With a little help from a dozen-byte assembly language LDIR and JP routine, I was up and running at 0A000H.

Try doing that with Turbo. But then again, have you ever tried to write a set variable routine in C?

■ ■ ■

(continued from page 5)

Figure 5

```
TYPE
   field=set of 0..255;
   param=RECORD
         {see above}
     END;
VAR
     dpbptr:^param;
     fieldptr:^field;
     reserved,disksize,blocksize,i:integer;
BEGIN
  dpbptr := PTR(bdoshl(31,0));
  blocksize := 1 shl (dpbptr^.BSH - 3);
  reserved := 0;
  fieldptr:=PTR(ADDR(dpbptr^.AL));
  FOR i:=0 TO 15 DO IF i IN fieldptr^ THEN reserved := SUCC(reserved);
  disksize := blocksize*(dpbptr^.DSM + 1 - reserved);
  write(^M^J'Disk size = ',disksize,'k''s --- ');
  fieldptr := PTR(bdoshl(27,0));
  FOR i:=reserved TO dpbptr^.DSM DO
```

The S-100 Bus

By Dave Hardy

736 Notre Dame
Grosse Pointe MI 48203

One of the most frustrating things that can happen to a small computer is the notorious "dead box" problem. If you own or use any kind of machine, it has probably happened to you.

What Is A Dead Box?

"Dead box" is a composite term—kind of a cross between dead machine and black box, and is a perfect description of what can happen to a computer that has no (or very little) self-diagnostic ability. A good example of this is a machine that auto-boots from a floppy disk, and simply gives you a blank screen if it is unable to read the disk.

I usually solve these problems with a small axe or short-handled sledge. However, if you would like to confront your dead box problems in a more constructive manner, the following circuits may come in handy.

Many of the older S-100 machines have LEDs on their front panels so you can monitor, examine, and modify the machine's memory. In the days before floppy disks, the purpose of these front panels was to help you bring up the machine. But the front panel indicators are also a valuable debugging tool. By watching the lights, you can tell, more or less, what's going on in the machine.

Normally, these flashing lights on the front panel of an S-100 machine are almost useless. Unless your machine has features like a run/stop switch, an examine circuit, and a few other things, all the lights can do is indicate some kind of bus activity. But, if your machine is dead, the lights can at least let you know if there's SOMETHING going on.

Bus Line Monitor

For that reason alone, you many find the circuit in Figure 1 useful. It is a general purpose bus line monitor that can be attached to any address or data line in an S-100 machine, and most of the status and control lines. For 25 cents, you can't go wrong.

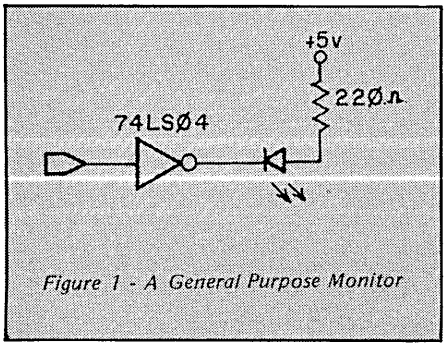


Figure 1 - A General Purpose Monitor

If you have an X-Y oscilloscope and want to see a more descriptive picture of what your computer's bus is up to, try the circuit in Figure 2. Using two inexpensive 1408-L8 digital to analog converters, this circuit will provide a two-dimensional display of your machine's address bus activity. Although it is not as sophisticated as a $10,000 state analyzer, it isn't as expensive, either. By watching the display when the machine is working properly, you can get a pretty good idea of what part of RAM the machine is operating in, and also see areas that the machine "hangs" in, or where the PC spends most of its time.

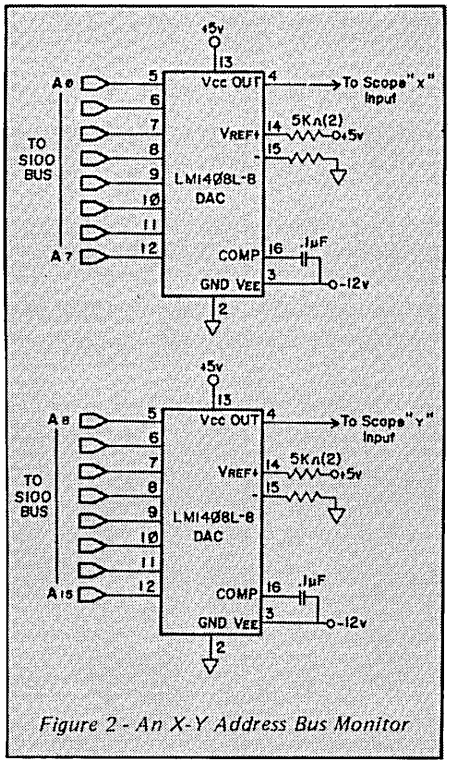


Figure 2 - An X-Y Address Bus Monitor

Dead Software

Almost as bad as a dead box is a program that makes your machine act like one. If you write your own programs, especially in assembly language, you have probably, at one time or another, loaded up some code that sent your machine's program counter into another galaxy, far, far, away.

Figure 3 shows a circuit that won't help after the program blows up, but it will at least tell you if your program has reached a certain address. The circuit is the equivalent of DDT's "break" command, only this is implemented in hardware. It is from the book "Interfacing to S-100/IEEE-696 Microcomputers" by Sol Libes and Mark Garetz, and it will stop an S-100 system if a hardware error occurs on one of the slave processors.

Multiple Processing

The future of the IEEE-696 (S-100) bus seems most promising in the field of multi-processing. Operating systems like TurboDos, that allow up to 16 slave (but functionally independent) processors to exist in a single frame, are making the S-100 bus a much-used base machine in many multi-processor applications that were formerly restricted to minicomputers or expensive networking systems.

Ironically, the thing most users object to (the high cost of an S-100 machine) is the biggest advantage of an S-100 multi-processing system. Although the initial cost of starting an S-100 system is relatively high compared to an equivalent "starter" machine (e.g., Kaypro or Xerox), the benefits of multi-processing quickly become apparent when additional users are added to the S-100 frame.

After setting up the main S-100 frame with the required operating system, the net cost of each additional user in a multi-processing S-100 system can be as low as $300, which is significantly cheaper than anything else around.

Unlike many popular multi-user systems (that is, systems that share a single processor among all of its users), multi-processing systems don't bog down with heavy user demand. Each processor is available exclusively to its user, so no CPU time-sharing is necessary.

The only thing that can slow down a multi-processing S-100 machine is heavy use of its shared resources, especially its disk drives. In my experience, this has not been a problem with TurboDos-based machines.

There are great advantages to being

(continued on page 11)

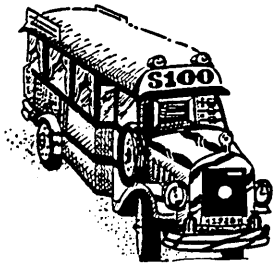
(continued from page 9)

able to perform multiple jobs at the same time, even in a home computer. If you have an S-100 machine and want to expand your computing power without taking out a new mortgage, look into multi-processing.

Next Time

In the next S-100 Bus, we'll continue examining multi-processing, and look at some alternatives to it that can run in an S-100/IEEE-696 environment, including some ways to simultaneously run different types of processors in the same S-100 box.

I look forward to reader mail with questions, suggestions, hints, and tips about the S-100 Bus.



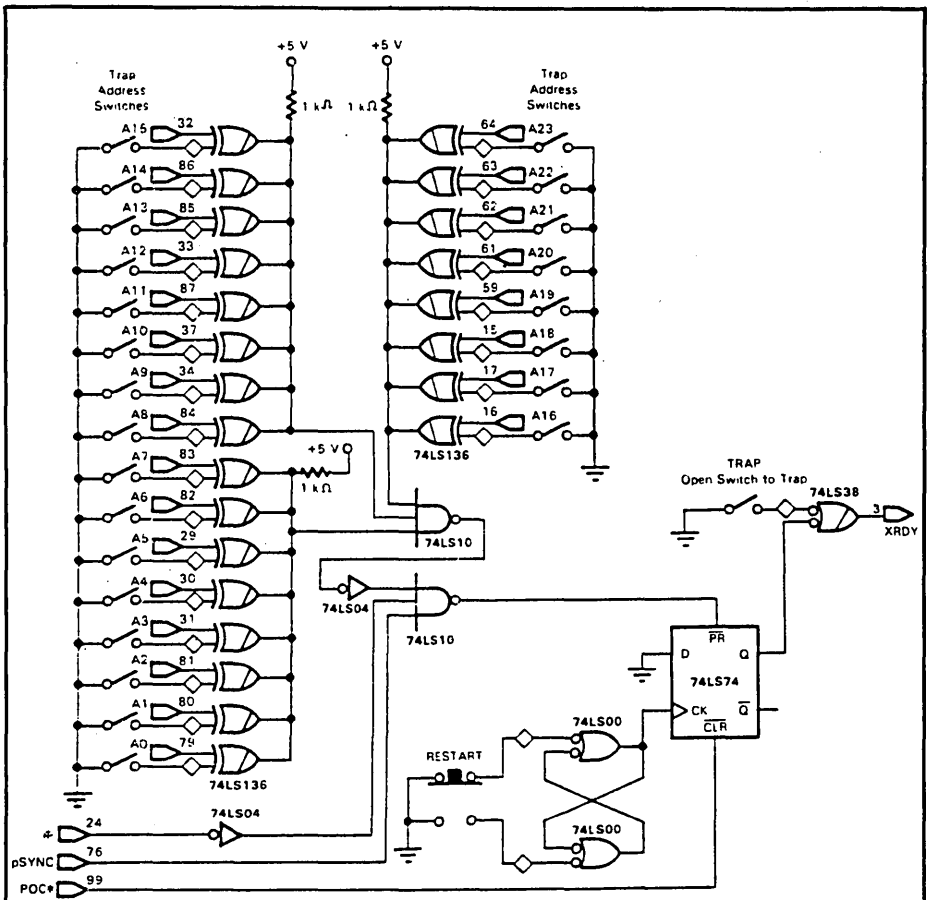


Figure 3 - An S-100 Breakpoint Trap Circuit

In The Public Domain

By Sol Libes

Box 1192
Mountainside NJ 07092

Call me crazy. I'm back in the magazine publishing biz—something I swore I would never do again. After Ziff-Davis closed Microsystems magazine I was depressed for weeks. Countless letters and phone calls from subscribers (there were over 31,000 plus another 25,000 newsstand) made me feel even worse. Everyone kept urging me to start it up again.

I kept remembering what my wife, Lennie, and I went through when we started Microsystems in late 1979, and I thought, "No, not again." After all, we wanted to live normal lives. But there has been something missing from my life the last several months. The passing of Microsystems left a void. There was no other magazine catering to advanced micro users the way Microsystems did.

Ziff-Davis would not sell Microsystems back to us, so we decided to start all over again at square one—down in the basement, on the ping-pong table.

This means, I regret to say, that this is my last column in Micro C—for a while, at least. I think Micro Cornucopia is a terrific magazine. However, publishing and editing my own magazine is very time consuming, so I must withdraw from other commitments wherever I can.

Adventures In Publishing Land

Let me tell you about some of my adventures in the world of big time magazine publishing.

Six years ago I realized there was a need for a magazine for users of CP/M and S-100 systems (this was long before Kaypro, Xerox, and Osborne entered the marketplace). I tried to interest several magazine publishers in the idea, but they all just laughed. I believed in the need for such a magazine, however, so I decided to do it myself. The first issue of Microsystems came out in January 1980.

In late 1980, it was apparent that the magazine was becoming a big business. I decided to sell it to Creative Computing and remain as the editor. Microsystems was published six times a year, and thus left me with time to pursue my other interests. My wife and I returned to our primary occupations at the community college where I teach electronics and she teaches math.

My avocations included being founder and president of the Amateur Computer Group of New Jersey (1,400 members strong), and being active in the SIG/M and PC/Blue Public Domain Software Libraries. I also kept myself occupied by writing a column in Byte magazine and writing a few books (e.g., Interfacing To the S-100/IEEE-696 Bus, published by Osborne McGraw-Hill).

In late 1981, Ziff-Davis, which published 60 magazines at the time, bought out Creative Computing, and as part of the deal acquired Microsystems magazine. Microsystems had a circulation of about 25,000 and was a small (but prestigious) book in an organization in which most magazines had circulations of well over 100,000 and as high as 600,000 (e.g., "Popular Electronics," which changed names last year to "Computers & Electronics," and which Z-D also canceled this year).

Z-D decided to invest in Microsystems to try and make it into another "big seller." We went monthly, and our staff went from two people (myself, working part-time, and an assistant) to ten people (four of whom sold advertising). We continued to grow in circulation and size. From a 90-page issue published bimonthly, we expanded to 180 pages monthly, and reached a circulation of over 55,000. (I was amazed that we reached this high a circulation. After all, how many advanced micro users are there?) But this was just not enough for Ziff-Davis, and in late '84 they decided to close Microsystems down.

Have You Ever Wondered About . . .

Having spent two years in the world of big magazine publishing, I have learned something about the publishing biz. I could probably write a book on the subject, but I thought I would give you a few insights here.

Have you ever noticed that many magazine publishers offer huge discounts to get new subscribers, and then seldom, if ever, offer discounts to subscribers who renew? In other words, once they suck you in they zing it to you. Also, do you realize how far in advance you've been receiving your renewal notices? I recently got a magazine renewal notice 12 months before it was due to expire!

A Few Other Tidbits

Have you ever wondered why so many publishers say "allow 6-8 weeks before you receive your first issue"? And why so many have a subscriber address in Boulder, Colorado, while their offices are actually somewhere else? Let me tell you some of the reasons.

(continued next page)

(continued from page 13)

First of all, most of these publishers use a subscription service. This turns out to be much cheaper for them (if they have 100,000 or more readers) and relieves them of the problem of dealing with subscribers.

The largest such company is A.C. Neilson, located (as if you didn't know already) in Boulder, Colorado. They are highly automated, dealing with hundreds of millions of subscriptions annually. They are more concerned with providing low cost service to publishers than they are with providing good service to subscribers. A publishing executive once told me that they can live with a 20% subscriber complaint rate from the subscription service they used.

Most large subscription services send the subscription cards they receive out of the country (typically to the Far East) for keyboard entry. This usually introduces a 2-3 week turnaround time, causing most of the 6-8 week delay. Also, since most of the operators keying in the data are unfamiliar with our language, they make a lot of data entry errors. Needless to say, the data is rarely checked for accuracy.

Thus, if you want to avoid problems, type your name and address on the subscriber card, or print very clearly, and do not write any messages on the card—this just confuses the operators.

If your address label is incorrect, don't try dealing with the subscription service. Instead, write directly to the publisher. His or her name and address is usually on, or right after, the table of contents page. Always write to a specific person and complain loudly.

In Closing

For a sample copy of my new magazine, Micro/Systems Journal, send me $3 (cover price is $3.50) and I'll send it out first class. You will not have to wait 6-8 weeks. A subscription is $18 (1 year/6 issues) or $32 (2 years/12 issues). I, like Dave Thompson, do not use any publisher's gimmicks. Dave and I are publishers, not because we're out to make our first million, but because we believe there is a need for our magazines.

So long . . . and keep hacking!

David Thompson's note: I have mixed feelings about Sol's disappearance from Micro C. I've really enjoyed his interesting and easy-to-edit offerings, and he is definitely an authority on public domain software. However, Micro C is not Microsystems—they are different animals, both in style and content. There is still a need for Microsystems, especially as Sol moves strongly into MS-DOS and UNIX. You have only to read PC Tech Journal to see the need for another magazine in this arena.

I wish Sol and Lennie the very best.

■ ■ ■

C'ing Clearly

By Gary Entsminger

Micro C Staff

C'ing Clearly will take a slight detour this time from its usual path to examine computer recreation with a practical twist. Next trip we'll return to C'ing seriously.

Two subjects: "The C Puzzle Book" by Alan Feuer and pre-processor macros.

Macroing In The C Puzzle Book

Tony mentioned he'd be reviewing *The C Puzzle Book* but he didn't get to it, so here goes. Obviously, the book is not new—it's been around since '82, but it still offers insight into this high-level, low-level language, and illustrates the use of the macro.

To sum it up quickly, it's a fun workbook for "The C Programming Language" by Kernighan and Ritchie. It's written in a very similar style, which means it's a little stiffer than it needs to be, but very informative. And recreational. (If you like puzzles, you'll love this.)

The puzzle in Figure 1, entitled "The Pre-processor Doesn't Know About C," should give you a feeling for the book's flavor and illustrate the dangers of parameterized macro processing. But first, a little background on macroing (or how to keep yourself in knots while eating brown rice).

Macro Processing

Every C compiler has a pre-processing phase that alters source code before passing it on for compilation. Its two most important functions are macro substitution and file inclusion.

Macros can improve a program's readability and efficiency, and can be handy as building blocks for parsing command line arguments, debugging large programs, and writing compilers.

The fundamental macro allows no arguments, and simply substitutes a token for a name:

```
#define X 25
```

where "X" is the name and "25" is the token (or substitution). On this level a macro isn't much different from a constant definition, except both name and token are character strings.

Argumentative Macros

It's harder to write a macro processor that allows arguments. Neither the Code Works' Q/C, Small C, nor Software Toolworks' C/80 supports parameterized #defines (Aztec C does). But since all three of these compilers provide compiler source code, you could expand them to allow arguments. Here's the form:

```
#define indentifier(indentifier, ... ,
   indentifier) token-string
```

But beware! Parameterized #defines are tricky.

When the compiler sees a macro call, it places the name and definition (or translation) into an evaluation area.This area looks like a stack. All arguments to the macro are also placed in this area unless the argument is itself a macro. (This is a nested macro—note that macros rarely nest in captivity.)

When the compiler sees a nested macro it creates a new stack, and the inner (new) macro is evaluated completely. Its output is then placed on the original stack, and work is resumed on the outer macro. The outer macro never sees the inner one, just its translation. Of course, the inner macro may have called other macros. (This process of calling oneself is called recursion.)

Conditional Macroing

A third level of macro processing allows conditional #defines of the form:

```
#ifdef indentifier
...
...
#else (optional)
...
#endif
```

A conditional #define checks to see whether a name has been defined, and if it has, compiles designated parts of the program. (Q/C, C/80, Small C, and Aztec C permit conditional #defines.)

C'ing Pascal Clearly

Macros have few grammatical restrictions, so if you were a Turbo Pascal programmer by nature and wanted to retain your favorite stylistics, you could by #defining.

For example, you could use Pascal block delimiters in C.

```
#define then
#define begin {
#define end   ;}

and then

if (i > 0) then
        begin
                a = 1;
                b = 2
        end
```

Compiler control lines of the form:

```
#include ""filename''
```

are also available in C. The call, #include, replaces the line with the contents of the file "filename."

Turbo Pascal includes a similar compiler directive also called "include" which performs a similar substitution. Included files cannot be nested in Turbo Pascal, but can be with some C compilers—Aztec C is the only CP/M-80 compiler I know of that comes with #include nesting.

The Wrap Up

If you prefer textbooks with a serious bent (as opposed to seriously bent textbooks) then *The C Puzzle Book* is probably not your cup. But if you want to try your hand at some serious C play this might be for you. It's a reasonably priced $12.95, and a terrific learning tool. Available from Prentice-Hall.

The Challenge

An expanded pre-processor that handles macros would be a great addition to the Small C compiler—a great Micro C community project. Anybody up for it?

■ ■ ■

Figure 1 - The Puzzle

What does this program print?

```
#include <stdio.h>
#define FUDGE(k)        k+3.14159
#define PR(a)   printf("a= %d\t",(int) (a))
#define PRINT(a)        PR(a); putchar('\n')
#define PRINT2(a,b)     PR(a); PRINT(b)
#define PRINT3(a,b,c)   PR(a); PRINT2(b,c)
#define MAX(a,b)        (a<b ? b : a)

main()
{

        {
                int x=2;
                PRINT( x*FUDGE(2) );
        }

        {
                int cel;
                for( cel=0; cel<=100; cel+=50 )
                        PRINT2( cel, 9./5*cel+32 );
        }

        {
                int x=1, y=2;
                PRINT3( MAX(x++,y),x,y );
                PRINT3( MAX(x++,y),x,y );
        }

}
```

To solve the puzzle, expand the macros, working inside out. But be careful! Macros can be tricky.

```
int x=2;
PRINT( x*FUDGE(2) );    To understand the effect of a
                        pre-processor macro, expand it
                        in place.

PR(a); putchar('\n')    Always expand the leftmost macro.
                        First, substitute the macro replacement
                        string for the macro call.


PR( x*FUDGE(2) ); putchar('\n')

                        Then substitute the argument(s) in the
                        call for those in the replacement string.

printf("a= %d\t",(int)(a))

                        Expand the leftmost macro, PR this time.

printf(" x*FUDGE(2) = %d\t",
        (int)(x*FUDGE(2)))

                        Substitute the macro arguments.

printf(" x*FUDGE(2) = %d\t",
        (int)(x*k+3.14159))

                        A macro name that occurs between quotes
                        is not expanded. However, macro
                        arguments are expanded wherever they
                        occur in the macro body. Thus,
                        x*FUDGE(2) replaces a in the macro
                        PR, but FUDGE(2) is left unexpanded
                        in the format of the call to printf.


(int)(x*2+3.14159)      Replace the formal parameter k by the
                        actual parameter. Surprise! First
                        multiply, then add (then truncate).
```

Nabbed by a parenthesis. The unwanted interaction between the replacement string and its context in this problem is avoided if FUDGE(k) is defined to be (k+3.14159).

End of Listing

J. B. FERGUSON, INC.

817-640-0207

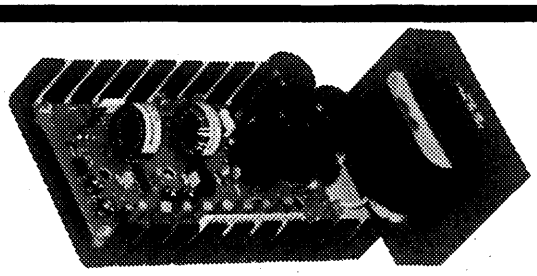
OTRONA 8:16 8086-2 co-processor computer board. Recently, Otrona Advanced Systems Corp. went bankrupt and we were able to acquire nearly all the stock of their 16-bit boards. These boards will operate in a stand-alone configuration. These boards have 256K bytes of 150ns 64K parts, an 8086-2, a socket for an 8087, a socket for a TI9914 IEEE 488 controller, a socket for a Z8530 serial communication controller, two sockets for 2732's or 2764's, and all the TTL to make it work. We have machine-drawn schematics and a good theory of operation for this equipment. These boards are to be sold as is. The dimensions are 4.5" x 10.5".

8:16 with 8086-2 ..... $ 65.00
8:16 without 8086-2 ..... $ 45.00

XEROX 820-1 ..... $ 45.00

I just purchased the entire final run of the 820-1 computer boards. These boards are complete with the exception of the ROMS and the 1771. These boards have all the Z80 I/O devices, all the crystals, all the connectors, the baud rate generator, the RAMs, and all the TTL. Compare this to B.G.'s $30. bare board!! These boards are to be sold as is.

Crydom S312 solid state relay, 120 VAC, 2 A, 0 volt switch ..... $ 10.00
820-1 / BIGBOARD-1 ASCII keyboard, w/cable, NEW! ..... $ 55.00
Bigboard serial I/O cable ..... $ 15.00
CP/M Primer (Xerox CP/M handbook) ..... $ 7.50



IMPROVED!

UNINTERRUPTABLE POWER SUPPLY

UPS 5-200 **$195.00**

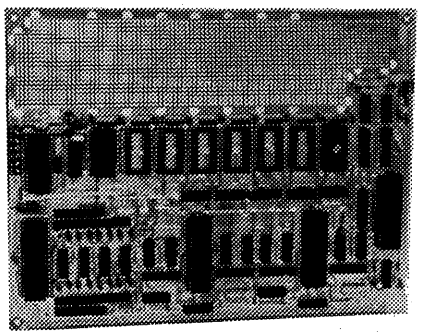
+5V @ 8.5 AMPS, +12V @ 7.5 AMPS — 12V @ .5 AMP
TWO +24 VOLT OUTPUTS @ 1.5 AMPS EACH (2A PEAK)

This supply is designed to power entire computer systems with Winchester and floppy disk subsystems. When the AC power fails, power can be maintained by a single 24V battery. The switchover time from AC to battery is zero, power is never interrupted.

24V 4.5 AH Panasonic LCR series battery $65.00

- High Reliability (1 year limited warranty)
- High Efficiency (75% typ)
- New design is very quiet.
- Quick response OVP
- Fold-Back current limiting
- Drop-out immunity for up to 250 mS (without battery)
- Uninterruptable (with addition of battery pack)
- Built-in battery charger
- Very Small 3.5"H, 10.0"L, 4.25"W

GENERAL PURPOSE INDUSTRIAL CONTROLLER



LB-1

$75.00

Bare Board & Doc

- Uses Z80,Z80A, or Z80B family of components
- Dual RS232 serial I/O W/Current loop option on the "A" channel (both strappable as terminal or modem in the RS-232 mode)
- 16 programmable Baud rates on each serial channel
- 32 bits of buffered parallel I/O assignable on 4 bit boundaries, W/Strappable Handshake Polarity (2xZ80-PIO)
- 4 Channels of counter-timer, 4 inputs, 3 outputs
- Accommodates 16K mix of RAM or ROM or EEPROM on 2K boundaries in 2716 or 2016 or Xicor EEPROM
- 2.3" x 10" sea of holes prototyping area on .1" grid
- All I/O is mode 2 interrupt compatible
- Watchdog timer for long-term unattended operation

ROM monitor and source listing ..... $ 35.00
Assembled and tested 4 Mhz LB-1 ..... 250.00

FERGUSON ENGINEERING
817-640-0207
P.O. BOX 300085 ARLINGTON, TEXAS 76010

TERMS: Master Card/VISA add 3%, money order, certified check, UPS cash COD. Allow 2 weeks for personal or company checks to clear. Shipping handling extra. Texas order add 5% tax.

The Xerox 820 Column

By Mitchell Mlinar

1225 Fonthill Ave.
Torrance CA 90503

It's official: Xerox has announced it is halting production of the 820-II and the 16/8. They say there are enough of these models in stock to last quite a while. Of course, "a while" means "when an IBM compatible appears" (what else?!?).

According to my sources, Xerox has been talking to Olivetti. Since Olivetti makes the AT&T personal computer, it seems likely that Xerox is cooking up an IBM compatible. (Editor's note: Their first model will probably have single-sided, single density drives, and run at 2.5MHz. The software package will be MS-DOS and a built-in typewriter function.)

Xerox has said it will still support the Xerox 820-II and 16/8 in service and some software. (I wonder what that means! Incidentally, Xerox no longer supports or sells 820-I software. Absolutely NOTHING is available from them.)

Surplus Boards

Up until now, there have been plenty of 16/8 boards on the surplus market at $50 and 820-II boards for $175. Now is the time to get them before there are no more left. Xerox 820 boards are readily available for $35-50, and there are more of them than either the -II or 16/8. Take your pick, but pick fast!

There is only one problem: the 820-II (16/8) requires either a floppy daughter board or a rigid daughter board. Rigids are no problem from the Dallas surplus outlet, but floppy daughters are almost impossible to find. If anybody knows a good source, please let me or Micro C know.

16/8

Thanks to an unnamed supplier, I am now the proud owner of CP/M-86 and MS-DOS for my 16/8 board. As I dig into it, I will let you know what I find. I do have one immediate comment: be careful before plunking your money down. Although the operating systems are generic, there is little generic software in CP/M-86 and even less in MS-DOS. CP/M-86 programs should run without a problem, but the amount of CP/M-86 software available is limited. MS-DOS has plenty of software, but it is strongly IBM flavored. The 16/8 can read/write IBM disks, which is a definite plus (and practically a requirement). But if you want it just to have it (the price was right in my case), don't expect the broad base of software that's available for CP/M-80.

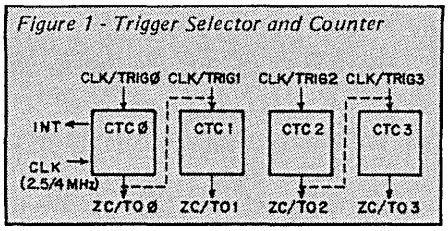
Z80 Support Chips

This column will begin a series on three chips that every 820 has, but which are not always understood: the CTC, PIO, and SIO. I'll concentrate mainly on operation and programming, with an occasional reference to a specific I/O pin. If you are an experienced programmer, you'll enjoy this discussion. If you are not experienced, then hang in there, you might learn a few things.

Z80 CTC

The Z80 CTC (Counter-Timer Circuit) is a special purpose chip which has four programmable counter/timer channels. Each of these channels is independent of the others, although they can be interconnected for some applications (as in the 820).

Figure 1 - Trigger Selector and Counter



Each of the channels consists of a trigger selector and a resettable counter (see Figure 1). The trigger preselector sends a "decrement" signal to the counter, which keeps track of the count and generates any output signals. The trigger selector is detailed in Figure 2, showing the multiplexer (which selects between the CLK/TRIG or pre-scalar inputs) and the pre-scalar types (divide by either 16 or 256).

There are two ways to send a decrement signal to the counter from the trigger selector: via the pre-scaled clock, or directly from the CLK/TRIG input pin for that channel.

The Pre-scaled Clock

The pre-scaled clock is the computer's clock (2.5MHz, for example) divided by either 16 or 256. This means that a decrement signal for a 2.5MHz clock (400 ns cycle time) can occur either every 6.4 us (400 ns x 16) or 102.4 us (400 ns/256). With a 2.5MHz clock, the CTC timer decrement trigger can occur either 156,250 times per second or 9,765 times per second.

The second way to send a decrement signal to the counter is directly from the CLK/TRIG input pin. Depending on the programming, a decrement will occur every time this pin sees a rising or falling edge (there is no pre-scaling).

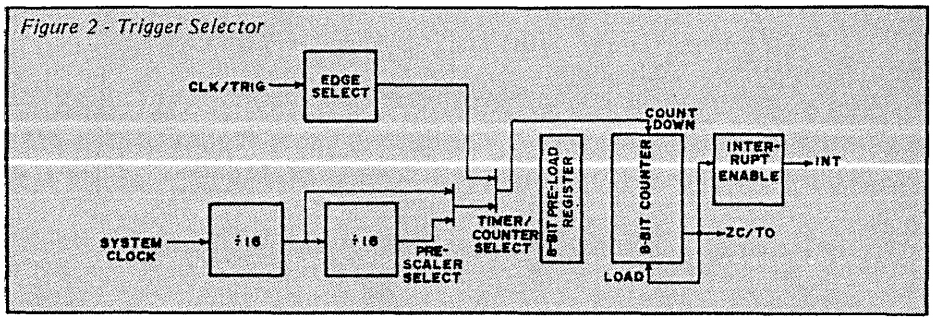
The counter portion of the circuit is resettable, and it counts to zero from a preset value. The value can be anything between 1 and 256. (Since an 8-bit data path really only has values from 0 to 255, the CTC treats 0 as 256.)

The counter deducts one for each decrement signal received from the trigger selector. When zero is reached, a momentary high signal is sent out the normally low ZC/TO (Zero Crossing/Time Out) output pin, and an interrupt is generated for that channel (if interrupts are enabled). Then the counter resets itself to the starting value, and begins counting down all over again.

One nice feature of the ZC/TO output pin is that you can connect it to the CLK/TRIG pin of another counter/timer circuit as is done in the 820 (shown by dotted lines in Figure 1). This daisy-chaining increases the timer period.

(continued next page)

Figure 2 - Trigger Selector



(continued from page 19)

What Does All This Mean?

About now you are saying, "So what?" (or "Whew! That was a mouthful."). Why do we need a timer at all? There are many reasons, but let's take an example we all know and love—the Xerox 820. It uses a one second interval timer for determining disk deactivation and for a real-time clock. When no disk activity has occurred for about 10 seconds, the drives are deselected to extend head life (a software routine counts to 10).

As with any clock, a time-base is required (like a quartz crystal on a watch) and the system clock (at either 2.5MHz for the -I or 4MHz for the -II) does fine. As I said earlier, even by using the timer pre-scalar, the best you can do is get a decrement signal about 9,765 times per second. Since the decrement signal is connected to the counter portion, which has a maximum count of 256, this single channel will give a "tick" about 38 times per second (9,765/256) instead of once per second.

Getting Out The Ticks

There is a way out. By daisy-chaining so one timer/counter drives another timer/counter, it is possible to get down to one tick per second. (Figure 1 shows the 820 connections between CTC channels as dotted lines.)

Setting up the individual CTC time equations:

```
CTCa = (400 nSec x 256) x COUNT1
   (using the 256 pre-scalar)
CTCb = CTCa x COUNT2
   (using the CLK/TRIG pre-selector)
```

Taking the one second "tick" off of CTCb,

```
tick =
(400 nSec x 256) x COUNT1 x COUNT2
```

Now, we select values of COUNT1 and COUNT2 to get as close to one second as possible. The Xerox monitor chose COUNT1 = 105 and COUNT2 = 93. The question is, are these the best values?

Just grabbing values will probably give a solution, but mathematics is useful (sometimes). Since (400 nSec x 256) = 0.0001024 seconds, then COUNT1 x COUNT2 = (1/0.0001024) or 9765.625. Since we are stuck with integers, we either round up to 9766 or down to 9765. Interestingly enough, 9765 = 105 x 93.

What about 9766, which is actually the closer choice? Well, prime factoring 9766 gives 2x2x19x257. Clearly, 257 cannot be used in our counters (argh, missed by 1!); hence, 9765 must be used. In any daisy-chained CTC, taking the prime factors is the best way to determine what values, if any, are possible.

Programming The CTC

Programming of the CTC is fairly easy. Since each CTC channel is independent, there is an I/O port associated with each of the four channels. These are:

```
CTC0:  18H  (24 decimal)
CTC1:  19H  (25 decimal)
CTC2:  1AH  (26 decimal)
CTC3:  1BH  (27 decimal)
```

Programming a channel consists of writing two words to the appropriate port. The value applies only to the specific channel addressed. The first word is a byte value as shown in Figure 3.

The second word (if any) is the time constant value selected.

The Trigger bit value applies only to "timer" mode and tells how to start the timer. Usually "automatic start" is selected, unless the interval timer must start at a prescribed time or event AND something is connected to the CLK/TRIG pin.

The Reset bit is only 0 when re-programming an operating channel, but I won't go into that here.

The Vector bit is 1 when programming a channel and is ONLY 0 for Channel 0 when setting the interrupt vector (for the Z80 IM2 mode).

It is possible to complete the Xerox example with the actual bit assignments (assuming that the IM2 vector has already been set). In the example using CTCa and CTCb, Xerox assigned CTC2 to CTCa and CTC3 to CTCb.

The first word is sent to CTC2, which is the 256 pre-scaled timer with a value of 105 for the counter.

```
MVI A,27H<which is 00100111 binary
OUT 1A   <send to CTC2 port
MVI A,105<the counter value in decimal
OUT 1A   <send this to CTC2, too
```

The value of 105 is clear, but what about 27H? Using the bit designation shown earlier, this means there is no interrupt (don't want one here, but at the end of the daisy-chain); timer mode; 256 pre-scalar; falling edge (does not apply here, so could be either value); automatic trigger (nothing is ever going to come into the CLK/TRIG pin unless something is connected to it); time constant follows (to give it the 105); software reset; and it is a control word.

Programming CTC3

The first counter is now happily humming along and generating a signal about once every 0.011 seconds. CTC3 needs to be programmed now.

```
MVI A,0C7H<which is 11000111 binary
OUT 1B    <send to CTC3 port
MVI A,93  <counter value for this channel
OUT 1B    <send it, too
```

Figure 3 - Programming The CTC

```
bit:    7 6 5 4 3 2 1 0  (highest is bit 7, lowest is bit 0)
WORD1:  I M P E T C R V

where
I(nterrupt) = 0 to disable interrupt
              1 to enable interrupt
M(ode)      = 0 for timer mode (use system clock)
              1 for counter mode (use CLK/TRIG pin)
P(rescalar) = 0 for divide by 16  <ONLY works in>
              1 for divide by 256 <timer mode.  >
E(dge)      = 0 for falling edge <ONLY works in>
              1 for rising edge  <counter mode.>
T(rigger)   = 0 for automatic start of timer
              1 for CLK/TRIG pulse to start timer
C(onstant)  = 0 if no time constant follows (rare)
              1 if time constant follows (usual choice)
R(eset)     = 0 for continued operation (rare)
              1 for software reset (usual choice)
V(ector)    = 0 for vector (Channel 0 only)
              1 for control word (usual case)
```

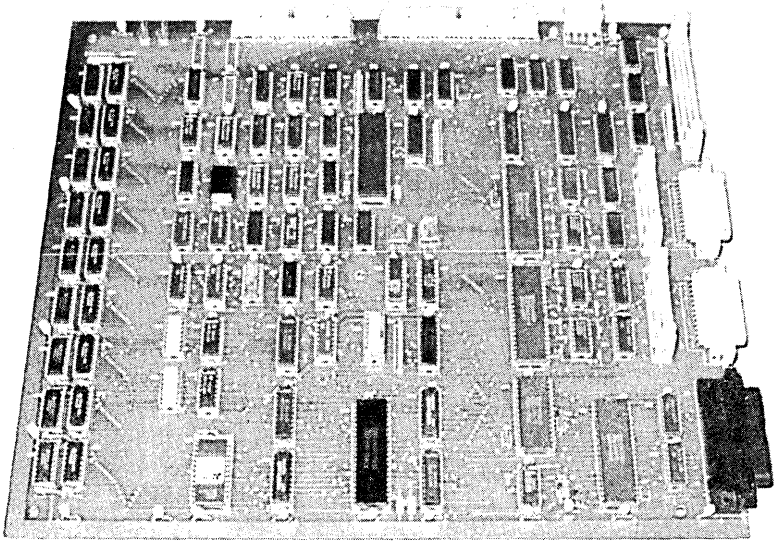
Again using bit designation, we enabled interrupt for this channel (to tell the rest of the computer about the one second tick); counter mode (the signal comes from the CLK/TRG pin connected to the ZC/TO pin of the previous channel); pre-scalar 16 (does not apply); falling edge (could also be rising edge in this case); automatic trigger (does not apply); time constant follows (the counter value); software reset; and control word. The value of 93 is the counter value.

A full-blown interval timer is now active which flags the microprocessor every second. This flag is known as an interrupt, and I'll talk more about that next time.

■ ■ ■

Colonial Data SB-80-II

The "Better Board" Gets BETTER!



$298.00
ASSEMBLED & TESTED
NOT A KIT!

Standard Features Include:
- 4MHz Z80A Processor
- 64K Memory (Expands to 128K)
- Supports 8" and 5¼" drives simultaneously
- Auto Density Selection (single/double/quad)
- 2 Serial Ports/1 Centronics Parallel Port

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!

SIZE: 12" x 13"
POWER: +12V, −12V, +5V
WARRANTY: 90 days Parts and Labor

■ 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 5¼" drives simultaneously
Automatic Density Selection (Single/Double/Quad)

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

■ 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.)

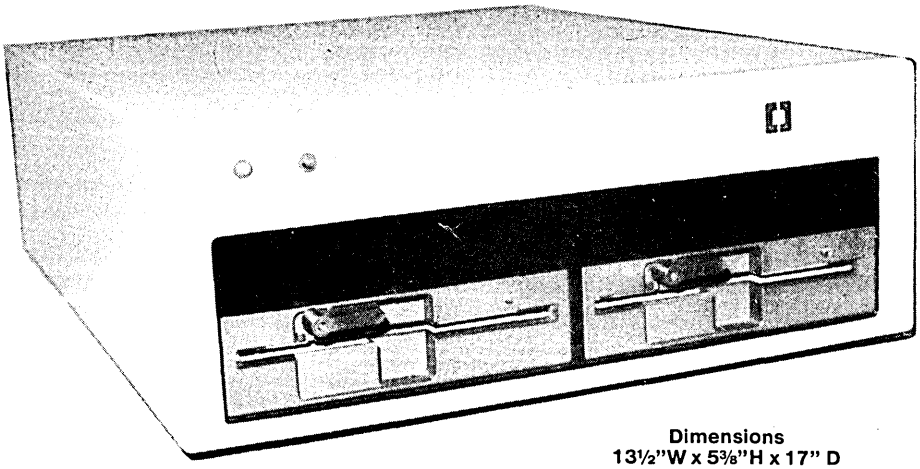
SB80-II SYSTEM OPTIONS:	PRICE
ADDITIONAL 2 SERIAL PORT (INSTALLED)	$ 60.00
ADDITIONAL 64K MEMORY (INSTALLED)	$ 60.00
SASI INTERFACE OPTION (INSTALLED)	$ 35.00
XEBEC HARD DISK CONTROLLER	$290.00
CP/M 2.2 OPERATING SYS W/MANUAL	$ 75.00
CP/M 3.0 OPERATING SYS W/MANUAL	$275.00

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.

MasterCard VISA

Colonial Data Services Corp.

80 Pickett District Road,
New Milford, Conn. 06776
Telephone (203) **355-3178**



Dimensions
13½"W x 5⅜"H x 17" D

FULL 90 DAY PARTS & LABOR WARRANTY

Colonial Data
SB-80-II

COMPLETE SYSTEMS!
4 MHZ Z80A CPU
64K RAM (EXPAND TO 128K)
DUAL 5¼" DISK DRIVES
2 SERIAL PORTS
1 PARALLEL PORT

MODEL IIA - 2/380K DSDD DRIVES	**$ 895.00**
MODEL IIB - 2/780K DSQD DRIVES	**$ 995.00**
MODEL IIC - 10 Meg H/Disk w/380K Floppy	**$1895.00**
MODEL IID - 10 Meg H/Disk w/780K Floppy	**$1995.00**

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

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

PICK YOUR SYSTEM! CHOOSE YOUR OPTIONS!

The Original "Better Board" SB80 from Colonial Data has just gotten BETTER YET! More on-board features like SASI interface circuitry, 4 Channel Counter Timer, Memory Expansion to 128K and MORE! All made to run the most popular programs under the CP/M Operating System.

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

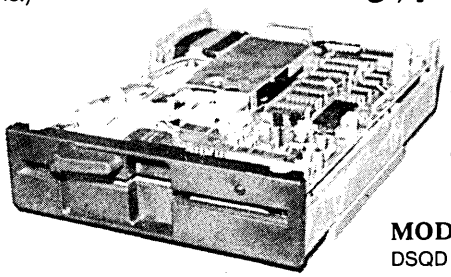
Colonial Data Cabinet Kits (Build Your Own Systems!)

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 floppies. 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**

Shugart 5¼" Disk Drives



MODEL 455
DSDD Floppy **$195.00**
Over 380K Capacity 2 For **$370.00**

MODEL 465
DSQD Floppy **$225.00**
Over 780K Capacity 2 For **$430.00**

O.E.M. Quantity Inquiries Invited! Call or Write for Details!

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 2% (up to a maximum of $20.00) for UPS (Brown) shipping within the United States.

Colonial Data Services Corp.

80 Pickett District Road,
New Milford, Conn. 06776
Telephone (203) **355-3178**

The Slicer Column

By Laine Stump

Micro C Staff

I was amused by Dave's admission in the February issue to listening to music while typing. For some time now, I have been listening to music while writing. There is a slight difference, though. While Dave is listening to crashing waves and harps, I am typing to the sound of decadent punk rock on the campus FM station (that's the only thing they play at 3 in the morning). And you all wondered what was wrong with me . . .

Give Me A Break

Finally, some first hand news about great new toys. After spending three days in Yellowstone Park in the middle of a snowstorm (spring break), I packed up my Slicer, my PC Expansion Board, and my Memory Expansion Board (and a genuine IBM keyboard and monitor borrowed from the local computer store) and holed up in a spare bedroom to play. I came out of the room with four things: an opinion, a Pascal program that uses the SC2681 UART chip to send and receive files, a story to tell, and a big, fat smile.

MS Is NOT A Disease

I don't know if I should admit this, but I played around with MS-DOS quite a bit, too. I'm not sure what to make of the current trend toward badmouthing MS-DOS. It has a few inconsistencies, but the concept is just wonderful. Besides, every mother's dog is using it, and that makes it a great development environment if you want to make a lot of money (open your eyes and smell the royalty checks, people . . .).

Anyone who has ever used UNIX long enough to get used to tree'd directories (like a tree'd raccoon, only not quite as cute), command search paths, and byte oriented files can appreciate my frustration with CP/M and my interest in MS-DOS. On the other hand, anyone who has ever tried to read the MS-DOS Programmer's Reference Manual (Hebrew/American edition) can understand why I still have reservations about MS-DOS.

Half of the manual is filled with documentation of version 1.x function calls that are annotated with "Don't use this function, use function z instead." If you want to do any assembly language programming for MS-DOS, you'll first have to spend $85 extra to buy MASM (or figure out where to get CHASM, a "freeware" assembler) and then take about a month off from everything else to decide which system call you REALLY want to use to open a file. I think I'll just stick to C and Pascal on MS-DOS (at least for now).

Back to my defense of MS-DOS. I haven't noticed any glaring problems with it (although I've heard others say they have), and it DOES exist and must be dealt with. The Slicer implementation is clean and well done (although disk accesses are slightly slower than a Slicer running CP/M-86), and I can think of no better way to ease the pain of entering the real world (you know, the one where you actually get paid money for your work) than to use MS-DOS on the Slicer as a development system.

Touchy subject. I won't say any more for fear of bodily injury.

The Problem

I have evaluation copies of several programs (editors, compilers) written for the IBM and wanted to spend my time "in hiding" checking out just how compatible the Slicer video board is. As luck would have it, all the programs were for MS-DOS and, although I have MS-DOS, all my pre-written source code is on CP/M disks. My commitment to the "never type anything twice" philosophy meant that I needed to transfer the files onto MS-DOS electronically, but I had no communications programs for MS-DOS and no utilities for reading CP/M disks on MS-DOS (or the other way, either).

The Solution

Since my Big Board was sitting sadly in the corner and I had Turbo Pascal for MS-DOS, I decided to write a simple communications program called SHIP (Figure 1) to send files out a serial port and receive files from a serial port. It would show a complete lack of spirit and enthusiasm to use one of the ports on the Slicer board that is already supported in the Slicer ROM, so I decided to use Port A of the SC2681 on the Expansion Board.

The Weapons

Some of you may balk at my choice of Pascal for writing a hardware control program, and I would agree if this were any old Pascal. But it's not. Turbo has a predefined array of type BYTE called "port." This is the programmer's gateway to the 8086 I/O space. It works very simply; to output a byte to an I/O port, just use the statement:

```
port[x] := y;
```

To input a value from a port use:

```
z := port[x];
```

Using Turbo Pascal and the port array makes writing I/O type routines trivial and allows you to easily add lots of bells and whistles (since you have all those predefined procedures for positioning the cursor, formatting output, etc.). I have used this feature of Turbo extensively for experimenting with new chips and have found it nearly invaluable. I almost always convert the final result to assembly language and put it in ROM, but Turbo shortens the investigation stage quite a bit.

SHIP includes the standard "Big Four" routines of interfacing: initialization, read, write, and status. Other than these four procedures, the rest of the program is totally hardware independent. This means that I can compile and run it without modification on the Slicer under CP/M-86 or MS-DOS, and I have to change only four procedures to run it on my Big Board.

Details Of The Quest

Before blindly spitting characters out a port, you must set it up for the mode of operation you want to use (bits/character, stop bits, interrupts, etc.). The SC2681 has more modes and features than I could describe in two columns, so I'll just talk about the important ones.

The first thing I do in the InitPort routine is turn off all interrupts. This is done by sending a byte of all 0s to the Interrupt Mode Register (IMR). Each bit in the IMR indicates that the SC2681 should generate an interrupt on a certain condition. Interrupts would just cloud the issue, so, for now, I'm not using them.

After turning off the interrupts, I set the number of bits/character, stop bits, and type of parity. These, and other modes, are controlled by the SC2681's two "mode registers." These registers are accessed by first setting the "mode pointer" to mode register 1 with a command sent to the command register, then outputting the two mode bytes in sequence to the mode register.

Finally, I set the baud rate. This involves two steps—selecting the baud rate set, and then selecting the baud rate within that set. The function of all the registers is covered in the SC2681 spec. sheet (you received a copy with your Slicer), so I won't go into any more detail here.

Due to time limitations, I chose to "hardwire" all of the initialization except for the baud rate. But the stop bits, etc. could all be set when running the program, just like baud rate. I leave this enhancement as an exercise for the reader (oh, how I Love to say that!).

On Speaking Terms

Once the port is initialized, input and output are simple. You just wait until the TxRdy (Transmit Ready) or RxRdy (Receive Ready) bit goes on in the status register, then output to or input from the data port.

To test these routines, I put in the Terminal procedure to allow the Slicer to be a dumb (?) terminal to the Big Board. After some fooling around with the initialization mode bytes, I had the Big Board and the Slicer talking.

Wait For Me

The Send and Receive procedures were simple extensions of the existing routines. Send worked with MODEM740 on my Big Board right away, but Receive lost characters while it was busy writing to the disk. Fortunately, MODEM740 has options for using XON/XOFF protocol during informal file transfers. I took advantage of this in Receive by saving each line of the file in a buffer, sending an XOFF, writing the line, then sending an XON to start the Big Board up again.

First I tried stopping every 128 bytes instead of every line, but I still lost characters, so I had to use a special mode of MODEM740 that automatically stops at the end of every line, waiting for XON. After I did this, I got perfect transfers every time.

Just to be consistent, I put XON/XOFF checking in Send, too. I automatically stop and wait for an XON at the end of a line. This is because I know the other end wants to stop at the end of the line anyway. If I didn't anticipate this, I could send some characters that would arrive after Receive had sent the XOFF, but before Send had received it. Receive would assume that it had stopped all transmissions and would then probably lose the extra characters. Not good.

The Finished Product

SHIP has a few problems. The worst are:

1. I must type at both keyboards to get anything done.
2. It only works with text files.
3. It has no error checking.

Since I need an MS-DOS modem program anyway, I'll probably end up adding XMODEM file transfer capability to it someday. For now, though, it does everything I need. I wanted to transfer source files from CP/M to MS-DOS, and I can. I have used it at 9600 baud without dropping characters.

As written, SHIP runs on port A of the expansion board, but this can be changed to any other SC2681 port by just changing the port address equates. I have used it on port A of the Slicer ($80-$8A), and nothing should prevent it from running on port B of either board.

Sidelines

Doing all this fooling around gave me a good chance to check out the Slicer PC Video Board. I tried out three different editors written specifically for the IBM PC, and all of them ran. There was a minor problem when scrolling backwards with the Turbo Pascal editor and Z (a VI clone included with Aztec C), but that problem will be corrected long before you read these words.

The real shining light of editors was the PC version of VEDIT, though. It is incredibly fast. The other editors took a small, though noticeable, period of time to repaint the screen after a "page" command; VEDIT seemed to respond instantly (like mashed potatoes, but more impressive and less filling). Rather then relying on ROM calls, VEDIT places characters directly into video memory. This makes the program very hardware specific, but using ROM calls is hardware specific, too. The best way to output a character is through an operating system call, but that isn't nearly as fast, and we must outdo the competition, mustn't we?

Of course, the main reason VEDIT is so fast is that the memory on the video board has a 16-bit data bus and is running at 8MHz. This makes it considerably (about 2 times) faster than any of the PC clones around. After three days I was thoroughly spoiled.

Out Of The Blocks

The first time I plugged everything in, I couldn't get the video board to sign on as the console device. But for once I had read the manual (several weeks earlier) before diving in. I remembered something about needing a jumper somewhere, so I took a quick scan back through the manual. I found I had to jumper pins 1 and 2 of JB2 (and UNjumper pins 3 and 4) on the main Slicer board in order to allow using the video board as the console device. After I made this change, everything worked just fine.

The manual states incorrectly that ESC> E clears the screen and homes the cursor. This is not so. The screen is cleared, but the cursor remains in the same place. The cursor must then be homed with ESC> H. This may seem like a trifling point, but if you tell SETUP only about ESC> E, many of the programs using ROM calls to clear the screen (e.g., SLIFORM and SETUP itself) will not work properly. I asked about this when I told Earl (Hinrichs) about the scrolling problem, and he said the clear command originally did home the cursor, but it turned out to be much more useful to allow clearing without homing, so he changed it. Anyway, remember to tell SETUP that the clear screen command is ESC> E ESC> H.

(continued on page 27)

Figure 1 - SHIP.PAS

```
Purpose: To allow simple communications between the Slicer and
another system capable of sending and receiving characters
on an RS-232 port.

Method:  In Receive mode, SHIP accepts characters from port A
of the Expansion Board and puts them into a file. This
continues until a key is struck (on the Slicer). The
file is then closed. No error checking can be done with
this primitive method, so you should check the file for
errors after you have received it.

In Send mode, SHIP simply opens the requested file and
sends it out port A of the Expansion board, one character
at a time until the entire file has been sent.

Terminal mode is mainly for debugging and making sure that
both ends are cooperating. Terminal mode just gets
characters from the keyboard and sends them to the other
end while receiving characters and displaying them on the
screen.

PROGRAM ship (input, output, workfile);

CONST

    XON  = ^Q;  { codes for stopping/starting character stream }
    XOFF = ^S;

    modereg    = $200;      { SC2681 mode }
    statusreg  = $202;      {    "   status }
    baudreg    = $202;      {    "   baudrate }
    commandreg = $204;      {    "   command }
    datareg    = $206;      {    "   data }
    ACRreg     = $208;      {    "   baudrate set }
    IMRreg     = $20A;      {    "   interrupt mode }

    TxRdy = 4;              { status mask for Transmit Ready }
    RxRdy = 1;              { status mask for Receive Ready }

    IMRBYTE   = $00;    { turn off all interrupts }
    COMBYTE   = $15;    { point to MR1, enable Tx & Rx }
    MODEBYTE1 = $93;    { use RTS/CTS, no parity, 8 bits }
    MODEBYTE2 = $1F;    { 2 stop bits }
    SET2      = $80;    { select baudrate set 2 }

    BAUD300   = $44;    BAUD600   = $55;
    BAUD1200  = $66;    BAUD2400  = $88;
    BAUD4800  = $99;    BAUD9600  = $BB;
    BAUD19200 = $CC;

VAR
    workfile : text;

    { ct & ct2 are global so they will be static variables  }
    { this is because static variables are sometimes faster }
    ct, ct2 : integer;
    buffer  : array[1..2048] of char;
    baudrate : integer;
    bits7    : boolean;
    selection : char;

{-----------------------------------------}
{ initialize port for reading and writing }
{-----------------------------------------}
PROCEDURE initport ( baudrate : integer );

    begin
    port[IMRreg]     := IMRBYTE;    { turn off ints }
    port[commandreg] := COMBYTE;    { point to modebyte1 }
    port[modereg]    := MODEBYTE1;  { set modes }
    port[modereg]    := MODEBYTE2;
    port[ACRreg]     := SET2;       { use baudrate set 2 }
    CASE (baudrate) OF              { set requested baudrate }
        300   : port[baudreg] := BAUD300;
        600   : port[baudreg] := BAUD600;
        1200  : port[baudreg] := BAUD1200;
        2400  : port[baudreg] := BAUD2400;
        4800  : port[baudreg] := BAUD4800;
        9600  : port[baudreg] := BAUD9600;
        19200 : port[baudreg] := BAUD19200
        end     { case baudrate }
    end;

{-----------------------------------------}
{    return TRUE if char ready, FALSE if not    }
{-----------------------------------------}
FUNCTION ReadPortStat : boolean;

begin
ReadPortStat := ((port[statusreg] and RxRdy) <> 0)
end;    { ReadPortStat }

{-----------------------------------------}
{ read a byte from port and return it to caller }
{-----------------------------------------}
FUNCTION ReadPort : char;

    begin
    REPEAT UNTIL (ReadPortStat);    { wait for char }
    IF bits7 THEN
        ReadPort := chr(port[datareg] and $7F)
    ELSE
        ReadPort := chr(port[datareg])
    end;    { ReadPort }

{-----------------------------------------}
{           write a byte to port          }
{-----------------------------------------}
PROCEDURE WritePort ( thisbyte : char );

    begin
    REPEAT UNTIL (port[statusreg] and TxRdy) <> 0;
    IF bits7 THEN
        port[datareg] := ord(thisbyte) and $7F
    ELSE
        port[datareg] := ord(thisbyte)
    end;    { WritePort }

{ - - - beyond here is hardware independent - - - }

{-----------------------------------------}
{    prompt for a baudrate and init the port    }
{-----------------------------------------}
PROCEDURE SetBaud (var baudrate : integer);

var yn : char;

    begin
    writeln;
    write ('Baudrate: ');
    readln (baudrate);
    InitPort(baudrate);
    write ('Strip high bit? ');
    read (kbd,yn); writeln (yn);
    bits7 := (upcase(yn) = 'Y')
    end;    { SetBaud }

{---------------------------------------}
{   send chars typed at console to port }
{   while echoing received chars        }
{---------------------------------------}
PROCEDURE Terminal;

VAR done : boolean;
    ch   : char;

    begin
    writeln('Terminal Mode, baudrate is ',baudrate);
    writeln('Type control+_ to end');
    writeln;
    done := FALSE;
    REPEAT
        IF (KeyPressed) THEN    { char typed ? }
            begin
            read(kbd, ch);
            IF (ch = ^_) THEN
                done := TRUE
            ELSE
                WritePort(ch);  { send it }
            end;    { if keypressed }

        IF (ReadPortStat) THEN  { char received ? }
            write(ReadPort)     { display it }
        UNTIL (done);
    end;    { terminal }

{---------------------------------}
{   receive a file from the port   }
{---------------------------------}
PROCEDURE Receive;

var filename : string[80];

    begin
    write('Name for Received file: ');
    readln(filename);
    assign(workfile, filename);
    rewrite (workfile);
```

```
    write ('Start sending from other end, ');
    writeln ('press a key on this keyboard when done');
    ct := 0;
    WHILE (not Keypressed) DO
        IF (ReadPortStat) THEN
            begin
            ct := ct + 1;
            buffer[ct] := ReadPort;      { save in buffer }
            IF (buffer[ct] = ^M) THEN
                begin
                WritePort(XOFF);         { turn off other end }
                FOR ct2 := 1 to ct DO    { dump buffer to file
                    write (workfile, buffer[ct2]);
                ct := 0;
                WritePort(XON);          { turn back on }
                end;     { if received char = ^M }
            end;     { if character ready }

    IF (ct > 0) THEN              { write out partial line }
        FOR ct2 := 1 to ct DO
            write(workfile, buffer[ct2]);
    close (workfile)
    end;     { Receive }

{--------------------------------}
{    send a file out the port    }
{--------------------------------}
PROCEDURE Send;

    var ch : char;
        filename : string[80];
    begin
    write('Name of File to Send: ');
    readln(filename);
    assign(workfile, filename);
    reset (workfile);
    write ('Set up other end to receive, ');
    writeln ('press a key on this keyboard when ready');
    REPEAT UNTIL (Keypressed);
    WHILE (not EOF(workfile)) DO
        begin
        IF (ReadPortStat) THEN       { check for XOFF }
            IF (ReadPort = XOFF) THEN
                REPEAT UNTIL (ReadPort = XON);
        read(workfile, ch);          { get a char }
        WritePort (ch);              { send it }
        IF (ch = ^M) THEN            { IF eoln wait for XON }
            REPEAT UNTIL (ReadPort = XON)
        end;     { while not eof }
    close (workfile)
    end;     { Receive }

{---- main ----}

begin
SetBaud (baudrate);
REPEAT
    writeln;
    write ('<B>audrate, <T>erminal, <S>end, <R>eceive, <Q>uit: ');
            read(kbd,selection); writeln(selection);
            CASE (upcase(selection)) OF
                'B' : SetBaud (baudrate);
                'T' : Terminal;
                'R' : Receive;
                'S' : Send;
                'Q','E','X' : ;
              ELSE
                writeln ('Bad Option, Try Again');
              end   { case selection }
            UNTIL (upcase(selection) in ['Q','X','E']);
        writeln ('Bye now, have a good afternoon.')
        end.
```

End of Listing

THE SLICER COLUMN

(continued from page 25)

C Sick

The underlying reason for writing SHIP (besides giving the PC board and the expansion board a workout) was to send over some C programs that I had on CP/M to compile them with different C compilers on MS-DOS. Unfortunately, I spent so much time writing SHIP that I never got the time to do any C work. Next time for sure, though. I have been having too much fun writing C programs lately to pass up such a great opportunity. Yes, that's right. I have (gasp!) C Sickness! (That's it, Martha! Pack up the kids, we're getting OUT of here!!) I'm sorry, Philippe . . .

Now that I have an expansion board to play with, I would also like to figure out how to use its other two serial ports (Zilog Z8530). Naturally, this chip isn't included in the Zilog manual I have, and for some reason, I didn't get any sheets for it with the expansion board. Maybe I can get some info on that soon, though.

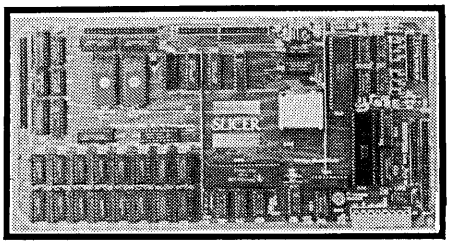
Self Congratulations

About the time you're reading this, I'll be graduating from college. I would like to take this opportunity to tell myself what a fine job I've done and wish myself well in the future. (Thank you.) (You're welcome.) I just hope I don't lose my humility after I leave school. *(Editor's note: I'm speechless.)*

■ ■ ■

SLICER. . . Growing to meet your needs!

The Slicer

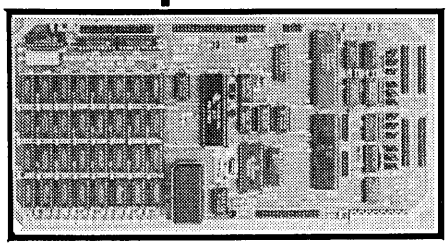


REAL 16-BIT POWER ON A SINGLE BOARD
Featuring the Intel 80186

- 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 . . . . April Special $895

The Slicer System Expansion Board



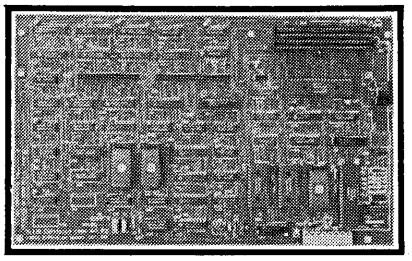
FOR EXPANDED MEMORY, ADDITIONAL PORTS, AND REAL TIME CLOCK

- Up to 256K 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 . . . . April Special $650

The Slicer PC Expansion Board

New!!



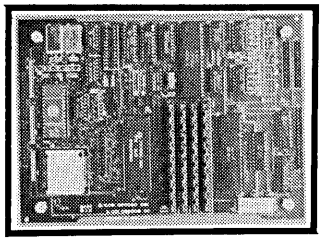
GIVES YOUR SLICER HIGH PERFORMANCE VIDEO CAPABILITY

- 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

The μ Slicer 188

New!!



REAL 8/16-BIT POWER ON A SINGLE BOARD
Featuring the Intel 80188

- Complete microcomputer on high quality 5-3/4" x 7-3/4" multilayer board
- 256K RAM, plus up to 8K 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 5-1/4" and 3-1/2" 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 $700

SLICER™

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

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.

SLICER Special April Prices!

(Continued through June 30)

Slicer Single Board Computer

	April Specials!
Assembled and Tested 8 Mhz	$895!
Full Kit	$715!
Easy Kit	$400!
Bare Board	$125!
New!! DOS KIT includes Bare Board, CPU and choice of MS-DOS or CP/M-86	$295!

The Slicer System Expansion Board

	April Specials!
Assembled and Tested	$650!
Full Kit	$450!
Memory Board Kit	$300!
3-Port Kit	$200!
Bare Board	$75!

The Slicer PC Expansion Board

Assembled and Tested	$600
Full Kit	$550
Easy Kit	$400
Not-So-Bare Board	$200

New!!

Western Digital 1002-SHD Hard Disk Controller

Assembled and Tested ..... Only $200!

New!!

The μ Slicer 188

Assembled and Tested ..... Only $700!

Slicer Enclosure System

Enclosure (alone)	$125
with 135W Power Supply	$245
with P.S. and two 80 track 5¼" Disk Drives	$695
10MB (formatted) 1/2 height 5¼" Hard Disk	$700
PC Compatible Keyboard	$150
PC Compatible Amber Monitor	$175

New!!

Super Enclosure System

Enclosure, Power Supply, 1/2 height
80 track floppy, 10 MB (formatted)
5¼" Hard Disk, Western Digital
1002 SHD Controller
All for Only ..... $1395!

New!! 8087 Math Co-Processor Board

(Requires removel of CPU socket)
This board requires a C-Step CPU and an 8 Mhz 8087. Kit with all parts except CPU and 8087 ..... $300!
8 Mhz 8087 ..... Call

Try our New SLICER Bulletin Board System 300/1200 Baud at (612) 788-5909

SLICER™

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

PAYMENT METHOD: Mastercard, Visa, check, money order, or COD (certified funds or cash). Please allow 4 to 6 weeks for delivery ARO. Minnesota residents please add 6% Minnesota sales tax. Normal shipping within the continental USA (via standard U.P.S. service) is now included on all boards and software.

MORE Listing

This is the listing that was left out of Issue 23's Slicer column. MORE was written by Laine Stump as an example 8086 assembly language program. It lists a file to the screen a page at a time.

```
;*****************************************************************
;*      MORE.A86 - print a file to the console 1 page at a time  *
;*                 similar to UNIX's more.                       *
;*      Assemble with:                                           *
;*                      A>asm86 more                             *
;*                      A>gencmd more                            *
;*                                              Laine Stump 2/6/85 *
;*****************************************************************
;
LINES   equ     24      ;lines per page
CR      equ     'M'-64  ;carriage return
LF      equ     'J'-64  ;linefeed
ESC     equ     '['-64  ;escape
EOF     equ     'Z'-64  ;end of file character
;
;       Define a 'BDOS' instruction to save typing and trees
;
CodeMacro       BDOS    FTNNUM:Db
        DB      0B1h            ;MOV CL,
        DB      FTNNUM
        DB      0CDh            ;INT
        DB      224             ;to BDOS
        ENDM
;
;       BDOS functions
;
CONINF  equ     1       ;input from console to register AL
CONOUTF equ     2       ;output DL to console
PRINTF  equ     9       ;print string @DX until '$'
OPENF   equ     15      ;open file with FCB @DX
READF   equ     20      ;sequential read from file w/FCB @DX
;
        CSEG

START:  CMP     FCB+1,' '               ;see if they typed a filename
        JNZ     OPEN

        MOV     DX,offset INFO          ;IF no filename
        BDOS    PRINTF                  ;THEN give command line syntax
        RETF                            ;and return to CCP

OPEN:   MOV     DX,offset FCB           ;attempt to open the file
        BDOS    OPENF
        CMP     AL,255                  ;255 if unsuccessful
        JNZ     TYPEFILE

        MOV     DX,offset NOFILE        ;IF we can't open
        BDOS    PRINTF                  ;THEN say so
        RETF                            ;and return to CCP
;
;       type out the file 24 lines at a time
;
TYPEFILE:
        MOV     LINECT,LINES-1          ;reset line count

TYPE1:  MOV     DX,offset FCB           ;read a record from the file
        BDOS    READF
        OR      AL,0                    ;see if EOF yet
        JNZ     DONE
        CALL    TYPE128                 ;IF not, THEN type these 128 chars.
        JMPS    TYPE1                   ;and go get some more

DONE:   RETF                            ;done with file, go back to CCP
;*****************************************************************
;
;       send 128 characters @FCB to console, keeping track of lines
;
TYPE128:

        MOV     CX,128                  ;CX is LOOP counter
        MOV     SI,offset DMA           ;SI points to data to output

TYPE2:  CLD                             ;get a byte from DMA into AL
        LODS    DMA
        CMP     AL,EOF                  ;IF character is EOF (^Z)
        JZ      TYPE5                   ;THEN we are done with file

TYPE3:  MOV     DL,AL                   ;put character into DL for BDOS
        PUSH AX ! PUSH CX ! PUSH SI     ;save important registers
        BDOS    CONOUTF                 ;output the character
        POP  SI ! POP  CX ! POP  AX     ;restore registers

        CMP     AL,LF                   ;IF character is LineFeed
        JNZ     TYPE4
        DEC     LINECT                  ;THEN update line counter
        JNZ     TYPE4
        CALL    PAGE                    ;IF done with page THEN wait
TYPE4:  LOOP    TYPE2                   ;go output another

TYPE5:  RET
;*****************************************************************
;
;       wait for character from console, then reset LINECT
;
PAGE:   PUSH CX ! PUSH SI               ;save everything important
        MOV     DX,offset MORMSG        ;say '-- More --'
        BDOS    PRINTF
        BDOS    CONINF                  ;wait for a character
        MOV     LINECT,1
        CMP     AL,CR                   ;IF CR THEN just show 1 more line
        JZ      PAGE1
        MOV     LINECT,LINES-1          ;ELSE show an entire new page

PAGE1:  MOV     DX,offset COVER         ;erase -- More --
        BDOS    PRINTF
        POP  SI ! POP  CX               ;restore everything
        RET
;*****************************************************************
;
        DSEG
        ORG     5Ch
FCB     RB      36                      ;default CP/M FCB

        ORG     80h
DMA     RB      128                     ;default DMA address

        ORG     100h
LINECT  RB      1
INFO    DB      'Usage:  MORE d:fid.ext$'
NOFILE  DB      'File does not exist$'
MORMSG  DB      CR,ESC,'BO-- More --',ESC,'C0$'
COVER   DB      CR,'            ',CR,'$'

        END
```

Z sets you FREE!

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

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

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

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

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

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

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

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

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

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

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

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

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

Echelon, Inc.

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

Integrated BIOS for BB II

This BIOS adds special features for floppy and winchester users.

Floppies

Read and write almost any 5" and/or 8" disk format
40 formats included
(Kaypro, Osborne, IBM . . .)

Use any type of 5" and/or 8" drive (SS, DS, 40- 77- or 80-tracks)
New SYSGEN works directly between 5" and 8" disks.

May be configured for our new 256K RAM disk, ZCPR2, Centronics, CP/M in ROM, etc.

Winchester

Winchester formatter and new Winchester SYSGEN (can make the Winchester drive A:)
Supports XEBEC and ADAPTEC controller.
Subdivides into any specified number of drives.

Also includes code for the New BB II monitor
(with all known bugs fixed)

Price: **$99.95**

BIGBOARD II 256K RAM card

less memory chips
includes test software

Price: **$99.95**

SASI Interface for BB I and Xerox 820 I

Includes pcb with components, BIOS, Formatter, and Sysgen. Winchester formatter automatically assigns alternate track(s) for track(s) with bad sector(s). Using the Xebec controller version F.

Price: **$99.95**

Ask us about a bios for CP/M 3.0!!!
Banked version plus time and date clock for the Bigboard II.

Meet me at SOG for a systems generation workshop and good deals on software.

ANDY BAKKERS

De Gervelink 12
7591 DT Denekamp
The Netherlands
(..31-5413-2488)

Please pay with US—$ Money Order.

Create graphics masterpieces with...

Rembrandt

Complete Business Graphics Toolkit™

NEED GRAPHICS? You don't need a new computer. You need REMBRANDT. The software package that unleashes all the graphics power built into your Kaypro Computer.

Until now, accessing Kaypro graphics required advanced programming efforts. Now the REMBRANDT Business Graphics Toolkit gives you three easy-to-use tools that allow even the most inexperienced user to quickly master Kaypro graphics.

KGRAPH™ enables quick and easy creation of business graphics including horizontal and vertical bar charts, pie charts and xy plots (scatter-graphs) — KGRAPH uses hand entered data or reads numerical data from just about any source including dBase II, spreadsheet, Mbasic and Wordstar files.

KBOARD™ is the full-screen graphics editor for your Kaypro computer. Create graphic screens, save and recall them to and from disk. Layout forms, design logos, draw pictures. It's easy and fun to use!

KBRIEF™ produces electronic on-screen "slide shows" with absolutely no programming required! KGRAPH and KBOARD files are easily sequenced using nine special effects!

REMBRANDT is complete with printer routines so graphics can be reproduced on virtually every dot-matrix or daisy-wheel printer.

REMBRANDT, the Complete Business Graphics Toolkit costs just $79.95.
A demonstration disk is just $5.00 applicable to the purchase price.
See your Kaypro dealer for a demonstration.
For Kaypro 2-84, 2X, 4-84, 4E, 4X, 10, 12X and Roble.

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™

Now available for the IBM-PC 75 formats just $39.95

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	LNW-80	Cromemco w/Int'l Term	Xerox 820 I SD
IBM PC-DOS 1.0 & up	TRS-80 with Omikron CP/M	Cromemco CDOS SSDD	Xerox 820 II DD
IBM PC-DOS 2.0 & up	TRS-80 III w/Memory Merchant	Cromemco CDOS SSSD	Zenith Z90
IBM CP/M-86	TRS-80 IV with CP/M+	TI Professional CP/M-86	DEC VT180
Morrow MD2	Heath Z100	Actrix	NEC PC-8001A
Systel II	Heath w/Magnolia CP/M	Lobo Max-80	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,

VISA MasterCard

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:

formerly DG/SYSTEMS

DIGITAL RESEARCH TECHNOLOGIES, INC.

22458 Ventura Blvd., Suite E
Woodland Hills, CA 91364

All programs also available for OSBORNE and DEC RAINBOW computers. Dealer inquiries invited.

Kaypro Column

By Dave Thompson

William Fankboner stirred up a good deal of controversy with his letter in Issue #21. A number of folks suggested that if he didn't like our documentation, then he should do something about it.

Well, he did. Very nicely, I might add. Figure 1 is a copy of the illustration that he drew to show the II to 4 modifications. We have put it in the latest Pro-8 manual and we are publishing it here.

Is It A II-83?

Before you start digging into your board, make sure your Kaypro II-83 is really a II at heart. Remove the top from your Kaypro and look closely at the board. There will be two 20-pin ICs with paper stuck on top. The one nearest the front of the computer will be marked 81-149 or 81-232. 81-149 means you have a II board. 81-232 means you have a 4 board. (If you have neither, then you have an 84 board.)

If you have a 4-83 board you don't need to modify or purchase anything to use two 390K drives (double-sided, double density). Just get a formatter from someone who has a 4-83, and you are on your way. Or you can purchase a Pro-8 ROM and run any mix of single-sided, double-sided, or quad density disks. If you purchase or build a decoder board, then you can use three or four drives rather than just two. (See our ad for info on the Pro-8 version 2. Dana's made it even better.)

If you have a II-83 board (81-149) you'll need to do the II to 4 upgrade before you can use the Pro-Monitor 4 or the Pro-8. We have the 74S04 ICs for $1.50 each, postpaid.

5MHz Revisited

This is another spot where a picture is worth at least a thousand words. Most people who are planning on speeding up their 83 Kaypro II or 4 get a plug-in board from someone, and off they go. The boards usually cost between $75 and $100.

Otherwise you can add the jumpers shown in Figure 2 (or Figure 3 if you are unsocketed) and spend your money on a faster ROM (only needed if you really have a II, see above) and Z80B. We have the Z80Bs for $12. For a faster ROM you have your choice of the Pro-Monitor II, Pro-Monitor 4 (if you have a 4 or are doing the II to 4 upgrade in Figure 1), or Pro-8.

Double Duty

A lot of folks do the speedup and the Pro-8 at the same time, since the Pro-8 ROM is a fast part. I highly recommend that you do the II to 4 upgrade (if necessary) first. Connect your board to the power supply, drives, etc. to check out your work.

Once you've verified that the Pro-8 is running, then do the speedup. If there's a problem, you won't have to check both mods to find it.

One part of the speedup that many people leave out (including the speedup kit manufacturers) is the CAS-MUX change. You could do the CAS-MUX change even if you aren't speeding up your system. Your Kaypro will run more solidly at 2.5MHz after this simple change.

IBM's Kaypro Clone

I understand that at a Washington, D.C. show during the first week of March, Kaypro displayed its 286i AT clone. IBM had a very large booth, but no ATs to demonstrate. Some people (wise guys, no doubt) stopped at the IBM booth asking to see the Kaypro clone. IBM didn't see any humor in the question, no humor at all. However, the story is definitely generating some chuckles around Kaypro.

A lot of anxious people placed orders for the 286i contingent on delivery within 90 days. It seems they had tried placing orders with IBM, but the delivery dates were too far out (and who knows if an IBM will work—what with their reputation and all . . .).

The rumor I'm hearing is that IBM has warehouses full of XTs because sales of the 8088 system with a winnie died when the AT was announced. So, hoping to clean out the XTs, IBM stopped shipping the hard disk version of the AT. But dealers were adding winnies to the floppy-based ATs, so the XTs still sat.

Finally, IBM simply shut down all AT shipments in hopes that within nine months the XTs will disappear. Those warehouses full of XTs might be just the medicine for Kaypro's financial ills.

286i

Speaking of Kaypro's AT clone, I thought you'd like some details. The 286i retails for $4550. It comes with 512K of RAM expandable to 15 meg. (Yeah, I know, there is absolutely no way anyone could ever use or afford 15 meg of RAM, but then I felt that way about 256K just a year ago.) The system comes with color graphics standard (it's not standard on the AT), MicroPro software, and 8 slots, 5 of them empty.

It comes with two floppies, 1.2 meg each. The drives can supposedly read 360K disks, but as far as I understand they can't write them. A number of shops have started offering to "upgrade" your AT to 360K drives so you'll have total compatibility with PCs and all those warehoused XTs.

The 286i comes with a built-in hard disk controller card. Kaypro is refusing to get into the hard disk battle, so it is letting you add your own. I'll be taking a look at the drive market to see if there are any reasonably priced drives that are also dependable.

I saw Kaypro's AT clone at the West Coast Computer Faire and was surprised at how close a copy it is to the real AT. If you saw both of them with their lids off, you'd be excused for not knowing which one was which. The physical design of the 286i appears identical to the AT.

Kaypro has already shipped about 500 units (as of early April) with major shipments scheduled for June 1.

Kaypro Sales

Kaypro sales says that its number one product (about 60-70 per day) is the 2X Business Pack. The second most popular product is the K16. They announced that the K16/2 (has two 360K floppies, no winchester) is $2550 vrs. about $3200 for the K16 with one floppy and a 10 meg winnie.

K16 Problems

A number of K16s, both winchester and floppy based units, have video circuits that go to lunch on random occasions (we're not talking short coffee break here). Those displays that are flaky appear to be very sensitive to static elec-

(continued on page 35)

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
256K	215K RAM disk 32K printer buffer ~~$425.00~~	$395.00
512K	470K RAM disk 32K printer buffer ~~650.00~~	545.00
512K	430K RAM disk 64K printer buffer ~~650.00~~	545.00
1mb	942K RAM disk 64K printer buffer ~~1,200.00~~	795.00

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 yours need change.

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

	4MHz Kaypro Floppy Disk	4MHz Kaypro RAM Disk
Recalc 14K Perfect Calc	9:31.25	1:17.78
Load LADDER.COM	9.38*	2.12*
Load Printer Buffer		24.61*

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:

	DR LOGO 320k IBM PC	TLC LOGO 64k Z80	Apple LOGO 64k Apple IIe
Circle test	10 seconds	3 seconds	22 seconds
PolySpiral1	17	4	11
PolySpiral2	out of stack	7	out of stack
Square Test	27	10	41
Four Bugs (req. 4 turtles)	78	6	N/A

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 ~~299.95~~ 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, ... ~~245.00~~ 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 dumpscreen to printer, each color prints a different pattern on a standard dot matrix printer ..... ~~$50.00~~ $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 SSDD on DSQD without a problem.

64k Dynamic RAM Chips:
200NS ..... $1.75 ea.
TMS 9918 Color Chip & Crystal 29.95

Infocom 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

Invisiclues ..... $7.95 ea.

6MHZ Z80 Support Chips
Z80B CPU $12.00
Z80B PIO 12.00
Z80B SIO/O 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

NEW
IBM-TTL Compatible Monitor Adapter, Kaypro 2-84, 4-84, 10 and 2X $49.95

Zenith Monitors
ZVM 122 Composite Amber $119.00
ZVM 123 Composite Green $114.00
ZVM 124 IBM-TTL Amber $179.00
ZVM 135 Composite Color/RGB/Green Hi Res. $500.00

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

(continued from page 33)

tricity. If you shuffle across the rug and touch any part of the K16—keyboard, cabinet, printer cable—the screen image disappears.

Kaypro purchased a static zapper and installed it at the end of the assembly line. Now that they're zapping all assembled units, it should stem the flow of defective ones.

If you have the problem, as MicroSphere did, Kaypro will send you a new video board. That swap didn't totally cure MicroSphere's K16, but it made the unit a lot less sensitive to static, and the screen doesn't go away nearly as often.

Kaypro 2000

Just when you thought that Kaypro had released all their new systems they surprise you with another (actually, this is one they've been working on for well over a year, so I can't say I'm too surprised).

Anyway, this one is an 11 pound system that's very similar to the Data General. Like the Data General it has the Citizen 3½ inch drive and an 80 by 25 LCD display. It runs 123 and flight simulator and comes with the Star Burst software package (whatever that is). Screen contrast (a real problem for the LCD units) is supposed to be better than on the early Data General displays.

The 2000 comes with batteries (4 hrs. per charge), charger, 256K of RAM, and a removable keyboard for $1995.

They are finishing up a separate base unit that will include a power supply, standard video, and a 360K 5¼" drive, and they will also offer a built-in 1200 baud modem.

84 Video Fix

The Kaypro 2-84 and 4-84 have a very slow video scroll because the processor is readdressing video RAM every time it sends it a character. This is slow. So slow, in fact, that some can't display serial data at 1200 baud without dropping characters. Plus, an original Kaypro II running at 4MHz is a lot snappier than the new 2s because the new processor is spending so much of its time dinking around in screen memory. (Let's see now, tell the 6845 that I'm going to send a character to RAM, send a couple bytes of address, and send the character. Now tell the 6845 that I'm going to send a character to RAM . . .)

However, the 6845 video controller (or pin compatible 6545) is very smart. It knows how to automatically increment the video RAM address so the processor can dump characters into video RAM just as fast as it can output them. So why didn't Kaypro take advantage of this feature? Bad timing.

Some video controllers work just fine in this auto-increment mode. Others get out of step, seem to lose track of what the processor is doing, and characters start showing up in strange places. It's this timing problem that Kaypro was trying to avoid when they wrote the very slow video code.

Remember the slow disk write code on the old II-83 and 4-83? That was another example of a hardware timing problem that they tried to solve by slowing down the software. They fixed the hardware with the modification to U87 (see issue #11), but the only way to get around the slow code is with a Pro-Monitor.

Anyway, there is still a video timing problem on the 84 boards, which makes the choice of video controller very important. On the Pro-884 Max ROM we give you a choice of fast or slow video. If your controller and processor work well together, then you can use the fast video (the system acts like it's running 10MHz). If not, then you use the slow. You'll see garbage on the screen if fast doesn't work (though the file you are editing will be clean).

With all this in mind, you'll understand why I was excited when Chuck Weingart called to say he had fixed his garbagy video. Cold.

He replaced his original 6845 with a faster 6545A. Then he replaced the Z80A with a Z80B. Separately, neither helped, but together they work flawlessly. No guarantees, of course, but when you see how responsive the Kaypro becomes with the fast scroll you'll understand what drove Chuck to find a solution.

■ ■ ■

Figure 2

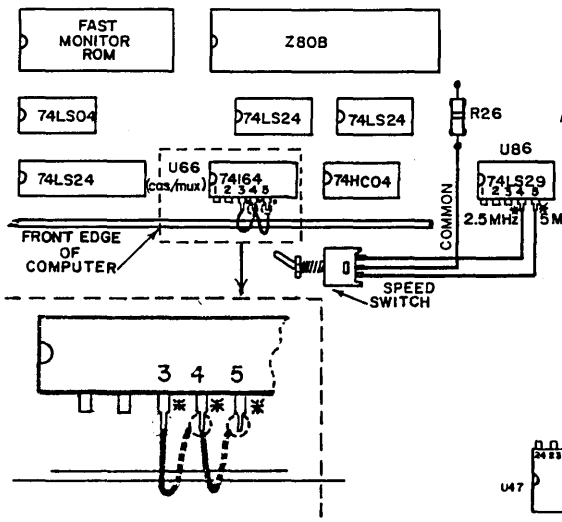
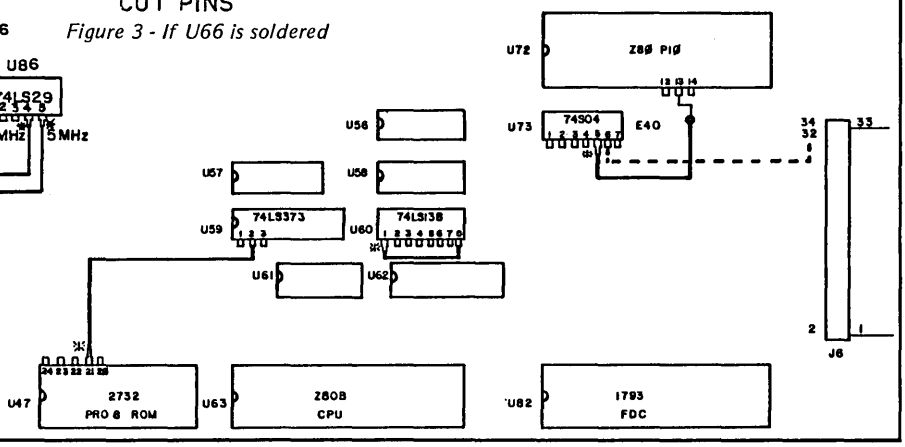


Figure 3 - If U66 is soldered

Figure 1 - Kaypro II To Kaypro 4 Upgrade
(Make these changes if your original monitor ROM is marked 81-149.) Heavy solid lines show the three jumpers you need to add on the top of the board. The dotted line is a jumper that goes underneath the board from U73 pin 6 to J6 pin 32. There are asterisks (*) next to the three pins that must be pulled up before you connect jumpers to them. (Don't forget to change U73 from a 74LS04 to a 74S04.).



Soldering: The First Steps

By Lewis Sternberg

535 NW 15th
Corvallis OR 97330

One of these days you'll want to do an upgrade, and good soldering techniques will save you time and money.

What You'll Need

Solder
Soldering iron
Sponge
Forceps
Knife
Diagonal wire cutter
Wire stripper
Desoldering tool
Isopropyl alcohol
Small stiff plastic brush
Luck

Solder

The solder should be 60/40 with multiple core rosin flux. (Flux facilitates flowing.) Don't use acid core solder—the acid will corrode the joint.

Editor's note: Cheap solder (i.e., 50/50), large old irons, corroded tips, and tins of flux (even radio flux) are no-no's. The only thing wrong with this article is that Lewis can't take a bit boffer to you (it's a baseball bat with spikes commonly used to straighten out programmers who write buggy code) when you use acid flux and a plumber's torch on your board.

When you purchase a new soldering tool, read the instructions on tinning the tip. Tinning the tip properly is almost as important as using the correct solder. I prefer an iron tip instead of copper, as they don't corrode as quickly. But with either tip, you must tin the tip (coat it with solder) the instant it is hot enough to melt the solder.

Irons, Not Guns

For $10 you can get a 15 watt pencil iron or one with a 15/30 watt switch. More watts heat the iron faster and can handle heavier soldering jobs, but most professionals use 15 watt units (or soldering stations) exclusively.

If you can afford it, a Weller soldering station is the best. It has a built-in stand, interchangeable temperature controlled tips, and an isolation transformer.

Soldering guns are too hot, too clumsy, and they are dangerous to ICs. Guns are for old Heathkit radios and subway riders, not for PC boards.

Sponge, Etc.

Use a clean, damp sponge to remove old solder and burned flux, which prevent new solder from flowing properly.

You'll also need hand tools—forceps (hemostats), an X-acto type knife, diagonal cutting pliers, and a wire stripping tool. Also, a little vice is handy for holding small parts. (A little vice might be kinda fun if they don't put the clamps to you.)

If you never make mistakes and never change circuits, then you won't need a desoldering tool. I do. My favorite is a "Solda-pullit" desoldering pump. The conductive model doesn't hold a static charge, so it won't zorch those spendy little ICs.

Wire

You'll need two kinds: stranded and solid. Solid wire is easier to use, but stranded wire is best if the wires will be flexed.

Insulation

You can buy wire insulated with either enamel, plastic, or teflon. I strongly recommend teflon. It doesn't scratch off, isn't too expensive, and doesn't smell bad. My favorite is Wire-Wrap wire.

Surface Preparation

Soldering involves a strange alchemy of several metals (copper, tin, lead) and flux.

There's no room for dirt, oxidized solder, or burned flux on the tip of your iron or on your circuits. Quickly wipe the iron's tip across a damp sponge when it first reaches soldering temperature and then immediately coat the tip lightly with solder. You will need to repeat the wiping and coating process while you are working so that the tip remains bright and shiny.

Clean up the circuit with isopropyl alcohol and a stiff plastic brush. Isopropyl alcohol evaporates readily so you won't need to dry off the board, but this alcohol is not good to breathe, so work in a well ventilated place, preferably outdoors. If the joint you're soldering is really corroded, it may be necessary to scrape it clean with a knife or small file.

Remember, the first thing that gets soldered is the iron. When it gets hot enough to liquefy the old solder, clean the tip with the sponge.

Also, solder is miserable glue. Make sure the pieces being soldered don't depend on the solder to stay together.

Surgery

Assemble your tools and parts on a clean work surface (Formica is very good) in a well-lit area. Prepare the surfaces for soldering. (For your first few solder joints try something disposable, like two pieces of wire.)

While the soldering iron is heating up, cut off a piece of solder 5" to 50" long. Wrap it around your index finger so it's firm but doesn't cut off your circulation. Extend it about an inch from your finger.

Test the temperature of the iron by touching the tip with the end of your solder. The solder should melt immediately on contact. If it doesn't, wait a bit.

As soon as the tip's hot enough (and it's bright and shiny), touch the tip of the iron to the surface to be soldered. Take your strand of solder and touch it to the tip of the iron as close to the soldering surface as possible. As soon as the liquid solder wicks onto the surface, remove the soldering iron and inspect your work. The solder should be hard already, and should look like the wax around the wick of a newly lit candle. You shouldn't be able to tell where the solder ends and the wire begins.

If a solder blob is just sitting on top of the joint, then the joint didn't get hot enough. This situation is called "the cold solder joint." Experts ruin their eyes looking for cold joints before (or sometimes after) they apply power to newly assembled boards.

Unsoldering

The trick to unsoldering ICs is to get the joint good and hot (even use a little added solder to help conduct the heat AND KEEP YOUR TIP CLEAN AND SHINY). Then force the Solda-pullit (solder sucker) down over the top of the iron, jerk the iron out of the way, and hit the button on the sucker. Once you get your timing down on this operation you can sometimes lift 16-pin ICs off the board with your fingers after you've finished. Usually a couple of pins will still stick after the first pass, and you can ei-

ther heat the holes and try to wiggle the chip out, or resolder the stubborn pins and then use the solder sucker again.

If you don't plan to reuse the IC, you can cheat by simply cutting all the pins off the chip and then remove the pins one by one by heating them and then pulling them out.

Heat Damage

ICs, transistors, diodes, and plastic of all kinds can be damaged by high temperatures. If you know that it's going to take more than one second to solder a joint—for example where a semiconductor lead is soldered to a large ground line—then "heat sink" the semiconductor's leads by clamping a forceps between the component and the joint.

Finally

Now that you've read all this you're probably wondering if soldering is totally beyond you. You've got to get a tiny joint hot enough to take solder without cooking a delicate board or IC.

But are you going to hang up your new iron and trudge down to your local computer shop for that long-anticipated 5 MHz speed-up? Of course not! At least not without a fight.

You just need a little heat and a little practice (which is what they told you when you bought the yogurt maker last year). Just get a surplus PC board with ICs soldered to it ($5 max at a swap meet) and away you go. Try adding components to the board. Try pulling up single pins and running jumpers to other pins or pads. Then, for your final exam try unsoldering several ICs from the board without damaging them and without lifting runs on the board. Pretty soon, you'll be an expert.

■ ■ ■

Eight Inch Drives On The Kaypro

By Dana Cotant

Micro C Staff

I've received numerous requests for an eight inch adapter board for the Kaypro. A couple of boards are already available, but they're limited to single density, and usually support only one eight inch drive. I wanted a system that would support two eight inch drives—single-sided single and double density, and double-sided double density. And I wanted to make the same upgrade available for Big Board users, so they could use five and eight inch drives simultaneously. Well, I've done it.

Why Eight?

There's only one "standard" format in floppy disk drives—an eight inch single density called IBM 3740. CP/M public domain software is almost always distributed in this format (SIGM and CP-MUG software is distributed only in IBM 3740), and anyone marketing CP/M software provides it first in IBM 3740. So if your computer can read and write the standard, you can obtain virtually any CP/M software.

More Storage

Another advantage of eight inch drives is storage capacity. Until the recent development of high density five inch drives, eight inch drives offered the highest capacity of any floppy disk drives. On a single-sided disk, single density offers only 241K, but double density increases that to about 600K. Double-sided double density disks have a capacity of over 1200K (1.2 Meg.).

Disk I/O is faster on eight inch drives. Five inch double density drives transfer data to and from the processor at a rate of 250 Kbits per second. Eight inch double density do it twice as fast. The faster transfer rate provides a dramatic speed improvement in disk operations.

Hardware

The most unusual feature of the board is the two sets of drivers for the interface lines. One set powers the five inch drives, and the other set powers the eight inch drives. For this reason, both types of drives have their own terminators. This eliminates compromises in the terminator's position on the cable. It also allows the Kaypro to be disconnected from the eight inch drives for easy portability.

The heart of the board is an SMC 9229 which handles data separation, write precompensation, and head load timing. I chose it because of its similarity to the 9216 data separator used in standard Kaypros. It is 100 percent digital, so there is no chance of drift from analog components. Its digital design also lets you change write precompensation without an oscilloscope.

The board can be installed with no soldering, and provides all the signals for the eight inch and five inch interfaces as well as four drive select signals. Drives A: and B: are the Kaypro five inch drives, and C: and D: can be either five inch or eight inch drives.

The eight inch drives you use can be any kind, but they'll need their own power supply because their requirements cannot be met by the Kaypro's supply.

Software

The software is ROM based and Pro-8 compatible.

The single-sided double density format uses 16 512 byte sectors for a total data capacity of 596K. This format is also compatible with the Big Board II and the Slicer.

The double-sided double density format has the same sector configuration as the single-sided double density, but every other track is on the opposite side of the disk. This "cylindrical" configuration is the fastest method of implementing double-sided operation since it reduces the amount of track to track seeking required. All formats including eight inch are automatically determined when the disk is accessed.

The ROM

The ROM for the Big Board is much closer to the Kaypro ROM than to the PFM monitor. Most of the low and intermediate device drivers are located and executed in ROM. Thus Big Board owners can have a 63K CP/M system even with double density and multiple drive types.

Both the Big Board software and the Pro-8 support the same disk formats. Five inch 48 tpi drive capacities are 191K (single-sided) and 390K (double-sided). Five inch 96 tpi drives have a 784K capacity.

The eight inch formats are the same as previously described. The drive types can be mixed and in any order. CP/M can be booted from either five inch or eight inch in any of the six formats.

Operation

At reset, the monitor sets up the keyboard and the floppy disk controller for interrupts. Then the processor waits for the first interrupt. If there's a disk in drive A:, the monitor autoboots CP/M. If you hit any key before a disk is inserted into drive A:, a debugging monitor is loaded into RAM and executed.

The debugger is a subset of PFM with memory dump, edit, and port I/O commands. You also have access to all the ROM based functions from the RAM based debugger.

The BIOS supports interrupt mode 2 including CTC disk drive time out and interrupt driven keyboard operation. It also implements the I/O byte. Other options include serial or parallel printer drivers and function key translation on keyboards that send special characters with bit seven set.

System Requirements

New Kaypros (2-84, 4-84, or 2X) are ready to run the board and software without any modifications. Old Kaypro 4s or Kaypros that have already been upgraded to a Kaypro 8 can run the board with no modification for single density eight inch, but will need to be sped up to 4 or 5MHz to run eight inch double density. Kaypro 2s will also need to be upgraded to Kaypro 4. Instructions for the upgrades are included with the board.

Big Boards will also need to be running at 4MHz or faster to run double density eight inch. Since the adapter board has its own 16MHz oscillator, it is very easy to upgrade to 4MHz.

The system will reside in the first two ROM sockets. If you want, you can plug a modified PFM ROM (that will run with the adapter board in place) into the third ROM socket. It will be available soon from Micro C. It will operate single density eight inch only. ZCPR 1 in ROM will also be available for the fourth ROM socket.

The modifications are outlined in the system manual.

Other Considerations

The market is swamped with inexpensive disk drives you can use with this system. Eight inch single-sided drives are selling for less than $100, and double-sided for about $200. All of the five inch drive types go for around $100. Stay away from Remex and Tandon 100-4 drives.

You can pick up a switching power supply for less than $50, but make sure you know the power requirements of your drives before you buy the power supply. Tandon 848 drives require more +24V than other eight inch drives. If you are a novice at putting together disk drive subsystems you might want to purchase an enclosure with the power supply and cables included.

Micro Cornucopia cannot possibly support the large number of different double density formats. According to Brian Garrison of Emerald Microware, Micro Solutions is working on a version of their UNIFORM program to support some other eight inch formats, but it is not yet available.

In the meantime, if you are transferring software between different eight inch computers, go to IBM 3740 single density format. After all, that's the reason for having a standard format.

■ ■ ■

Kaypro BIOS Patch

By Ray Rizzuto

450 Forrest Ave. Apt. N311
Norristown PA 19401

I recently bought a quad-density, double-sided drive and installed it as my A drive, leaving my B drive single-sided. I had previously modified my Kaypro for new drives, and had been using the Pro-8 ROM for several months.

Once I had the system all SYSGENed, I started transferring information from my single-sided disks to the quad drive. Trouble. Occasionally, the drive "locked up" while accessing the single-sided drive. I checked out back issues of Micro C, and found that someone had written about the same problem (Issue 21, Tech Tips).

I couldn't find the pattern—but I did notice that it locked up only on drive B. I even had the problem while using Wordstar to edit a file on drive B.

Later I surmised that my Kaypro was getting stuck trying to read the back of the single-sided drive. I put my logic probe on the side select signal (E40) during a file transfer, and found that the lock-up occurred after the system accessed the back side of drive A and then tried to write to drive B. Although the disk in B is single-sided, the side select was still set for the back side. So the system looks on the back side of a single-sided disk for the next sector, and not finding it, simply times out (15 seconds), and then resets the drive. Once the drive is reset, the controller looks on the front side of the disk and away it goes.

Fix

I patched CP/M's BIOS (basic input/output system) to always select the front side when it accesses a drive. If the track and sector are not there, then the system looks on the back side. So I patched the BIOS call SELDSK (select disk) so it always selects the front side of the disk.

Of course, if you really need to read or write the back side of a disk, this BIOS patch might seem counter productive, but the drive controller checks for the correct track and sector before doing a read or write and selects the back if the correct sector is on the back side.

The patch does slow down the first access to a reselected drive if the wrong side is selected (e.g., A bottom = > B top = > A top), but the delay isn't anywhere near the 15 seconds you get otherwise.

The following listing is the patch I used. It also contains the LISTST patch which is already present in the CP/M configured by PRO-8SET.

■ ■ ■

Kaypro LISTST BIOS and Pro8 Disk Patch (KPATCH.ASM)

```
;  The Kaypro IV has a bug in the BIOS function 14, LISTST.
; This bug causes the print buffer function in MDM7nn not to work
; properly.
;  According to the DRI Alteration Guide, this function is
; supposed to return a value of 0FFH in register A if the printer
; is available, and 00 if it is busy. The Kaypro BIOS (which in
; turn calls the ROM) does not return the zero in A, but it does
; return with the zero FLAG set. This can be fixed by the BIOS
; patch which follows:
;
;
; The disk patch forces the selection of the top surface of a disk
; whenever the drives are selected through bios call SELDSK. This
; should fix the problem of a single sided drive (or a single sided
; disk in a double sided drive) *hanging the system when transferring
; data from a double sided drive.
;
; 1/8/85 Ray Rizzuto

                ;
FA00 =          BIOS        EQU     0FA00H    ; Beginning of BIOS jump vectors
                ; THESE 2 EQUATES ARE SPECIFIC TO THE CPM FOR A KAYPRO II/4
FB65 =          LISTST      EQU     BIOS + 165H
FB80 =          SELDSK      EQU     BIOS + 180H
                ;
                ; BIOS0 = BOOT (COLD BOOT)
                ;
FA2D =          BIOS15      EQU     BIOS + 3 * 15
FA1B =          BIOS9       EQU     BIOS + 3 * 9
                ;
FA2D            ORG         BIOS15    ; Put jump to patch here
FA2D C3EEF9     JMP         PATCH
                ;
FA1B            ORG         BIOS9
FA1B C3F4F9     JMP         DSKPAT

; PATCH IS PUT AT THE END OF THE CCP. IF A TRANSIENT PROGRAM MAKES USE OF
; ALL OF THE CCP AREA, THESE PATCHES MAY BE DESTROYED. THIS IS AT BEST A
; "KLUDGE".
;
F9EE =          SPARE       EQU     BIOS - 18 ; Hole for patch
                ;
F9EE               ORG      SPARE
F9EE CD65FB     PATCH:      CALL    LISTST    ; Call the ROM
F9F1 C0            RNZ              ; OK except when zero
F9F2 AF            XRA      A       ; Clear accumulator
F9F3 C9            RET              ; And return
F9F4 DB1C       DSKPAT:     IN      1CH       ; GET SYSTEM PORT
F9F6 E6FB          ANI      0FBH    ; LOWER SIDE SELECT
F9F8 D31C          OUT      1CH     ; RETURN TO SYSTEM PORT
F9FA C380FB        JMP      SELDSK  ; CONTINUE WITH NORMAL DISK SELECT

                ; Since the bug is in the BIOS, the patch requires
                ; generating a new version and placing it in the system
                ; tracks of your disk. To accomplish this, take the
                ; following steps:
                ;
                ; (1) Assemble the code above using ASM or MAC. The
                ; output will be KPPATCH.HEX. Note that this file has all
                ; text commented out, so it should assemble without
                ; editing.
                ;
                ; (2) Use SYSGEN to get a copy of CP/M in memory.
                ; To do this, insert a disk containing both SYSGEN
                ; and DDT into drive A. Then type:
                ;
                ;  SYSGEN
                ;
                ; When SYSGEN prompts with
                ;
                ; SOURCE DRIVE NAME (OR RETURN TO SKIP),
                ;
                ; respond with an  A. This will cause SYSGEN to
                ; generate a copy of the system tracks in low RAM.
                ;
                ; (3) When SYSGEN prompts for the destination drive,
                ; answer with a RETURN to reboot. The CP/M copy will
                ; remain in RAM.
                ;
                ; (4) Save the copy of CP/M to disk by typing:
                ;
                ;  SAVE 36 CPM.COM
                ;
                ; (5) Type the following commands exactly as they
                ; appear:
                ;
                ;  DDT CPM.COM
                ;  IKPPATCH.HEX
                ;  R2580
                ;
                ; This causes DDT to load CPM.COM, then overlay it
                ; with the patches. The "2580" is the offset required
                ; to compensate for the fact that we are using a copy
                ; moved to low RAM. Just to be sure, type:
                ;
                ;  L1FAD
                ; and  E
                ;  L1F6B
                ;
                ; You should see JMP F9EE at the first address, and
                ; the short subroutine at the second.
                ;
                ; (6) Exit DDT with G0 (that's a zero) or ^C.
                ; The patched copy will remain in RAM.
                ;
                ; (7) Now run SYSGEN again. This time, respond to the
                ; first prompt with a RETURN (the system is already
                ; in RAM). At the second prompt, you can either respond
                ; with an "A", which will load the system onto the
                ; system tracks of the same disk, or "B" to load it onto
                ; any disk you desire to put in drive B.
```

End of Listing

Alternative Power Supply For The Kaypro

By Eric J. Torney

7 Hart Street
San Rafael CA 94901

Micro Cornucopia Issue 16 contained a letter asking about alternative power supply requirements for a Kaypro. I read this with interest since my electricity had been off for three days straight.

I can read by the light of a kerosene lantern, and the fire burns whether or not there is electricity. I was ready to work, but my Kaypro just gave me a blank stare.

Your note about a battery powered Kaypro stuck in my mind all through the rest of the year, and just in the past few weeks the power failed again. I immediately ordered +12 volt 5 amp, +5 volt 5 amp, and -12 volt 1 amp voltage regulators. I scrounged around through the basement to see what I had on hand: some 12 gauge romex house wire with three conductors, some aluminum sheet metal from an old panel, a metal electrical box, some wire nuts, miscellaneous pieces of wire, wire clamps, two heat sinks, a matched 4-conductor plug set (disk drive power type), and miscellaneous hardware and capacitors.

Getting Started

I hooked up a charger to my old boat battery, and then got to work designing a battery backup power supply.

When the voltage regulators arrived I started assembling. It took just an hour to make a cover for the wiring box. Then I used a mica insulator and heat sink grease as I mounted the -12V regulator and its heat sink to the cover. (Editor's note: Most +5V and +12V regulators can be mounted directly to grounded heat sinks without an insulator since their cases are normally grounded.)

Hooking Up the Batteries

Since I needed two batteries, I took my newly charged boat battery and hooked it up to my car battery via jumper cables, yielding a +12 volt, common, and -12 volt contact.

I nervously connected the romex house wire from the batteries to the voltage regulators, still remembering the last automotive/microchip circuit design I made that turned into a burning glob. Then, testing the connection with my voltmeter, I was pleased to find +12, +5 and -12 volts, right where they were supposed to be.

Kaypro Connections

I took the cover off my Kaypro and found a good connection point for splicing into the power lines. These contacts, conveniently labeled +5, ground, +12, and -12, were easy to solder to.

With short lengths of 16-gauge multistrand wire I carefully connected a female plug to these contacts, using the same plug configuration that the disk drives have for +5, ground, and +12. The -12 was input to the otherwise unused location.

Then I assembled the output from my power supply to the male plug to match. I chose the male/female plug arrangement to avoid inadvertently inserting the wrong plug into a disk drive.

I left the Kaypro end of the plug loose inside my system, meaning that I have to take off the cover to connect up. It would

be easy, though, to mount a connector on the rear of the system so it is accessible from the outside.

Testing My Work

After carefully examining all the battery connections, I took a deep breath and made contact with the plugs. In a few seconds I was rewarded with a message to insert a disk. After putting a disk in, I made a few commands to fill up the screen.

Unfortunately, the screen was all distorted and the display warped during disk activity. I measured the voltages on the main board and found +9.5V on the +12V contact, +5V and -12V supplies were okay.

I had about 50 feet of wire between the battery and the voltage regulators. When I shortened the wire to approximately 10 feet, the +12V supply increased to 11V and everything worked fine.

After finishing the project, I found that my costs totalled $32.50. Even if I had added the cost of the parts I had around the house, it probably wouldn't have topped $40, excluding the batteries.

Parts List

1. 1 +12 volt 5 amp voltage regulator with heat sink
2. 1 +5 volt 5 amp voltage regulator with heat sink
3. 1 -12 volt 1 amp voltage regulator
4. 3 0.33 mFd capacitors
5. 3 0.1 mFd capacitors
6. 1 male, 1 female 4-contact plug (disk drive type)
7. several feet of 12 gauge (or heavier) wire
8. wire nuts for #12 solid wire
9. length of #16 multi-strand wire for connection to plugs
10. misc. hardware and metal box with aluminum cover
11. two 12 volt batteries

■ ■ ■

48 Lines On A BBI

By David Griesinger

Lexicon
60 Turner St.
Waltham MA 02156

This mod is lots of fun and results in a computer which is wonderful for programming. The mod requires extensive hardware changes, a bunch of software patches, a reconfigurable editor (I use PMATE), and a medium or high persistence monitor (an Amdec amber or any P39 green). If your green monitor smears badly when you scroll, it's P39. If it doesn't, you'll need a different monitor. Note: P4 and P31 are the standard short persistence phosphors.

Background

The idea for this modification began with some observations about my Amdec.

The horizontal scan lines were painfully noticeable. With full vertical height all the letters were too high, and in normal mode (amber characters on black background) each character was composed of small disconnected dots. In black on amber mode the characters looked fine, but the background was full of disconnected lines. It seemed my choice was either dotty or caged characters. I wanted a better solution.

I turned down the vertical height to compress the scan lines, but the text filled only a little more than half the screen. I wondered if I could use interlaced scan to fill the missing scan lines.

Interlacing

Interlace doubles the number of vertical scan lines by slightly displacing the scan pattern on alternating scans. In other words, instead of writing every horizontal line every time down the screen, it writes half the lines (1,3,5,7,9 . . .) on the first pass, and then the other half (2,4,6,8 . . .) on the second pass.

Interlace reduces effective scan rate from 60Hz (60 times per second) to 30Hz, which is why it shouldn't be attempted with a short persistence monitor. If the alternate frames are identical, the interlace simply fills out the missing space between scan lines, considerably improving the display.

You can interlace by adding 74LS157 to control the extra section of U51.

This puts an adjustable extra delay in the vertical sync pulse every other frame. This modification is simple, and highly recommended before you tackle "48 lines."

I work from the top of the board by lifting pins and soldering jumpers to the ICs themselves. This is fast and reversible, but it requires a temperature controlled soldering iron.

Assuming you have a fully socketed BBI, the will, and the equipment, you can make this modification.

Preliminary Instructions

I'll describe which pins to lift and which connections to make by using a table, beginning with U10, and working from the front to the back of the board. As you go, you'll find many of the changes have already been done. So the table is only about half as long as it looks.

Any pin which needs to be lifted (pulled out of the socket) will be flagged with an L. Connections are shown by "-" between entries. The piggyback chips get the same U numbers as the chips beneath them (but the piggyback's number has an apostrophe appended to it).

If you're fast at stripping hook-up wire, you can make these changes in an evening.

24 Lines Interlaced

Piggyback a 74LS157 on U50, connecting pins 8 and 16 only. This chip is now U50'.

If you follow Figure 1 and you're lucky, you'll have adjustable interlace. Set the pot so the scan lines just disappear in the middle of the screen.

48 Lines

The real pay-off to interlace is to use every scan line, and display 48 lines of text. You'll need to add more video memory, and change the character ROM addressing, the scroll register, and the software.

The extra screen memory is simply piggybacked on the existing screen memory, and mapped from 2000h to 3800h. If you're using two or more 2716 ROMs in the BBI you'll have to copy them to 2732s, since the remapping affects the ROM sockets. 2732As are a good idea at 4MHz anyway.

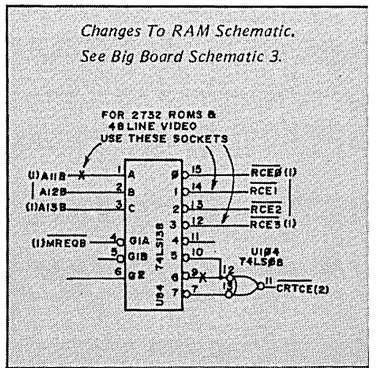
The monitor software can be patched, but even without changing the ROM, you can use the screen. Most editors can be re-configured to use the whole screen. PMATE is easy to reconfigure and quick to scroll using the BBIOPATCH from Sage Microsystems. The new system runs most old programs without any changes—Pacman, Aliens, and Games simply run in the top half of the screen.

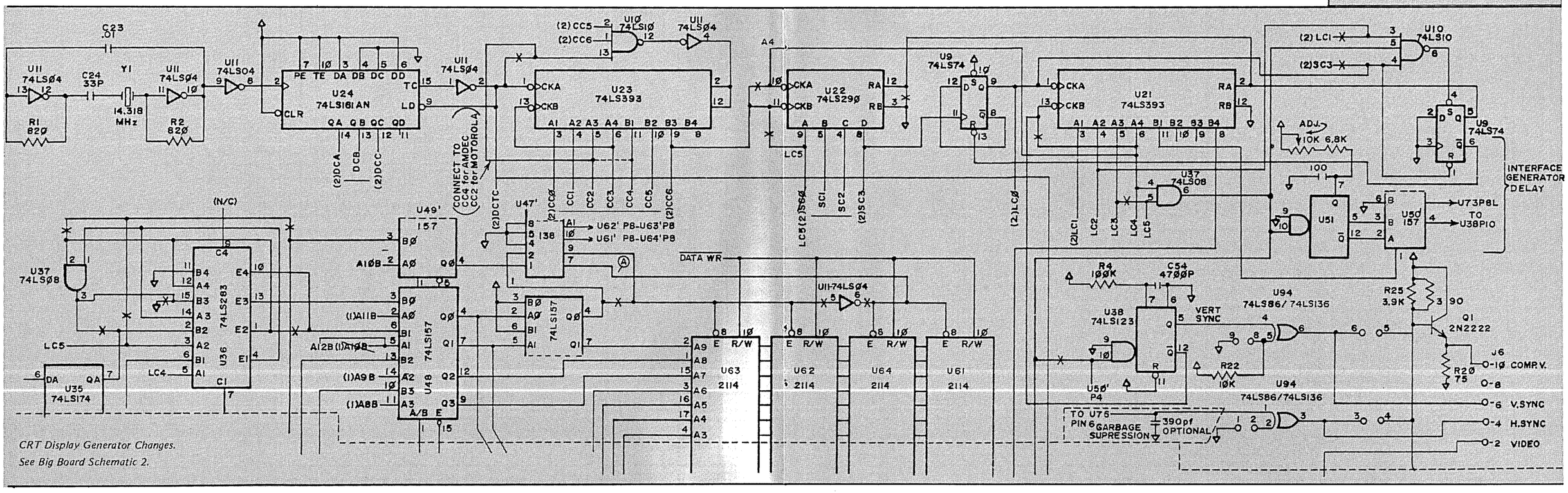
You need four more 2114 screen memories, two 74LS157s, and one 74LS138. The 157s piggyback on U49 and U50 (connected only to power and ground), and the 138 piggybacks on U47. The screen memories should be soldered (every pin except pin 8) to the chip underneath. The combination runs quite hot, but seems to work. I tried CMOS 6514s as screen memories, but they don't work.

The Mod

Figure 2 is the complete 48 line modification including the changes for interlace.

(continued on page 47)

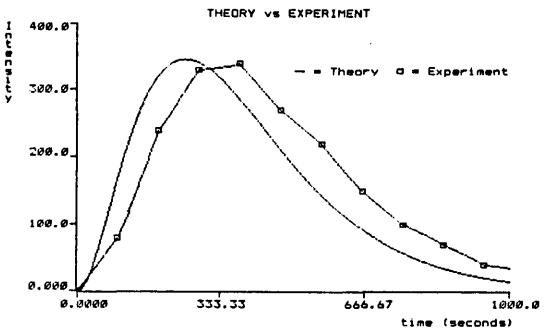




GRAF 2.0 $29.95

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

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



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

MSC Microcomputer Systems Consultants

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

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

64K Memory Chips!
150 nanosecond
Only $1.99 each!

4116 200ns
ONLY $.55

ADVENTURE GAMES BOOK

Written by Kim Schuette over 300 pages of maps, listings and general hints on almost every computer game on the market. Like **Zork I, II, & III, Adventure, Ultima, Secret Agent, Dark Crystal**, etc. - Over 75 in all.

* * only $16.95! * *

Z80A Support

Z80A DMA	$3.95
Z80A CTC	$2.50
Z80A PIO	3.50

Qume Trak 242 - 8" half-ht. double-sided double-density **$349.00** ($659.00 List)

TERMINAL CITY

Televideo 921	$495.00
Televideo 925E	$669.00
Televideo 950	$935.00
Televideo 970	$1129.00
Wyse 50	$545.00
Wyse 75	$739.00
Qume QVT 108	$695.00
ADDS Viewpoint - your're choice - A1, A2, or 3A+	$569.00

HARD DISK SPECIALS

Miniscribe 5¼" 10 Megabyte Half Height. (As used in the Eagle PC and Televideo PC)

1 Year Parts & Labor!

12.8 Megabytes unformatted with an 18 msec. track-to-track access time. This is the 10 meg used by IBM for the PC-XT version. Standard ST-506 interface. IN STOCK ~~$699.00~~

Sale Price $539.00

ORANGE COUNTY COMPUTER

VISA **(714) 895-5033** MasterCard

5122 Bolsa Avenue, #108
Huntington Beach, CA 92649

Greywolf MARKETING

203-928-3654
or **928-6555**

NEW Computers with
WORDSTAR ∘ MAILMERGE
SPELLSTAR ∘ CALCSTAR
INFOSTAR ∘ BASIC

MBC 1200 $799

8-Bit Integrated Computer with High-Resolution Display

- Two Z-80A CPUs (main and secondary) with no-wait mode and large 64KB RAM/4KB ROM memory capacity for fast execution
- High-resolution 640 x 400-dot matrix display with full graphics functions
- Choice of 33-line or 40-line text mode
- CP/M operating system with editor, assembler and all standard utilities
- Easy-to-use Sanyo Graphic BASIC
- One (MBC 1200) or two (MBC 1250) internal double-sided, double-density, double-track, 5¼" slim-type mini floppy disk drives with 640KB formatted capacity
- Specially-designed detachable keyboard with coiled cable for easier operation
- Interfaces for one Centronics printer and one RS-232 port.

★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[00] to:
P.O. Box 689, Putnam, CT 06260
PHONE ORDERS ONLY: 203-928-3654.

SUPER SYSTEM 800–2
for Sanyo MBC 550

Will Run The Sanyo Video Board

$1395.00 Complete

Software Features:

- 100% disk compatibility across MS-DOS and PC-DOS, including ability to read, write and format disks from many computers including the IBM PC, the DEC Rainbow, and Tandy 2000
- Built-in RAM disk program offering disk access which is hundreds of times faster than floppies
- SortDir program to alphabetize and sort the large directories that these drives can hold
- ASCII Screen Dump program
- Calendar Program, a personal datebook which displays a two-month calendar on the screen
- Certificate to purchase a top-selling data base, Personal Pearl, for only $99.95, a savings of $200.00. Personal Pearl is a product of Pearlsoft, Inc.

PLUS **FREE**
Clock/Calendar Board
($139.95 Value)

Plus **Free**
Video Display Monitor
($125.00 Value)

Plus **Free**
256K of Memory
($120 Value)

Hardware Features:

Operates as a standard 555-2 with two 360K drives

Flip a switch and run your Super Sanyo System with dual 811K drives

Works with Sanyo's New Video Graphics Board

Runs both MS-DOS 1.25 and MS-DOS 2.11 operating systems

Runs copy protected software, such as Michtron's Demon Seed and Cash Man

Clock/Calendar Board

Sanyo Software:

Wordstar, Easywriter, Calcstar, Sanyo Basic, MS-DOS 1.25, MS-DOS 2.11

EasyPac or StarPac (optional)

48 LINE VIDEO

(continued from page 45)

Wrap Up

Plug in your new or old ROM in U68 and boot. If you use SWP you should have already made the patches to the video, and you'll be running right away. If not, I'm making a new ROM available which fits in the lower half of a 2732a. You can plug it into socket U68.

The SWP monitor should be patched as shown in Figure 3.

The steps with comments indicate what to look for if you want to change the source code in CRTOUT. You must also change CRTBAS=20h, CRTEND=38h, and CRTMEM=2000h.

Once you finish this mod, you'll have a much friendlier programming environment.

■ ■ ■

Figure 1 - Chip Changes For An Adjustable Interlace

```
U10   P13L - U23P5 (or P11)  ;this modification speeds up the display
      P12  - U11P3L
      solder 15k in parallel with r5
U11   P3L  - U10P12

Test and readjust your monitor before continuing.

U21   P11 - U50'P1
      P13L - U38P12

U37   P6 - U51P10

U38   P10L - U50'P4
      P12 - U21P13L

U50'  P8 - P15  ;74LS157 solder P8 to U50P8 and P16 to U50P16
      P1 - U21P11
      P2 - U51P12
      P3 - U51P5

U51   P5 - U50'P3
      P6 - P8 - P9
      P7 - .001uFd to P8
      P7 - 6.8k + 10k trimpot to P16 (mount on top of chip)
      P10 - U37P6
      P12 - U50'P2
```

Figure 2 - Interlace Wiring From The Interlace Modification

```
U9:  P1L-U10P4L-U21P1

U10: P3L-U21P4
     P4L-U21P1-U9P1L
     P12-U11P3L
     P13L-U23P5

U11: P3L-U10P12
     P5L
     P6L

U21: P1-U10P4L-U9P1L  ;character address reassign
     P2-U22P12L
     P4-U10P3L
     P6-U22P10L
     P11-U50'P1
     P13L-U38P12

U22: P9-U37P5L-U36P3L  ;LC5
     P10L-U21P6
     P11L-U23P9
     P12L-U21P2

U23: P5-U10P13L
     P9-U22P11L

U34: P10L-U49'P3  ;scroll register re-wire

U35: P3-U49'P2
     P6-U83P6
     P7-U36P2L
     P11-U48P5L

U36: P1L-P12-U37P2
     P2L-U35P7
     P3L-U37P5L-U22P9
     P8-P11
     P10-U48P6
     P11-P8
     P12-P1L-U37P2
     P15L-U37P3

U37: P2-U36P12-U36P1L
     P3-U36P15L    ;LC5
     P5L-U36P3L-U22P9
     P6-U51P10

U38: P10L-U50'P4
     P12-U21P13L

U47: NO CHANGE

U47':P1-U49'P4    ;74LS138 MEMORY ENABLES
     P2-U50P4L
     P4-P5-P8
```

(Listing continued)

48 LINE FIGURE 2

(continued from page 47)

```
        P7-U61P8      ;this is under U61' - solder carefully
        P9-U63P8
        P10-U62'P8-U63'P8
        P11-U61'P8-U64'P8

U48:    P5L-U35P11
        P6-U36P10

U49:    P1-U49'P1     ;solder pins 1&15 as well as 8&16
        P15-U49'P15   ;when you piggyback these chips

U49':P1-U49'P1        ;74LS157
        P2-U35P3
        P3-U34P10L
        P4-U47'P1
        P15-U49P15

U50:    P4L-U47'P2

U50':P1-U21P11        ;74LS157 - interlace delay select + inverter
        P2-U51P12
        P3-U51P5
        P4-U38P10L
        P6-P8-P15     ;P6&P7 act as an inverter for the alternate
        P7-U73P8L     ;frame signal from U21P11 - applied to char ROM

U51:    P5-U50'P3     ;15K IN PARALLEL WITH R5! (sweep speed-up)
        P6-P8-P9
        P7-(.001uf to P8)-(6.8k in series with 10k trimpot to +5 P16)
        P9-P8-P6
        P10-U37P6
        P12-U50'P2

        ;video RAM

U61,U62,U63,U64 - solder all pins but 8 to U61',U62',U63',U64'
  pin 8 as above to U47'

U73:    P8L-U50'P7    ;character ROM

U68:    P21L-U70P21L-U83P2 ;monitor ROM (low half 2732a)

U68P21 can be left high (not lifted) if you wish to use the old
2716 monitor ROM or a 2732a with the monitor in the upper half.
The old monitor ROM can still be used (with some difficulty)
with the 48 line display, especially if you patch the SWP
software to overlay the old monitor when you boot.  The old
monitor must go in socket U68.  Sockets U67 and U69 are not
useful after the changes to U84.

U70:    P21L-U68P21L-U83P2 ;monitor ROM #2 if used

U83:    P2-U68P21L-U70P21L ;addresses for ROMS and video memory
        P6-U35P6

U84:    P1L     ;This re-maps the memory space
        P10-U104P12L

U104:P12L-U84P10
```

Figure 3 - Patching the SWP Monitor

```
Do ddsysgen to get the image in RAM.  Now use DDT to change:

28b9  0f-1f  f2b9  ;a little insurance
28bb  30-20
299c  30-20
29a0  30-20
29a3  0c-18  ;24*128
29a9  17-2f  ;LD A,23
29e0  30-20
29e3  3b-37
29eb  3c-38
29ee  30-20
29f5  1f-3f  ;LFEED: AND 1FH
2a16  1f-3f  ;AND 1FH
2a65  18-30  ;sub 24
2a69  18-30  ;add 24
2a6b  60-40  f46b

Now use ddsysgen to restore on disk.
```

End of Listing

Adding An 8" SSSD Drive To A Morrow MD-2

By Mark Sihlanick

121 Twin Creek Terrace
Forest VA 24551

As popular as the smaller disk drives are becoming, eight inchers still reign, at least in the CP/M world. I'm a satisfied owner of a Morrow Micro Decision MD-2 with 5" drives, but I couldn't stand the thought of being locked out of all that public domain software on 8" disks. So I modified my MD-2 to allow the addition of a single density 8" drive as drive C.

Types Of MD-2s

There are two revisions of MD-2s. While the following modification should work on either, it has been verified only on Revision 1. Significant changes in board layout and circuitry will make my instructions incorrect for REV 2 boards.

To help you tell which machine you have, the Revision 1 board has an expansion drive connector on the rear and is marked (strangely enough) REV 1.1. The REV 2 board replaces the drive expansion connector with a Centronics printer port and signs on with REV 2.x or 3.x at turn-on.

Hardware Modification

The stock MD-2 uses an NEC 765 (same as Intel's 8272) with an external TTL data separator. Rather than modify the existing data separator, I used a new one from Western Digital's FD179X Application Note. My only modification to the original 1793 circuit was to invert the RD DATA signal to match the 765 by using the other output on the 74LS123.

The circuit shown in Figure 1 detects when drive C is enabled by monitoring the HD load signal for drive C brought out on pin 12 of the expansion drive connector. It then switches in the external data separator and sets the 765's clock to the 8 MHz needed for eight inch operation.

For disk writes, the circuit forces the MD-2's circuitry to act as though double density 5" were being selected. The resulting 500 KHz signal to the NEC 765 WRITE CLOCK is what's needed for the 8" single density drive.

This design allows the 8" drive to be turned off but remain connected without upsetting the operation of the 5" system.

Construction

It should be obvious that this modification will void your warranty. But if you have a REV 1 MD-2, your warranty is almost certainly void anyway, so you may as well dig in and go to it! A word of caution: believe everything you have ever heard about static zapping expensive electronic parts, and exercise caution when making these modifications.

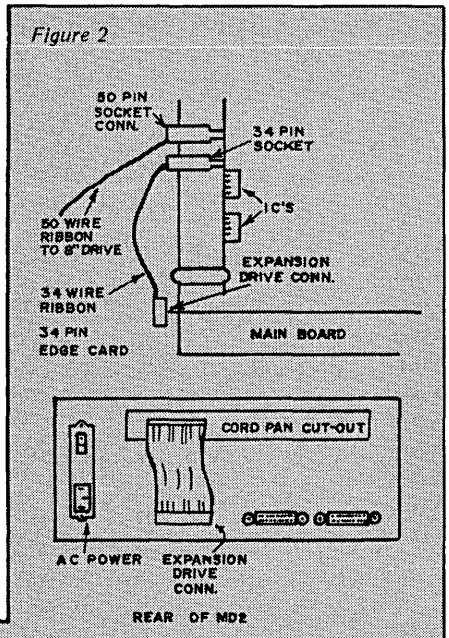


Figure 2

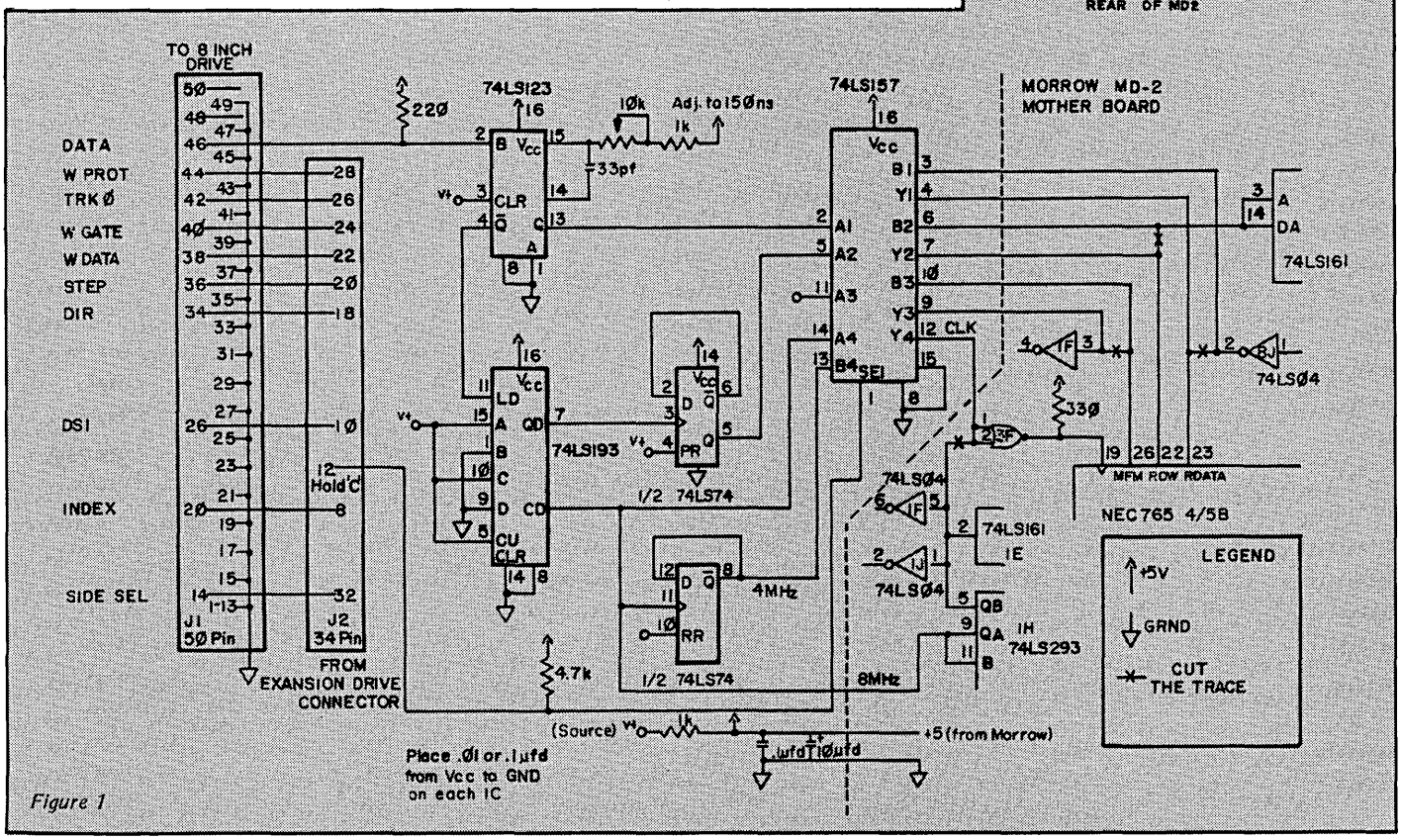


Figure 1

MD-2 Disassembly

1. Turn off the computer and remove the power cord and any RS-232 cables that may be connected.
2. Remove the four screws that hold the cover to the chassis.
3. Remove the four screws from the rear panel that hold the AC cord pan, and remove the cord pan.
4. Carefully remove the three screws that hold in each disk drive. (I do this by dangling the computer over the edge of the desk and carefully backing the screws out from the bottom.)
5. As each drive is unscrewed, disconnect the DC power cable from the jack near the power supply.
6. Disconnect the ribbon connector from the back of the drive and carefully lift out the drive.
7. Gently unglue the disk drive ribbon cable from the PC board, disconnect it from the main board, and put it aside.
8. Do not worry about keeping the A and the B drives separate since they are strapped identically and can be interchanged.
9. Disconnect the main power cable from the PC board.
10. You now have gone too far to back out, so stand the unit on end and remove the four screws that secure the PC board to the chassis.
11. Remember every warning you have ever read about static.

Daughter Board

I chose to mount the board containing the new circuitry in place of the cord pan on the rear of the computer. The cut-out in the rear panel gives room to bring out the 50 conductor cable to the 8" drive and the 34 conductor cable to the expansion drive connector. See Figure 2.

The circuit is constructed on perfboard using point to point wiring.

MD-2 PC Board Changes

These changes allow the external 74LS157 to select between the internal data separator, the NEC 765 clock source, WRITE CLK, and the external 8" equivalents.

Refer to Figure 1 and carefully cut the traces as indicated by the Xs. Take your time and be sure you have the right point. Then carefully scrape off some of the solder mask to allow connection to the runs.

When the daughter board is installed, check the wiring and reassemble the computer. You do not need the modified software to access the 5" drives.

Jumper Settings For Shugart SA-800/801

The disk drives used with an NEC 765 must have the stepper energized continuously. Therefore, be sure that jumper positions HL and DS are open. In addition, the remaining jumper positions are:

T1,T3,T4,T5,T6	PLUGGED
DS1	PLUGGED
DS2,DS3,DS4	OPEN
RR,RI	PLUGGED
HL,DS	OPEN
Y	OPEN
Z	PLUGGED
C	OPEN
A,B,X	PLUGGED

(continued next page)

(continued from page 51)

Software Modifications

There are three areas of the MD-2 BIOS that must be patched: the Disk Parameter Header (DPH), the Disk Parameter Block (DPB), and what Morrow calls the MTAB (which contains the constants associated with the disk drive). Plus, you need to add a sector translation table (XLT) for 8" disk drives. We will first locate the patch (and add) areas, discuss the contents of the patch, and finally make the changes (and addition).

Finding The Patch Area Locations

First make a new copy of the Morrow CPM distribution disk. Follow all the steps outlined in the instructions, but answer '3' when asked the number of drives to be used. You must have DDT-.COM and SYSGEN.COM on this disk. Use this new disk for the rest of software modification.

Morrow sets aside a portion of the BIOS to be used for foreign drive translation tables as well as terminal configuration space. This is the area that Morrow-supplied programs such as XER.COM or OSB.COM use. Its position varies depending also on how much patch area was required to set up the BIOS for the particular terminal you are using. It is easiest to find this area by snooping around with DDT.

Start looking at about FA80 and you should see something like Figure 3. (The symbol (R) signifies pressing RETURN.)

Write down the address of the FF that just precedes the Room Left Byte (in this case FB08) and the value of the Room Left Byte (BE).

To find the location of the DPH and DPB in the BIOS we will use Figure 4, a program that uses the CPM SELDSK function to return the address of the DPH for disk C. Since the program is short it will be keyed in directly in HEX under DDT, then run, and the registers examined.

Now that you've perused Figure 4, let's have a quick recap. In this example we have found for drive C, the DPH starts at F76E, DPB is at F7CB, CSV starts at F98B, ALV starts at F972, the Translate Table starts at FB08, and MTAB starts at F845.

Translate Table Patch

Morrow has a specific way of entering extra sector translation tables. The first byte is a designator for the type of drive format, and the next word is the length of the table. The table itself comes next, followed by an FF and then a space remaining byte.

For SD 8" with six sector interleave, the translation table for the Morrow MD-2 is found in Figure 5.

DPB Patch

The DPB for 8" single density is:

```
HEX FOR PATCH      DESCRIPTION
1A00        DW   026  ;sec per track
03          DB   3    ;Block shift
07          DB   7    ;Block mask
00          DB   0    ;Ext mask
F200        DW   242  ;dsk size-1
3F00        DW   63   ;direct.size
C0          DB   192  ;alloc 0
00          DB   0    ;alloc 1
1000        DW   16   ;check size
0200        DW   2    ;track offset

Note: The HEX patch data is in form
to be patched in, IE low byte
first for all DWs.
```

For a full discussion of these parameters see Digital Research's CP/M 2.2 Reference Manual.

DPH Patch

The DPH must be patched to show the address for the new XLT, and the size of the ALV must be increased by 7 to account for the larger number of 1K sectors used in 8" SD.

Since the CSV space is larger than needed, we can steal from it for the ALV. This will change the CSV address in the DPH from F98B to F992 (F98B + 7).

The new XLT address to be put into the DPH is the address of the first sector in the new table. Since the first three bytes in the XLT are used for identification and table length, the first byte begins at FB0B (FB08 + 3) in this example. (Be sure and use the value you found for your system in place of FB08.)

MTAB Patch

Morrow packs a lot of information about the disk drives in the table entitled 'MTAB.' Take a look at the distribution copy of the BIOS that came with your machine. You will note that there are 9 bytes for each drive. Figure 6 shows what these bytes become for SD 8".

If you wish to change the constants for different step rates, for example, refer to a 765 or 8272 data sheet. Remember that the switch to the 8" drive occurs after all the head positioning. Therefore, SRT/HUT and HLT should be figured for a 4 MHz clock rate.

Making The Patches

The symbol (R) signifies pressing RETURN. Use SYSGEN to create the disk image of the CBIOS shown in Figure 7.

Now swap the disk from drive B into drive A, press RESET, and reboot the system. With the 8" drive hooked up, and with a formatted disk in it, try to pull a directory listing. If there are problems, reinspect the wiring. Use STAT, STATUS, INFO, or DUU to inspect the disk parameters to confirm they were patched correctly.

Other Sector Sizes

By changing the appropriate locations in MTAB and changing the XLT and DPB to match, single density 256, 512, and 1024 byte sectors can be supported.

Formatting

The standard Morrow FORMAT.COM will not format 8" disks. This means you will have to buy formatted disks, or use those formatted on another machine. (One bright note: disks formatted with either a 1771 or 1793 will work with the NEC 765.) Single density 8" disks are normally sold pre-formatted so the lack of a formatter is not usually a problem.

Double Density 8"

The approach used in the Morrow to access the disk is not fast enough to keep up with the higher data rates associated with double density 8". The simple data separator described here will not work for double density anyway.

Operation

After using the 5" drives, the 8 incher seems painfully slow and noisy. Changing the sector size to 1K bytes brings the speed up to parity with the five inchers. This modification, in conjunction with the use of a multiple format disk program such as Uniform, allows maximum interchangeability of software.

■ ■ ■

Figure 3 - Finding Terminal Configuration Data

```
A>DDT<R>
DDT VERS 2.2
-DFA80<R>
FA80 0A 45 78 63 68 61 6E 67 65 20 64 69 73 6B 65 74 .Exchange diskett
FA90 74 65 73 20 61 6E 64 20 70 72 65 73 73 20 5B 52 tes and press [R
FAA0 45 54 55 52 4E 5D 00 FF 1B 3D FF 00 00 00 FF 00 ETURN]...=......
FAB0 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 ................
FAC0 00 28 00 01 02 03 04 05 06 07 08 19 1A 1B 1C 1D .(..............
FAD0 1E 1F 20 09 0A 0B 0C 0D 0E 0F 10 21 22 23 24 25 .. ........!"#$%
FAE0 26 27 28 11 12 13 14 15 16 17 18 FE 1A 00 2A 59 &'(...........*Y
FAF0 54 FF 1A FF FC FA FF FA 02 FB 05 FB 1B 2A FF 1B T............*..
FB00 59 FF 1B 54 FF 1B 2A FF FF BE 00 00 00 00 00 00 Y..T..*.........
FB10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
FB20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
FB30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

Normal Morrow XLT                    Terminal Configuration Data
                  Room Left in Bytes
```

Figure 4 - Locating DPH and DPB in the BIOS

```
The current values are unimportant
-S100<R>
0100 XX 2A<R>
0101 XX 01<R>                    The program used is:
0102 XX 00<R>
0103 XX 01<R>                LD   HL,(1)     ;GET WARM BOOT ADDR
0104 XX 18<R>                LD   BC,018     ;OFFSET TO SELDSK
0105 XX 00<R>                ADD  HL,BC
0106 XX 09<R>                LD   BC,02      ;SELECT DSK C:
0107 XX 01<R>                LD   E,1
0108 XX 02<R>                CALL JUMPHL     ;INDIRECT JUMP
0109 XX 00<R>                NOP             ;ROOM FOR BREAK
010A XX 1E<R>                NOP             ;POINT
010B XX 01<R>          JUMPHL: JP   (HL)     ;INDIRECT JUMP
010C XX CD<R>                                ;TO SELDSK
010D XX 11<R>
010E XX 01<R>
010F XX 00<R>
0110 XX 00<R>
0111 XX E9<R>
0112 XX .<R>
-G100,10F<R>           THIS RUNS THE PROGRAM AND HALTS AT 010F
*10F
                    DUMP THE REGISTERS, ADDR OF DPH IS IN HL
-X<R>
C0Z0M0E0I1 A=10 B=F74E D=F7CB H=F76E S=0100 P=010F NOP
                NOW INSPECT THE DPH FOR C:
-DF76E<R>
F76E C3 FA ..
F770 00 00 00 00 00 00 60 F8 CB F7 8B F9 72 F9 C3 FA ...

  This gives us the following information for drive C:
     DPH ADDR F76E  (Also addr of XLT pointer)
     DPB ADDR F7CB
     CSV ADDR F98B
     ALV ADDR F972

     MTAB is easy to find since Morrow places its address right
     after the CPM jump table... so get warm start address..
           Warm Start   (Start of BIOS is Warm Start-3 or F200)
-D00,04
0000 C3 03 F2 00 00

-HF203,041<R>       ADD OFFSET TO MTAB(41H)
F244 F1C2           MTAB POINTER IS AT F244
-DF244,F245
F244 33 F8<R>       MTAB FOR DRIVE A STARTS AT F833 BUT WE WANT
                    MTAB FOR DRIVE C SO MUST ADD 18 BYTES TO THIS
                    ADDRESS SINCE EACH MTAB ENTRY IS 9 BYTES LONG
-HF833,012<R>
F845 F821
```

Figure 5 - MD-2 Translation Table

```
Marker as SD 8''    Size of Table    Table
0A 1A 00 01 07 0D 13 19 05 0B 11 17 03 09 0F 15 02 08 0E 14 1A 06
0C 12 18 04 0A 10 16 FF A1

End of Table Byte    Space Remaining Byte  (equal to original
                       space remaining less the 1D bytes used
                       or BE - 1D = A1 in this case.)
```

(Figures continued)

Figure 7 - Creating a Disk Image of CBIOS

```
A>SYSGEN<R>
SOURCE DRIVE NAME (OR RETURN TO SKIP)A<R>
SOURCE ON A, THEN TYPE RETURN<R>
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)<R>
A>SAVE 48 CPM64.COM<R>

Then use DDT to find and patch the areas in the SYSGEN image.

A>DDT CPM64.COM<R>
DDT VERS 2.2
NEXT    PC
3100    0100        Look for the start of CPM jump table . . .
D2300,2310<R>       here it is as shown by all the C3 Jumps

2300 C3 C9 FC C3 52 F2 C3 DE F2 C3 E6 F2 C3 48 F3 C3
-HF200,2300<R>      This calculates offset between SYSGEN and BIOS
                    addresses.
1500 CF00           The offset is CF00

-HF76E,CF00<R>      Use the offset to find the DPH
C66E 286E           This is it, now to patch, remembering that low
-S286E<R>           order bytes are first when patching words.
286E C3 0B<R>       New XLT
286F FA FB<R>       ADDRESS

2870 to 2879        No changes

287A 8B 92<R>       Change size of ALV
287B F9 .<R>        Done with DPH patch

-HF7CB,CF00<R>      Calculate address of DPB
C6CB 28CB           It is 28CB
-S28CB<R>
28CB 28 1A<R>       Change sectors per track
28CC 00 00<R>         "
28CD 04 03<R>       Change block shift
28CE 0F 07<R>       Change block mask
28CF 01 00<R>       Change extent mask
28D0 5E F2<R>       Change disk size
28D1 00 00<R>         "
28D2 7F 3F<R>       Change directory size
28D3 00 00<R>         "
28D4 C0 C0<R>       Alloc 0 happens to be the same
28D5 00 00<R>         "
28D6 20 10<R>       Change check size
28D7 00 00<R>         "
28D8 02 02<R>       Track offset the same
28D9 00 00<R>         "
28DA 28 .<R>        Start of DPB for D: so stop.

-HF845,CF00<R>      Calculate position of MTAB
C745 2945           It is at 2945
-S2945<R>
2945 04 84<R>       Start patching MTAB
2946 5A 22<R>
2947 58 01<R>
2948 05 01<R>
2949 6F BF<R>
294A 03 23<R>
294B 05 1A<R>
294C 1C 07<R>
294D FF .<R>        Done with MTAB patch

-HFB08,CF00<R>      Calculate position of XLT
CA08 2C08           Start patch
-S2C08<R>
2C08 FF 0A<R>
2C09 B8 1A<R>
2C0A 00 00<R>
2C0B 00 01<R>
2C0C 00 07<R>       Continue patching in
 |   |  |  |       the XLT until
2C25 00 FF<R>       End of table mark
2C26 00 A1<R>       Space remaining (be sure and use your value here)
2C27 00 .<R>        Done! You may however wish to modify the
                    signon message (at FC00 in RAM, 2D00 in SYSGEN
                    image) to show the modifications.

-^C                      Use Control C to exit DDT and
A>SAVE 48 CPM8.COM<R>    Save the patched result.
                    And  use  SYSGEN to put the patched BIOS on  a
                    formatted disk in drive B.
A>SYSGEN CPM8.COM<R>
SYSGEN VER 2.0 MD1.2
DESTINATION DRIVE (OR RETURN TO SKIP)B<R>
DESTINATION DRIVE (OR RETURN TO SKIP)<R>
A>
```

Figure 6 - Byte Equivalents on 8" SD

Byte 0	84H	Foreign Drive, Single Sided, Mot. Cont. 001
Byte 1	22H	Non virtual drive, Single Density, 80 track (although this makes no difference), 128 byte per sector, single sided, drive C
Byte 2	01H	Motor wait time to minimum since drive runs continuous
Byte 3	01H	Head settle time to minimum
Byte 4	BFH	SRT/HUT to 765
Byte 5	23H	HLT/ND to 765
Byte 6	1AH	End of Track Sector Number
Byte 7	07H	Gap Length 3
Byte 8	FFH	Current Track

Figure 8 - Parts List

Description	Quantity
34 conductor ribbon cable	about 8-10''
34 pin card edge ribbon cable connector	one
34 pin header	one
34 pin ribbon cable socket	one
50 pin card edge ribbon cable connector	one
50 pin header	one
50 pin ribbon cable socket	one
50 conductor ribbon cable	as needed
74LS74 dual D FF	one
74LS193 binary counter	one
74LS123 dual one-shot	one
74LS157 or 74LS257 quad 2 to 1 mux	one
4.7 K resistor	one
1 K resistor	two
220 Ohm resistor	one
10 K pot	one
33 pF mica capacitor	one
.1 uFd capacitor	four
10 uFd	one

Keeping Real Time: The Ztime-I

Review By Ron Biedenbach

185 Hillcrest
Amherst NY 14226

One shortcoming of 8-bit computers is their inability to time/date stamp files. One solution is the Ztime-I calendar/clock by Kenmore Computer Technologies.

Ztime-I is based on the National Semiconductor 58167 and is available in four flavors: bare board, kit, assembled and tested, and assembled, tested, and trimmed for time accuracy.

Easy To Assemble

I assembled the kit, which includes the PC board, all parts, instruction manual, and support software on an 8" disk. It was easy, and took about an hour. The only thing that confused me was the value of the capacitors. They're marked with the industry standard, but that's Greek to me. Luckily, an EE friend was nearby to translate.

When I finished the assembly, I unplugged the Z80 from my Xerox 820-II and plugged it into the clock board. Then I plugged the clock board into the Z80 socket on the computer.

I turned the system on, booted (so far so good), and ran the date configure program to modify DATE.COM and SETDATE.COM to the physical address port to which the clock is tied. Then I ran SETDATE.COM to set the time. From that point on I had a battery backed real-time clock.

It's been running for over a month, and it's lost only two minutes—about 20 seconds a week. (Editor's note: If you find those two minutes you've lost, please save them. You never know when you'll need a couple of extra minutes.)

Enhancements

Kenmore Computer Technologies claims a variable capacitor can be installed and adjusted to improve the accuracy to within +- 8 seconds a month. I haven't tried it.

Kenmore supports 8" single density, Kaypro, Osborne, and Xerox. If you have some other format, you can get support software via modem.

They also plan to develop and support system specific programs which incorporate the Ztime board. It's rumored that future software will include an archive program, an RCP/M-BBS system, and dBASE II modules.

Available from: Kenmore Computer Technologies, 20 Landers Rd., Kenmore, NY 14217, (716) 877-0617. Prices for the Ztime-I range from $29 to $99, depending on the package.

■ ■ ■

component supply, inc.

1771 Junction Ave. • San Jose, CA 95112 • (408) 295-7171 **Hours: 10:00 am - 5:00 pm / MON-FRI**

Shipping charged on all orders. Minimum order: $15.00. Please call on all orders as items are limited to quantities on hand. Prices subject to change without notice. *NO OPEN ACCOUNTS / NO FOREIGN ORDERS, PLEASE!*

SHUGART SA604 HARD DRIVES

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 packing and guaranteed by us for 90 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 50 or 60W power supply, you will need a 130W supply.

Performance Specifications

Formatted (33 sectors track)	
Per Drive	5.40 Mbytes
Per Surface	1.35 Mbytes
Per Track	8.45 kbytes
Per Sector	256 bytes
Transfer Rate	5.0 Mbits sec
Access Time	
Track to Track	16.2 msec
Average	99 msec
Maximum	215 msec
Average Latency	8.33 mesec
Start Up Time (Tvp)	12 sec

Functional Specifications

Cylinders	160
Tracks	640
R W Heads	4
Disks	2
Index	1

DC Voltage Requirements:
+12Vdc ± 5% 1.8 A typical (4.0A max. starting for 6 sec.)
+5Vdc ± 5% 0.9A typical (1.5 A max.)

Ask about quantity pricing. **$95.00**

POWER SUPPLIES:

Power One G5-35/OVP, 5V@35A, OVP preset @6.2V, 17"x5", 18 lbs. **$70.00**
ACDC OEM 5N17-1, 5V@17A, 10"x5"x5", 14lbs. **$45.00**
Trygon LBS6-24-OV, 4.8-6.8V@24A, 6"x17"x8", 36 lbs. **$60.00**
Tokin ST0510-S2, +5V@10A switcher 8½"x4¾"x1¾", 2 lbs. **$30.00**
Sanyo 5V@6A Switcher 7½"x4"x1¾", 1 lb. **$25.00**
Lambda LXS-CC-5-OV, 5@16A, 5"x5"x9.25", 15 lbs. **$45.00**
Lambda LXS-D-5-OV, 5V@27.5A 7½"x7½x9¼", 22 lbs. **$60.00**
Sorenson SSD9-50, 9V@50A swticher, 7½"x5½"x12¼", 13 lbs. **$175.00**
Standard Power 200 B 12, 12V@5A unregulated 4½"x8¼"x5½", 12 lbs. **$20.00**
Dacom 2489-01-C, 12.5V@1A unregulated 5"x5"x3", 2 lbs. **$10.00**
ACDC 15N5-1, 15V@5A, 10"x4"x5", 6lbs. **$75.00**
Standard Power200 B 16, 16V@12.5A unregulated 4½"x8¼"x5½", 12 lbs. **$15.00**
CEA CEA6A20V252, 20V@2.5A, 5"x5"x7¼", 11 lbs. **$25.00**
Sorenson SLC20-5, 20V@5A, 5¼"x3½"x8½", 4 lbs. **$40.00**
Sorenson SSD24-8.5, 24V@8.5A 5¼"x3½"x10", 5 lbs. **$90.00**
Astec AA11261, 5V@6A 28V@1A, unregulated switcher, 7½"x4½"x2" **$50.00**
Compower 70040 +5V@3A, +12V@5A, −12V@.1A, 5"x10¼"x2" **$50.00**
Osborne Power Supply **$25.00**

DISK DRIVES:

- Tandon TM 502 12 megabyte unformated **$325.**
- Syquest SQ 306R Removable 3" 5 megabyte ST506 type interface. ECL 06 **$350.**

SURPLUS Tandon TM 55-2, 5¼", double sided, 48TPI 40 track, 6 msec. track to track* **$ 99.**
SURPLUS Shugart SA455, 5¼, half height, 48TPI, 40 track, 6msec., tracktotrack, dbl. sided **$ 99.**
NEW Remex ⅔ height with full height face plate, 48TPI, 40 track, doublesided. May be used in place of TM 100-2* **$ 80.**
NEW Shugart SA455 / Panasonic JA551/2N 5¼", half height, 48TPI, 40 track, 6 msec., double sided.* **$124.**
NEW TEAC FD55B half height, 48TPI, 40 track, 6 msec., double sided. Power: +12V @ .25A typ., +5V @ .38A typ.* **$139.**
NEW TEAC FD55F, half height, 96TPI, 80 track, double sided **$169.**
NEW TEAC FD35F, 3½", 80 track, 135TPI, double sided **$195.**
NEW DTC Hard Disk Controllers for PC **$250.**

* All drives with asterisk are compatible with the IBM PC™ as 360K or 380K drives. Half height drives may required mounting brackets or filler plates not supplied with the drive.
All drives carry a 90-day guarantee.

MISC. ITEMS:

MOTORS:
TRW 403A117-3, 24 V, 5600 RPM **$7.50**
Canon EN35-T101Z1A, 12VDC, 3400 RPM, 12 g/cm, 110 ma **$10.00**

STEPPER MOTORS:
Copal Electric SH-65, 40 60Ω, 12 V, 1.8° **$2.00**
for Shugart 8" Drives **$7.50**

AC ADAPTORS:
Viewsinics VSADP-20, 9 VAC, 150 ma **$2.00**
Basler Electric BE24V20, 24V, 20VA **$2.00**

LCD DISPLAYS:
Toshiba LT 8026-35, 16 Line X 46 Column **$7.50**

DISKETTES: Almost New 5¼" **$ .65**

DOCUMENT CARRIERS,
2-pocket leatherette 8½"x11 **$1.95**

SPEAKERS,
Matsushita EAS4P15SA, 8Ω, 15/8" **15/$10.00**

LED HOLDERS,
Siemens 2004-9019 Right Angle **10/$1.00 200/$10.00**

TERMINALS, Wyse 100 **$195.00**

LINE FILTER/FUSE HOLDER Combination, Corcom 6J4F1881 **$5.00**

MONITORS, Osborne 5" Green Screen **$35.00**

I.C.'s:

Available for immediate delivery —

F-8	Single Chip Microcomputer		$1.00				
10101	1.00	10158	2.00	SCN2652A	Signetics		15.00
10102	1.00	10160	2.00	2708	1024x8 EPROM		2.50
10104	1.00	10161	2.00	2716 450ns			3.50
10106	1.50	10162	2.00	2732A-3 300ns			4.00
10107	1.50	10170	2.00	2764 300ns			5.00
10113	2.00	10172	2.00	27128	300ns		16.00
10117	2.00	10175	2.00	2758	1024x8 EPROM 54 only		3.50
10121	2.00	10176	2.00	2901	bit slice microprocessor		3.50
10130	2.00	10180	2.00	LM 304	Adjustable Neg. Volt. Reg.		1.50
10132	2.00	10182	2.00	CA 3075	FM dtector, lim., audio preamp		5.00
10133	2.00	10197	2.00	CA 3082	7Hi. Cur. NPN Transistors Com. Coll.		1.00
10134	2.00	10212	2.00	LM 309H	+5V 200 ma. Voltage Regulator		1.50
TID 126	Texas Inst.		1.00	LM 310N	Linear Follower		1.50
TIL 111	Opto. Isolator		1.00	LM 339	Linear Comparator Quad		1.00
1400S	16K Static 55ns.		4.00	DS 3486	Four Three State 422/423 Receiver		3.00
HP146818	R.T.Clock & RAM		5.00	DS 3487	Four Three State 422/423 Receiver		3.00
TIL 156			1.00	4N26			1.00
AF 132CJ	Nat'l Modem Filters		5.00	4N37			1.00
TD9 185030	32x8 Prom		2.50	MK4027N-3	4Kx1 200 NS		1.50
1702	256x8 Ims EPROM		1.50	TMS4045-20NL			4.00
1793-02	floppy disk controller		18.00	4164 200ns or Foster			3.75
2016P-7	2Kx8 Static 70ns.		3.50	RM 4136 DC	Quad Linear Op Amp		1.00
ULN2032A	Sprague		1.00	14412	11.00	14516 Harris	1.25
2101-2	2.00	2104-4	2.00	4528	1.00	14539	1.50
21FO2	2.00	2108-2(A6L)	2.00	4529	1.00	14553B	5.00
2104	2.00	2108-4	2.00	UCN 4810A	10 bit Fluorescent Dis. Driv.		2.00
KR 2376 ST	SMC keybd. encoder		10.00	TL 494CN	Switching Regulator Circuit		3.50
P 2405	Intel		2.00	COM 5016	15.00	COM 5025	15.00
				MM 5204	EPROM		1.50
				MM 552D	analog switch		2.00
				MM 5314	Clock Circuit		4.50
				MM 5316	Clock/Watch Circuit		4.50
				MM 5318	Clock Circuit		4.50
				MM 5330			6.00
				MM 5370			6.00
				MM 5375AA			4.50
				MM 5375AB			4.50
				6522A	Periph. Interface Adapt.		6.00
				6264LP-12	8Kx8 Stat. 120ns Tested/Desoldered		6.00
				6545A-1	CRT Controller		19.00
				6800-L	2.50	6845	12.00
				68010L8	16 bit/Virtual Memory		40.00
				6810	128x8 Static Ram		1.50
				68450-8	DMA Controller		130.00
				68452	Bus Arbitration Module		28.00
				7220	Graphics Controller		36.00
				74116	1.50	74LS189	8.00
				74150	1.25	74LS244	2.00
				74159	1.50	74LS259	2.50
				74181	2.00	74LS273	1.25
				74198	1.25	74LS299	1.50
				74251	2.50	74LS366	.40
				74390	1.00	74LS377	1.00
				74C154	2.00	74LS379	1.00
				74C193	.50	74S74	.50
				74C89	4.00	74LS280	1.50
				74LS147	2.00	74S373	2.00
				74LS181	2.00	74S374	2.00
				74S571	512x4 Prom		2.50
				SN 76005N	D 5-10W Pwr. Amp.		4.00
				76477			3.50
				765A	Floppy Cont.		18.00
				C80186-3	8Mhz 16bit CPU		85.00
				8080A			3.50
				8086			22.00
				8088			22.00
				81LS97	Octal 3 state buffer		1.25
				8202	15.00	8237A-5	18.00
				8212	1.00	8278	10.00
				8216	1.50	8284A	5.00
				8251	4.00	8286	6.00
				8253	6.50	82S123	2.50
				8255	4.00	82S130	3.50
				8259	6.00	82S137	3.50
				8275	27.00	82S141	3.50
				82S16	3.50		
				82S23	32x8 Prom		2.50
				82S90			3.50
				8741			27.00
				8755			20.00
				93453C	Fairchild 1024x4 Prom		4.50
				AM9517-5PC	DMA Controller		15.00
				96L02			1.50
				Z80 SIO (MK3887)		Mostek	9.00
				Z80 CTC (MK3882)		Mostek	2.00
				Z80 PIO (MK3881)		Mostek	2.00
				Z80 CPU (MK3880)		Mostek	2.00
				Z80A CPU			2.50
				Z80A DMA			9.00
				Z80A SIO/0			10.00
				Z80A SIO/2			10.00
				280A CID			12.00
				Z80A DART			7.00

CAPACITORS:

.1 uf monolythic caps. **100/$8.00 1,000/$60.00**

Pascal Procedures

By John P. Jones

6245 Columbia Ave.
St. Louis MO 63139

Because there's so much Pascal software in the public domain, you often need to sift the wheat from the chaff.

If the software you find is not exactly what you need, you are faced with the problem of modifying the code to fit your application. If the original was written in Pascal/Q Version 7.342C (Infinity Vaporware, Atlantis) and you only use Imaginary Pascal-7 (Thoughtless Products, Gotham City), you will most likely have to translate portions of the code.

By the way, even though Pascal/Q is only available on 12", half density, triple-sided diskette and requires that all variable names be entered in Greek, at $19,999.95 it's a reasonable buy. The price includes shipping by Percolator Timewarp.

Not Getting Lost In The Translation

Here are some basics for translating between compilers.

First, try compiling the code. If you're lucky, the author will have used "standard" Pascal, and it will compile correctly. More likely, though, there will be compiler errors.

Especially if you are new to Pascal, get a reference manual for the source compiler. It may be critical. The differences are often subtle and difficult to find.

Getting the program to compile correctly involves changing syntax for similar but not identical procedures, writing new procedures to emulate those your present compiler lacks, and editing identifier names. For instance, some compilers use as few as eight significant characters in names, but an author will occasionally use more than eight, so that INPUTFIL and INPUTFILE may actually be the same. You might even have to break the source into "include" files.

Expect to find differences in file I/O. Some compilers use GET/PUT for sequential I/O, while others have extended READ/WRITE. Remember that GET and PUT use a pointer variable to access the file, so the statements:

```
f^ := var;  put(f);

are equivalent to: write(f,var);

and var := f^;  get(f);

are equivalent to: read(f,var);
```

This may seem backwards, but there is an implied GET when a file is reset, so file input is a "look ahead" operation. Some compilers may only allow writes to files that have been opened with REWRITE.

More Differences

RESET/REWRITE will probably be different, and if any random file I/O is used, expect to rewrite an entire section of the program.

You will have to totally reconstruct non-text data files. Text files and untyped files (which some compilers don't support) are the only means of transferring data between compilers, since all other files are written using the same binary format the compiler uses for internal data storage.

String procedures are another problem area because they lack a defined standard and have unique extensions. Some compilers allow the "+" operator to concatenate strings, while others use only the CONCAT procedure. The range of allowable operations between STRINGS, character arrays, and CHAR variables will also differ widely.

Comparisons

JRT (now Nevada) Pascal source is particularly difficult to translate because of its non-standard syntax and relaxed type checking. Translations among Pascal/MT+, Pascal/Z, and Turbo Pascal are somewhat easier, but can still be a chore. Pascal/M to MT+ or Turbo translations are somewhere in between. Sad to say,

(continued next page)

Figure 1 - Comm Program Rewrite

```
{$c-}                          {Disable ctl char interpretation }
program basic;
const
  baudrate = 10;               { 2400 baud using COM8116 }
  brport = 0;                  { SIO channel A baud rate register }
  serialio = 4;                { SIO/A data I/O }
  serialstat = 6;              { SIO/A control/status }
  serialctl = serialstat;
  initvals : array[3..5] of byte = ($c1,$44,$ea);    { initialization values
                                                       for register 3-5 }
  rdabit = 1;                  { receive data available status bit }
  tbebit = 4;                  { transmit buffer empty status bit }

var
  ch : char;

function txok:boolean;
{ Read SIO status, return TRUE if xmit buffer empty }
begin
  txok := port[serialstat] and tbebit <> 0;
end;

function rxok : boolean;
{ Read SIO status, return TRUE if receive char available }
begin
  rxok := port[serialstat] and rdabit <> 0;
end;

procedure sioinit;
{ initialize sio registers }
var i : byte;
begin
  port[brport] := baudrate;  { set COM8116 rate register }
  port[serialctl] := 0;      { disable SIO interrupts }
  for i := 3 to 5 do
  begin
    port[serialctl] := i;            { register address }
    port[serialctl] := initvals[i];  { register data }
  end;
end;

procedure send_to_8052;
{ send a disk file to BASIC }
var
  line : string[128];      { longer than needed for insurance }
```

(Listing continued)

(continued from page 57)

I've had the most trouble translating from 16-bit (MS-DOS) Turbo to 8-bit Turbo.

Once the program compiles correctly, it MIGHT run correctly. If not, you'll have to look for the subtle differences noted above. Example: for Turbo Pascal, input from a TEXT file "looks ahead" at the following character to determine EOF and EOLN. Some other compilers do not.

The final step in translation is optional. After the program is running correctly, it can be optimized for your compiler. Procedures can often be replaced with "built-ins" not available in the source Pascal. Using them can save code and increase efficiency.

By this time, you may be wondering if it's easier to just start from scratch and write your own code. In some cases it is, but often the algorithms and logic in the source program are a real help.

Communications

My current project at work involves a custom micro based on the Intel 8052. To debug the hardware, it was cheaper for us to use the version of the chip which has BASIC in the mask ROM (8052 AH-BASIC) since the interpreter has all the facilities needed to directly access memory. (It uses memory-mapped I/O.) To use the BASIC interpreter, all you need, in addition to program RAM for the 8052, is a serial terminal.

An alternative to a dedicated terminal is another computer running a terminal emulator program. I first brought up the board using a communication program running on a Kaypro 4.

As testing became more involved, we needed to be able to load/save BASIC programs from the Kaypro's disk. Unfortunately, the program's send/receive facilities were incompatible with the BASIC. Rather than try to modify the comm program (no source) I wrote the program in Figure 1.

A Word Of Explanation

The comments should explain most of the program, but I should point out a couple of things. Because of its lack of hardware scroll assist, the Kaypro can't run this program faster than 2400 baud. My Big Board will run it happily at 9600 baud. The Save command buffers the program in memory rather than writing directly to disk for two reasons.

First, since the ROM BASIC does not toggle output on and off with ctl-S, ctl-Q the way CP/M does, writing to memory was an easy way to avoid missing characters during disk writes. Second, the memory buffer provides an opportunity to strip the echoed "LIST" command and the trailing "READY" from the BASIC source. Of course, I can save the memory file to disk once the transfer is completed. The saved input files can be

(Listing continued)

```
  bas_prog : text;          { use text files }
  fn : string[14];          { file name }
  i : integer;
  ch2 : char;               { temp storage }

begin
  repeat
    writeln;
    write('Name of BASIC program file: ');
    clreol;
    readln(fn);                 { get BASIC program filename }
    if pos('.',fn) = 0 then fn := fn + '.bas';   { I don't like to type }
    assign(bas_prog,fn);
    {$i-}                       { disable I/O check since don't want to }
    reset(bas_prog);            { halt program if file not there }
    {$i+}
  until ioresult = 0;           { loop til get good file name }
  repeat
    readln(bas_prog,line);      { get a line of BASIC }
    line := line+^m;            { append a <CR> as terminator }
    for i := 1 to length(line) do     { send line, char by char }
    begin
      while not txok do;        { empty loop waiting for xmit ok }
      port[serialio] := ord(line[i]);  { ship out the char }
    end;
    write('+');                 { let us know a line was sent }
    delay(500);                 { give BASIC a chance to store the line }
  until eof(bas_prog);          { send the whole file }
  close(bas_prog);
  while rxok do ch2 := chr(port[serialio]);    { clear SIO input buffer }
  writeln;
  writeln('File sent');         { signal done }
  ch := ch2;                    { echo last char BASIC sent to screen }
  write(ch);
end;

procedure get_from_8052;
{ capture BASIC program from a LIST command }
const
  buffsiz = 8191;               { only have 8K ram on the beast }
var
  line : string[127];           { longer than needed }
  bas_prog,temp : text;
  fn : string[14];
  i,j : integer;                { input & output capture buffer pointers }
  ch2 : char;
  buffer : array[0..buffsiz] of char;

procedure getline;
{ pull a line of source from the input buffer }
begin
  line := '';    { clear assembly area }

  while (buffer[j] in [^J,^M]) and    { skip leading <CR> <LF> and }
        (j < i)  do j := succ(j);     { check to be sure don't overrun buffer}

  while (not (buffer[j] in [^J,^M])) and (j < i) do  { copy til <CR> or <LF> }
    begin
      line := line+buffer[j];   { append the char }
      j := succ(j);             { bump the buffer pointer }
    end;
end;
```

sent practically verbatim to BASIC.

I have since extended the program to display the disk directory, "TYPE" files, and dump files to the printer. The program could be further extended to become a full blown communication utility. In fact, the Borland SIG on CompuServe has a Turbo version of MODEM7 available for downloading (it's specifically written for 16-bit Turbo!). A print program translation to Turbo that I mentioned in an earlier column is also available on the Borland SIG.

```
begin
  writeln('Reading BASIC program into memory.');
  fn := 'LIST'+^m;                { set up and send LIST command to BASIC }
  for i := 1 to length(fn) do
  begin
    while not (txok) do;
    port[serialio] := ord(fn[i]);
  end;
  i := 0;                         { input buffer pointer }
  repeat
    if rxok then                  { char by char capture of program }
    begin
      buffer[i] := chr(port[serialio]);
      i := succ(i);
      if buffer[i-1] = ^m then write('+');   { if got <CR> was complete line }
    end;
  until (keypressed) or (i>buffsiz);  { don't know really when BASIC is
                going to quit, so when +'s stop, hit a key to exit loop }
  read(kbd,ch);
  writeln;
  write('Name of BASIC program file: ');    { get name & open output file }
  readln(fn);
  if pos('.',fn) = 0 then fn := fn + '.bas';
  assign(bas_prog,fn);
  rewrite(bas_prog);
  j := 0;                     { buffer pointer for output }
  getline;                    { get and throw away echoed 'LIST' }
  getline;                    { get first real program line }
  repeat
    writeln(bas_prog,line);       { write to file }
    getline;                      { get next line }
  until (line='READY') or (j >= i);   { til get BASIC's READY output or
                                        buffer overrun }
  close(bas_prog);
  write('>');                 { simulate BASIC's prompt }
end;


begin
  sioinit;   { initialize baud rate and SIO }
  while rxok do ch := chr(port[serialio]);   { clear SIO's input }
  port [serialio] := 32;          { send space to BASIC for auto baud }
  repeat
    if rxok then                  { character available from BASIC? }
    begin
      ch := chr(port[serialio]);  { yes, grab it and echo }
      write(ch);
    end;
    if keypressed then            { something we need to send ? }
    begin
      read(kbd,ch);               { yes, get it, then check for command char }
      if not(ch in [^L,^S,^Z]) then
      begin                       { if not command char, ship it out }
        while not txok do;
        port[serialio] := ord(ch);
      end;
    end;

    case ch of                    { process potential command char }
      ^L : send_to_8052;          { ^Load command }
      ^S : get_from_8052;         { ^Save command }
    end;
  until ch = ^Z;                  { ^Z is exit program command }
end.
```

End of Listing

Turbo Tips

Turbo Pascal Version 3.0 is now being shipped. The 16-bit version boasts substantial performance improvements and extensions. (A friend's IBM PC can now keep up with my 5MHz Big Board.) The 8-bit version has fewer changes. Some bugs have been fixed in MARK/RELEASE and the overlay handler, and several procedure/functions have been added:

Exit—exits the current block (a cheap GOTO). Ovrdrive—specifies where overlays reside, replaces Y compiler directive. Paramcount—returns number of parameters in command tail. Paramstr(N)—returns nth parameter from cmd tail. Seekeoln, Seekeof—skip trailing whitespace. Fourth parameter to BLOCKREAD/WRITE returns actual # records read. Facilities are provided so you can write your own runtime error handler.

I've found no significant differences in the speed of compilation or execution between the 8-bit versions 2 and 3.

Warning

There have been changes in the way that both the CON and TRM devices handle input. You may need to modify any programs you have that specifically access these logical devices.

Pages 260 and 262 of the manual mention a "P" compiler option used for passing command line parameters when operating in Memory mode. This option is not available in the 8-bit version.

The manual has grown to over 370 pages, and much of the expansion is IBM PC specific. This manual has the same consistent quality and detail I have come to expect from Borland.

Borland will give a $39.95 trade-in credit towards the purchase of version 3.0 if you return your original version 2 diskette with your order.

■ ■ ■

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 many come with source.

8" Users Disks $15.00 each

USERS DISK #1
1-Two fast disk copiers
2-The manual for Small C+
3-Crowe Z80 Assembler
4-Two disk formatters
5-Modem 7
6-Othello
7-Serial print routine-Port B

USERS DISK #2
1-Two single disk drive copy programs, both with source
2-Crowe Z80 Assembler source
3-New Crowe.COM file, debugged version
4-New CBIOS with parallel print driver & other extensions for CP/M 1.4 & 2.2
5-Disk mapper with source

USERS DISK #3
1-EPROM burning software for BB 1
2-Reset bit 7 (unWordStar a file)
3-Disk file CRC checker
4-New fast copy program & source
5-DU77, disk inspector/editor
6-FINDBAD, isolates bad disk sectors
7-Print fancy page headings

USERS DISK #4
1-CBIOS, custom bios for Tandon drives
2-ZCPR, dynamite CCP checks drive A for missing .COM files; improved commands
3-ZCPRBLOC, identifies CCP location

USERS DISK #5
1-CAT, disk cataloging routines
2-Modem 7 for Port A
3-Modem 7 for Port B
4-PACMAN, the arcade game
5-FAST, buffers the disk to speed up assemblies
6-NOLOCK, removes BB 1 shift lock
7-VERIFY, cleanup & verify a flaky disk
8-DUMPX, enhanced for BB 1
9-UNLOAD, create .HEX file from .COM file

USERS DISK #6
1-REZ, 8080/Z80 disassembler, TDL mnemonics
2-PRINTPRN, prints Crowe listings
3-RUNPAC, run-time utility package for 8080 assembly language programs. Has 51 functions. Includes source which assembles under ASM.

USERS DISK #7
1-CHNGPFM, PFM monitor mods
2-TERM, terminal routines let you set up BB as simple terminal, as a file receiver, or as a file sender
3-Checkbook balancing package
4-Disk Utilities - copy to memory, from memory, and dump.

USERS DISK #8
1-BDSCIO, custom BDSC I/O for BB 1 (both .h and .c)
2-YAM, Yet Another Modem program in source & .COM form. Turns BB into paging intelligent terminal, complete with printer interface, baud rates to 9600.
3-ROFF, text formatter
4-SIGNS, prints large block letters

USERS DISK #9
1-ADVENTURE, expanded 550 pt version
2-Keyboard translation program
3-CBIOS, serial & parallel printer interface
4-EPROM programming package for BB II, for 2732s only

USERS DISK #10 - Lots of Disk Utilities
1-REBOOT, sets up the CP/M auto load
2-SWEEP, directory/file transer routine
3-A, Lets BB I recognize a double sided drive as one drive with 494K of usable space
4-FIX, super disk utility, does everything, much easier to use than DU77
5-Compare files routine
6-UNERA, retrieve erased files
7-FIND, check all drives on system for a file
8-MENU, menu program for CP/M
9-NEWCAT, enhanced disk catalog program
10-Single drive copy program that does track by track copies rather than file by file

USERS DISK #11 - Printer Utilities
1-Microline 92 printer routine
2-Graphics display package for MX-80 with Graftrax, very fancy
3-Epson MX80 setup for BB 1 with 59.5K CP/M
4-Epson MX8 setup for any CP/M, lets you set print modes.
5-Micro Tek print driver, Ports A & B

USERS DISK #12 - Games for BB I
1-ALIENS, a fast, exciting arcade game
2-ZCHESS, chess with a 1-6 level look ahead
3-MASTERMIND, match wits with the computer
4-BIO, Biorhythm charts complete with graphics on the BB I
5-LIFE, so fast it's real animation!
6-CRAPS, see how much you'd lose in Vegas
7-WUMPUS, a caver's delight, kill the Wumpus or be killed
8-PRESSUP, similar to Othello
9-Games, 7 games in one program, includes blackjack, maze and animal

USERS DISK #13 - General Utilities, BB 1
1-ZZSOURCE, disassembles to real Zilog mnemonics
2-EX14, superset of submit or supersub
3-MOVPATCH, lets you use MOVECPM on other copies of CP/M
4-XMON, 3K expanded BB I monitor, use in ROM or as overlay .
5-CURSOR, prompts you for cursor char you want
6-UMPIRE, very fancy RAM test
7-ZSIDFIX, display improvement for ZSID
8-PIPPAT, modify PIP so you can reset system from within PIP
9-@, Lets you use the BB as a calculator, including HEX
10-SORT, sort package written in C80.

USERS DISK #14 - BB II Software
1-PRO32, latest 2732 reader & programmer
2-SMODEM2, lets BB II talk to Hayes Smartmodem
3-GRAFDEMO, demonstrates BB II graphics (in BASIC)
4-ATTRTEST, demonstrates BB II graphics (in JRT Pascal)
5-INITSIO, initializes port B for 300 or 1200 baud
6-MENU, displays menu of .COM files, enter number to run file
7-SETCLK, sets realtime clock built into BB II
8-PRINT2, modified print which accesses BB II clock
9-BOX, draws a thin line box on screen determined by HL and BC
10-ALIENS, space invaders arcade game
11-LISTSET, printer interface, auto-enables RTS, ignores DCD.

USERS DISK #15 - Word Processing
1-EDIT, very fancy line editor similar to EX (Unix). Includes help menu, programmable key, and full manual on disk.
2-TED, simple minded line editor, easy to learn & use. Very fast.
3-TTYPE, typing training program written in BASIC
4-TINYPLAN, very simple-minded spreadsheet. Whets your appetite for a fancy one.
5-C80 Text Utilities
6-CHOP, cuts off file after N bytes
7-ENTAB, replace spaces with tabs where possible
8-MS, double or triple spaces a file to output
9-RTW, removes trailing spaces from file
10-TRUNC, truncates each line to specified length
11-WRAP, wraps at column 80, plus pretty pretty printing, page #s . . .

USERS DISK #16 - BB I Modem Software
1-RCPM27, list of U.S. bulletin boards
2-SMODEM, interfaces BB I with Hayes Smartmodem
3-PLINK66, easy to use with non-CP/M host, for port A
4-BBPAT, menu selection of BAUD rate, bits/char, parity, & stop bits
5-MODEM 7+, Modem 7 plus BBPAT, lets you talk to anything from port A

USERS DISK #17 - Small C version 2
SMALLC2, this substantially expanded version of Small C now includes for, goto, label, switch (case); external declarations; new preprocessor commands; expanded I/O includes redirection; initializers; plus 12 new expressions. The I/O and runtime libraries have been greatly expanded (including printf). Source & documentation on one full disk.

USERS DISK #18 - FORTH
IFORTH, this is Idaho FORTH which can be burned into ROM or loaded from disk. It replaces the PFM monitor & handles all the monitor functions. See issue #11 FORTH column for more info about IFORTH and this disk.

USERS DISK #19 - BB I Double Density
New BB I Monitor, BIOS, character ROM, Winchester Interface, ZCPR, and formatter from Trevor Marshall. See BB I expansion article in Issue #11.

USERS DISK #20 - Assemblers
CROWEASM: This is the Crowe assembler modified so that it runs on any CP/M system (including the BB I, BB II, Xerox . . .). Includes .COM .Z80 and .DOC files.
LASM: This assembler is similar to the ASM that comes with CP/M except that it can link files at assembly time.
PRINTPRN: Print routine for CROWEASM .PRN files.
LIBRARY: Utilities which let you combine many files into one, then you can run, type, or extrace any file within the larger system.

USERS DISK #21 - Winchester Utilities
BACKUP: Helps you back-up the winchester onto multiple floppies. Creates a catalog of the files on each disk and includes the date of the latest backup. Will not back-up an unchanged file more than once. Plus many more super features.
FLOPCOPY: Lets you make floppy copies (with only one floppy drive) by using the winchester as a buffer.
BIGBURST: Backs up a very large winchester file onto multiple floppies. Joins the copies to recreate the original file.
MULTCOPY: Use this like PIP but it prompts you to change 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 TYPE, ERA, DIR, and PIP.
UNSQ: This is the latest, greatest file unsqueezer. Enter UNSQ *.* and it will check every file on the disk. All squeezed files will be unsqueezed.

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 screen dump fromWayne Sugai(with source); modifications for the SWP package including ZCPR, a new monitor, and a clock/calendar from Mitch Mlinar; 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 printer 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 - BBII CP/M 3.0 Banked BIOS/ Winchester Support
CP/M 3.0 Banked BIOS implementation for the BBI. Roy Epperson's software to support the Adaptec ACB-4000 SCSI and the Rodime R204 5" Winchester on the BBII (see issue #19). Plus more Winchester programs.

USERS DISK #27 - BYE Remote CP/M System
BYE programs to run your BBI, BBII, or XEROX 820-I as a remote CP/M system using a Hayes Smartmodem compatible modem. 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 DISK. Also, Larry Blunk's documentation and software for implementing extended single density (334K) on eight inch disks.

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

FORTHwords

By Arne A. Henden

7415 Leahy Road
New Carrollton MD 20784

I've finally gotten around to reviewing Thinking FORTH. It's taken me this long not because Leo Brodie wrote a terrible book, but because I've been so busy applying using it that writing the review had to wait. But first, I have a bone to pick.

Free FORTH

I'm sick and tired of people berating FORTH when they haven't even tried it. It reminds me of kids who hate peas without ever trying them, just because their friends don't like peas. Therefore, I am releasing a subset of UNIFORTH into the public domain.

Called the UNIFORTH Sampler, the freeware version follows the FORTH-83 standard, and includes an assembler, floating point, and a video editor. Try it! If you like it, send a contribution or an order for the Professional Series. If you don't like it, you haven't spent a penny. Check your local bulletin board, or send $35 to Unified Software Systems for the latest disk in your format. Versions are now available for Z80 CP/M 2.2 and the IBM PC (DOS 2.x); others will be released when there are requests for them.

Thinking FORTH

Leo Brodie is well known for his Starting FORTH, one of the best introductory language texts ever written. Though it is billed as a sequel, Thinking FORTH is an altogether different beast.

Brodie's latest text is concerned with programming techniques rather than language details. He uses FORTH as a vehicle to teach his principles. You don't need to know much FORTH to follow the text, and those of us who use structured programming techniques with other languages will also learn lessons. However, if you are reluctant to use FORTH, don't read this book, or you may become a convert!

Inside The Book

Thinking FORTH contains 300 pages, including eight chapters, five appendices, and an index. Brodie's clear style is enhanced by 15 or so cartoons and several detailed figures. Programming hints are scattered throughout the text.

A unique feature of Thinking FORTH is the set of interviews that Brodie did while writing the book. He quotes often from users, vendors, and Charles Moore to explain his points.

There are several detailed FORTH examples, including: a telephone rate calculator, a Roman numeral printing routine, a tiny video editor, and a listing of his DOER/MAKE construct for vectored execution. All code follows the FORTH-83 standard.

Summary Of Chapters

The book starts with the philosophy of FORTH. Is it a high level language? How much of the underlying structure should be hidden to the user? How efficient is FORTH in designing and executing applications?

Chapter 2 details the analysis phase of software design. Brodie points out both the value and the limitations of planning. Stressing simplicity, he suggests defining the decision rules and data structures before programming.

Chapter 3 concerns the preliminary design and decomposition phase. FORTH is an extremely modular language, and good decomposition is essential. Brodie shows how the traditional application design process falls short when applied to FORTH.

Design and problem solving are discussed in Chapter 4. What techniques are best for solving programming problems in FORTH? How can the FORTH syntax be used most effectively in the final application? What data structures should be used?

Implementation is covered in Chapter 5. This involves a detailed discussion of FORTH programming style: the naming conventions, screen layouts, commenting, load blocks, etc. On this controversial subject, Brodie makes several good points, but relies too heavily on the programmer's ability to choose short, yet useful, names.

Factoring is the topic of Chapter 6. This is the art of breaking your program into useful fragments, separating the reusable parts from the unique.

Chapter 7 deals with the data stack and execution states. Brodie presents a simple stack helpsheet. He suggests methods to keep the data and return stacks clean, and how to avoid using variables. His DOER/MAKE construct demonstrates one approach to vectoring execution and using state tables.

Brodie feels that control structure usage should be minimized, and tells why in Chapter 8. He suggests using decision tables instead of CASE statements. I use both, and find that using CASE is often easier than designing a decision table.

Wrapping Up

Thinking FORTH is unique in the software industry, and I commend Brodie for his approach. The text elegantly demonstrates the power of FORTH, how to approach a problem using FORTH, and how to write code that can be read and debugged. This is one text that should be in everyone's library.

There are a multitude of books on FORTH, but the following texts make up a good nucleus: Starting FORTH (Brodie), as an introductory text; Thinking FORTH, as a style manual; and The FORTH Encyclopedia, as a reference guide to Fig-Forth. Now all we need is a book on advanced FORTH. I would rather have someone other than Leo Brodie write it, though, as a deep text deserves the experienced hand of an implementer rather than a user.

Next Time

Three FORTH computers have been sent to me for review. These boards execute FORTH as soon as they are turned on, and are ideal for OEM and controller applications. I'll describe each in detail, as well as FORTH engines in general.

■ ■ ■

BDOS Vectors (Mucking Around Inside CP/M)

By Tom Geldner

3746 29th St.
San Diego CA 92104

This month's topic is how to beat a dead horse into the ground. Actually, we're going to continue examining locations within CP/M. Next time, we'll do something practical.

At least this column now has a focus. We are going to dedicate our efforts to "The Intermediate to Advanced Pascal or BASIC or dBASE II or Something Programmer Who Wants to Learn More About CP/M and/or How to Do Fancy Stuff With It and/or Get Thoroughly Confused."

Back To The Bee-Doss (BDOS)

Last issue, we talked primarily about how much TPA (Transient Program Area) was available for your programs. Also, by locating the bottom of the BDOS we were able to locate, by inference, the CCP and BIOS. In case you forgot how we did this, Figure 1 is a quick review.

Note that the addresses in the memory map are "offsets" from a known location: the beginning of the BDOS.

As you'll recall, we determined the location of the BDOS by examining its entry vector at memory locations 0006 and 0007, using the value at 0007 (most significant byte) to show where the BDOS started.

Editor's note: Transient programs (WordStar, Perfect Calc . . .) can use all the memory between 100H and the bottom of BDOS (called the transient program area or TPA). These programs use the BDOS vector (at 0007H) as a pointer to the highest address they can use.

Since the console command processor (CCP) resides below the BDOS, its space can be used by the transient program (that's OK, since you won't need it again until you return to CP/M). When you exit a program the CCP is read from the disk and written back where it belongs. This action takes place during a "warm boot."

What's A Vector, Mr. Wizard?

Here I was with my first article, talking about vectors and addresses and stuff, and a reader had the nerve to ask what a vector was. The word doesn't come up very often in high-level language programming, so the question makes sense.

A vector is a fixed location in memory that contains a memory address that may vary. In the case of the BDOS entry vector, we can look at memory locations 0006 and 0007 to find the location of BDOS.

The entry vectors are usually preceded by an assembly language JMP (say JUMP) instruction. JMP aaaa is similar to BASIC's GOTO xxxx where xxxx is a line number. In the case of the JMP instruction however, the aaaa is a memory address. For example, a disassembly listing of the BDOS JMP in your CP/M system might look like:

```
JMP D406
```

And in HEX format:

```
C3 06 D4
```

Note that the address D406 is in standard byte-reversed format with the least significant byte first, most significant second. C3 is the HEX representation of the JMP instruction. Any program that encounters this instruction will continue execution at address D406.

Tom Tackles Turbo

Now that we know what a vector is, let's go back to figuring out what the BDOS vector is. Last time, we were supposed to have had an S-BASIC example, but somehow, typesetting gremlins sent it to Source Code Heaven. (Editor's note: that example is alive and well and residing at the end of this article.) This time, we have a Turbo Pascal program that does the same thing, but first, Figure 2 shows a procedure that makes things easier to understand.

Unlike S-BASIC, Turbo Pascal has no equivalent to a HEX$ function that returns a HEX string representation of an integer value. So, we supply our own. We take the most significant byte and divide it by 16. We convert this value to a hex number or letter by locating its position in the array constant HexLtr. The result is stored in HexStr at the first position. We'll do the same with the remainder (mod 16). Then we repeat the whole process on the least significant byte.

Now that we have our little hex converter, we can make sense out of the example in Figure 3.

Absolute Variables

Some explanations. First, the integer variable BdosJump is positioned at address 0006 (the BDOS jump vector) using the reserved word "absolute." An absolute variable is a variable whose value reflects that of a particular memory location. Absolute variables behave somewhat like PEEK or POKE depending on what side of the assignment statement they are on. Figure 4 illustrates this.

While it is safe for absolute variables to be on the right of an assignment statement (:=), be careful when the absolute variable is on the left since the wrong value in the wrong place could have disastrous consequences.

OK, so moving right along. First, we use our DisplayHex procedure to display the HEX value of BdosJump. Then, the most significant byte of BdosJump is obtained with the HI function and multiplied by 100H. We multiply by 100H because the BDOS starts on an even page of memory (xx00H).

Then, we again display the result using DisplayHex. The BDOS Entry Vector (BdosJump) will usually be an address 6 higher than the start of the BDOS due to the Digital Research serial number.

Fly In The Ointment

There's always a catch! Using the BDOS entry vector as a method for determining TPA space is fine and, in fact, is the correct way of doing it. But, it's not always the correct way to actually locate the BDOS! Here's why.

There are a number of programs (resident system extensions or RSEs) like debuggers, keyboard translators, or screen dumps that have to reside in RAM while other programs are running. (My company, Xpert Software, makes two such products, XtraKey and XScreen.)

One thing these programs have in common is that they relocate themselves into high memory, just below the CCP. They load at 100H just like any other program, look at the BDOS entry vector, subtract 800H to find the beginning of the CCP, and then relocate themselves just underneath the CCP.

To keep from being overwritten by subsequent programs, these programs take the original BDOS entry vector, subtract 800H (for the CCP), subtract the

size of the new resident (RSE) program, and then stick the result back into address 6 (BDOS).

Figure 5 gives another example. Start by assuming that the original BDOS jump vector points to address D406. Now, run an RSE that requires just under 2K worth of working code. (The values on the left side of Fig. 5 are actual numbers taken from my computer, Zorba the Lunch Box.)

If the original BDOS vector is D406, as in Figure 5, the RSE would change the vector to C406. Programs that dynamically allocate storage space (WordStar, Perfect Writer, and others) will check the (new) BDOS vector and determine how much TPA there is so they won't crash into the presumed BDOS. (Turbo Pascal-compiled programs won't do this automatically. They merely assume that the BDOS is still in whatever location it was in during compilation.)

Since C406 is obviously not the real BDOS, any program trying to JMP to the BDOS would get into trouble unless provision is made to redirect the jump to the real BDOS. This must be done by the RSE itself. What actually happens is that when a program jumps to the fake BDOS address, the RSE usually contains a JMP instruction at that address to the real BDOS (unless it does what the BDOS was supposed to do).

OK, back to the fly in the ointment. Since we have a fake BDOS vector, our offset of 800H no longer applies to the CCP, and our offset of E00H no longer

(continued on page 65)

Figure 1

```
---------------
|     BIOS    |
--------------- xx00H + E00H
|     BDOS    |
--------------- xx00H
|     CCP     |
--------------- xx00H - 800H
|             |
|     TPA     |
|             |
--------------- 0100H
   Base Page  |
--------------- 0000H
```

Figure 2

```
procedure DisplayHex(InValue: Integer);
 var HexStr: String[4];
 const HexLtr: array[0..15] of char =
                  '0123456789ABCDEF';
 begin
 HexStr := '0000';
 HexStr[1] := HexLtr[(Hi(InValue) div 16)];
 HexStr[2] := HexLtr[(Hi(InValue) mod 16)];
 HexStr[3] := HexLtr[(Lo(InValue) div 16)];
 HexStr[4] := HexLtr[(Lo(InValue) mod 16)];
 Write(HexStr);
 end;
```

Figure 3

```
var BdosJump:   Integer absolute $0006;
    BdosStart:  Integer;

{insert procedure DisplayHex here}

begin
  Write('BDOS Jump Vector = ');
  DisplayHex(BdosJump);
  BdosStart := Hi(BdosJump)*$100;
  Write(^M^J,'BDOS Start = ');
  DisplayHex(BdosStart);
end.
```

Figure 4

```
BASIC
  SomeInteger = PEEK(6)
Turbo
  SomeInteger := AbsoluteIntegerAt6


BASIC
  POKE(6,SomeIntger)
Turbo
  AbsoluteIntegerAt6 := SomeInteger
```

(Listings continued)

(continued from page 63)

Figure 5

```
Zorba                    Offsets
FFFF ----------------
     |      BIOS      |
E200 ---------------- xx00H + E00H
     |                |
     |      BDOS      |
D400 ---------------- xx00H
     |      CCP       |
CC00 ---------------- xx00H - 800H
     |      RSE       |
C400 ---------------- xx00H - E00H
     |      TPA       |
     |                |
     ---------------- 0100H
     |  Base Page     |
     ---------------- 0000H
```

Figure 6

```
var
  WarmBoot:   Integer absolute $0001;
  BdosJump:   Integer absolute $0006;
  TpaEnd,
  BiosStart,
  BdosStart,
  CcpStart:   Integer;

{insert procedure DisplayHex here}

begin
  BiosStart := Hi(WarmBoot)*$100;
  BdosStart := BiosStart - $E00;
  CCPStart := BdosStart - $800;
  TpaEnd := (Hi(BdosJump)*$100)-$800;
  Write('BIOS Starting Address = ');
  DisplayHex(BiosStart);
  Write(^M^J,'BDOS Starting Address = ');
  DisplayHex(BdosStart);
  Write(^M^J,'CCP Starting Address = ');
  DisplayHex(CcpStart);
  Write(^M^J,'TPA Ending Address = ');
  DisplayHex(TpaEnd);
end.
```

applies to the BIOS! So how do we know where the CCP or BIOS actually start? I thought you'd never ask.

Warmus Booticus Vectoritis

We have another jump vector at memory locations 0001 and 0002. This vector points to the warm boot routine in the BIOS. Once again, by taking the most significant byte of this vector (the byte at address 0002) and multiplying it by 100H (appending 00H) we can determine where the BIOS starts. The BIOS warm boot JUMP vector is rarely, if ever, changed by a program.

Subtracting the appropriate offsets from the warm boot vector, we can obtain the real locations of the CCP and BDOS. So why didn't we do that in the first place? Why fool around with the BDOS vector? The answer is that we were trying to determine TPA size, and to do so, we needed to take into account the possibility of RSEs or other things that may affect memory availability. So, we have two different approaches:

1. Use the BDOS vector to determine TPA space.
2. Use the warm boot vector to find the actual BDOS, BIOS and CCP.

Figure 6 illustrates the correct method of locating the various CP/M working parts. One final thought. By running this program and comparing the values of TpaEnd and CcpStart, you should be able to tell how much room in memory a Resident System Extension is using. If an RSE is not present, then TpaEnd and CcpStart should be the same.

Coming Attractions

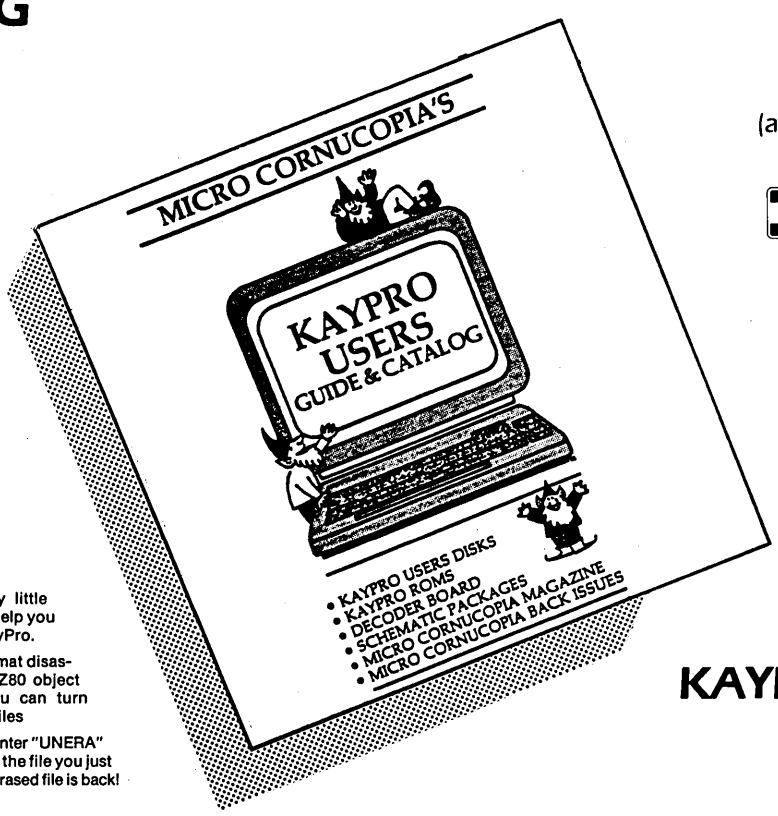
Next time, we will do something practical with what we've learned. This will include some direct BDOS and BIOS calls as well as a mini-tutorial I call "Intro to Assembly Language 1A."

■ ■ ■

You may have noticed in Tom's last article, " "Running In CP/M's TPA (Issue 23, page 67), that Figure 1 was missing. This is it.

```
PRINT "TPA space (from 100H to CCP) in HEX =";HEX$((PEEK(7)-9H)*100H)
PRINT "TPA space (from 100H to CCP) in DECIMAL =";256*(PEEK(7)-9)
```

FREE CATALOG



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 Kaypro II, 4 and 10

KayPro Disk K1
Modem software

KayPro Disk K2
Utilities

Really oodles of spiffy little (and big) programs to help you get full use of your KayPro.

DASM: A true Zilog format disassembler for 8080 and Z80 object (.COM) files. Now you can turn .COM files into .MAC files

UNERA.COM: Simply enter "UNERA" followed by the name of the file you just erased and presto, the erased file is back! A lifesaver!

FINDB54.COM: Checks an entire disk, reports bad sectors, and then creates a special file containing those sectors. You save a bundle on disks.

CAT2: This is a group of programs which create and maintain a single directory of all the programs you have on all your disks. Even keeps track of which programs are backed up and which aren't.

UNSPOOL.COM: Use your KayPro II and print files at the same time. Doesn't slow down system response!

DUMPX, DU-77, COMPARE, SUPERSUB, FORMFEED, DIRDUMP, . . . and all have documentation on disk.

KayPro Disk K3
Games

KayPro Disk K4
Adventure

This disk contains one 191K game. Adventure. ADV/.COM: This is the latest, greatest, most cussed adventure ever devised by half-mortals. This is the 550-point version so the cave is greatly expanded and the creatures are much smarter.

KayPro Disk K5
MX-80 Graphics

KayPro Disk K6
Word Processing Utilities

KayPro Disk K7
Small C Version 2 Compiler

KayPro Disk K8
Small C Version 2 Source

KayPro Disk K9
ZCPR

KayPro Disk K10
Assemblers

KayPro Disk K11
Library & Checkbook Programs

CHECKS: This has been a very popular group of programs. Categorizes checks so you can keep track of which are tax deductible and which get charged to which projects. Includes source and example check files. Very powerful.

LIBR: This is a complete set of library routines which let you group files into a single file called a library. Then CP/M sees them as a single file, but with the library routines, you can list them out separately, run them separately, or divide them up again. Almost like a unix environment.

DISPLAY, VLIST, PGLST: Additional screen and print utilities.

KayPro Disk K12
FORTH

KayPro Disk K13
Source of fig-FORTH

KayPro Disk K14
Smartmodem Program

KayPro Disk K15
Hard Disk Utilities

KayPro Disk K16
Pascal Compiler

KayPro Disk K17
Z80 Tools

KayPro Disk K18
System Diagnosis

Just as we finished editing the routines on this disk, we received a copy of KayPro's diagnostic disk. The memory test and drive exercise routines on this disk are more powerful than KayPro's versions. (Plus, it's only $12) Setup for KayPro II & 4.

KayPro Disk K19
Prowriter Graphics

KayPro Disk K20
Color Graphics Routines

KayPro 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 KayPro's new and old. You can buy a similar package elsewhere for $60.

KayPro Disk K22
ZCPR (Again)

This disk is filled with ZCPR files. You get ZCPR for the KayPro II, KayPro 4, and the KayPro 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 "B:" 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 scrolls text 24 lines at a time, and a new LIST command outputs a file to the printer.

KayPro Disk K23
Fast Terminal Software & New BYE

KayPro Disk K24
MBASIC Games & Keyboard Translator

We sifted through many, many games before coming up with these gems. All will work on any KayPro 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 (pardon the pun). Hunter tries to shoot 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 KayPro's keypad! You can define 8 keystrokes at up to 63 characters each.

KayPro Disk K25
Z80 Macro Assembler

KayPro Disk K26
EPROM Programmer & Character Editor

KayPro Disk K27
Typing Tutor

A complete typing tutor for beginners and experts. Written in Australia, it comes complete with source. This was customized for KayPro II, 4 and 10 by Barry Cole of WLAKUG.

The documentation says you can learn to touch type in 8 hours (probably a little longer for mortals).

KayPro Disk K28
Modem 730

KayPro Disk K29
Turbo Pascal Games 1 With Source

KayPro Disk K30
Turbo Pascal Games II With Source

KayPro Disk K31
Turbo Bulletin Board

Complete Bulletin Board Package for only $12

KayPro Disk K32
Forth-83 Much Fancier Forth

PRICELESS

UPGRADES

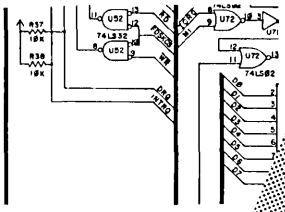
Schematic Packages

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 get detail information on your processor board that's available nowhere else.

For instance, those of you with the 10 and new 84 systems get a thorough rundown on your video section complete with sample video control programs in assembly language and Pascal. Of course, 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) $20
Kaypro 10 (pre-84) $20
Kaypro 84 series (II & 4) $20
All prices include postage.



ROM'S from Micro Cornucopia

Pro-8 Version 2 for Kaypro 4

NEW!!

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. Screen dump with selectable dump character.
2. Select slow or fast step rate for each drive.
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. And, it's still only $49.95 for ROM, disk, and manual!

Of course, you get all the original Pro-8 features such as: user selectable cursor (blinking or not), ignores nulls, and your choice of 1-4 drives of the 191K, 390K, and 784K variety. (Use of 3 or 4 drives requires drive decoder.)

Installation requires no cuts or jumpers. The ROM simply plugs into a Kaypro 4-83 (or II-83 with a Kaypro 4 processor board). If you have a Kaypro II with the original II processor board then you must do the II to 4 upgrade. See issue #22 for details. Your Kaypro II has the original II board if the monitor ROM (a 20-pin chip with paper stuck to its top) is marked with 81-149. The 4 ROM is marked 81-232.

If you already have a Pro-8, you can upgrade to version 2 for half price. Call or write for details.

Pro-884 Monitor

NEW!!

The long-awaited PRO-8 ROM for the Kaypro 2-84 and 4-84s is ready! You have not saved your nickels and dimes in vain!

The PRO-884 ROM does everything your old Kaypro ROM does, plus it:

1. Gives you 784k bytes of storage per disk (with quad-density 80-track, double-sided drives).
2. Uses any combination of Kaypro II, IV, or quad-density drives.
3. Includes software to let you format and copy II, IV, or quad-density disks.
4. Comes with complete printed instructions on ROM installation and tips on installing quad-density drives.
5. Runs up to four drives with the Plus-4 Decoder Board.

Installation of ROM takes just five minutes and requires no cuts or jumpers.

Pro-884 Max Monitor

NEW!!

The ROM that does everything but wash the dishes is here! There's so much stuff in this ROM, even we didn't think it would fit!

The PRO-884-MAX ROM does everything the PRO-884 does, PLUS IT:

1. Emulates a DEC VT-52 or Heath H-19 video terminal (both the display and keyboard).
2. Does FASTER screen output.
3. Has ZCPR1 in ROM, eliminating the need for a system disk in drive A: during a warm boot.
4. Dumps the screen to your printer at the touch of a single (user definable) key.
5. Sends characters going to the screen to your printer regardless of the program you are running.
6. Displays the time on the Kaypro's status line (Kaypro 4-84s only). Because of battery backup, the time does not have to be re-entered each time you power up.
7. Inserts the time and date into text while you are running your favorite text editor (4-84 only).

Prices:

Pro-8 Version 2 Pkg. 49.95
Pro-884 Pkg. 59.95
Pro-884 Max Pkg. 79.95

PLUS-4 Decoder Board

With this nifty little plug-in board, your Pro-8 ROM can access up to four 5¼" drives. You just plug a four-drive 34-pin cable into this board and you can add up to two additional drives.

Now you can run any mix of 191K, 390K, and 784K drives as drives A, B, C, and D. You can run your original drives as A and B then add 380K or 784K drives outboard as C and D. You can even run four half-wides inside your original Kaypro!

The Plus-4 Decorder Board for only $39.95

Watch for 4-84 and 10-84 compatible ROMs coming soon.

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 can locate 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 chunge 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 up 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 cr
**L5 cr*
(ddt's response)

The first line of the response will be a JMP D600 or a JMP D800. The JMP D600 means that you have a low (minus) version, and the JMP D800 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. Otherwise, we'll just guess that you need the normal Max.

On Your Own

By Hampton Miller

PO Box 816
Carpinteria CA 93013

Editor's note: The following is excerpted from the "On Your Own" session at SOG III. Hampton Miller led off by discussing the reasons he was working through a broker rather than directly for the client, and how he was going to be publishing and promoting his book.

Of course, the SOG is not an official event (it's only semi-official), so members of the audience were free to pitch in with their own ideas and experiences, which, of course, they did. It was a very enlightening session played to a standing room only crowd.

Here's Hampton's story followed by comments from the audience. (If you're interested in what Hampton is doing this year, attend his session at SOG IV.)

When I first arrived in California I was really hurting for work so I tied in with a broker. I limited myself a lot when I specified I wouldn't do anything related to the military, but the broker found a place that was doing some interesting communications projects, so I went to work.

Plus 30 Percent

A broker collects his fee from the client, not from the engineer. So he just totals your charges, adds 30 percent, and then bills the client. All three parties seem mutually satisfied with this arrangement.

(Being a broker can be pretty lucrative if you're getting 30 percent from 10 or more people. Of course, you have to be established and have really good contacts with the business community. Businesses want stability.)

If you are a really good designer and charge a lot, businesses usually won't hire you directly. They'd much rather go through a broker and pay the extra money. They trust a broker because they know he won't put in a flake who would hurt his reputation.

As a consultant, I work for an hourly fee, fill out time sheets, get them signed by the client supervisor, and give them to my broker. A week later I have a check in hand. The broker bills the client monthly. If there is any question about the work, the client talks to the broker, not me.

All I have to do is what I'm very good at—software engineering. Without the broker, I'd have to be a negotiator, lawyer, and engineer, and I'd still get burned. But problems with clients rarely get out of hand because the broker mediates disputes.

What causes problems? Misunderstandings, mostly. A good way to help prevent misunderstandings (whether or not you're working for a broker) is to get everything in writing. Sam Baldwin said that verbal agreements aren't worth the paper they're written on. Even (especially) when you are working for friends, you really need to have a written agreement. If you don't, you'll lose a friend.

Anyway, how do you find a good broker? One way is to ask around. Almost everyone has a friend, or a friend of a friend, who works or has worked through a broker. Personal references are your best clue to who's reliable and who's not. My broker is Mini-Systems Associates, 634 Venice Blvd, Marina del Rey, CA 90291.

A lot of executive recruiters keep resumes on file, and they sometimes function as brokers, but you need to watch them with a jaundiced eye.

Books

So you've decided to start writing and publishing your own books. What do you write about? Look at the computer market right now. On one hand, there is the flood of beginner's texts, and on the other hand, there are the incomprehensible tomes by people trying to convince you how smart they are. There is almost nothing in between.

That leaves a very large market of all those people who have read all those introductory texts and are sick of them. They are waiting to go one step farther. At the very least, you can take some clever program you've written, or some neat piece of hardware, and narrate it step by step. Don't leave anything out, and let them in on special tricks you use that make all the difference.

Pricing

By self-publishing, you don't have to charge $34.95 per book. You can charge $12 or $15, still turn a handsome profit, and reach far more people. A 200-page book costs $2.16 each when you print 10,000. If you only print 200, they cost $9 each.

Get the price sheet from Ken at Maverick Publications in Bend, OR. The number is 503-382-6978. I publish through Maverick, and they do the whole thing for you—typesetting, printing, binding, the whole works—for not a whole lot of money. If you send them everything on disk, it's 20 percent off. Ken does it out of his house (he has a building in the back). He has a Z80 system and three typesetting machines—good stuff.

Audience Questions And Comments

"Once you write your book and publish it, how do you sell it?"

Hampton: Advertise, advertise, advertise. Micro Cornucopia has very reasonable rates. Second, sharply aim your books and then do direct mail.

A member of the audience added that book reviews are very important. Most magazines are looking for books, so send copies to key places like Byte. Send releases to smaller publications, and be sure to follow up with phone calls.

Computer shows are great, according to another SOG attendee who sold about 350 copies of his book at the LA Computer Show. He paid expenses the first two days, and then made money the next two days. He also sells books via his bulletin board.

The consensus was to do it any way you can. It's very hard to get mass distribution from the major publishers. In fact, they've hired a bunch of writers to knock off whatever they need, and they'll put it out for less than you're charging. You can approach them, but you may be better off on your own.

Distribution

An audience member told of an experience he had with a guy who was supposed to be distributing a book for him. "The only response we saw through him was while we were advertising in Publisher's Weekly at $2,000 per page. That was real expensive. After the ad stopped (it sold 1,500 copies the first two weeks) we didn't sell a single copy through the distributor. He was supposed to be national, but he didn't push it."

Keep as much control of your book as possible, because once it gets into someone else's hands there is no guarantee it will be distributed, or that anyone will

ever see it. Pushing your own book is very important. You need ads, reviews, and testimonials. You can get testimonials before the book even gets published. Then keep track of who purchases the book and use that information with the testimonials in your ads.

Putting Together A Book

Parachute Press prints an excellent book on self-publishing by Dan Pointer. He says that if you are compiling a lot of information from magazines, books, or whatever, you should lay out all your material on the floor, cut it up, and make a huge outline out of it. Then gather it up section by section, enter and edit the information, add your own comments, and you have a book. If you don't quote directly, then you don't have a problem with copyrights.

Often, you can work out some kind of deal with a small printer. Don't settle for royalties—you want a bigger chunk. So make a cooperative deal with the printer. But remember, distribution is the hardest thing to do, and can rarely be trusted to an outside party.

Information Sources

Finally, Hampton recommended everyone read "The Secret Money Machine" by Don Lancaster. Filter what he says down to what is applicable for you, and carry a small salt shaker—he's very opinionated.

Another good book is "New Start Publications." It's four years old, so it's a little dated, but still worth reading.

■ ■ ■

TECHNICAL TIPS

DDTing Inverse Video

I really like the new 3.3 version of WordStar that came with my new '84 Kaypro 2, but the inverse video menus don't work on my older, non-graphic Kaypro 4. Anyone who doesn't want the inverse video may be interested in how to turn it off using DDT. Here's how the session should go:

```
A>DDT WS.COM
DDT VERS 2.2
NEXT PC
4600 0100
-D267 26D
0267 06 1B 42 30 1B 42 31 ..B0.B1
-S267
0267 06 00
0268 1B .
-GO
WARM BOOT

A>SAVE 70 WS.COM
```

The sequence of bytes at address 0267 says to send 6 bytes, (Esc)B0 (Esc)B1, to the screen, turning the inverse video on in the graphics Kaypro. Changing the 06 to 00 tells it to send zero bytes. No more inverse video!

Michael Snyder
1010 Grayson
Berkeley CA 94710

Resetting BBI's Reset

I know it's only a minor nuisance, but some of us BBI owners have to to hit RESET every time we turn on our computers.

All it takes to correct this is to extend the automatic power-on reset. I increased C141 from 68uF to 150uF and the job was done. There is no reason why the same effect couldn't be achieved by increasing R48 instead, but the capacitor is easier to get to.

Hal Vikks
Address Withheld by Request

Ringing Your Bell And . . .

The following suggestions will get the "bell" working in dual density (mine only worked in single density), and will put the Olivetti PR2300 printer on line with improved print quality.

BELL: If you can't ring your bell (in dual density) try the bell circuit in Micro C, Issue #13, pg. 36 (it works fine in single density). Then add a jumper from TB1-8 or pin 10 (U111) to pin 2 (trig.) of the 555. The bell should now work in both single and double density.

Evidently dual density uses pin 10 (bit 4) instead of pin 9 (bit 5) of the system PIO, and opposite logic as well. The logic to ring the bell is:

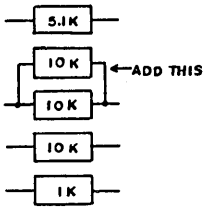
	Pin 9	Pin 10
Single Density	Lo to Hi	HiZ
Dual Density	HiZ	Hi to Lo

PRINTER: The Olivetti PR2300 has worked reliably for many months; the print quality is so-so. Ink ampules have been hard to find, but it's fast and very quiet.

The SWP printer driver (supplied with dual density) works well and the "Parallel Printer Cable" connections and jumpers from Appendix A may be used as shown with one modification: you must add a Timing Circuit to the STROBE IN line such as the one in Issue #18, pg. 57.

Use the jumper connections shown in Appendix A (not issue #18). Install the Timing Circuit between pin 34 (J5) and Pin 1 on the printer connector. Get +5V from pin 18 on the printer connector. You can wire wrap the Timing Circuit, and attach it to the printer cable near J5 with double-back tape.

PRINT QUALITY: The print can be made darker (it was too light) by adding a 10k parallel resistor to the circuit board (the resistors on the board are not numbered). In the upper left corner of the board:



You can now control the print intensity from very light to reasonably dark.

Lynn P. Smith
3051 Shirley Drive
Newbury Park CA 91320

Loading KSTROKES On A 4-84

A few months ago I purchased your Kaypro disk #24 primarily for the KSTROKES program, but was dismayed to find it wouldn't run on my Kaypro 4-84. The Micro C techies suggested the problem might be caused by a non-standard version of CP/M 2.2G I got from Kaypro.

The problem goes like this: originally, none of the KSTROKES programs would load. Whenever I tried to load one of the KSTRO*.COM programs, I received the message "Cannot load KSTROKES—reset system and try again." The problem is in the value given to the address FBASE in the conditional EQU on lines 18 to 20 in the KSTROKES.ASM file.

I now have a fix that might help others: on line 19 of the KSTROKES.ASM file, change 0E806H to 0E606H. Make sure you set KAYPRO2 EQU FALSE on line 7, KAY484 EQU TRUE on line 8, and KAYTEN EQU FALSE on line 9. Then assemble and load this source file to get KSTROKES.COM. Fixing the COM file with DDT is possible but tedious, as FBASE is used to define several other addresses in the source file, and finding everything with DDT takes a while.

Richard M. Warner
430 O'Keefe, Apt. 210
Palo Alto CA 94303-2140

Z-time For BBII

The Z-time calendar/clock from Kenmore Computer Technologies works fine on a Xerox 820, but when I hooked it up to my Big Board II, no go. I contacted Dave Schnabel at KCT, and he suggested the following fix:

1. Remove PAL U23 from its socket and bend out pin 8 which is the signal NOT BIORQ.

2. Connect a short piece of wire (about 4.5 inches) to pin 8 of PAL U23. Replace the PAL in its socket, or better yet, connect the piece of wire to the bent-out pin 8 of a second 20-pin socket. Replace the PAL in the new socket with the "flying lead" and plug the entire assembly into the existing socket for U23 on the BBII.

3. Connect the other end of the wire to the pad labeled "B" in the NOT IORQBRD line on the KCT board.

TECHNICAL TIPS

4. Jumpers are also required from pad A to the unmarked pad in the NOT IORQBRD line, and from pad C to the unmarked pad in the NOT RDBRD line, all on the KCT board.

This whole process takes about five minutes. A remarkable piece of silicon, the 58167 chip used in the KCT board makes for a very accurate calendar/clock which does not need to be reset after every boot.

Robert Bose
6821 Sally Lane
Edina MN 55435

Matter Of Grave Concern

I recently installed your updated version of ZCPR for Kaypros (disk K-22) on my older (pre '84) Kaypro 4. The submit file made the process a snap. The enhancements and additional built-in commands all worked perfectly, but I discovered one small compatibility problem. All of the messages sent to the console by ZCPR were followed by an accent grave which gave DIR displays a very messy appearance.

A look at the source code showed that ZCPR marks the end of its character strings by setting bit #7 of the last character HI. According to Bill Kennedy's modification comments, this led to some problems with Kaypro 10s (because of the graphic characters), so he altered the messages. Instead of adding 80H to the last character in the string (DB 'ALL,"?'+80H) each string is followed by an 80H (DB 'ALL?',80H). This results in a NULL (00H) being sent to the console as the last character in the string.

This NULL is sent even if you are using the PRO-8 monitor ROM. The only reason for this I can think of is that the PRO-8 system probably does not clear bit #7 before checking for a NULL character.

I assume (from other comments in the ZCPR source code) that a NULL sent to the console is represented with a space (20H) on most Kaypros. On my Kaypro 4, however, it is represented with an accent grave (60H). This is even documented in the user's guide (vers. 5, rev. 1, page 56).

I eliminated these extra NULLs (thus clearing up my messy displays) by simply changing the source code back to its original condition (adding 80H to the last character in message strings) and then reassembling it. You could also either change the string terminating character from 80H to OAOH (space + bit #7 set HI), or replace the character ROM to alleviate the problem (my character ROM is labeled 81-146).

Joe Fitzpatrick
257 W. Laurel Drive
Altadena CA 91001

Cheating Linefeed

Borland's TLIST.COM program (the one that comes with Turbo Pascal) will print without the extra linefeed per page if you fool it with a page-length directive on the top line of your program listing. Put (*.PL65 *) (one line less than the actual page length for 11" paper) on the first line, and TLIST behaves quite properly.

Joseph Mortensen
4214 Chelsea Court
Midland MI 48640

BBII ROM Monitor Fix

The Big Board II contains an error in the ROM monitor that will drive word processing users crazy. Likewise, any program that uses "Clear-To-End-Of-Line" or "Clear-To-End-Of-Screen" will experience the same trouble.

The two Clear functions erase the data, but they also can reverse the video attribute, creating stripes. When you're editing in WordStar, for instance, you can get reverse video patches all over the screen during editing.

The cause is the Vertical Sync interrupt routine which may occur during the clear process. When this happens it is likely that the incorrect attribute will be stored in the video memory. A change to the CLRLINE routine in the monitor will correct this. The change is shown in Figure 1.

In order for any change in the ROM to work, the code at memory locations 0009 hex through 000D hex must be NOPed out (replaced with 00H). This code is used to perform a CRC check on the ROM. When any changes are made, the CRC check will no longer work. If not removed, it is not possible to get the system to come up.

The fix changes the order in which the screen is cleared. In the original software, the cursor position is cleared first, then all others in order. In the new software the order of clearing is reversed, so the last byte to be cleared is cleared first, and the cursor position last.

The fix shown may be changed directly in ROM if so desired, but don't forget to NOP out the code between 0009h and 000Dh.

Also, don't forget that 2-byte values are entered in reverse order. For example, the code at 035E should be entered as follows:

```
035E  11
035F  FF0
0360  5F
```

Joseph L. Kappes
880 Reynard Avenue
Cincinnati OH 45231

■ ■ ■

Figure 1 - BBII ROM Monitor Fix

```
035E   11 5FFF     LD      DE,CHRMEM-1     ;COMPUTE LAST POSITION
0361   19          ADD     HL,DE           ;TO BE CLEARED
0362   09          ADD     HL,BC
0363   EB          EX      DE,HL           ;DE = CHAR MEMORY
0364   21 1000     LD      HL,4096         ;HL = ATTRIBUTE MEMORY
0367   19          ADD     HL,DE
0368   3A FFAB     LD      A,(BLANK)
036B   12          LD      (DE),A          ;STORE A BLANK IN LAST
036C   3A FFAC     LD      A,(ATTRIB)
036F   77          LD      (HL),A          ;STORE ATTRIB IN LAST
0370   0B          DEC     BC
0371   78          LD      A,B
0372   B1          OR      C               ;DEC BYTECOUNT
0373   C8          RET     Z               ;RETURN OF DONE
0374   D5          PUSH    DE              ;ELSE SAVE POINTERS
0375   C5          PUSH    BC
0376   54          LD      D,H
0377   5D          LD      E,L
0378   1B          DEC     DE
0379   ED B8       LDDR                    ;FILL LINE WITH ATTRIB
037B   C1          POP     BC              ;RESTORE POINTERS
037C   E1          POP     HL
037D   54          LD      D,H
037E   5D          LD      E,L
037F   1B          DEC     DE
0380   ED B8       LDDR                    ;FILL LINE WITH SPACES
0382   C9          RET
```

(continued from page 1)

sign and software) which are scheduled to run in the July and August issues of Byte.

The board, called the DSI 32 (DSI stands for Definicon Systems, Inc.), will be sold as a kit, which when assembled will plug into and run with any IBM PC, XT, or AT clone. The 32032 will run under MS-DOS so you can use WordStar to write your source code, then compile, assemble, and run your software under the 32032. All the data files will be completely PC compatible.

He will be selling two kits:

1. For $995 you can purchase a 6MHz version complete with floating point processor, 256K of RAM, and a selection of public domain software.
2. For $1495 you get the 10MHz kit with a floating point processor, 1 meg of RAM, and the public domain software.

Trevor says that the 6MHz version of the 32032 is really loafing at 8MHz (National rates them at 6MHz so it can sell the more expensive 10MHz parts), so we will have to look at a speedup mod for the slower boards. (Micro C continues as the magazine for speed freaks.)

The public domain software includes the Small C compiler and a Pascal P-code interpreter.

C, Fortran, and Pascal are the commercial compilers currently running on the board. FORTH and a BASIC interpreter should be available shortly.

Trevor has found that the 32032 running under MS-DOS is three to four times faster than the same processor running under UNIX. Nevertheless, he is planning to make UNIX available for his system.

Benchmark

Trevor did a floating point benchmark processing an array of 40,000 32-bit floating point numbers. He got the following times:

IBM XT	11.46 seconds
IBM AT	17.73 seconds
DEC 11-750	.83 seconds
DEC 11-780	.50 seconds
DSI 32	.97 seconds

All of the systems have hardware floating point, and the 68000 has times very similar to the AT. The AT is slower partially because its floating point processor is running at 4MHz instead of 4.77 (the 80287 is not as fast even at the same clock speed as the 8087). The time shown for the DSI 32 was while running at 10MHz. The 6MHz version would still be under 2 seconds.

The 32032 has a fully linear architecture (no segmented addressing, hurray), and it has a very powerful instruction set. For instance, the C compiler turns most C commands into one or two assembly language instructions. And, because the compilers are highly optimized for the instruction set and the instruction set has been optimized for compilers, the compiler output is as clean as hand written assembly language.

32032 At The SOG

Trevor and his group will be doing four sessions at the SOG.

1. 32-bit processors in general. What they can do, benchmarks, strengths, weaknesses.
2. 32032 assembly language. A detailed look at the instruction set and the architecture of the chip.
3. A designer's eye view of the DSI 32.
4. Dave Rand, a member of Trevor's group, will discuss the latest Z80 and other public domain software including NSWEEP (which he wrote). He will concentrate on the little known features in NSWEEP. (Come to the SOG—you might be very surprised what you'll take home with you.)

A Second Class Magazine

If you'll look closely at the bottom of the masthead on page 1 (you know, the who's currently who at Micro C), you'll see a long, convoluted message that says we have applied for a second class mailing permit (and you thought the bulk of this operation was first class).

Second class is delivered almost as fast as first class (the post office says it gets the same handling) at the price of bulk rate. We will be sending out the $16 subscriptions second class as soon as they approve our application (could be as long as six months). Of course, they could choose not to approve it (they are the government, after all), but according to the local staff there shouldn't be any problem.

The only difference between first class and second class, as far as you are concerned, is that first class is forwarded free. Second class costs you.

Anyway, for those of you lucky enough to live in the U.S. I'd suggest you renew (or subscribe) at the $16 rate. It'll save you money, and as Micro C continues to get bigger, it'll save us money, too.

Multi-year Subscriptions

Three gripes have turned up pretty consistently on the renewal forms.

1. The return envelope is too small. This is really a test of your mental acuity. (The trick is to fold the form in half, then in thirds—but don't tell anyone I told you.)
2. Micro C is getting too thick to three-hole punch.
3. We don't offer a multi-year subscription, so people have to keep filling out our silly renewal form (the one I read every comment on).
4. We should go monthly (but this is #4 and I promised you only three).

All right, already. Those of you in the U.S. (that's us) can have one year for $16, two years for $30, and three years for $42. That's not much of a break, but it will save you a lot of trauma trying to get that large form into that small envelope once a year.

Now it's possible that within the next three years we might go more than 6 times a year. If that happens we'll have to come up with an equitable way to raise our prices so everyone gets tréated (overcharged) equally.

Note to the weird person who folded his renewal form into a paper airplane: Straighten up and fly right, fella. (It's people like you who encourage weird editors like me.)

Selling Out

"Don't sell out to Ziff-Davis!" is a frequent comment on the renewal forms lately. Don't worry. A few days ago, three Micro C staffers (Dave Pogue, Gary Entsminger, and I) went to Eugene, Oregon, to visit a computer magazine. The magazine is called "Programmer's Journal," and it's aimed at the collegiate PC programmer.

Programmer's Journal (PJ) is two years old and is having major financial problems (it has no money and its current

owner probably won't continue to finance it after the next issue).

The story goes like this: at the end of its first year, PJ's founding editor sold the magazine to Avante Garde (an Apple software house). Avante Garde purchased the magazine because it thought it could immediately resell PJ to Hayden (you know, the book publisher).

Well, Hayden didn't buy. Reportedly, it wanted to broaden the mag rather than keep it PC specific, so when it encountered resistance from PJ's staff, Hayden backed out.

So Programmer's Journal has remained on the block. Its staff of three includes Greg Estes (who has stayed on part-time as editor), a circulation person, and an advertising sales person. It has about 1,700 subscribers, and the latest issue is 40 pages. Subscription rate is $24 per year (6 issues). Advertising is $700 per page. They have no money in the bank to fulfill current subscribers. And, of course, they are looking for a buyer.

Greg mentioned some ideas he had for expanding PJ. He is writing a formal journal—definitely not a light, bright piece. However, the marketing gal has been pushing him to aim at a less formal audience, since she's been pitching to a light, bright (PC World) audience.

Greg tried to get someone in Avante Garde to listen to his ideas about marketing (how to reach the real audience), but he struck out. So he has continued to produce a formal journal, while marketing has continued to pitch the novice audience. (Sounds like a large corporation, doesn't it?)

The Microsystems Experience

It turns out that Sol had the same experience with Ziff. He disagreed with the way Microsystems was being run. He wanted his magazine to remain a journal—plain and meaty. Ziff made it pretty.

I've talked to other ex-editors, and universally they've said that the only thing businesses understand is mass appeal (which means they homogenize anything controversial or unique into me-too bland). These editors have also said that anyone who sells a magazine should clear out completely—immediately.

"Don't make the mistake of keeping any part of the action," they tell me. "Clean out your desk and disappear. Don't even let them send you a copy. You won't be able to stand it."

Well, it'd be next to impossible to clean out my desk (Chris has given up even trying to organize it), I'm not moving out of Bend, and I couldn't stand the thought of someone homogenizing Micro C (much less beating it senseless).

Plus, I really enjoy doing Micro C. You should hear the ideas that go through this office in one week. It's an incredible experience—like rummaging through the universe's possibilities box (if not better).

Drives

Every day two or three people call to ask what kinds of drives I recommend. Well, I lean toward the cheapest drives (there is a lot of abject poverty in my recent past).

We have a lot of drives around Micro C and use most of them daily. We have a lot of Tandons (from the original Kaypros), but most of them now occupy shelf space. We have a lot of Japanese drives, and all of them have been good, some of them outstanding. (You know Shugarts are now made in Japan by Panasonic, and I understand the latest Tandons are made somewhere over there.)

Anyway, I'll rate our drives for you. Please understand that there is more fantasy in this than fact (look, I'm an editor and this is an editorial, so it's only reasonable). I've set an absolute range of 0 to 10 with only a few excursions.

The Ratings

TEAC—A bit noisy, but really dependable. I've run some of these for over a year and they haven't flinched. The data connectors on the 55 series are backwards from other drives, but that's usually just a nuisance. They rate an 8.5.

Mitsubishi—Generally quiet and quite solid. Rate a 9.

Shugart—Very quiet, flawless per-

(continued next page)

(continued from page 73)

formers, so far. I haven't run these very long yet, but Bruce loves his. If they stand up well (they should) they are a definite 10 in my book. I'll give them a 9 until they've had a few more months to prove themselves. Shugart has just gone out of business, but they are reportedly selling the quads for $70 each (in lots of 10 or more) and the double-sided double density for $75 each (they have fewer of the double-doubles). I'm working on a phone number, but you should find them if you talk to the marketing department at their main plant (wherever that is). Shugarts are rebranded Panasonics, so choose either brand.

TEC—We've had good luck with a couple of them, but other people haven't been as happy. Because of the gossip, give them a 7.

Tandon—The early full-heights were pretty good (look how many are still rasping around in old Kaypros). But they're noisy and eventually die. They get a 5. The later half-heights have had head problems (can't read or write) which limits their usability (a bit). They rate a -1. (The Japanese models may be better. If not, Big Blue is in for a surprise, as they just signed a contract for a bunch.)

Remex—I rate the ones we got about equal with my Tandon half-heights (-1).

Epson—Really solid little performers that are laying data on most of the Kaypro disks we ship (and have been for a year). The push-button sometimes doesn't eject the disk, but that's no biggie. Very quiet and absolutely reliable. Give them a 9.5.

Double-Sided Vrs. Single-Sided

If you have a Kaypro II and plan to replace your drives, I suggest you get double-sided 48 tpi drives—even if you aren't planning to upgrade to a 4 or 8. The single-sided drives have a felt pad which shapes the disk around the head. That pad needs to be replaced every so often, and I haven't found anyone who stocks them. On the double-sided drives, the two heads simply press the disk between them.

Theoretically, the single-sided drive should write data more solidly on the disk. However, I have seen worn and dirty pads that not only don't hold the disk properly, but that also do nasty things to its back side.

Don't worry about compatibility. Just plug a double-sided drive into a single-sided machine and it will run single-sided. Then if you want to upgrade to a Pro-4 or 8 . . .

Fairely Poor

MicroSphere didn't go to the West Coast Computer Faire—which is no big deal, I suppose, but I thought you'd like to know why.

MicroSphere's 6 by 6 foot booth was $504 this year, the same as last year (they paid in advance, so not going was no small decision), but everything else had changed.

The Faire had been moved from the Civic Center to the Moscone Center. MicroSphere's booth was moved four times (the latest relocation was behind a large post). This year's Faire was Saturday through Tuesday rather than Thursday through Sunday. The charge for power was $60 (it was free last year), the cost of a phone doubled, the table and chairs were extra (they were free last year), and every time something was moved to or from the booth it had to be handled by a union member. The major forums cost extra (they were free before), and the user groups weren't contacted until the last minute about holding meetings (those who were even contacted).

Micro C held user meetings the last three years, for instance, and we weren't contacted at all. When I called them the first of March to ask about the schedule of user group meetings and regular forums they said they hadn't finished working it out.

In fact, they didn't release a schedule until March 13, and then only to people who yelled and screamed. How are they supposed to attract attendees when they don't even know what they are offering?

I can't give you an actual figure on the comparative numbers of booths, but I do know that there was a lot of empty space in the Moscone Center. I also know that the prices will be higher next year and that the rules have changed.

Next year exhibitors will pay $15 per square foot rather than $12. Also, anyone who has a booth this year will not be able to have a 6 by 6 foot booth next year. The minimum size for old-timers will be 8 by 10 (for $1200). This year the 6 by 6 booths were about $500.

I heard several reports of exhibitors going into the office to register for next year, only to turn tail when they heard the new rules.

Of course, these are the little guys, and who needs them? Right?

dBASEd Findings

I received a number of cards, letters, and calls from helpful souls who have themselves faced the dragon (dragon Tate) and have survived. One suggested I make sure my dBASE had been installed on the copy of CP/M I was running (you know, run the install program). Others commented on my strange version numbers. For the record, I am running versions 2.4 and 2.3b. The 1.4 and 1.37 listed in issue #22 were errors (my own).

Well, I have been able to make the system work properly (no more dropping back into CP/M) and in the process discovered three things.

1. Although you are supposed to have over 1,000 bytes space for variables, mine dies when the variables take over 830 bytes in version 2.3b. It's less than that for 2.4.

2. When I run a program that uses most of the variable space, "ESC" out of the program, and then re-start with a DO command the program will bomb. If, instead, I "QUIT" dBASE after the "ESC," re-start dBASE, and then re-start the application, the program will run.

3. We can make new entries in a large indexed file much more quickly using version 2.3b than using 2.4.

I got a call from a dealer who said that version 2.43 (the latest) had been recalled by Ashton-Tate and that there was no word on when it would be re-released (probably as version 2.5). Ashton-Tate had promised me the 2.43 upgrade (I've bought three copies already), but maybe I'm glad they haven't sent it. On the other hand, if I had zillions of users and could charge $200 for copies of bug fixes I'd be tempted to come out with a new fix every few months.

David J. Thompson
Editor & Publisher

WANT ADS

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.

Disk Service Manual, disk drive tutorial, printer & plotter manual, copier manual, computer phreaking!! Much more! FREE information. Consumertronics Co., Attn: Computers, 2011 Crescent, Alamogordo, NM 88310.

Keyboards for Computer Builders – 83 keys, full ASCII; upper/lower case, all control characters, numeric pad, CAPS-LOCK, REPEAT, self-test! Brand new, hundreds sold already to builders of Apples, Big Boards, Xerox 820s. Parallel output, positive TTL logic, strobe. Uses only 106mA of +5 volts. Custom case available. 90 day warranty unmodified. Keyboard $35. Documentation (21 pgs.)/cable package $5. Spare custom CPU/ROM $4. UPS included. Call/SASE for detailed spec sheet. Electrovalue Industrial Inc., Box 376-MC, Morris Plains, NJ 07950. (201) 267-1117.

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 printer capable of 132 columns for maximum usage. Terminal installation program module included. Supplied on 8" SSSD, 5.25" Kaypro and many others (please write). Special introductory offer by ABLE DATA SOFTWARE, INC., PO Box 86923, Station C, North Vancouver, BC V7L 4P6. Only USA $19.95 postpaid check or money order.

Public Domain UG Software Rental: CP/M UG Vol 1-92 on 46 8" Flippies, $45, SIG/M UG Vol 1-209 on 100 8" Flippies, $99.50, PICONET Vol 1-34 on 17 8" Flippies $25, Pascal-Z UG Vol 1-25 13 8" Flippies $25, UG Games 20 Vols of the best $25, UG Business 20 Vols of the best $25, UG Utilities 10 Vols of the best $25. Rental is for 7 days after receipt with 3 more days grace for return. Credit cards accepted (preferred). 5" disk formats, 170 available. Downloading-disk format conversions. Call. User Group Software Automatic Update Service, $7.50 per 2 volume set PP.619-727-1015 24 hrs. 619-941-0925 info. 9-5. P.J.'s National Public Domain Software Center, 1533 Avohill, Vista, CA 92083.

8" Drive Cleaning Kits – 12 cleaning disks and carrier jacket made by Datalife, regular price, $29.95. Close out price just $6.00 each plus $2.00 postage. Limit 2. P.D. Software, 1533 Avohill, Vista, CA 92083.

Motorola 68000 Versabus Systems. Include CPU, I/O, 512Kb RAM, disk controller, 13 Mag hard disk—$3,000. Hugh Shane, 7 Green Meadow Road, Pleasantville, NY 10570. (914) 769-4299.

Lomas Data Products Thunder-186 S-100 Board for Sale: Includes 8MHz 80186, 256Kb memory with parity, floppy disk controller, I/O ports= two serial, one parallel, concurrent CP/M-86 and MS-DOS, all manuals. Brand new. $750 or best offer. Dan Blumenfeld, 3900 Chestnut Street #803, Philadelphia, PA 19104. (215) 898-1956.

New 5.25" Half Height Disk Drives. TEC-FB 501 SSDD with documentation $89.00 each—2 Drives in cabinet with power supply and connectors $225.00. Shugart SA 800-2 Disk Drives reconditioned 60 day warranty with documentation $79.00. LDL Electronics, 1-305-747-7384.

BBI, 5MHz, All Options, 2 Shugart Drives, Power 1 switching P/S, Odd Ball Electronics cabinet with fan, keyboard, user disks, Pascal, Forth. Very reliable system—$500. Okidata 82A printer—$310. Kevin Tyrrell, 1221 Colorado SE, Grand Rapids, MI 49506. (616) 241-1902.

For Sale: Slicer computer: Assembled complete (less 80186 & RAMS). ROMS, source disk and documentation. $450 or best offer. BBI computer system: Assembled 2.5 MHz, system and disk power supplies, enclosure, fan, power line filter, 2 Shugart 8 inch drives, disk cables, source disk, Big Board CP/M, JRT Pascal, user disks Nos. 1, 2, 7, 8, 20, B10, B15, B17, B18, 9, and blank diskettes. All documentation. System needs slight work. $650 or best offer plus shipping. Xerox Computer: Assembled and complete, untested. $300 or best offer. Slicer and Xerox systems prices include shipping. Miscellaneous chips and boards for sale, write for list. Warren E. Greenberg, 145 Cottage Road, West Roxbury, MA 02132.

Spring Sale! Sale/Retail: IBM/XT 10mb controller 590/250, Otrona Attache 1800/2995, Quantex 150 cps matrix printer 700/1195, Qume Sprint 9-45 cps daisywheel 2495/1265, Shugart 712 10mb 1/2 height 5" 465/695, Xebec S1410 250/495, BBII A & T 450/995, Ferguson cabinet 5", 8", & BB 100/400, Morrow MD-2 650/1299, Smith-Corona TP-1 daisywheel 250/595. Polygon Industries, P. O. Box 24615, New Orleans, LA 70184. (504) 282-5372.

Teeny-Weeny Basic—A 1K BASIC interpreter/editor. How much power can fit into 1K? A LOT! Full integer arithmetic functions with 26 variables, random function, single-dimension array, parenthesis nesting, string variable I/O, abbreviated commands, multiple statement lines, error handling. PRINT, INPUT, IF, GOTO, LET, RUN, LIST, SYSTEM. External LOAD and SAVE. Full documentation. Sample programs show TWB's power. Source available. Convinced? Try it! Just $17.50 postpaid (Texas add 5.25%). Specify 8" SSSD or Kaypro SSDD. Glen McEowen, 3801 Glenmont Dr., Fort Worth, TX 76133.

Compilers—Used MicroSoft Basic Compilers MS-DOS $199. CP/M $189. Mike Loth, Box 847, Steamboat, CO 80477. (303) 879-2056.

WD2797 Floppy Controllers, 12.00 U.S. (Surplus, not used). BBI modification doc included. Complete Dynadisk kit (sockets soldered, never used) $40.00 U.S. M. Voakes, 555 Brookhaven Cres., Waterloo, Ontario N2L-4R6.

Mince-Scribble no longer distributed commercially. Want legitimate copy with source and documentation for Kaypro/CPM if the price is right. Also want BDS-C. Write: Mike Perry, 6035 40th NE, Seattle, WA 98115.

300/1200 auto-dial auto-answer smartmodem. Hayes compatible. $199. KEYTRONIC keyboards. New. Parallel ASCII. $10 each. Two for $18. B.W. Systems, Box 9791, Austin, TX 78766. (512) 255-8350.

Yet another computer "garage sale": Qume 5.25" DSDD half-height floppies, $70 each. Dysan alignment disks, new, $25. Tandon 602 5M hard drive, ST-506 equivalent, $150. The stuff works; no "as-is" surprise packages. Monitors, chips, etc., cheap. Please request list. Noor Singh, P. O. Box 807, Santa Cruz, NM 87567. (505) 753-2211, eves.

5¼" hard disk controller, DTC-510A, BIOS source on 8" floppy, $125. Qume 5¼" DSDD half height floppy, 2 for $125. Many S-100 boards, please request listing of excess equipment, Noor Singh, P.O. Box 807, Santa Cruz, NM 87567. (505) 753-2211.

■■■

THE CULTURE CORNER

Submission Requirements

In the last few months we've been reorganizing the submissions section of Micro C. We've hired a couple of ex-computer salesmen to categorize submissions before the editorial staff sees them. (This is a make-work project, folks — the used car lot wouldn't take them back.) Since these guys don't have the slightest idea what the articles are about, we ask you to print the proper category at the top of your submission.

Categories

Every article must fit into one or more (or none) of the following categories. (There can be exceptions.)

☐ Technical Fiction

Nearly all of the material published on computers fits under this category, and Micro C is no exception. Material which seemed reasonable when we thought it up is often wrong by the time we hit (liter) the streets. The problem is that designers are no longer creating systems to match our descriptions. This is why there are so many undocumented features and so many unfeatured documents.

☐ Humor

Humor has no place in a formal technical magazine like Micro Cornucopia. It may show up because of an editorial oversight (by our very active editorial oversight committee), but all of our humor is intentionally unintentional.

☐ Reviews (Rave)

See Technical Fiction.

☐ Reviews (Unbiased)

See Reviews (Rave).

☐ Inscrutable Tomes

You can always spot someone who has just finished a long, arduous, exhausting, debilitating, confusing, boring stint in academia. He's the guy who appends a 20K bibliography to his techtip.

He has two measures for his work — obscurity and length. He has spent three or four years of his life learning how to turn a simple idea (his research project) into a book-length epistle that will be read by two people: his advisor and his typist. Neither will understand it.

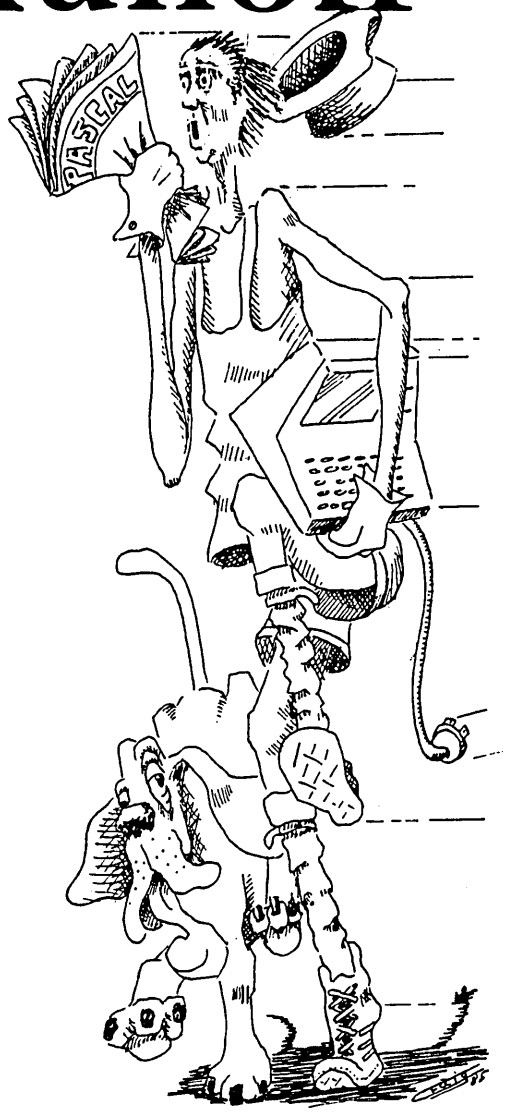
Academics have written manuals on such graduate level topics as "Distinguishing CP/M's Ed from Mister Ed" (it's a horse of a different color), and "Communicating with Surley Waiters in Assembly Language" (a hex on your baud, bud).

☐ Practical Topics

These really don't fit in a publication such as ours. Send these to Digital Navel Review (if you can stomach it) or Micro Fillings Amalgamated (a real mouthful).

Meanwhile, keep those cards and letters and articles coming, folks. If you dredge up something really good for this column, PLEASE send it in. After seeing this, the entire staff is signing up for a refresher at the funny farm.

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.

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 _____

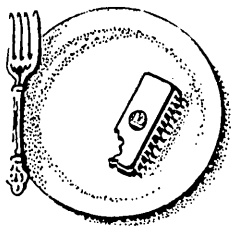
MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

Order No. (503) 382-5060

Tidbits

By Gary Entsminger

Future Tense Editor



Syntel
530 Pylon Dr.
Raleigh NC 27606
919-828-4626

Bulletin Board

Bruce has Micro C's first public bulletin board up and running. It's written in Turbo Pascal, and the source is available on Kaypro Disk K31. Micro C is now online 24 hours a day at 300 or 1200 baud. Call us at 503-382-7643.

Program listings referred to in Micro C and new programs will be available on the bulletin board. To find out what's new, exit to CP/M with the C command (from the BB), type

CD NEW

and then D

for a directory. You'll be able to download what you need. Make sure you're using 8 bits per character.

Several new programs are on board already—all in source.

SHOW.MAC, written in Z80 assembler, is a TYPE lookalike that scrolls forward and backward. If you'd like to expand it into an editor, give it a try.

LINK.MAC, also written in Z80, links .REL files. It's very primitive—won't handle embedded DS statements—but it should be fun to expand.

PRINT.MAC, also Z80, loads as much of a file into memory as it can, then prints it (allowing your disk drives to shut off). It's only 2K.

SHIP.PAS, written in Turbo Pascal, is a simple communications program for sending and receiving files through serial ports. For more info, see Laine's Slicer column this issue.

There's lots more, so check it out. We'd like to hear from you.

DSD—Full Screen CP/M-80 Debugger

Soft Advances has lowered the price on their sophisticated CP/M debugger from $195 to $125.

DSD maintains a full screen of six independent windows: displaying instructions, registers, stack, memory, command line, and echo line.

For more info, contact:

John Otken
Soft Advances
P.O. Box 49473
Austin, TX 78765
512-478-4763

Submissions—Writing For Micro C

We're hearing from lots of you—so many, in fact, that a few submission guidelines will improve our information exchange.

First—if you're sending an article, please submit it on disk as well as on paper. As usual, we'll acknowledge your submission with a free disk of software (your choice). This will let you know your hard work has reached us safely. If you forget to tell us which disk you want, we'll send you a coupon which you can use when you feel like it. Also, put your name, address, and phone number on the disk and on the printout so we can get in touch with you if we need to.

About content—overexplain everything (let me repeat that: overexplain everything), and show us every step. We want to make sure we understand what you're doing. It's a lot easier for us to cut than to add. We especially like illustrations and schematics, but make sure references to the art work are clearly stated in the article.

About subject—we'll look at anything (we might laugh under our breaths, especially if it's from Laine, but certainly not out loud).

Surplus

Jim Ferguson (you know, the BBI designer) has several hundred Otrona Attache 8086 16-bit add-on processor boards designed to work with their Z80A system. With 256K already soldered onto the board, they look like bargains at only about $45 each. Call Jim for details.

In order to use this board you'll have to either find or write the software to enable it to talk to the main board, and vice versa. (If you get it talking to the BBI, Kaypro, or Xerox write in and tell us how you did it!)

Ferguson Engineering
P.O. Box 300085
Arlington TX 76010
817-640-0207

Also, Syntel has 300/1200 baud auto answer modems (not auto dial) for $129. They're going like hotcakes (they're priced like them, too), but you still might be able to get one.

An 11-Pound Baby Kaypro

Just when you thought David Kay was misleading Kaypro into oblivion (1985 first quarter earnings at Kaypro were $72,872 down from the 1984 first quarter $2.8 million), he responds with masterful touches. In March, he introduced the AT clone, and now he's out with a portable PC clone.

"It has everything an IBM PC has except a standard CRT," he says.

It's the Kaypro 2000, and it has 256K RAM (expandable to 640K), an 80 character, 25 line LCD screen, 3 1/2 inch disk drive (with 720K capacity), and a rechargeable battery—all for $1995.

The microprocessor is an 8088, and standard software includes MS-DOS.

Bugg Music

Richard Bugg's Band (Richard wrote "Kaypro Composite Video Output" in #22), Cosmic Debris, has released its second album, While You're Asleep. It's a warm electronic album. The group's first (3.7K) was a hit in Oklahoma City and in parts of the Midwest. Can't wait to hear it.

Ampro At SOG

Dave Pogue and I spoke with Rick Lehrbaum, founder and vice president of engineering at Ampro, this week, and I'm delighted to report that they're planning two workshops for SOG IV: one on the SCSI multi-master bus expansion for any Z80, and one on their new Little Board/186.

The new Little Board has the SCSI/Plus Multi-master Bus and an 8MHz 80186 microprocessor, and is the same size as the original Little Board. Even the connector locations, pinouts, and I/O signals are essentially identical to those of the other Ampro board.

This looks like a real hummer in a small package. Look out, IBM.

Rick expects to be in full production by mid-summer, with a 128K board selling for $549, and a 512K board selling for

(continued on page 81)

(continued from page 79)

$749. They'll have boards at SOG IV, so you'll be able to put together a super little system (Z80 or otherwise) for a reasonable price.

He's sending one to Micro C, so we'll have our little system together by SOG.

C Language Conference

OK, East Coast C'ers, if Bend's too far for you, or if you're just dying for a follow up to SOG IV, Computer Language is sponsoring a C seminar/workshop September 16-18 in Cambridge, Mass.

C has just been standardized by a special committee of the American National Standards Institute (ANSI), and committee chairman Jim Brodie will be speaking at the seminar on the state of C.

P.J. Plauger, co-author of "Elements of Programming Style," heads an early list of speakers. Attendance fee for the seminar is $695 ($595 until June 30). (On the other hand, if you sent your $600 to Ampro . . .) For more information contact:

Computer Language
131 Townsend St.
San Francisco CA 94107

Proportional Spacing For WordStar

If you're using WordStar 3.3 and need proportional spacing, Chaucer Software's little program might be your ticket.

It requires a letter-quality printer, and runs on all Kaypros including the 16. It's $19.95 from:

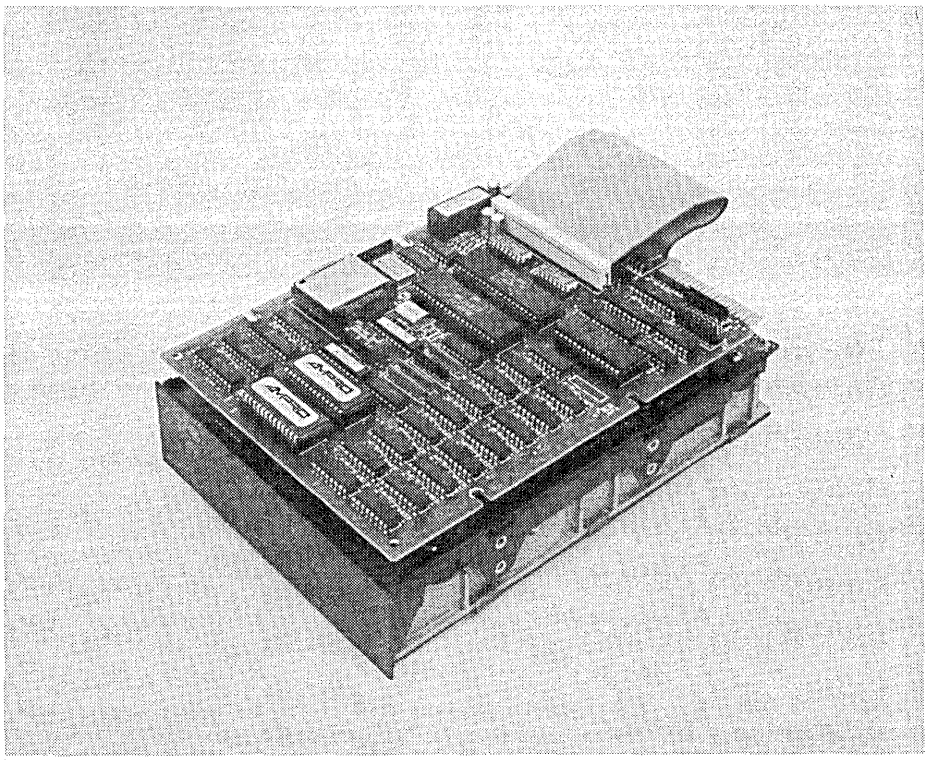
Chaucer Software
P.O. Box 2308
Princeton NJ 08540
609-734-9016

No Z800, But . . .

We've just heard about a new processor from Hitachi that is upwardly compatible with the Z80. (Thanks to Allan Emord of Albuquerque, NM.)

This microprocessor, dubbed the 64180, is a high integration VLSI containing a 64K Z80 CPU, serial port, two 16-bit timers, onboard MMU, and two DMA channels that can directly address the 512K of physical memory (bypassing the MMU).

Due on the market soon, it should sell for around $20. Who knows—this could be the beginning of a Z80 revival (and the demise of $20 bills from general circulation).



Little Board ™/186 shown with a 10 MB Xebec OWL

Canon Drives On The Kaypro

Note: To use Canon drives on a Kaypro you need to set the DIP switches on the units as follows:

SW1 is the drive select.
SW2—set 1,5,6 on.
SW3—set 2,4,5 on.

And that's about all the tidbits fit to print in this issue. See you at the SOG.

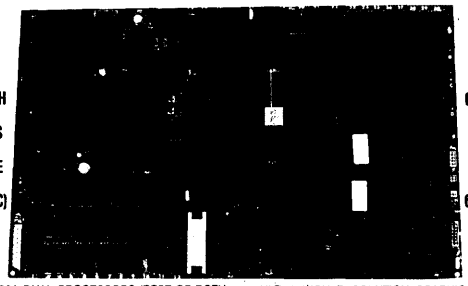
■ ■ ■

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

THE CYPHER™

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

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



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

- 68000 & Z80A DUAL PROCESSORS (BEST OF BOTH WORLDS! OPTIONAL Z80 H.)
- 256K to 1 MEGABYTE MEMORY. (4164 OR 41256 DRAM).
- DOUBLE DENSITY FLOPPY DISK CONTROLLER (8", 5¼" OR 3½" WD 2793).
- DMA CONTROLLER FOR FAST IMAGE TRANSFERS TO/FROM VIDEO MEMORY. (INT 8237).
- 2 RS232 SERIAL PORTS (ZSIO).
- 24 BIT ADDRESS MANAGEMENT FOR Z80.
- 4 LAYER P.C.B. (9¼" x 14¾").
- RUNS CP/M-80 2.2, CP/M-80 3.0, CP/M-68K, CYPHER DOS, RAM DISK, 68000 BASIC IN ROM, NEC 7220 TERMINAL EMULATION.
- ULTRA HIGH RESOLUTION GRAPHICS, 128K, PROGRAMMABLE UP TO 1024 x 1024 RESOLUTION (NEC 7220, GREAT FOR CAD SYSTEMS!).
- REAL TIME CLOCK (MULTITASKING CAPABILITY!).
- TWO CHANNELS OF D/A AND A/D, 12 BIT RESOLUTION (MUSIC! ROBOTICS! LAB WORK!).
- 16K TO 64K MONITOR EPROM.
- 4K TO 64K STATIC RAM.
- PROGRAMMABLE BAUD RATE GENERATOR.
- PARALLEL ASCII KEYBOARD INPUT.
- FULL 68000 EXPANSION BUS (60 PIN HEADER, BUFFERED).

LOWER PRICES! NOW 1 MEGABYTE CYPHER AVAILABLE.

- MANUAL **$20.00**
- BARE BOARD, EPROMS, Z80 BIOS, 68000 BIOS, Z80 MONITOR, 68000 MONITOR AND UTILITIES **$399.95**
- COMPLETE "CYPHER" WITH **256 K DRAM**, 128K VIDEO DRAM, NEC 7220, REAL TIME CLOCK, A/D D/A DISC CONTROLLER, SERIAL I/O, ASSEMBLED AND TESTED **$1249.95**
  PARTS KIT **$1149.95**
  EXPANDED MEMORY OPTIONS – INQUIRE
- COMPLETE "CYPHER" WITH **1 MEGABYTE DRAM**, 128K VIDEO DRAM, NEC 7220, REAL TIME CLOCK, A/D D/A DISC CONTROLLER, SERIAL I/O, ASSEMBLED AND TESTED **$1399.95**
  PARTS KIT **$1299.95**
- KEYBOARD **$94.95**
- SWITCHING POWER SUPPLY **$154.95**
- CASE **$94.95**
- HARD DISC INTERFACE PLUG-IN CARD **$150.00**

VISA

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

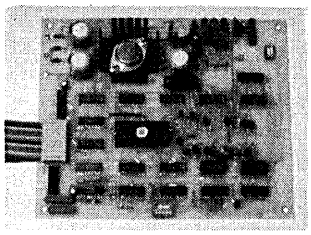
PRICES SUBJECT TO CHANGE WITHOUT NOTICE
CP/M IS A TRADEMARK OF DIGITAL RESEARCH.

MOTOROLA INTEL

MOTEL COMPUTERS LIMITED
174 BETTY ANN DRIVE, WILLOWDALE,
TORONTO, ONTARIO, CANADA M2N 1X6
(416) 221-2340

CP/M EPROM PROGRAMMING SYSTEM

2708
2758
2716
2732
2764



2516
2732A
27128
27256
27CXX

- STAND ALONE BOARD – ELECTRONIC SWITCHING OF EPROM TYPES
- USES 24 VOLT XFMR FOR POWER – ALL SUPPLIES/TIMING ON BOARD
- DESIGNED WITH EASY TO GET PARTS – COMPREHENSIVE 30 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 WIRES AND A GROUND – NO HANDSHAKE LINES REQUIRED.
2 – CENTRONICS PRINTER PORT – 8 OUTPUT DATA BITS AND ONE INPUT DATA BIT (BUSY LINE) – 9 WIRES AND A GROUND.

*** * CONTROL PROGRAM COMMANDS * ***

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

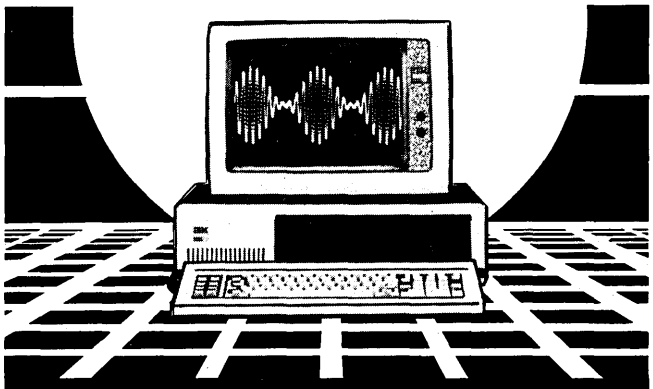
BARE P.C. BOARD WITH COMPLETE DOCUMENTATION AND SOFTWARE ON 8" SINGLE DENSITY DISKETTE **$69**

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

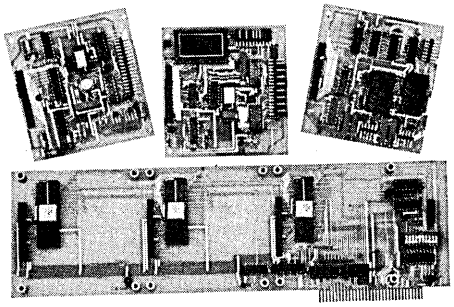
TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL
ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752–7218

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

WE BRING ENGINEERS, SCIENTISTS AND THE IBM-PC TOGETHER



Modular Data Acquisition System



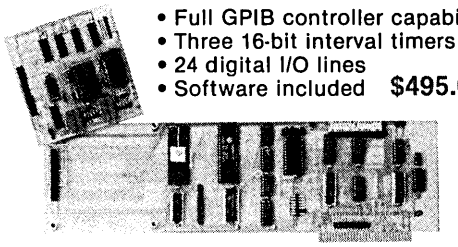
- 12-bit resolution
- 24-KHz throughput rate
- 16 channel SE, 8 DI
- LABSTAR Software

$690.00

- 8-bit resolution
- 30 KHz throughput rate
- 16 channel SE, 8 DI
- LABSTAR Software

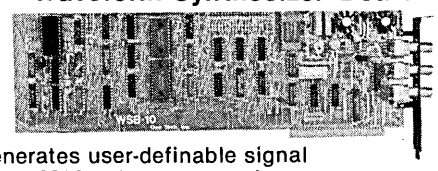
$490.00

IEEE-488 GPIB Multifunction Board



- Full GPIB controller capability
- Three 16-bit interval timers
- 24 digital I/O lines
- Software included **$495.00**

Waveform Synthesizer Board



- Generates user-definable signal
- Up to 2000 points per envelop
- 200ns maximum output rate per point **$795.00**

QUA TECH, INC.

478 E. Exchange St.
Akron, OH 44304
(216) 434-3154

Benchmarking The 68000 and 80X86

By Luis Basto

12707 Poquoson Dr.
Austin TX 78759

What's the fastest 16-bit chip around? It depends on whom you're listening to.

Intel has published reports comparing the speeds of its 80*86 family and Motorola's 68000. Their reports claim the iAPX286 is three to six times faster than the 8086 and three times faster than the 68000. Motorola decided to study Intel's benchmark results, and they found some inconsistencies in Intel's comparisons. Here's food for thought:

1. Intel used the fastest iAPX286 they make (8MHz), but not the 12.5MHz Motorola 68000.

2. Intel used a record area of 64K for the linked list benchmark (which is the maximum memory all 80*86 chips can address without segment switching) and used a 16 Megabyte area for the 68000.

3. None of Intel's benchmarks handled the case of crossing a segment boundary. Obviously, many applications require more than 64K RAM. Crossing a segment boundary means more overhead (slower operation) for Intel's parts.

Figure 1 - Intel Vrs. Motorola Benchmarks

		MC68000 12.5	MC68008 10	8086 10	iAPX186 8	iAPX286 8	MHz
A	I/O Interrupts	25.6	57.6	43.2	50.0	96.8	us
B	I/O Processing	259.2	573.6	396.0	446.2	357.3	us
E	String search	127.0	372.6	201.0	249.8	128.4	us
F	Bit Manipulation	55.4	116.1	127.1	158.2	97.9	us
H	Linked List	116.8	281.6	269.0	259.2	199.8	us
I	Quicksort	13.9	31.0	38.3	45.2	36.1	ms
K	Bit Matrix	289.1	555.6	938.5	724.7	508.8	us

Intel Vrs. EDN Benchmarks

EDN published a list of benchmarks which the major chip manufacturers can use to compare parts. Figure 1 gives the results used in the Motorola report, using the fast chips.

From these results one concludes that the 286 can't be three to six times faster than the 8086. In fact, the 8086 beats the iAPX286 in the I/O Interrupts benchmark and finishes close behind in three others. In all cases, the 12.5MHz 68000 was faster than the iAPX286.

It's worth noting that the iAPX186 is slower than the 8086 in five of the seven benchmarks. Even if you extrapolate the iAPX186 to 10MHz, it's not much better than the older 8086. (What about the 8088? It's in their benchmark report for the Z80.)

EDN asked Intel to send in the code for their benchmarks, but Intel refused. Motorola interpreted Intel's refusal to mean that the code for the iAPX286 was so long and clumsy Intel would be embarrassed to see it in print.

Why The Discrepancy?

One explanation might be the segmented architecture of the 80*86 family. The maximum memory address in that case is 64K. Since the iAPX286 has an onboard MMU (memory management unit), the MMU takes over and updates the segment registers when the software addresses an out-of-boundary location. This creates a significant overhead when compilers operate on large data areas.

The 68000 can address anywhere in its 16 Megabyte address space without any overhead. Even when an external MMU was added to the system, the 68000 ran faster than the 80286 in five ot the seven benchmarks.

Benchmarks are, well, they're benchmarks, and obviously they're only one consideration for designers. But they're food for thought.

Editor's note: Of course, there's more to a microprocessor's success than benchmarks. The Intel-Motorola battle illustrates how marketing moxy can outweigh performance in the battle for industry's pocketbooks.

In 1981, when the Motorola 68000 was gaining momentum, Intel president Andy Grove called in Regis McKenna, a public relations hotshot from Palo Alto, California.

Grove, McKenna, and six Intel managers met to develop a new marketing strategy for Intel. Their project was codenamed CRUSH. Very simply, its intention was to stop the movement of designers from the Intel chips to the newer 68000 series.

After surveying the market, they concluded that if customers compared the 8086 to the 68000, chip to chip, "Intel would have trouble." The 68000 was becoming more and more popular among software-oriented companies, while the 8086 was holding its own among hardware-oriented companies. (See "The Last Page" this issue for details.)

The CRUSH strategy was to play on customers' fears. They wanted people to worry about the consequences of committing themselves to Motorola. After all, the 68000 had very little software, no peripheral chips, and no development system. And Motorola hadn't clearly defined its future. Would customers get stuck with an orphan if they went 68000?

During the next quarter, Intel gave 50 half-day seminars to potential customers, and thereby won the positioning battle. Motorola is only now beginning to catch up in the home computer market, with new machines coming from Amiga, Atari, and Apple.

■ ■ ■

MSX In The USA

By Trey Weaver

8428 Lacebark Lane
Liverpool NY 13090

MSX is a new operating standard for home computers in the U.S. that specifies everything down to the physical addressing of the I/O devices, integrated circuits, and the size of the cartridge slot.

Hardware

CPU—Z80 Running at 3.58MHz.

Memory—ROM 32K (MSX system software), RAM 64K.

Video—Yamaha 9938 video display processor.

Tape—Cassette FSK format (1200/2400) baud.

Sound—GI AY-3-8910 (3 voices and 8 octaves).

Joystick—Atari type.

Expansion Slots—Software cartridge, 2 min without disk, 1 with internal disk.

Printer—8 bit parallel.

Clock—CMOS battery backed up (optional).

Disk—8", 5.25", or 3.5" MS-DOS format (optional).

RS232—(optional).

A typical MSX computer will sell for about $200, including graphics and a built-in printer port. By summer just about every computer manufacturer in Japan will be selling MSXs in the U.S.

Memory

All MSX computers have 32K ROM with BASIC (designed to follow the GW-BASIC standard). The ROM also includes various system calls.

MSX computers use a memory bank select structure (slots). There are 64 of these banks; each is 16K bytes long. Any four of these banks can be mapped into the Z80 operating area at a time, and can be mapped back out by software. This allows a full 1Mbyte expansion and unbelievable flexibility!

Video

Get a load of these display modes.

Text I—40 characters per line, 24 lines per screen.

Text II—80 characters per line, 24 or 26 lines per screen, 4 colors out of the 512 colors.

Multi Color—64 x 48 blocks, 4 x 4 blocks, 16 colors.

Graphic I—256 x 192 blocks, 8 x 8 blocks, 16 colors, 32 sprites, 4 sprites per horizontal line, 256 patterns.

Graphic II—256 x 192 blocks, 8 x 8 blocks, 16 colors, 32 sprites, 4 sprites per horizontal line, 768 patterns.

Graphic III—Same as graphic II but with 8 sprites/line.

Graphic IV—Bit mapped, 256 x 212 pixels, 16 colors out of 512 colors, 8 sprites per line.

Graphic V—Bit mapped, 512 x 212 pixels, 4 colors out of 512 colors, 8 sprites/line.

Graphic VI—Bit mapped, 512 x 212 pixels, 16 colors out of 512 colors, 8 sprites per line.

Graphic VII—Bit mapped, 256 x 212 pixels, 256 colors, 8 sprites per line.

Graphics modes Text I, II, and III require 16K of memory; modes IV and V require 32K; and VI and VII require 128K. Video memory is addressed by the video display processor and is not in the CPU addressing space.

Most of the U.S. versions should have a 32K video memory, with the option of expanding it to 128K.

MSX-DOS

MSX-DOS is the operating system for disk based systems. Microsoft has been tight-lipped about it, but we do know a little.

It'll be menu or icon based, but from a program's perspective will look like CP/M with BDOS calls. So an MSX computer will run lots and lots of programs—Turbo Pascal, for example. The format for the disk will be MS-DOS 1.0 compatible, with 360K bytes per double-sided disk.

In Addition

The keyboard has upper and lower case, 10 function keys, a graphics select key, cursor controls, delete, insert, and home keys.

Tablets, mice, RS-232 cartridges, games (some by Activision), expansion boxes, and music synthesizers are a few of the goodies already available for the MSX, and some companies have started "MSX Engine" ICs that incorporate Z80, interrupt control keyboard scan, bank select control, RAM interface, I/O interface, printer interface, RS-232 protocol, real-time clock, and sound generation into one integrated circuit.

I think this is one of the best values in a home computer. Where else can you get such good graphics, 80 columns, CP/M compatibility, and a printer port for $200?

Editor's note: I attended a session on MSX at the Computer Faire. Both speakers had been working closely with the Japanese computer manufacturers. Plus, I own one of the first MSX systems brought into the U.S., a Yamaha unit that interfaces with music synthesizers.

So I'd like to add a few thoughts to this article:

1. MSX has been the standard in Japan for several years, but the units that are popular there are little more than TRS-80 model 1s.

2. The speakers assured me that the systems coming into the U.S. this summer will be much fancier than their Japanese counterparts. They will have four or more MSX compatible slots rather than one. They will have parallel and serial ports built in. They will support 80 by 24 (or 25) video as standard (that is a recent decision). They will run all standard CP/M software, but the disk format will be the same as MS-DOS 1.0.

3. MSX is a standardized system, so all manufacturers have to agree when changes are made or features are added. The imported systems should be able to run all software (ROM, cassette, or disk based) and all accessories interchangeably. This interchangeability is one of the reasons it has been so difficult for MSX to support new processors. MSX will continue support of the Z80, but manufacturers are also looking at the Intel and Motorola worlds. It's likely that a standard plug-in board could contain a very powerful co-processor, and yet the board could be plugged into any MSX system.

■ ■ ■

SOG IV The Technical Conference For The Whole Family

It's registration time again and this year's Semi-Official Get-together will be grander than ever. Like last year, there will be two days of top notch technical presentations sandwiched between two days of fun. This year, however, there will be more things to build, more sessions to attend (not that the schedule wasn't full last year) and special events Friday and Saturday for non-technical family members.

SOG IV is being held at Central Oregon Community College in Bend, Oregon. The dates are Thursday, July 25 through Sunday, July 28.

Free Conferences

Friday and Saturday: Two full days of free technical conferences, demonstrations, workshops, and forums (plus the new-product-displays and swap meet). You'll have a great opportunity to exchange ideas with the Micro C editorial and technical staff, columnists, as well as other leaders in the micro industry. You'll get to talk with the main folks from Slicer, Ampro, Integrand, and Byte. Plus, you'll get the inside scoop from the book authors about the trials and tribulations of publishing.

Thursday Rafting

Once again we are kicking off the SOG with whitewater rafting followed by the Kickoff Cookout. If you are interested in safe thrills then sign up for one of these professionally guided trips.

The all day trip includes transportation from the college, box lunch, 4½ hours of whitewater (up to class 4), and the Kickoff Cookout. The Mackenzie River is famous for its whitewater and the road to the river winds along alpine wilderness.

The 2½ hour trip includes transportation from the college, 1½ hours on the river (up to class 3), and the Kickoff Cookout.

Or, you can choose to attend only the Kickoff Cookout (with the victorious rafters).

Thursday Evening

Following the Kickoff Cookout we'll adjourn to the college for a musical jam session. Bring your instruments and tin ears (or at least stop by for a laugh).

Saturday Evening

We're holding our SOB (Semi-Official Banquet) on Saturday evening. Our keynote speaker, Ezra Shapiro - technical editor for Byte, will follow the food (not just desserts).

Transportation

The nearest commercial airports are Redmond - RDM (15 miles), Eugene (120 miles), and Portland (165 miles). Shuttles run between Bend and the Redmond and Portland airports. Trailways Bus Lines also serves Bend.

If you need travel information, call Bend Travel, 503-388-3424 (they are really helpful folks).

Finally

If you haven't SOGged, then you haven't sogged. So don't miss this year's extravaganza. You'll have a lot of fun and learn a lot without getting soaked (unless you raft, of course).

If you're even considering coming, get this form filled out and in. We'll send you a free packet of information about the area. Also, if you're interested in staying in the dorm (holds two per room) you'll need to call Micro C to verify your reservation. We're limited to 25 rooms and they'll probably go fast.

REGISTRATION FORM

☐ Please send me the **FREE** SOG IV info pkg.

☐ I plan to attend the **FREE** conferences and activities

____ Number of people in my party

Please list any special (non-computer) interests you have, i.e. fishing, hiking, rock climbing, sky diving, etc. _____

_____

_____

		adult	und.12	number	total
Thurs.	All day raft trip	$65	$60		
Thurs.	2½ hr. raft trip	$25	$20		
Thurs.	Cookout only	$10	$7		
Saturday	Dinner	$15	$8		
Dorms	Double room only For the nights of 7/24-7/27	$75			
Food Ticket	3 meals on Friday Breakfast & Lunch on Saturday All you can eat at COCC Cafet.!!	$20	$12.50		
SOG IV T-Shirt	Indicate sizes: S ☐ M ☐ L ☐ XL ☐	$6.50			
				TOTAL $	

Make check or money order payable to: **Micro Cornucopia**

☐ VISA Card # _____

☐ Signature _____ Exp. date ____

Name _____ Ph. # ( ) _____

Address _____

City _____ State _____ Zip _____

MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709
503-382-5060

ORDER FORM

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?

Quantity	Description	Price Each			Total
		U.S.	Can & Mex	Other Foreign	
	SUBSCRIPTION (6 issues per year) ☐ New ☐ Renewal	1 Yr. $16 2 Yr. $30 3 Yr. $42	1 Yr. $22 (Air Mail)	1 Yr. $30 (Air Mail)	
	USERS DISKS – 8" SSSD, CP/M #'s	$15	$15	$15	
	CP/M 86 DISKS – 8" SSSD #'s	$15	$15	$15	
	KAYPRO DISKS for the II, 4, & 10 – 5" SSDD #'s	$12	$12	$12	
	KAYPRO SCHEMATIC PACKAGES ☐ II & 4 (pre-84) ☐ 2, 4, 10 (84) ☐ 10 (No Modem)	$20	$20	$20	
	FREE KAYPRO CATALOG				
	BACK ISSUES #'s	$3	$3	$5	
	T-SHIRT (specify S, M, L, XL)	$6.95	$8.50	$8.50	
	OTHER ITEMS:				

Prices include 1st Class postage (Air Mail for Other Foreign)

☐ Check or money order enclosed
(US funds only, payable on a US bank)

Make checks payable to:
MICRO CORNUCOPIA

TOTAL ENCLOSED

Card No. _____ - _____ - _____ - _____ **Exp.** _____ _____
☐ Visa ☐ MasterCard Signature

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

BACK ISSUES

$3.00 each
US CAN MEX

$5.00 each
Other Foreign

ISSUE NO. 1 (8/81)
Power Supply
RAM Protection
Video Wiggle
1/2 PFM, PRN
16 pages

ISSUE NO. 2 (10/81)
Parallel Print Driver
Drive Motor Control
Shugart Jumpers
Program Storage Above PFM
1/2 PFM, PRN
16 pages

ISSUE NO. 3 (12/81)
4 MHz Mods
Configuring Modem 7
Safer Formatter
Reverse Video Cursor
FORTH words begins
16 pages

ISSUE NO. 4 (2/82)
Keyboard Translation
More 4 MHz Mods
Modems, Lync, and SIOs
Undoing CP/M ERASE
Keyboard Encoder
20 pages

ISSUE NO. 5 (4/82)
Word Processing
Two Great Spells
Two Text Editors
Double Density Review
Scribble, A Formatter
20 pages

ISSUE NO. 6 (6/82)
BB I EPROM Programmer
Customize Your Chars
Double Density Update
Self-Loading ROM
Terminal in FORTH
24 pages

ISSUE NO. 7 (8/82)
6 Reviews of C
Adding 6K of RAM
Viewing 50 Hz
On Your Own begins
24 pages

ISSUE NO. 8 (10/82)
Drive Maintenance
Interfacing Drives
Installing a New BIOS
Flippy Floppies
C'ing Clearly begins
Xerox 820 begins
28 pages

ISSUE NO. 9 (12/82)
BB II EPROM Program
Relocating Your CP/M
Serial Print Driver
Big Board I Fixes
Bringing Up WordStar
Cheap RAM Disk
32 pages

ISSUE NO. 10 (2/83)
Saving a Flaky Disk
Hooking Wini to BB II
The Disk Inspector
JRT Fix
Serial Keyboard Interface
Pascal Procedures begins
36 pages

ISSUE NO. 11 (4/83)
BB I Expansions
BB II Details
Dyna, RAM Disk Review
Easier Reverse Video Cursor
PlannerCalc Review
KayPro Column begins
36 pages

ISSUE NO. 12 (6/83)
256 K for BB I
Bringing Up BB II
dBase II
Look at WordStar
Double Sided Drives for BB I
Packet Radio
5MHz Mod for KayPro
40 pages

ISSUE NO. 13 (8/83)
CP/M Disk Directory
More 256K for BB I
Mini Front Panel
Cheap Fast Modem
Nevada Cobol Review
BB I Printer Interface
KayPro Reverse Video Mod
44 pages

ISSUE NO. 14 (10/83)
BB II Installation
The Perfect Terminal
Interface to Electronic Typewriter
BB I Video Size
Video Jitter Fix
Slicer Column starts
KayPro Color Graphics Review
48 pages

ISSUE NO. 15 (12/83)
Screen Dump Listing
Fixing Serial Ports
Playing Adventure
SBASIC Column Begins
Upgrading KayPro II to 4
Upgrading KayPro 4 to 8
48 pages

ISSUE NO. 16 (2/84)
Xerox 820 Column Restarts
BB I Double Density
BB II 5"/8" Interface Fix
KayPro ZCPR Patch
Adding Joystick to Color Graphics
Recovering Text from Memory
52 pages

ISSUE NO. 17 (4/84)
Voice Synthesizer
820 RAM Disk
KayPro Morse Code Interface
68000-Based System Review
Inside CP/M 86
56 pages

ISSUE NO. 18 (6/84)
KayPro EPROM Programmer
I/O Byte: A Primer
KayPro Joystick
Serial to Parallel Interface
Business COBOL
60 pages

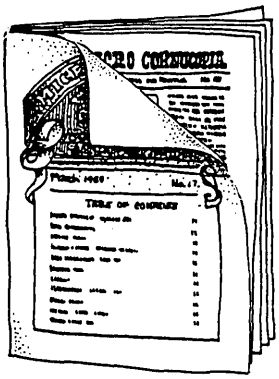
ISSUE NO. 19 (8/84)
Adding Winchester to BBII
6MHz on the BBI
Bulletin Boards, Getting
It Together
Track Buffering on the Slicer
4MHz for the 820-I
64 pages

ISSUE NO. 20 (10/84)
HSC 68000 Co-Processor
DynaDisk for the BBII
Serial Printer on BBI Sans S10
Cheap & Dirty Talker for KayPro
Extended 8" Single Density
72 pages

ISSUE NO. 21 (12/84)
Analog to Digital Interface
Communication Between
High-Level and
Assembly Language
Installing Turbo Pascal
Low Intensity BBI Video
Turbo Pascal, The Early Days
80 pages

ISSUE NO. 22 (2/85)
Converting a Xerox 820-II
to a KayPro-8
S-100 Expansion for Single Board
Z80 Systems
Sound Generator for the STD BUS
Reviews of 256K RAM Expansion
Boards
In the Public Domain
88 pages

ISSUE NO. 23 (4/85)
Automatic Disk Relogging
With CP/M 2.2
Interrupt Driven Serial Print
Driver
Low Cost EPROM Eraser
Smart Video Controller
Review: MicroSphere RAM Disk
Future Tense begins
88 pages



ADVERTISER'S INDEX

The Last Page

By Gary Entsminger

Micro C Staff

When you talk about the 16-bit world, you're usually referring to the Motorola 68000 and the Intel 8086 series. Both are powerful systems that we're going to be hearing a lot about this year.

8086 Vrs. 80286

Compatibility has been the strength of Intel's microprocessors since the 8008 of the '70s, and the 8086 family continues the tradition.

The 8086 series accesses memory by dividing the 16 megabytes into 256 64K chunks. (This is called segmented addressing.) The reason for this chunky addressing is that an 8086 creates an address by combining its 16-bit address register with 8 bits stored in an offset register. Four segment registers access physical memory.

In the past many routines were limited to 64K because programmers didn't want to deal with the offset (segment) register. The 80286 makes it easier on programmers by adding an invisible register set to handle offset addressing, so the four segment registers become invisible to the program on the 80286, and the programmer doesn't have to worry about them.

Their visible register sets are still identical, and therefore compatible. So, from a programmer's perspective, the 80286 is really just a modified version of the 8086.

In fact, the 80286 can operate in two modes—a fully compatible 8086 mode, with a 1 megabyte memory limit, or an enhanced mode with 16 megabyte memory limit and increased speed. This family compatibility makes the new Kaypro, IBM, and the Slicer kissin' cousins.

68000

Unlike the 80286, the 68000 has 32-bit data and address registers. Motorola brings out 24 of the 32 address bits giving you direct access to 16 megabytes of memory. (This is called linear addressing.)

The 68000 has memory-mapped I/O, and gives a program direct access to the entire 16-MB address space. The instruction set is simpler than the 80286 and is arguably a programmer's chip.

It has 8 data and 8 address registers, all 32-bit, and all general purpose. This bounty of registers virtually eliminates saving register values, thus saving time. And you can operate on all 32 bits with a single instruction.

The 68000 combines these classes of register transfer instructions—load, store, push, pull, and pop—into one single class: MOVE. Variations of MOVE can transfer 8-, 16-, or 32-bit data by simply changing a suffix. For example, the instructions to add 16-bit numbers:

```
MOVE.W  VALUE1,D0  Get first value
ADD.W   VALUE2,D0  Add 2nd to 1st
MOVE.W  D0,RESULT  Store result
```

could be changed to add 8-bit:

```
MOVE.B .......
ADD.B .......
MOVE.B ......
```

or 32-bit numbers:

```
MOVE.L ......
......
```

Compatibility

The 68000 was designed to interface directly to the 6800 line of 8-bit peripherals, in order to utilize existing circuits. A MOVEP instruction moves either 16 or 32 bits of any register to a port in 8-bit chunks.

Showdown—68000 Vrs. 80286

The 68000 and the 80286 are both speedy 64-pin microprocessors, and both are true 16-bit processors. Simplicity of instruction set, 32-bit registers, and linear access to memory are in Motorola's corner. But more registers require more silicon, and so the manufacturing cost is higher.

Intel's hardware-ease approach has kept it ahead in the marketplace. After all, you need hardware first. And the 80286 (like the 8086) has clean support for co-processors (the 80287 floating point and the 802730 text and graphics co-processors, for examples). But as hardware has become less expensive, the door has opened for Motorola. Chip to chip—this will be a fun competition to watch.

Wrap Up

So far, we've received several CP/M-MS-DOS file transfer programs, and should have one ready to release to the public domain real soon. There are also new CP/M and MS-DOS disks just around the corner, so stay tuned.

Meanwhile, IBM's stock is slipping, and several new boards look really interesting—Trevor's new 32032-based board has me jumping. It looks like we're going to be building some dynamite systems here this summer. At SOG, Trevor Marshall will be holding four sessions on this 32-bit chip (which is about as powerful as the DEC VAX 11-750), so come to Bend and see what's building.

■ ■ ■

CO-POWER

We've just added new features & reduced prices!

1024k RAM, MS-DOS and RAMDISK just $789.95 (most models)

We now have a one megabyte CO-POWER-Plus. Like CO-POWER-88, it is an 8088, add-on 16-bit coprocessor for many CP/M computers. CO-POWER-Plus expands from 256k to 1024k RAM. You can get all 1024k at once, or add it 256k at a time.

CO-POWER lets you upgrade without sacrificing your CP/M system. CO-POWER-Plus has just been updated to include many new features. Current specifications are:

- 5.33 Mhz 8088 processor
- Up to 1024k RAM, using 256k RAM chips
- Includes MS-DOS 2.11 with these features:
  - . new built-in ANSI screen driver
  - . new instructions for patching modem programs, including a patch for ASCOM.
  - . IBM-PC disk format, 5 1/4".
- New file transfer utility to convert CP/M data files to MS-DOS and vice versa. Runs under MS-DOS.
- Our CP/M RAMDISK program uses CO-POWER's RAM as a simulated disk drive for CP/M. Imagine what you can do with one Meg!
- Public domain disk available with MS-DOS RAMDISK program. Set up your CO-POWER RAM between MS-DOS and a ramdrive.
- For ATR; Kaypro 2,4 (1983); Osborne I; Morrow MD3; Xerox 820; Zorba computers.
- MS-DOS 2.11A version available for Kaypro 2/84, 2X, 4/84 and 10s. Includes a utility to run IBM-PC LOTUS 1-2-3! MS-DOS 2.11A CO-POWER-Plus with 1024k is $889.95.

CO-POWER-88 Owners: ACT NOW TO TAKE PART IN **TRADE-IN DAYS.** EXCHANGE YOUR CO-POWER-88 FOR A $200 CREDIT TOWARDS A CO-POWER-PLUS! CALL SWP SALES FOR DETAILS.

Pricing:

256k CO-POWER-Plus, MS-DOS 2.11	$549.95
256k CO-POWER-Plus, MS-DOS 2.11A (KP 2/84, 2x, 4/84, 10)	$649.95
each additional 256k	80.00
1024k CO-POWER-Plus, MS-DOS 2.11	$789.95
1024k CO-POWER-Plus, MS-DOS 2.11A (KP 2/84, 2x, 4/84, 10)	$889.95
256k CO-POWER-88 now just	$399.95

Note: CO-POWER-88 RAM is not expandable.

ATR

Ideal computer for custom systems!

SWP's ATR8500 is a unique CP/M computer. It can use either a terminal or an ATARI computer for the display device! (For ATARIs it is also a complete interface for peripherals.) It easily converts to an 8-bit/16-bit system by adding CO-POWER!

- -Z80a processor
- -64k RAM
- -2 Serial Ports
- -1 Parallel Port
- -Drive Port: runs up to 4, 5 1/4" drives
- -Z80 expansion bus
- -Hardware UART
- -Compact: 11" x 5 5/8"
- -Expandable w/ CO-POWER
- -Includes friendly CP/M Some utilties are:
  - CONFIG - Set up system
  - DISKDEF - read/write 40+ CP/M disks
  - MODEM - for commun.
  - DDINIT - formatting
- -Optional 2 ATARI ports, no additional charge
- -Optional DMA controller
- -Optional bank-switched RAM/ROM

A&T Board w/ CP/M just $349.95. Power Supply, & enclosure extra.

DUAL DENSITY

$149.95

Bigboard & Xerox 820-I

Add the most popular double density system to your computer. All you need is a bootable SD CP/M disk. Features:

- -Controller board with 1791, plugs into 1771 socket.
- -Changing the IOBYTE customizes system for nearly any printer.
- -8" SSDD storage is up to 674k. DSDD is double that. 5 1/4" SSDD storage is 188k and DSDD is 374k.
- -Includes format and sysgen programs.
- -Special features include enhanced pause, screen print and clock.

SWP
MICROCOMPUTER PRODUCTS, INC.

1000 W. Fuller
Ft. Worth, TX 76115
(817) 924-7759