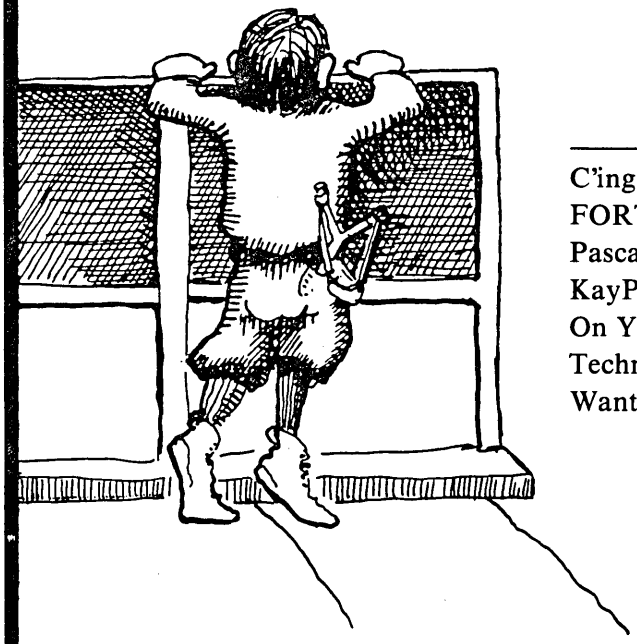
$3.00

MICRO CORNUCOPIA

THE SINGLE BOARD SYSTEMS JOURNAL

August 1983 No. 13

TABLE OF CONTENTS



REGULAR FEATURES

NEW LOWER PRICES

"THE ORIGINAL BIG BOARD"

OEM - INDUSTRIAL - BUSINESS - SCIENTIFIC

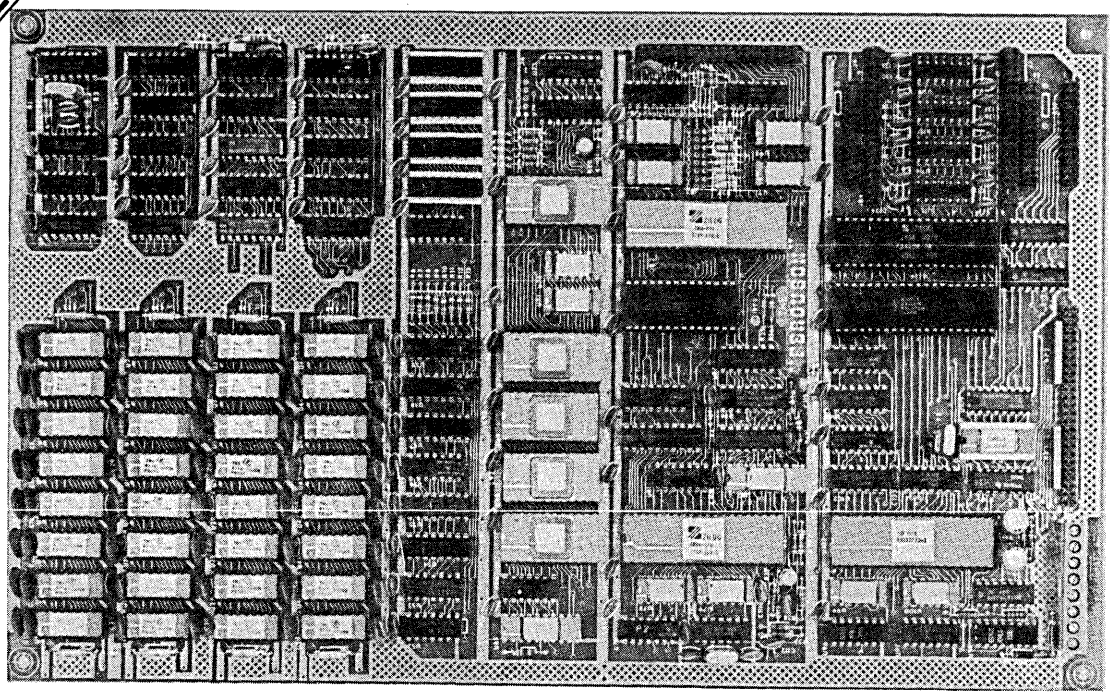
SINGLE BOARD COMPUTER KIT!

Z-80 CPU! 64K RAM!

(DO NOT CONFUSE WITH ANY OF OUR FLATTERING IMITATORS!)

NEW!

PARTIALLY ASSEMBLED KITS
For All Sockets Installed
And Soldered Add $50.
(Not For Blank PCB)



WANT MORE INFO?
Full Documentation and
Schematics — $5.

THE BIG BOARD PROJECT: With thousands sold worldwide and over two years of field experience, the Big Board may just be one of the most reliable single board computers available today. This is the same design that was licensed by Xerox Corp. as the basis for their 820 computer.

The Big Board gives you the right mix of most needed computing features all on one board. The Big Board was designed from scratch to run the latest version of CP/M*. Just imagine all the off-the-shelf software that can be run on the Big Board without any modifications needed.

$ 279 00 ** (64K KIT BASIC I/O)

SIZE: 8½ x 13¾ IN.
SAME AS AN 8 IN. DRIVE.
REQUIRES: +5V @ 3 AMPS
+ - 12V @ .5 AMPS.

FULLY SOCKETED!

FEATURES: (Remember, all this on one board!)

64K RAM

Uses industry standard 4116 RAM's. All 64K is available to the user, our VIDEO and EPROM sections do not make holes in system RAM. Also, very special care was taken in the RAM array PC layout to eliminate potential noise and glitches.

Z-80 CPU

Running at 2.5 MHZ. Handles all 4116 RAM refresh and supports Mode 2 INTERUPTS. Fully buffered and runs 8080 software.

SERIAL I/O (OPTIONAL)

Full 2 channels using the Z80 SIO and the SMC 8116 Baud Rate Generator. FULL RS232! For synchronous or asynchronous communication. In synchronous mode, the clocks can be transmitted or received by a modem. Both channels can be set up for either data-communication or data-terminals. Supports mode 2 Int. Price for all parts and connectors: $39.95

BASIC I/O

Consists of separate parallel port (Z80 PIO) for use with an ASCII encoded keyboard for input. Output would be on the 80 x 24 Video Display.

BLANK PC BOARD — $99.95

The blank Big Board PC Board comes complete with full documentation (including schematics), the character ROM, the PFM 3.3 MONITOR ROM, and a diskette with the source of our BIOS, BOOT, and PFM 3.3 MONITOR.

24 x 80 CHARACTER VIDEO

With a crisp, flicker-free display that looks extremely sharp even on small monitors. Hardware scroll and full cursor control. Composite video or split video and sync. Character set is supplied on a 2716 style ROM, making customized fonts easy. Sync pulses can be any desired length or polarity. Video may be inverted or true. 5 x 7 Matrix - Upper & Lower Case.

FLOPPY DISC CONTROLLER

Uses WD1771 controller chip with a TTL Data Separator for enhanced reliability. IBM 3740 compatible. Supports up to four 8 inch disc drives. Directly compatible with standard Shugart drives such as the SA800 or SA801. Drives can be configured for remote AC off-on. Runs CP/M* 2.2.

TWO PORT PARALLEL I/O (OPTIONAL)

Uses Z-80 PIO. Full 16 bits, fully buffered, bi-directional. Uses selectable hand shake polarity. Set of all parts and connectors for parallel I/O: $19.95

REAL TIME CLOCK (OPTIONAL)

Uses Z-80 CTC. Can be configured as a Counter on Real Time Clock. Set of all parts: $9.95

CP/M* 2.2 FOR BIG BOARD

The popular CP/M* D.O.S. to run on Big Board is available for $139.00.

BIG BOARD SOFTWARE SPECIAL — $149

Through special arrangement with CDL we offer a powerful package of TDL Z-80 software that has a suggested retail of almost $600. Includes: Extended Disk Business Basic, ZEDIT text editor, MACRO II Macro Assembler, LINKER, DEBUG I and DEBUG II. Supplied on 8 in. diskette with extensive manual.

PFM 3.3 2K SYSTEM MONITOR

The real power of the Big Board lies in its PFM 3.3 on board monitor. PFM commands include: Dump Memory, Boot CP/M*, Copy, Examine, Fill Memory, Test Memory, Go To, Read and Write I/O Ports, Disc Read (Drive, Track, Sector), and Search PFM occupies one of the four 2716 EPROM locations provided. Z-80 is a Trademark of Zilog.

Digital Research Computers

(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's (for the Big Board only) with a $75 deposit. Balance UPS COD. Add $4.00 shipping.

USA AND CANADA ONLY

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A FREE COPY IS AVAILABLE ON REQUEST.

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE ORIGINATORS OF CPM SOFTWARE
**1 TO 4 PIECE DOMESTIC USA PRICE.

MICRO CORNUCOPIA
P.O. Box 223
Bend, Oregon 97709
503-382-8048

Editor & Publisher
David J. Thompson

Graphic Design
Sandra Thompson

Technical Guru
Dana Cotant

Staff Assistants
Dorcas Dsenis
Alice Holbrow

Typography
Patti Morris & Martin White
Irish Setter

MICRO CORNUCOPIA is the single board systems journal supporting systems programming languages and single board systems—including the Big Board, the Big Board II, and the Xerox 820.

MICRO CORNUCOPIA is published six times a year by Micro Cornucopia of Oregon, P.O. Box 223 Bend, Oregon 97709

SUBSCRIPTION RATES:

1 yr.(6 issues)	$16.00
1 yr.(Canada & Mexico)	$20.00
1 yr.(other foreign)	$26.00

All subscription orders payable in U.S. funds on a U.S. bank, please.

ADVERTISING RATES: Available on request.

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

SOFTWARE, HARDWARE, AND BOOK VENDORS: We would very much like to review your CP/M compatible products for Micro C. Please send material to the Review Editor, Micro Cornucopia.

LETTERS TO THE EDITOR: Please sound off.

MICRO CORNUCOPIA

August 1983 The Single Board Systems Journal No. 13

Drip Dry!

The SOG

If nothing else, the SOG (Semi Official Get-Together) proved that you don't have to go to a big city convention to have technical discussions, see the latest equipment, enjoy good food and drink and make a splash.

The food was potluck, so variety was the name of the game. Of course it's a little hard to bring a warm pan of muffins when you have to travel two or three thousand miles, but people certainly went out of their way to bring something (all the way up to a seven foot circumference pizza from a local pizza parlor). A few Texas folks even suggested that a pizza that large could only have been flown in from you-know-where. But that was a lot of bull.

The technical discussions were first rate and we'll be covering some of them in future issues.

And as equipment went, the name of the game was bigger and faster. We had winchesters (bigger) attached to just about anything that moved. We had RAM disks and the Slicer (faster) for those who were jaded by floppies and Z80s.

Plus, we saw color sprite graphics (more about that in the next issue) running on the Big Board I and the Kaypro. This was the real Pacman, nibbling its way across the screen.

Of course, we can't forget the splash.

The raft trip

Friday afternoon, three large vans showed up at the house to pick up 55 merry (and somewhat anxious) rafters. 43 went on the white water run, 12 went slack.

After we reached the river, we donned life jackets that made us look more like robots (mechanical engineers?) than people. Once into the jackets, we were told that we would be perching precariously on the sides of the rafts.

We were supposed to paddle like mad to keep the raft under control in the rough waters and to keep our balance. If you stopped paddling in the white water chances were very good you'd lose your perch and fall either into the center of the raft or out of the raft entirely.

Anyone with the audacity to fall into the middle of the raft was asking for a bruised bottom (from protruding rocks). Anyone falling out of the raft would be retrieved (if possible) farther downstream.

Following these encouraging words, we divided up into groups of eight—seven SOGs and a guide—then each group climbed carefully into a raft.

Everyone got a paddle, and we spent the first 30 minutes getting used to paddling straight, in circles, and backwards before entering the first rapids.

Wet

Everyone had been warned that we might get a bit wet, but then it wouldn't have been a real SOG otherwise, right?

Those of us in the front of the rafts got soaked as we went through the roller coaster-like rapids. But everyone eventually got soaked as vicious (sneaky) splash fights broke out between boats.

A Rapid Description

White water is rated between 1 and 6. A 1 wouldn't wake a 5-year-old on an air mattress. Niagara falls is a 6.

Our first rapid was a 1½, the second and third were 2s. The fourth was a four so we pulled out the rafts, emptied them, and ported around the monster. The next series of rapids was rated at 3 and we went through them (not over or around, but through).

It's a pretty amazing feeling to look into the fangs of a large wave as you head straight into it. I was so awestruck that I quit paddling (a definite no-no) and got dumped into the middle of the boat.

Farther downstream, we went through a reversal (where the current

(continued on page 38)

LETTERS

Dear Editor,

Computer Dynamics sells used 8" Shugart or Tandon drives for $100 each (as pulled from working equipment) or $150 each aligned. They say that the drives work well with the Big Board and can be used double density. The company also sells a 12" B&W CRT with a keyboard for $125.

I don't know about their service, but I'll let you know if I have any problems. They can be reached at: Computer Dynamics, 105 S Main St, Greer, SC 29651.

Congratulations on a fine magazine, I hope to submit something when I have my system assembled and have something useful.

D. Gregory Bishop
348 Pictou Rd
Truro, Nova Scotia B2N 2T5

Dear Editor,

I have a CompuServe executive program available for the Kaypro. The price is $5.00 plus an additional $5.00 for the 5" disk and postage. Purchaser should specify 300 or 1200 baud.

William Hutchison
PO Box 278
Exton PA 19341

Dear Editor,

Keep up the good work, I wish you were monthly.

Question: I recently obtained the user disk with the UNIX style editor. Unfortunately, one backspace displays as two backspaces on the system. Any idea what might be going on? I am using the ADM-3 terminal emulation.

Another thing, as a BB II owner, I'd like to encourage you to keep after CAL-TEX to try to get out a theory of operation like the one for the BB I.

James Binkley
20820 SW Blanton
Aloha OR 97007

Editor's note:

We've fixed EDIT.COM's backspace problem, please return your user disk and we'll update the program free (and that goes for anyone else having this trouble). Thanks for bringing the bug to our attention.

The documentation problem with the BB II has left me in an uncomfortable position. I really like the BB II (I have two running here with winchesters) and I appreciate Bill Siegmund's desire to have things absolutely correct—he completely assembled and tested the first 30 unkits just to make sure they met his requirements. Twenty-eight of the 30 came up immediately, the other 2 had solder shorts.

However, there is so much need for more documentation that if I were Bill, I would be willing to release the information even if it didn't meet my standards for writing quality etc. But then, I'm not Bill.

Dear Editor,

Just a note to let you know that the articles on troubleshooting (Issues #4 and #9) have been invaluable.

I was also especially interested in the article in #8 on configuring MFE drives because I have two.

One correction to the article is required if both drives are double sided, double density. On drive 1, DI must not be jumpered, it must be cut.

I used the Hazeltine 2000 case from D&W associates and was fairly satisfied. I was disappointed that the keyboard was upper case only and that it was almost worn out (the keys were sticky). But everything fit well and the monitor works fine.

I am now interested in building a BB II system. I have the sparse literature from CAL-TEX but I need more information, such as a parts list, operation description, etc for evaluation. The package from DRC was so good, I am spoiled. Any articles or a copy of the parts list would be greatly appreciated.

I'm also curious whether the CO-POWER-88 board would work with the BB II.

Meanwhile, keep up the good work!

Victor Yarberry
7105 Winans Dr NE
Albuquerque NM, 87109

Dear Editor,

I have solved a couple of problems with my BB I.

The first problem caused my system to crash on random occasions, sometimes every 30 minutes or so, sometimes it would take 3 or 4 hours before it would die.

I carefully rechecked the power supplies for voltage levels and noise. Everything was OK. I used my heat gun (a hair dryer works well also) and warmed up the board an area at a time and then used a large can of freeze spray to cool selected areas, but I couldn't consistantly cause a failure.

Finally, I removed and reseated all the ICs in their sockets. That fixed the problem. Apparently oxidation had created low value resistors between the ICs and the sockets, and the mechanical action cured the problem.

The second problem was interfacing my old Centronics 779 printer to the centronics port as suggested by Mike Worley in Micro C Issue #7. I jumpered J3 pins 7 and 8 and used the printer's acknowledge signal in place of busy, which gave me a marginally usable printer.

However, I always got double line spacing (auto linefeed was turned off), and I occasionally didn't get the first character on a line. A borrowed Okidata worked just fine.

It turned out that the handshake signal was just too quick for the old printer. I unstrapped J3 pins 1 and 2 which gave me a low going pulse which I fed into a one shot which generated a 10 micro second low going pulse. With J3's pins 13 - 14 tied together and pins 15 - 16 tied together, the printer works fine.

I have appreciated the notes which appeared in Micro C. Those related to video components were most helpful when I switched from separate to composite display.

Charles Vanleeuwen
3395 SW 116th
Beaverton OR 97005

Dear Editor,

I have owned a Kaypro II since March and am very please with it. I am used to dealing with an IBM 3033 and hence I have had much to unlearn and relearn. I am still amazed at the great pile of software which comes with the machine. It will be a while before I am able to use all of it properly.

I agree with most of your points in the Kaypro reviews (issue #10 p. 26 and issue #11 p. 24), but I have had no heat problems nor drive problems. My Kaypro has been absolutely dependable.

I have one major complaint. If I accidentally hit two keys at once while holding down the cntl key, the system hangs up, and I have to hit the reset button to unhang it, thereby losing everything in memory.

My other complaint is the documentation. IBM offers some of the world's worst documentation, but the documents which come with the Kaypro appear to have been slapped together by non-writers who feel that the way to

teach someone English is to hand him a dictionary and say "start with page one and good luck."

Actually, the S BASIC manual is pretty good, and the others are pretty good reference manuals, but one does not learn to speak or write a language from lexicons and grammars. Hopefully someone will come up with a good book for Kaypro users. It is discouraging to browse through the hundreds of books in a Walden or Dalton bookshop and see scores of Apple titles and dozens of IBM PC titles but nothing for Kaypro.

I'm a technical writer, and my special interest is explaining technicalities to laymen. I started out as a layman with a strong background in theatre arts, so I have retained a lifelong sympathy with the plight of the slow-headed dumbbells who don't understand explainations from experts.

John Caffrey
6701 E Granada Rd
Scottsdale AZ 85257

Editor's note:

You've made a good point John, and it's one that needs to be made over and over again. There is a real shortage of "friendly," "handholding," "this is how you do it" information.

Perhaps you can help fill the void once you have your head above water with the Kaypro.

Also, you might be interested in a new volume that sounds like at least part of what you're asking for. It's called The Perfect Manual *for the Kaypro II, from PeopleTalk Associates, 4054 McKinney, Suite 209, Dallas TX 75204. Price is $17.95.*

This manual covers the Perfect package, with an initial chapter on use of CP/M. It is well written and will definitely get you started successfully with any of the Perfect programs.

It is also worth picking up if you are interested in trying your hand at self-publishing. It was done very inexpensively, but the information is so important and so well laid out that this book should do very, very well.

Dear Editor,

I have just finished the 256K modification to my BB I (from issue #12). On page 8 step 18, "Ui pin 8 to Ci+26 . . . " should read "Ui pin 8 to Ci+39 . . . " This change is pretty obvious but I though I should pass it along.

By the way, the system came right up after I remembered to install the new 64K chips in bank 0.

Bob Carol
216 Oswego
Huntington Beach CA 92648

Dear Editor,

I have located sources of BB I ICs and connectors and can make them available to readers very inexpensively. For instance, the power connector is only $2.75, and the Z80 B is only $11.00.

Stuart Russell
1105 11th St #11E
Boulder, CO 80302

Dear Editor,

I really enjoyed your last issue (#10), especially the article on the winchester drive. However, I had trouble reading the magazine since everything was upside-down except the cover.

·ʞoɹʍ pooƃ ǝɥʇ dn dǝǝʞ ʎɐʍʎuⱯ

Mitchell Mlinar
1013 W 210th
Torrence, CA 90502

Editor's note:

Thank you for your note, Mitchell. It appears that you received one of the copies we had specially printed for Australia.

Dear Editor,

Having been a self-employed consultant for the past 8 years, I was thoroughly tickled by the article by Hampton Miller. I now have a way to respond to those who ask me why I accept this insecure, unpredictable, and otherwise unstructured life.

Robert New
1638 South Grant Street
San Mateo, CA 94402

Editor's note:

Thanks for the comment Robert. Hampton's "On Your Own" in issue #12 has generated a great many rave reviews such as yours. (Sigh!)

Also, congratulations on getting unbanked versions of CP/M 3 running on both the BB I and BB II. They were a nice addition to the SOG.

Anyone interested in CP/M 3, either banked or unbanked, should get in touch with Robert. He says that unbanked CP/M 3 runs about 25 percent faster than 2.2, plus it has a better assembler (RMAC), better documentation, and it is easier to install.

CP/M 3 can be very cheap (as low as $7.50 per copy in 10,000 quantity). So folks might want to get together and pool their pennies.

Dear Editor,

I just wanted to drop you a note to let you know that the modified version of SpellSys is doing fine. You were successful in eliminating the "CNTL-Z" bug (caused by a non-standard file termination) so I no longer get garbage at the end of the documents produced by FinalWord.

At $29.00, the spell checker is a real bargain, even with the bug. Now that the bug is gone, I am thoroughly satisfied.

I am most impressed by the prompt and personal service I received from you. I doubt that many software outfits would offer this kind of attention.

I hope your company will come out with more great software like this (especially for word processing).

Alan Gomes
14612 San Bruno Drive #3
La Mirada, CA 90638

Editor's note:

You are very welcome Alan, but I can't really take the credit. Willis Gore, who wrote SpellSys, has been doing yeoman's duty making it compatible with all flavors of text editors. Plus, we both really appreciated your assistance in diagnosing the problem. Bringing up a really universal, and friendly, spelling checker is not much different than bringing up a stubborn Big Board. Both can be very frustrating and very exciting.

Dear Editor,

I am using Stieglitz's SCBIOS from user disk #2 and Worley's connector configuration from Micro-C #7, pg 2. However, for a while I was not able to make my prowriter 8510-AP parallel printer do a tap dance with the BB I's parallel port. It seemed to choke after receiving 40 or 50 characters and would lock up.

After much trial and even more error, I found that using the Prowriter's ACK line (pin 10) instead of the BUSY line (pin 11) solved the problem immediately.

Philip Plumbo
1128 Dayton Ave
St Paul, MN 55104

Dear Editor,

I have 300 used disks (8") that I will be glad to share with subscribers who are cheap like me. I paid 25 cents each and I can supply them verified in any single-sided format.

Also, I have software publisher's dual density package, all the Micro-Pro software, Basic-80, M80, ZSID, JRT 2.2, and JRT 3.0. I suppose I should become serious, but I can't. I'm having too much fun just learning.

Neil Joba
4774 Carter Rd
Fairport, NY 14450

(Letters continued on page 28)

The CP/M Disk Directory

By Don Brittain

4200 Spruce St. Apt 208
Philadelphia, PA 19104

Editor's note: The following article is about fixing a flaky disk. But in many ways, it is much more a treatise on how CP/M uses the directory to locate information on the disk. Once you have read through this a few times and digested at least part of it you'll appreciate what some good tools—both public domain (DUMPX and VERIFY) and commercial (Disk Inspector)—can do to make life with disks much easier. I use all three of these tools and find each absolutely invaluable in its own way. DUMPX and VERIFY are on user disk #5.

What do you do with disks which have bad sectors? Throw them out? I can't afford that. Ignore the bad sector messages? Could cause disasters at some future time. One easy solution, of course, is to buy SuperSoft's Disk Doctor, but at $100, this doesn't seem very reasonable to me—especially considering SuperSoft's reputation. The purpose of this article is to discuss another alternative.

Tools

To follow this article you will need an extended (track and sector) disk dump program, such as DUMPX on user's disk #5. It is possible to use PFM's disk read command, but this is awkward at best. Ward Christianson's DU or DUU will also work but they are not very easy to use.

Before we can take care of the bad-sector problem we have to understand the CP/M file structure. In this article I will only deal with single-density standard IBM 3740 because it is the simplest case, and it is a standard.

Starting simply

Single-density is simple because you don't have to distinguish between logical and physical extents: both refer to 16K bytes of data, which is precisely the amount of space controlled by one directory entry.

The amount of disk space a single directory entry controls is called a physical extent, while a logical extent always refers to 16K bytes of space. Each directory entry on a double density disk usually controls 32K or more (thus there would be two or more logical extents—each 16K—for every physical extent). STAT always lists the number of logical extents under the Ext heading.

Anyway, for each physical extent there is a 32-byte entry in the directory. This entry is called an FCB (file control block) and it tells CP/M where to find the file on the disk. *(Editor's note: At this point you should use DUMPX or PFM to dump the first sector of an active directory (track 2 sector 1).* This sector has room for 4 directory entries (physical extents). Each of these 32-byte entries contains the following information.

(1) The first 12 (of 32) bytes in the extent contain the name of the file. (Byte 1 contains a drive code, and the next 11 bytes contain an ASCII representation of the actual file name and type in uppercase letters.)

(2) The next byte tells CP/M which physical extent of the file this directory entry describes. It is zero if the file takes up less than one physical extent.

(3) Bytes 13 and 14 are reserved for system use and need not concern us here. (They should be set to 0.)

(4) Byte 15 contains the number of records this physical extent controls. (A record is 128 bytes.) This is the most accurate record CP/M keeps concerning the size of a file. STAT uses this number to calculate the number of records in a file.

(5) Bytes 16 through 31 contain the map. CP/M uses this space to record which groups (also called blocks) have been allocated to this physical extent. Thus there is room in an extent for 16 blocks.

Groups and Blocks

What is a group (or block)? Well, they are the same. A group (or block) is the smallest amount of space a directory entry can control, or equivalently, the smallest unit of space which CP/M can allocate to a file. For single density this is 1K.

If you type "STAT A:DSK:" one of the values displayed will be the number of records per block. On a standard single-density disk, CP/M allocates 8 records per block, and space on the disk is parceled out in 1K groups since (8 recs) x (128 bytes/rec) = 1024 bytes.

Note that for a single-density disk, STAT also reports there are 128 records/extent. Thus each (physical) extent controls 128 x 128 = 16K of space. This verifies that logical and physical extents coincide. Be forewarned that the word "extent" is used in the CP/M manuals without further qualification, and can refer to either logical or physical extents.

Groups (blocks) represent space on the disk, and as far as disk usage is concerned, CP/M get its information here.

The blocks are numbered, with the lower numbers corresponding to space near the outer edges of the disk (near Track 0). CP/M keeps track of which groups have already been allocated, and distributes disk space by using the lowest numbered unassigned blocks first.

This is why most disk activity takes place near the outer edges. This also explains why a file can be spread all over the place. If you erase several small files having low allocation numbers, then CP/M will allocate their blocks to the next file that needs disk space.

The program BITMAP from user's disk #2 (or CP/MUG) will display the current state of your system's allocation blocks: a 1 means the block has been assigned to a file, and a zero means it's free. In fact, your CBIOS contains a 'bitmap' (technically known as an allocation vector) for each disk drive in your system. It is this information that CP/M uses to make a changed-disk R/O if you don't warm boot. (If the new disk has the same allocation vector, then CP/M gets fooled. and if you perform a disk write, you get angry.)

Now, since a (logical=physical) extent in a single-density system consists of 16K, and an allocation block consists of 1K, it is necessary for each directory entry to control exactly 16 allocation blocks. But there are exactly 16 bytes in an FCB to be used for controlling disk space. How convenient!

Each block represents 1K of space and a single-density disk has approximately 250K of space. Thus we need to use one byte to describe 250 different things and 1 byte can store numbers from 0 to 255. Thus, all we need to do is number the blocks consecutively, starting from 0. Since CP/M doesn't count the system tracks, the blocks are numbered from 0 to 243, with blocks 0 and 1 corresponding to the directory space on track 2.

Double Density

It should now be apparent why double-density disk allocation is slightly more complicated. CP/M still leaves only

(or How to Fix a Flaky Disk)

16 FCB bytes to describe space on the disk, but for double-density disks these bytes have to describe 32, 64, 128, or 256K.

Furthermore, the amount of data in a record will no longer equal the amount of data in a disk sector. (By the way, the STAT D:DSK: command doesn't really tell you the number of sectors per track. It is really the number of records per track.)

Thus allocation blocks in DD systems must describe more space, and there may not be a simple way to to figure out what block a disk sector belongs to. The CBIOS must handle all the details.

Single Density Again

Meanwhile in single-density, each allocation block contains exactly 8 physical sectors, each sector 128 bytes. It would be nice, at least for the purpose of this article, if group 0 (i.e. allocation block 0) corresponded to sectors 0-7 on track 2, and group 1 corresponded to sectors 8-15, and so on.

However, you would not appreciate this simplicity when it came time to begin a 10-minute compile——the compile would probably take 1 hour. This is because the disk would have to make a complete revolution between each access to consecutive sectors.

The reason is, that after writing to the first sector, the disk controller chip has to do some bookkeeping, and by the time it hears the CPU tell it to write something to the next sector, that sector has just gone by. Thus the disk controller has to wait one full revolution.

Thus CP/M blocks aren't written onto consecutive sectors. Rather, they are written onto every 6th sector.

Group 0 contains sectors 1, 7, 13, 19, 25, 5, 11, 17 on track 2. Group 1 contains sectors 23, 3, 9, 15, 21, 2, 8, 14 on track 2.

Since there are 26 sectors per track, and there are 8 sectors per block, there are 3.2 blocks per tracks and some blocks contain sectors on more than one track.

By the way, the correspondence between blocks and physical tracks and sectors is known as the sector translation and the correlation must be contained in the CBIOS. CP/M 1.x only supported single-density disks and had BDOS do the sector translation calculations.

Curing Bad Sectors

Suppose you know that track 33 sector 0B (both numbers hex) is bad. This knowledge doesn't come from divine revelation, any reasonable CBIOS tells you exactly what track and sector is giving CP/M fits.

You can also use special programs designed to check disks to see where the bad sector lies.

The VERIFY program, for instance, will nondestructively test all tracks and sectors of a given disk and attempt to restore faulty sectors. Sectors that can't be restored are reported on the console twice; sectors that have been restored are reported only once. *(Editor's note: In two or three passes VERIFY can, many times, completely clean up a disk.)*

DUMPX has a validate command but it simply reports the bad sector and then dies rather than trying to fix it and move on.

Back to the Disk

Anyway, once you know where the problem is, (and VERIFY can't fix it) you need to prevent use of a bad sector by creating a CP/M directory entry which contains the faulty sector.

In order to create a directory entry you need to find which block contains the faulty sector. One way to do this is to use DUMPX (in command mode) to dump random blocks (or the specific blocks from the program that bombs) until you find the block which makes DUMPX crash. You'll have to reset and reboot but you'll know which group contains the bad sector.

Ed. note: You can also get a disk utility which tells you the block number when you read the track and sector. The Disk Inspector does this and gives you a very easy way to create and recreate directories at will. Or you can get down into the trenches and hand-calculate the exact block number, but it's not easy.

However, you know that a track contains 3.2 blocks and that block 0 is the first block on track 2. So it's pretty easy to estimate within 3 or 4 blocks since you know the track number.

Editing the Directory

Okay, so we now know which block contains the bad sector. What next? Simply create a directory entry which includes only the group containing the damaged sector. To do this you can use DUMPX to edit the directory.

```
(the '*' is DUMPX's prompt)

Dump groups 0 and 1:

        A>DUMPX<cr>
        *G 0<cr>

        (Group 0 is now displayed)

        *G 1<cr>

        (Group 1 is now displayed)
```

Look for an unused directory entry (one beginning with e5, or containing only e5's). This is where the new dummy directory will be placed. I usually look at the very end of the directory first (track 2, sector 14), since this almost always is all e5's, and a dummy FCB placed here will also appear last in CP/M's DIR.

Use DUMPX to edit the track and sector you wish to have contain the dummy file's FCB. For example, to remove block A0 from circulation, you could use DUMPX to make the following changes in the directory.

```
*READ TRACK 2 SECTOR 14 EDIT<cr>

(Track 2 sector 14 displayed here)

EDIT - 60<cr>

(assuming the 4th dir entry in this
 sector is free)

Now enter the following:
(each followed by a <cr>):

00 62 61 64 20 20 20 20
20 73 65 63 00 00 00 08
A0 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00.
```

Now type a period so that DUMPX will display the modified sector. The right-hand portion should have "bad-.sec" displayed (in lower case). This is the name of the dummy file, and since

(continued next page)

(The CP/M Disk Directory continued)

it's in lower case you don't have to worry about accidentally erasing it. (Indeed, bad.sec is going to be around forever, unless you hand edit it in the future or execute an ERA *.* command.) The 08 at the end of the first row tells CP/M that the dummy file contains 8 records (actually, any number between 1 and and 8 will do fine), and the A0 at the beginning of row 2 tells CP/M that these records are recorded in block A0H (which is 160 decimal).

Thus, as long as bad.sec is in existence, CP/M will not let any file be written onto the bad sector. Hold it a minute though, we're not done yet. DUMPX won't record any of our handy-work onto the disk until we tell it to. So, when you're satisfied with your directory FCB, type a "." and wait for the "EDIT-" prompt. Then type "WRITE" and that's that.

Any bad sectors created in the future can be taken care of simply by upping the record count in bad.sec's FCB and adding the offending group to the FCB disk allocation bytes. (Or by discarding the disk!)

Whew! Finally done. (It's really quite a routine.)

Some Additional Thoughts

CP/M can assign attributes to any files on a disk. (These attributes include read only and system status.) If you want your bad.sec file to be R/O, set bit 7 of the first filetype byte to 1 (e.g. change 73 to F3 in the above example). If you want the dummy file to have system status (so it won't appear in the directory), set bit 7 of the second filetype byte to 1 (e.g. change 63 to E3 above). It's best (unless you really know what you're doing) to leave the rest of the high bits in the filename set to 0.

If your system operates with double-density disks, then hopefully the above discussion will still shed some light on disk-space allocation. By examining the source of your CBIOS the whole thing should become crystal clear (well, at least translucent).

A good reference for disk space allocation in CP/M is the book *Inside CP/M: A Guide for Users and Programmers* by David Cortesi (Holt, Rinehart and Winston, 1982). This book is an excellent, overall reference for almost all aspects of CP/M (including the technical ones, such as disk parameter headers and disk parameter blocks). The 275-page appendix alone is worth the purchase price. (The book has 570 pages.)

Cheap Reads

You can use PFM to modify the directory if you don't have DUMPX. See "Undoing the Fatal Erase" on page 10 of Micro C issue #4. WARNING!! When you're through with the write command MAKE SURE to change byte F72F back to 88. (This is best accomplished by hitting the reset button.) If you don't make this change, the results could truly be disastrous—unfortunately I am speaking from experience.

■ ■ ■

256K in Detail—Part II

By Art Boehm

2000 29th Ave NW
New Brighton, Minn 55112
612-633-9292

The following covers the kinds of software needed to support 256K on the BB I. This includes bank select, of course, and it also includes ways to use the extra memory, such as disk track buffering.

Bank Select

To use the extra storage, you'll have to write your own code to do the bank switching; first set up the System PIO so that Port A bits 5 and 4 are outputs (without changing the other bits):

```
syspio  equ     01c         ;The System PIO Port Number
bitdat  equ     syspio      ;Data Port A
bitctl  equ     syspio+1    ;Control Port A
                            ;
setup:
  di                        ;Assure an indivisible operation
  ld      C,bitctl          ;Put the control port number in C
  ld      A,0CFh            ;Load mode control word
  out     (C),A             ;Put port A in Mode 3
  ld      A,008h            ;Load I/O selection control word
  out     (C),A             ;Make all but bit 3 outputs
  ei                        ;Allow interrupts

Then you can switch to any of the four banks with a simple:

switch:
  di                        ;Assure an indivisible update
  in      bitdat            ;Get the current data value
  and     0CFh              ;Clear the current bank value
  add     0#0h              ;Add in the new bank # ("#"= 0-3)
  out     bitdat            ;Replace the data in the port
  ei                        ;Allow interrupts
```

You might want to use the above code to initially select a bank (like 'bank 1', the one PFM initially selects) before changing the PIO control words with setup.

Disk Cache

It might be interesting in closing to mention that the thing that led us to search for more storage on the Big Board was not the immediate need for more execution storage but rather a way to beat the slow disk I/O. After all, a million clock cycles go by at 4 MHZ while the disk goes around trying to find the record you just read or wrote.

One solution is a solid state disk, but it is expensive and only speeds up that disc.

So, we are implementing a software disk cache in the extra storage that works roughly as follows: Every time there is a request to read a disk sector, we will check to see if the track containing it is resident in the cache. If it is, we will locate and transfer the sector from the cache to the program and return control. If the track containing the sector is not resident, we will read it from the specified disk.

Every time there is a request to write a disc sector, we will check to see if the track containing it is in the cache; if it is, we will update the sector in the cache; in either case, we will schedule the sector update on the specified disk.

Interrupts

Scheduling is done using CTC 0 and 1 as a millisecond timer to run an interrupt driven event list that pops up the next thing to be done just before it needs to be done. The things that get scheduled for a sector update include power on (if it was off), track seek, record ID read, and then sector write.

Between each of these steps, there is a fixed or computable delay of potentially long duration during which control is returned to the program that originally generated the I/O request. If more I/O requests are generated, they are processed in the cache and queued to the disk if necessary (up to some practical limit) using the event list.

The number of tracks that may be buffered in the cache is limited only by the available storage. At under 4 KB per track (including descriptor info), two 48K banks of storage could hold 24 of the 154 tracks in a two disc system. We estimate that half that many would easily cover the number of active tracks typically in use, but we will do some checking to be sure. If necessary, we could switch to buffering eight-sector granules to beat the statistics.

To manage the cache, we will use a modified least-recently-used algorithm to decide which track to eliminate from the cache to make room for a new one when the buffer is full. (The directory tracks will get more sticking power.)

If you are wondering what happens when you switch a disk, remember that we are using a trickle-through write algorithm, so that by the time you remove a disk, it should be fully up to date. (You wouldn't remove a disk from a drive while it was being written on, would you?).

That only leaves the cache invalidate problem (i.e. marking the buffered tracks of the removed disk as invalid).

Fortunately Shugart has come to our rescue, since by connecting the optional "DC" jumper on the disk board, interface pin 12 will go true (LOW) when you select the drive—if the door has been opened since the drive was last selected. All you have to do is connect pin 12 to another open pin on the system PIO, and you can check it before every disk operation.

Rolling Your Own

If it also isn't obvious, the disk cache requires a bit of modification to PFM, so it may not be for everyone. There are also as many cache replacement stratagies as stars in the sky. Also once you have the event list scheduling algorithm and extra storage, you have everything you need for a printer symboint; and if you run out of room in storage, and you want to get fancy, your symboint can use the disk.

The point is simple: there are a infinite number of things to do with the big board. The optional PIO is another target to which you can add a dayclock, a pair of proportional joysticks, a 4 channel analog switch (2X, 2Y), an op amp, a voltage reference, and an A to D converter, right? Or you could add a Centronics parallel interface with a Z8068 FIFO buffer (if you can afford the AMP 57 connectors) and include an auto-dialer (e.g. National TP53130) for a modem, a sound chip (more analog stuff), a TI speech synthesizer chip, digital and analog sensor I/O, and an EPROM programmer, to name a few.

(continued next page)

BB II Printer Interface

By John Taylor

Taylor Electric Co
1000 W Donges Bay Road
Mequon WI 53092

There have been quite a few questions from our customers on interfacing printers to the BB II. The A serial port comes set up as the printer port with 9600 baud, 7 data bits, 1 stop bit, and odd parity. It recognizes XON and XOFF. But that's it.

If you want to permanently change these parameters, you will find the initialization code in the file called ONSECT.MAC supplied on your CBIOS disk. You can also change the baud rate by running a small COM file. Just assemble and load the program in Figure 1, setting the value of BAUD to whatever you need.

If you want to change the printer driver from the serial port to either of the parallel ports, check the listing in Figure 3 and the cable wiring diagram in Figure 2.

This parallel driver uses bit 7 of the output port as a software controlled strobe. The hardware strobe line will not work with most printers without modification. Using bit 7 for strobe means that you can't use it to select an alternate character set on more printers.

The parallel driver is shorter than the serial driver, so it may be hand patched into the CBIOS in place of the serial code.

Figure 3 - Parallel Output Listing (on page 10)

■ ■ ■

Figure 1 - Baud Rate Routine

```
BAUD    EQU     1200    ; set to desired baud (300 min)
        ORG     100H
START   MVI     A,47H   ; set CTC to timer mode
        OUT     89H     ; CTCB0 = A port baud generator
        MVI     A,38400/BAUD ; baud rate time constant
        OUT     89H
        RET
```

Figure 2 - Printer Cabling for Big Board II Parallel Port

```
          BB II
PARALLEL CONNECTOR                PRINTER CONNECTOR

Data 0     1 ----------------- 2    Data 0
Data 1     3 ----------------- 3    Data 1
Data 2     5 ----------------- 4    Data 2
Data 3     7 ----------------- 5    Data 3
Data 4     9 ----------------- 6    Data 4
Data 5    11 ----------------- 7    Data 5
Data 6    13 ----------------- 8    Data 6
Data 7    15 ----------------- 1    /Strobe
ODAV      17 (N/C)
/TAKE     19 ----------------- 11   Busy

Ground     2 ---------------/- 25   Ground
Ground     4 --------------/

Ground     6 ---------------/- 27   Ground
Ground     8 --------------/

Ground    10 ---------------/- 28   Ground
Ground    12 --------------/

Ground    14 ---------------/- 28   Ground
Ground    16 --------------/

Ground    18 ---------------/- 29   Ground
Ground    20 --------------/

                              |-- 9    Data 7
                              |-- 30   Ground
```

NOTES: This has been tested with the OKIDATA 82A. The printer's /BUSY output must go high whenever the printer is unable to accept data (deselected, paper out, or buffer full). This configuration ties the printer's data bit 7 to ground. On some printers, this line must be connected to +5 instead to select the normal alphabetic characters. Graphics or alternate character sets controlled by data bit 7 are not supported by this implementation.

(256K in Detail - Part II continued)

Just Before My Time Runs Out

If you really want a good timer at 4 MHz, make CTC0 a timer with a prescaler of 16 and a count of 250; it then outputs 1 ms pulses. If you then strap CTC 0's output to both CTC 1 and 2's input, and strap CTC 2's output to CTC 3's input, and you make CTC 1, 2, and 3 all counters, and load CTC 2 with 100, then CTC 1 becomes a programmable millisecond timer (1-256 ms), and CTC 3 becomes a programmable tenth of second timer (0.1 to 25.6 seconds). Then the final touch is to add an OKI MSM5832 Dayclock chip (i.e. YY MM DD HH MM SS) with battery backup to the optional PIO, or whatever, so that you always know what time it really is.

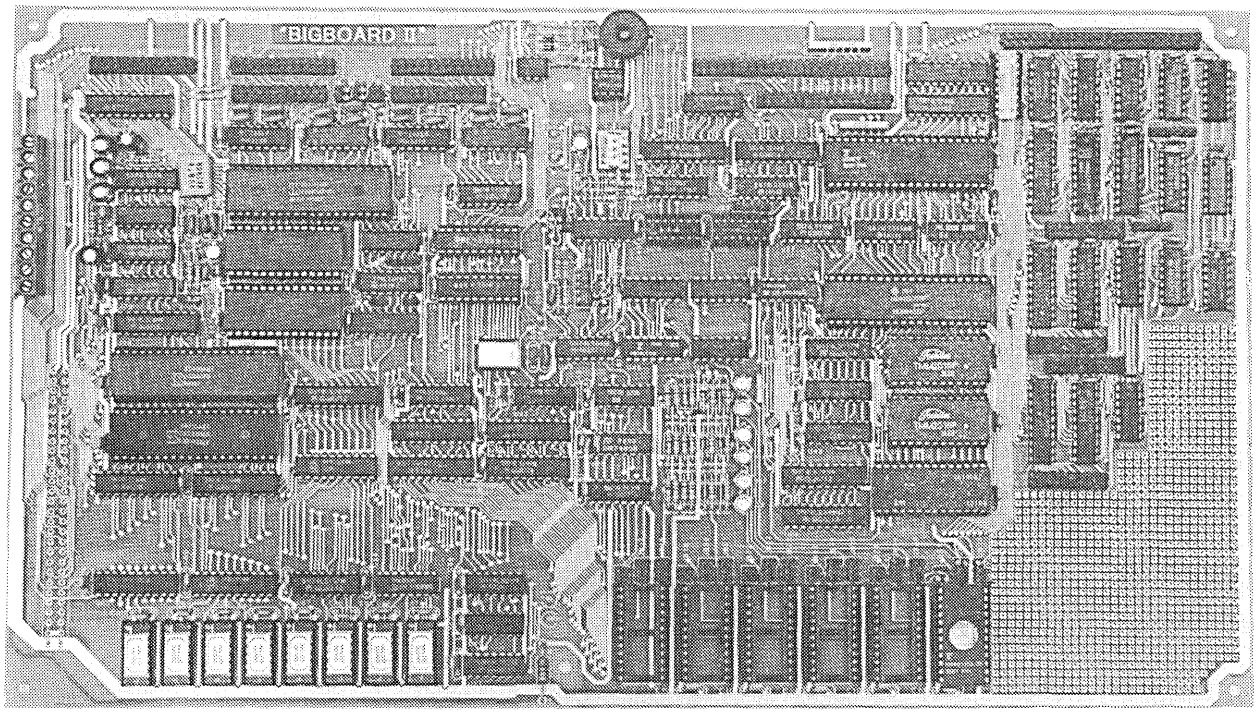
Editor's note: These articles by Art Boehm, plus Trevor Marshall's article in Issue #11 represent a real breakthrough for the BB I. These two fellows have opened a whole new area for hardware and software design. Have at it folks, and keep us posted. The next 12 months should see some very interesting developments on the BB I front.

■ ■ ■

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.

BB II PRINTER INTERFACE (continued)

Figure 3 - Parallel Output Routine

```
                        2 ;
                        3 ; Parallel Centronics output routine for Better Board CP/M.
                        4 ; (Equates set up for hand patching into distribution CBIOS).
                        5 ;
                        6 ; This code is released to the public domain 01/17/83 by:
                        7 ;
                        8 ;         Taylor Electric Co
                        9 ;         Computer Services Division
                       10 ;         PO Drawer 11N
                       11 ;         Milwaukee, WI  53201
                       12 ;         (414) 367-5123
                       13 ;
                       14 ; Version 1.00   821027
                       15 ;
                       16 ; NOTE: MSB is used as strobe, BUSY is attached to /TAKE/
                       17 ;       (BB Pin 19), ODAV is not used (BB Pin 17) and /OE/
                       18 ;       is grounded (BB Pin 20).
                       19 ;
0000                   20 FALSE:    EQU      0
FFFF                   21 TRUE:     EQU      .NOT.FALSE
                       22 ;
0000                   23 O2:       EQU      FALSE       ; true=J11 false=J9
FFFF                   24 O3:       EQU      .NOT.O2     ; output port 3?
                       25 ;
                       26 ; NOTE: The following addresses are the starting locations for
                       27 ;       the LIST and LISTST routines in the CBIOS.  They may
                       28 ;       be different for your version.  In any event, they can
                       29 ;       be located by examining the CBIOS jump table.  The jump
                       30 ;       at CBIOS+0FH goes to LIST and the jump at CBIOS+2DH goes
                       31 ;       to LISTST.
                       32 ;
E75A                   33 STATADDR: EQU      0E75AH
E769                   34 LISTADDR: EQU      0E769H
                       35 ;
                       36 ; ================================================
                       37 ;
E75A                   38           ORG      STATADDR
E75A DBD9              39 LISTST:   IN       A,(STATPORT) ; read status port
E75C 2F                40           CPL                   ; invert byte
E75D E604              41           AND      RDYFLAG      ; check /ODAV/
E75F 3EFF              42           LD       A,0FFH       ; return 0FFH if ready
E761 C8                43           RET      Z            ; 0 means ready
E762 AF                44 .NOTRDY:  XOR      A            ; return 0 if not ready
E763 C9                45           RET
                       46 ;
E769                   47           ORG      LISTADDR
E769 CD5AE7            48 LIST:     CALL     LISTST       ; check for printer ready
E76C B7                49           OR       A
E76D 28EB              50           JR       Z,LISTST
                       51 ;
E76F 79                52           LD       A,C          ; move character to A
E770 CBFF              53           SET      7,A          ; leave strobe off
E772 D3DB              54           OUT      (DATAPORT),A
E774 CBBF              55           RES      7,A          ; set strobe
E776 D3DB              56           OUT      (DATAPORT),A
E778 CBFF              57           SET      7,A          ; ..and turn strobe off again
E77A D3DB              58           OUT      (DATAPORT),A
E77C C9                59           RET
                       60 ;
                       61 ; ================================================
                       62 ;
00D8                   63 GENPIO:   EQU      0D8H
00D9                   64 STATPORT: EQU      GENPIO+1
                       65 ;
                       66           IF       O2
                       67 DATAPORT: EQU      GENPIO+2
                       68 RDYFLAG:  EQU      80H          ; /ODAV2/ = bit 7
                       69           ENDIF
                       70 ;
                       71           IF       O3
00DB                   72 DATAPORT: EQU      GENPIO+3
0004                   73 RDYFLAG:  EQU      04H          ; /ODAV1/ = bit 2
                       74           ENDIF
```

END

Mini Front Panel

By Peter Diemer

104 Sharlene Road
Ithaca, NY 14850

My Big Board has had some problems which cause it to crash every now and then. Until now, the only way I could tell that it was down was if it took too long to run through a procedure. Sometimes when I was doing a large assembly, that was a long time.

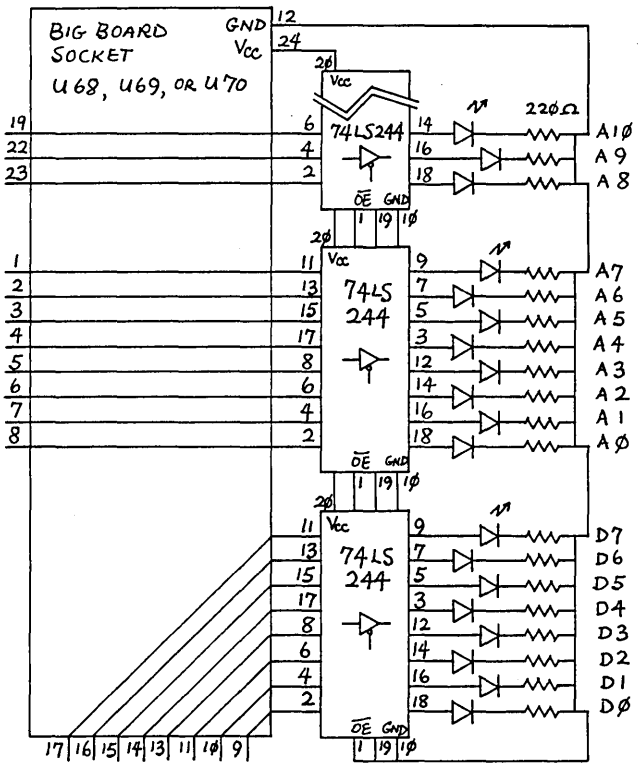
This front panel tells me immediately whether the system is running, and if it hangs, I know the status of the address and data bus.

One very easy way to access the address and data busses in the Big Board is to use one of the spare EPROM sockets U68 to U70. Such a scheme gives you A0 through A10 and all the data lines. You get better access to the address lines via the BB II ROM sockets.

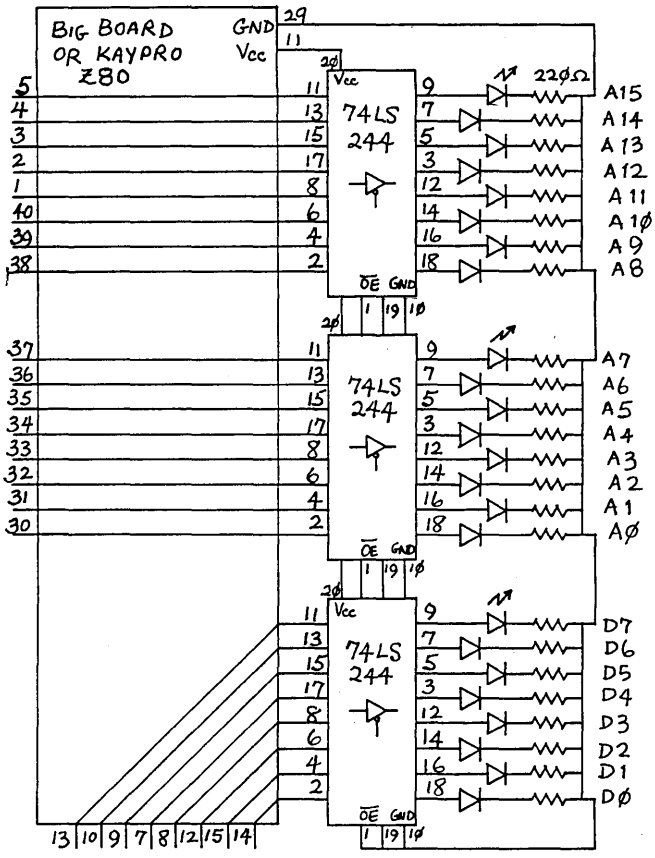
If you have a Kaypro or other system, probably the best way to tap into all 16 address bits is to remove the processor, plug a shortened wirewrap socket into the processor socket, and then plug the Z80 into the top of the whole mess. You can then use the exposed legs on the socket to solder your connections for the address and data lines.

■ ■ ■

Mini Front Panel ROM Socket Connection



Mini Front Panel Z80 Socket Connection



C'ing Clearly

Column By Tony Ozrelic

6708 Melrose
Los Angeles, CA 90038

For quite a while I was using the Code Works' Q/C compiler for most of my work. When Aztec C came out, I dropped Q/C and started using Aztec almost exclusively, making sure I kept tabs on Q/C's refinements and rationalizing Aztec's speed and size problems by saying, 'That's the way it's gotta be for a full blown compiler....

Q/C version 3.0 has just changed my mind. It still doesn't have floats or longs, but Jim Colvin has done an admirable job of putting the rest of the features of C into his compiler. I compiled an 8k C program with it and got a fast 15k .COM file for my troubles; Aztec got me a slower 20k .COM file.

I'm beginning to wonder how much more I'm getting for an extra hundred or so dollars (plus $50/yr support). I wonder when somebody's gonna come out with a $29.95 C compiler a la JRT Pascal?

Of course, it would have to do the full C spec as laid out by Kernighan & Ritchie, plus it would have to come with its own assembler/linker (or better yet, just make a direct .COM file). I know there're some brains out there (just look at all the public domain C code), but are there any guts? Guts are what it takes to become Rich And Famous, although brains do help. Doing it better and cheaper has made more people wealthy than any original thought in the world. Henry Ford invented nothing original—he just put already existing concepts to work.

Just a few more thoughts before I get off my soapbox—I hope whoever is writing the next C compiler will put two switches in their compiler just for me: an "s" switch to force the compiler to produce a small object file, regardless of speed, and an "f" switch to generate the fastest possible program, regardless of size.

No Communication

Faced with several hundred K of files to transfer between my BB and a TRS-80 model I and having only three wires to connect between them (RS-232 rec, xmit, and ground), the thought of having to write some huge assembly language program for file transfer filled my heart with nameless dread.

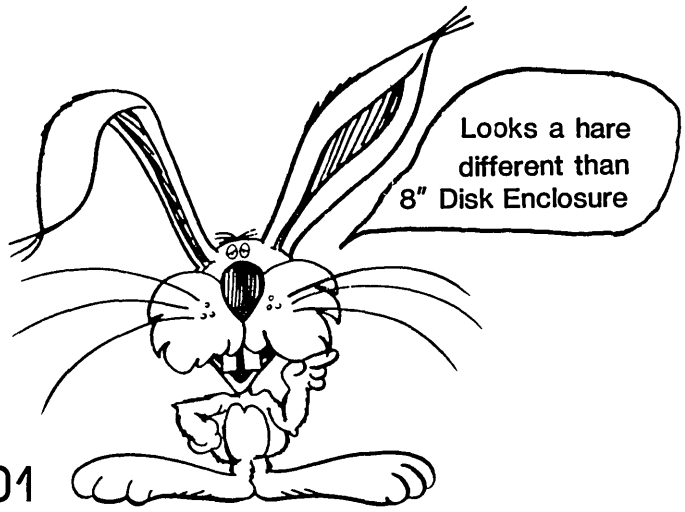
(continued on page 14)

LOW LEVEL DATA TRANSFER ROUTINES

```
;       PREC - A PRIMITIVE RECEIVE PROGRAM FOR CP/M
;       reader/punch I/O.
;
        org     100h
;
;       equates
;
dest    equ     9000h   ;destination of tiny loader code
conin   equ     1       ;bdos function for console input
conout  equ     2       ;console output
rdrin   equ     3       ;hi speed reader input
punout  equ     4       ;hi speed punch output
conio   equ     6       ;console output
bdos    equ     5       ;bdos function call address
;
;       Let's move the loader code up to dest, where it will be
;       out of the way - we'll load data from the other computer
;       starting at 100 hex.
;
beg:
        mvi     h,0     ;h is the counter (255 bytes max loaded)
        lxi     b,source;b=source address
        lxi     d,dest  ;de=destination
loop:
        ldax    b       ;a=*bc
        stax    d       ;*de=a
        inr     h       ;h++
        mov     a,h     ;is h==0?
        cpi     0
        jnz     loop    ;no, keep going till 256 bytes moved
;
;       Now that the move has been done, let's wait for a sync byte
;       from the other computer
;
sync:
        mvi     c,rdrin ;set up c for a call to the bdos
        call    bdos    ;get reader input
        cpi     0fh     ;is it a sync byte?
        jnz     sync    ;no
        mvi     c,punout;yes, send it back
        mvi     e,a
        call    bdos
        jmp     dest    ;now let's go load the data
;
;       This little loader picks up a byte from the reader device,
;       stuffs it in memory, and if the 'q' key has been pressed,
;       bails out to CP/M, where you can do a SAVE NB FILENAME.EXT
;       where NB is the number of 256 byte blocks, and FILENAME.EXT
;       is the file you want it saved under
;
source:
        mvi     c,rdrin                 ;get a char from the other computer
        call    bdos
        lhld    dest+(buf-source)       ;get pointer to storage area
        mov     m,a                     ;store character
        inx     h                       ;bump hl
        shld    dest+(buf-source)       ;store updated pointer
        mvi     c,conio                 ;look for a 'q' from the keyboard
        mvi     e,0ffh
        call    bdos
        cpi     'q'
        jz      0                       ;yes, all done, go back to CP/M
        mvi     c,punout                ;no, send byte back and get another
        call    bdos
        jmp     dest
buf:
        dw      0100h           ;pointer for storing data - starts at 100 hex
;
        end     beg
;
```

(C Routines continued on page 14)

SINGLE BOARD COMPUTER CHASSIS $275



2901

SINGLE BOARD COMPUTER CHASSIS — NON S100 — 2 EIGHT INCH DRIVES — REGULATED PS

Cabinet size: 20"w × 22.4"d × 7.5"h. Cabinet painted dove grey; front, back and side accents black. Mounts 2 SHUGART SA851R EIGHT INCH FLOPPY DISK DRIVES or MECHANICAL EQUIVALENT*. Lighted reset switch on front panel. Lamp supplied, but not connected. 4 DB25, 1 Delta 50, 1 Centronics connector cutouts, two switched ac receptacles, 6' three wire line cord, EMI filter, 70CFM fan with washable dust filter, line fuse, power switch on rear panel. P2901 removable module power supply: +5@3A, +12@5A, −12@.5A for computer; +5@2.5A, +24@2.5A−4A peak, −5@.25A for disk drives. Includes power cable from power supply to computer (unterminated wires at computer end) and power cable to 2 disk drives (specify single or double sided drives when ordering).

INTEGRAND

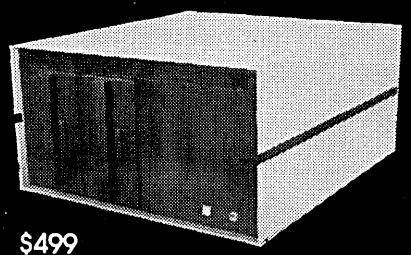
8620 Roosevelt Ave.
Visalia, CA 93291
209-651-1203

Main/Frames

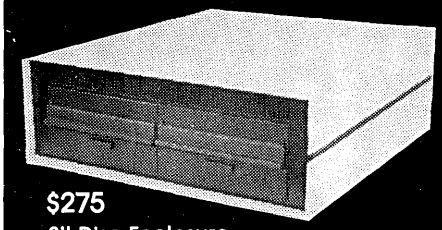
Main/Frames

from
$175

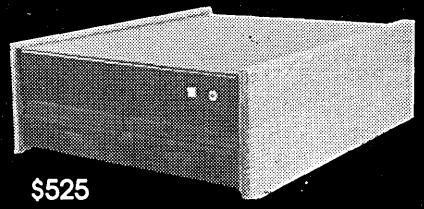
- 70 Models of Enclosures
- Assembled and tested
- Quasi-Coax Motherboards
- Power Supply
- Card cage and guides
- Fan, line, cord, fuse, power & reset switches



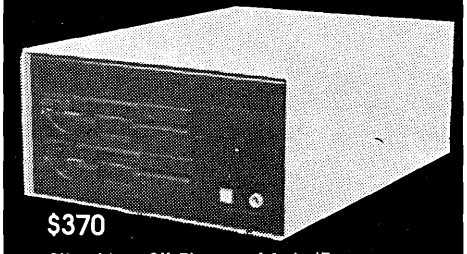
$499
8" Floppy Main/Frame



$275
8" Disc Enclosure



$525
Phase/80 8" Floppy Main/Frame



$370
Slim Line 8" Floppy Main/Frame

Write or call for our brochure which includes our application note: "Making micros. better than any ol' box computer"

INTEGRAND

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

We accept BankAmericard/Visa and MasterCharge

Since the TRS had a CP/M add-on but no software (other than ED, ASM, and PIP), I had to dope out a way to get the computers to do the work for me. A few lines of assembly code and a couple of little C programs go a long way—I can now talk to my Apple II, the TRS-80, Osborne, and Kaypro with little or no modification to the programs listed here.

The data transferred can be any size from 128 bytes to the capacity of the receiving computer's disk, and since handshaking is done byte by byte, you don't lose data while disk I/O is taking place.

Primitive Programs

The programs PSEND.C and PREC.ASM are the first step in transferring data, see Figure 1. Once you've got these going, you can send over a more sophisticated set of transfer programs.

PSEND sends a sync byte to the receiving computer, waits for an acknowledgement, and proceeds to send the file a byte at a time. At the end of the file, nulls are sent so that you can halt the other computer and save the data.

PREC is a simple loader that can be assembled by the receiving computer. The only things you need are ED, ASM, and LOAD, which already come with CP/M. The loader relocates itself into upper memory (9000h) and waits for the sync byte, sending an acknowledgement and then loading memory starting at 100h with the data from the sender. This is done until the "q" key is pressed on the keyboard, at which time control is returned to CP/M. The SAVE command is then used to save the data under a filename.

The first thing to send over is the compiled version of GREC.C. GREC is better thant PREC since it generates a checksum of the transmitted data and stops automatically at the end of the file.

The next thing to send is GSEND.C, so you can talk both ways. After this you can send whatever you like—I suggest a decent editor for starters.

Hardware Dependencies

Of course, you must have computers that have the hardware to communi-

(continued next page)

```
/*      PSEND - A PRIMITIVE SEND FILE VIA THE SERIAL PORT       */

#include "qstdio.h"

#define BELL    7               /*ASCII bell                            */
#define RDRIN   3,0             /*bdos number for reader                */
#define PUNOUT  4               /*bdos number for punch                 */

main(argc,argv)
int argc,argv[];
        {
        char buf[128];
        char c;
        int *file,i,j,k;
                                /*check to see if file has been         */
                                /*specified and if it can be opened     */
        if(argc!=2) {
                printf("? Usage:\npsend <file>\n",1);
                exit(0);
                }
        if((file=open(argv[1],0))==ERROR) cantopen(argv[1]);

        puts("\nprimitive CP/M file sender. Waiting for sync.");

                                /*send a sync byte and wait for a response*/
        while(1) {
                bdos(PUNOUT,0x0F);
                if(bdos(RDRIN)) break;
                }

        puts("Link established. Sending Data.");

                                /*read a 128 byte block                 */
                                /*and send it a byte at a time          */
        k=0;
        while(1) {
                i=read(file,buf,1);
                k+=i;           /*keep track of how many blocks sent    */
                if(i<=0) break; /*if i!=1, error or end of file         */
                for(j=0;j<128;j++) {
                        c=buf[j];       /*fetch, send, wait             */
                        bdos(PUNOUT,c);
                        c=bdos(RDRIN);
                        }
                }
        close(file);
        printf("No. of blocks sent=%d\n",k/2);
        puts("Run over to the other computer and hit the 'q' key, then");
        puts("type SAVE NB FILENAME.EXT, where NB is the number of blocks,");
        puts("FILENAME is the name of the file, and .EXT is the");
        puts("extension (.COM, .TXT, etc.)");
        puts("hit any key when ready");
        c=getchar();
        for(i=0;i<1000;i++) {           /*send some nulls to give poor  */
                bdos(PUNOUT,'\0');      /*human time to push button     */
                c=bdos(RDRIN);
                }
        putc(BELL);
        }



/*      GREC - A GENERIC RECEIVE FILE VIA CP/M's RDR:/PUN: DEVICES */

#include "qstdio.h"

#define BELL    7                       /*ASCII bell                    */
#define RDRIN   3,0                     /*bdos call to CP/M reader      */
#define PUNOUT  4                       /*bdos call to punch            */


main(argc,argv)
int argc,argv[];
        {
        char buf[128];
        char c;
        int *file,i,checksum,j;
                                        /*check for filename and see if */
                                        /*it can be opened              */
        if(argc!=2) {
                puts("? Usage:\ngrec <filename>");
                exit(0);
                }
        if((file=creat(argv[1],0))==ERROR) cantopen(argv[1]);
```

```
        puts("\nCP/M file receiver. Waiting for sync.");

                                        /*wait for sync character       */
        while(1) {
                c=bdos(RDRIN);
                if(c!=0x0F) continue;
                bdos(PUNOUT,c);
                break;
                }
        puts("Link established.");
        puts("File fetch in progress.");
        puts("Sit back and relax. I'll buzz you when I need you.");

                                        /*let's read in some bytes      */
        checksum=0;
        while((c=bdos(RDRIN))=='y') {   /*1st byte should be 'y' for yes*/
                bdos(PUNOUT,c);         /*echo it back                  */
                for(j=0;j<128;j++) {    /*collect 128 bytes             */
                        c=bdos(RDRIN);  /*fetch, store, sum, send       */
                        buf[j]=c;
                        checksum+=c;
                        bdos(PUNOUT,c);
                        }
                write(file,buf,1);      /*write 128 bytes to disk       */
                }
        printf("Checksum is %x - should equal other checksum\n",checksum);
        close(file);
        putchar(BELL);
        }

/*      GSEND - A GENERIC SEND FILE VIA THE SERIAL PORT         */

#include "qstdio.h"

#define BELL    7                       /*ASCII bell character          */
#define RDRIN   3,0                     /*bdos call parameters for reader*/
#define PUNOUT  4                       /*and punch                     */
main(argc,argv)
int argc,argv[];
        {
        char buf[128];
        char c;
        int *file,i,checksum,j;

                                        /*check for filename and see if */
                                        /*it can be opened              */
        if(argc!=2) {
                printf("? Usage:\ngsend <file>\n",1);
                exit(0);
                }
        if((file=open(argv[1],0))==ERROR) cantopen(argv[1]);
        puts("\nCP/M file sender. Waiting for sync.\n");

                                        /*send sync byte and wait for reply*/
        while(1) {
                bdos(PUNOUT,0x0F);
                if(bdos(RDRIN)) break;
                }
        puts("Link established. Sending Data.\n");

                                        /*now let's send the file       */
        checksum=0;
        while(1) {
                i=read(file,buf,1);     /*read 128 bytes                */
                if(i<=0) {              /*if all done, send an 'n' for  */
                        bdos(PUNOUT,'n'); /*no more blocks and exit     */
                        break;
                        }
                else {                  /*not done, send 'y' for yes, more!*/
                        bdos(PUNOUT,'y');
                        bdos(RDRIN);
                        }
                for(j=0;j<128;j++) {    /*send 128 bytes                */
                        c=buf[j];       /*fetch, sum, send, wait        */
                        checksum+=c;
                        bdos(PUNOUT,c);
                        c=bdos(RDRIN);
                        }
                }
        printf("Checksum is %x - checksums should be equal\n",checksum,2);
        close(file);
        putc(BELL);
        }
```

cate—there must be a reader/punch combination that is compatible, i.e. both serial ports or both parallel ports. Baud rate must be the same, as well as the number of bits transmitted (don't forget, you'll need 8 bits of data!).

Parallel logic levels need to be the same. Since transmit and receive are sometimes on different pins on serial ports, check with a voltmeter that you aren't hooking up transmit to transmit and receive to receive (if a voltmeter shows the same voltage on the same pins, you must swap the pin connections around).

Software Dependencies

Since the software is written in C, there isn't much to change; the programs were written for the Q/C compiler and so you must change the 'qstdio.h' standard I/O header to whatever your C compiler understands. Also, the low-level system I/O routines (read, write, etc.) may be a little different; Q/C expects to read and write 128-byte blocks and standard C takes 1-byte reads and writes.

Closing Comments

Some of you might ask, 'Why not use MODEM7 or some other program?', and I say, How do you get it to the other computer? If you have disk problems (different formats, 5¼ vs. 8 inches, etc.), what do you do?

If these programs seem simple, it's because they are. They are bare bones versions and can stand some bells and whistles. For instance, you could add multiple file transfers (GSEND *.COM, for example), faster transfer (do handshaking only at beginning of 128-byte block transfer), bigger buffers, checksum by block and retransmission of a block if an error is encountered, echo text to console—all kinds of stuff. Have fun. Send me your version and the fanciest will be published in this here column . . . (Hear, Hear!)

■ ■ ■

FORTHwords

Column by Arne A. Henden

7415 Leahy Road
New Carrollton, MD 20784
(301) 552-1295

I'm afraid this column is going to be more of an editorial than a tutorial. To be honest, I've been busy, with no time for developing any small applications. So instead, I have a few words to say about 16-bit microprocessors and their use for FORTH.

Which FORTH Chip Is First?

One of the reasons I got into the FORTH business was my unending fascination with state-of-the art microprocessors. People who work with me at Goddard are used to my daily tirades about the latest chips. One of the problems with keeping up with the industry is that software always lags far behind. With FORTH, however, I can create a new system using the crosscompiler and be playing on a development board in just a few weeks.

Therefore, I've implemented FORTH on most of the new generation CPU's. Some turn out to be good FORTH chips and others stink. My scientific biases are certainly present in answering the question I raised at the beginning, so take the following words with a shaker of salt.

Listed in Table 1 are the 16-bit chips on which I have implemented (at least on paper) a version of UNIFORTH, along with a Z80 fiducial. I won't consider any other 8-bit chips as they could easily comprise another full table. The ratings given are: A=excellent, B=good, C=fair, D=poor, relative to the rest of the listed chips. The paragraphs below detail each of the categories.

Category	Z80	Z800	LSI-11	Z8000	8086	80286	68000	16032
Multiple stacks	D	D	A	A	C	C	A	C
Hardware multiply/divide	D	C	A	A	B	B	A	A
Addressing modes	D	C	A	B	C	C	B	B
General purpose registers	D	C	B	A	C	C	A	B
String operations	B	B	C	B	B	B	C	A
Address space	64k	512K	4M	8M	1M	16M	16M	16M
Addressing implementation	LIN	SEG	SEG	SEG	SEG	SEG	LIN	LIN
Multiuser considerations	D	C	D	A	D	B	A	A
Small task support	A	A	A	A	A	A	C	C
Processor speed (MHz)	8	10	3	6	10	10	12	6
Processor pipelined?	N	Y	N	N	Y	Y	N	Y
Floating point processor	9511/12	9511/12	FIS/FP11	8070	8087	80287	68881	16081
Basic FPU architecture	D/C	D/C	C/B	A	B	B	A	A
Transcendental functions	C/D	C/D	D	C	B	B	A	D
Conventional O/S support	B	B	A	D	B	B	C	D
Language/applications	A	A	A	D	B	B	C	D
Literature	A	B	B	B	A	B	B	D

Table 1 - Subjective Comparison of Microprocessors

A=Excellent *C=Fair*
B=Good *D=Poor*

Device Attributes

Multiple stacks are very important for FORTH. Besides the data stack, you always need a return stack. Floating point/string operations often are implemented with separate stacks, and the interpreter pointer is a stack in disguise. The ratings in this category are very important to a clean and fast FORTH implementation.

You need at least 16-bit multiply and divide. The Z8000 even gives 32x32=64 bit! Emulating 16-bit multiplies on the Z80 takes some 500-1000 clock cycles, about 10x slower than microcode.

Addressing modes are very important. For example, to fetch the contents of a variable whose address is on the data stack top is a single instruction with the LSI-11.

You don't need 40 general purpose registers, but FORTH uses quite a few. Most implementations require at least 5: data stack, return stack, user pointer, interpreter pointer, and a working register. In addition, the 32-bit CPUs require an offset register. Several registers need to remain for users or else you will have overhead when saving and restoring FORTH registers.

String operations are not only used for comparison but also for block moves of text and lower/upper case translation. You can implement string operations on the LSI-11 and 68000 using the post-increment/pre-decrement addressing modes but not nearly as straightforwardly or as quickly as the 16032 primitives.

Standard FORTH demands 16-bit addressing, but for some FORTH applications such as image processing or large floating-point arrays, more memory space is essential. Of course, the linear addressing processors are most efficient; the worst is the poor memory management on the LSI-11.

For multitasking/multiuser FORTH, you need a quick way to save the processor state (status and registers). Related to this is the interrupt handling capability. Small task support refers to how easily it is to implement a 64K task space with 16-bit addresses for memory conservation (often needed for dedicated controller applications).

Clock rate does not tell all for these processors. The rates shown are for chips that you can currently buy from mail- order houses. Chips such as the 80286 perform numerous operations in parallel, so at a set clock rate, the 80286 outperforms any other chip.

Pipelining means that instructions are fetched during the execution of the previous instruction; this multiplexing usually halves the access time and allows slow RAM to be used with a fast processor. The Z800, for instance, allows the internal processor clock to be much faster than its memory cycle clock so a 10MHz part can use 2MHz dynamic memory.

To me, floating point support is very important. The 9511 is slow and has limited dynamic range; the 9512 has double precision arithmetic but no high-level functions. The 8070 and 68881 won't be in production until at least next year. The 8087 has partial transcendental functions that require about 2000 lines of code to fully implement. The FP-11, 8070, 68881 and 16081 have 8 registers; the 8087 also has 8, but they are really stack-oriented.

Of the processors listed, the 68881 sounds the best, as it not only has transcendental functions but also has branching based directly on the FPU's status. I'm happy to see that manufacturers are finally starting to support technical uses of their chips!

(continued next column)

Operating Systems

The final categories refer to conventional operating system support. Are reasonably priced, full function systems currently available? Are the major systems (such as MSDOS, CP/M or UNIX) available so that you can transport software? What about application programs like word processors? Finally, at least an Osborne-like book on assembly language programming should be available.

None of these categories really benchmark one processor against another. To do so, you need to compare apples with apples: same clock speed, system configuration, optimized operating system and language, and use your own application. Of the chips listed, only the Z800 and Z8000 do not have single board computers currently available for controller applications. All the processors are available on S-100 cards.

Conclusions

The 8086/80186/80286 family is probably your best bet for the present, with much hardware and software support.

The 68000/68881 pair will be hard to beat for FORTH, but the 68881 is not available yet and the software support is not here either.

The 16032 is the most powerful of the processors, but it is not a very good FORTH machine.

I like the Z8000, but the hardware and software support is really lacking.

The LSI-11 and DEC's new T-11 and F-11 chips are excellent FORTH machines but they have terrible memory management and no outside vendors.

So, I guess the answer to the rhetorical question is: there is no perfect general purpose FORTH chip among the current crop of 16-bit microprocessors. Consider your application, your programming ability, and the other uses you will have for the chip when making your decision. The 16-bit processors are all better than any of the 8-bit units, so you can't go wrong with any of them!

■ ■ ■

Pascal Procedures

Column by Rex Buddenberg

1910 Ash St
North Bend, OR 97459

First of all, in this column I'd like to mention the Pascal/Z User Group and follow that with an overview of Modula-2 a new Pascal-like language written by Niklaus Wirth (the developer of Pascal). Finally, I'm including a couple of routines which I find very useful.

Pascal/Z User Group

Charlie Foster runs a garage enterprise known as the Pascal/Z User Group. Don't let the name throw you, he doesn't get paid or subsidized by Ithaca and he has no particular company loyalty. Indeed, he has included JRT Pascal as a User Group disk.

He was the first to tell me that Jim Tyson (JRT) is writing a compiler for Modula-2—to be released in the fall for under $100. (Of course, knowing JRT, no one is taking bets on which fall.)

If you send $9.00 to 7962 Center Parkway, Sacramento, CA 95823, you'll get his bi-monthly newsletter. The newsletter will tell you about his user group offerings and how to get on the Sacramento Microcomputer Users Group (SMUG) RCPM. The library is on that system.

Modula-2

Meanwhile some folks are wondering why Wirth decided to write a new language.

Variable insulation. Pascal does a marvelous job of keeping local variables within subroutines, But any variable that must go into the main program must be declared globally—which means that ALL of the procedures can change any global variable.

Modula-2 uses actions called "import" and "export" that limit the scope of global variables. This increases the modularity of subroutines so you can swipe that neat sort algorithm from another program without having to check all of its variables against your globals. Great for team programming.

Input/Output. Wirth originally only defined data inputs and outputs for sequential devices (like magnetic tape). So, every Pascal compiler writer came up with his or her own extensions to support I/O.

Modula-2 contains NO I/O routines! None. These routines are all in library modules. You get to write your own drivers to handle disks, mice, light pens, and modems. Since you get the source of the library you can rewrite the routines to match your hardware.

String handling. In its original form, Pascal supports fixed length character arrays. This isn't a whole lot better than FORTRAN, for Pete's sake. Virtually every implementation has some string handling added; again none of it is standard. Modula-2 tries to address this.

Low level code. A great many of the Micro-C users are interested in software that lets them twiddle with the hardware.

Pascal insulates you from that hardware quite effectively. So, if you want to write a CBIOS or real time controller, Pascal is not the language of choice.

For these applications you will find yourself in C or assembly language. There are some ways to make Pascal access a particular byte of memory, but it's not easy. Modula-2 on the other hand is designed to allow "controlled low-level access" to the hardware.

Upward path. The skills you have learned in Pascal will be easily movable to Modula-2.

Getting It

Jim Tyson of JRT is, of course, writing a version. Additionally, Volitions Systems of San Diego supposedly has a Xerox 820 version on the street now. It should be easy to get running on any Big Board derivative. Logitech in Palo Alto is also working on a compiler. Meanwhile, you can pick up the Wirth book, *Programming in Modula-2*.

Figure 1

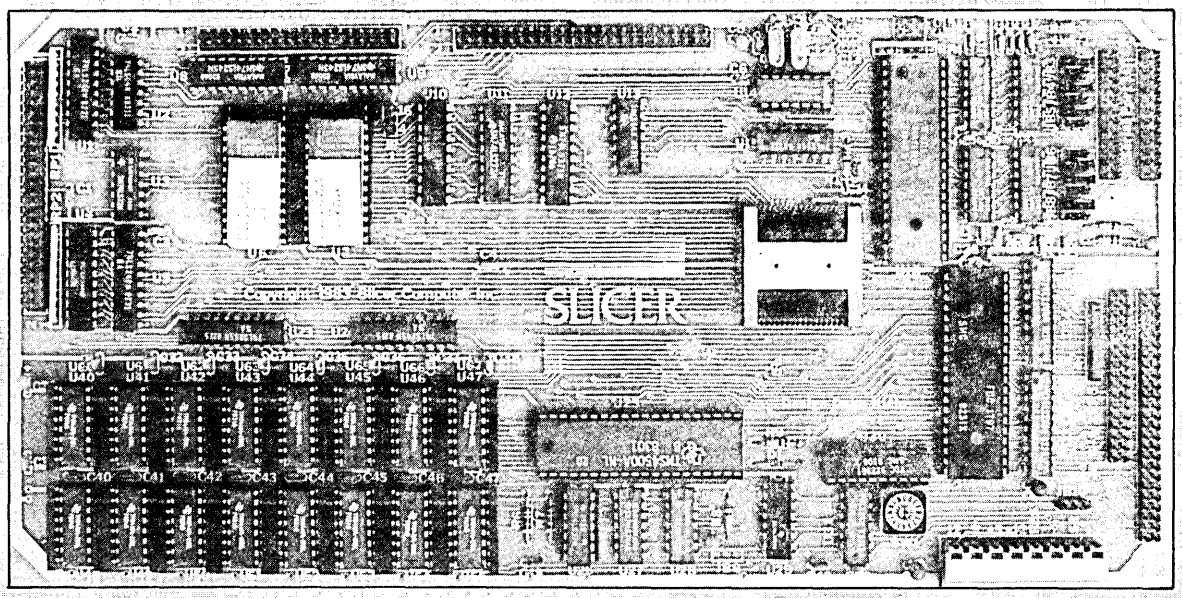
A couple of notes. These functions are part of a celestial sight reduction program that I have used extensively in my teaching of celestial navigation the past couple of years. They work fine for that application. Pascal, like BASIC, uses radian trig—multiply a value times 0.0174532925 to get degrees. These conversions came from an old North Star user group library. Pssst ... pass 'em along.

■ ■ ■

Figure 1 - Extended Trig Functions in Pascal

```
{global definition must include...}

var
    x: real;

{some coaching for those who haven't seen Pascal before:
  - curly brackets {} enclose comments (like this) that
    you don't want to compile
  - the following functions accept a real number (x:real)
    and return another real number}

function ARCSIN (x:real):real;
begin
   {arctan is built into Pascal, although some compilers
   call it "atan." }
   arcsin := arctan(x/sqrt(-x*x+1))
end; {function arcsin}

function ARCCOS (x:real):real;
begin
   arccos := -arctan(x/sqrt(-x*x+1))+1.5708
end;
```

Step up to a 16-BIT Powerhouse!



The "SLICER" is a HIGH PERFORMANCE single board computer based on the new, highly advanced Intel 80186 CPU. The board has these advanced features:

- *Full 8MHz 16-bit microprocessor having complete software compatibility with the 8086 and 8088*
- *Two full function RS232C serial ports with baud rates individually controlled by software.*
- *Baud rate for console port is acquired automatically.*
- *Floppy disk controller allows the combination of 5¼ and 8" disk drives, single or double sided, single or double density.*
- *SASI port for hard disk controller with data transfer rate of up to 2Mb per second.*
- *Memory capacity of 256KB ram on board plus up to 32KB of EPROM.*
- *90 line expansion interface with 20 bit latched address bus, 16 bit data bus and all important 80186 control signals.*
- *A high performance ROM monitor is included with all systems.*
- *BIOS for CP/M86* operating system included.*
- *Power requirements:*
  + 5 volt, ± 5% at 3 ampere max.
  + 12 volt, ± 10% at 60 milliampere max.
  – 12 volt, ± 10% at 50 milliampere max.
- *All this on a board the width of a 5¼" drive and only 11½ inches long.*
- *Sold in various kit forms from $140-$895. Assembled and tested $1,075. Quantity discounts available. Prices valid through July.*
- *Complete documentation included.*

Available Now!

This system is marketed and supported exclusively by:
***Note:* New name, address and telephone number.**

SLICER

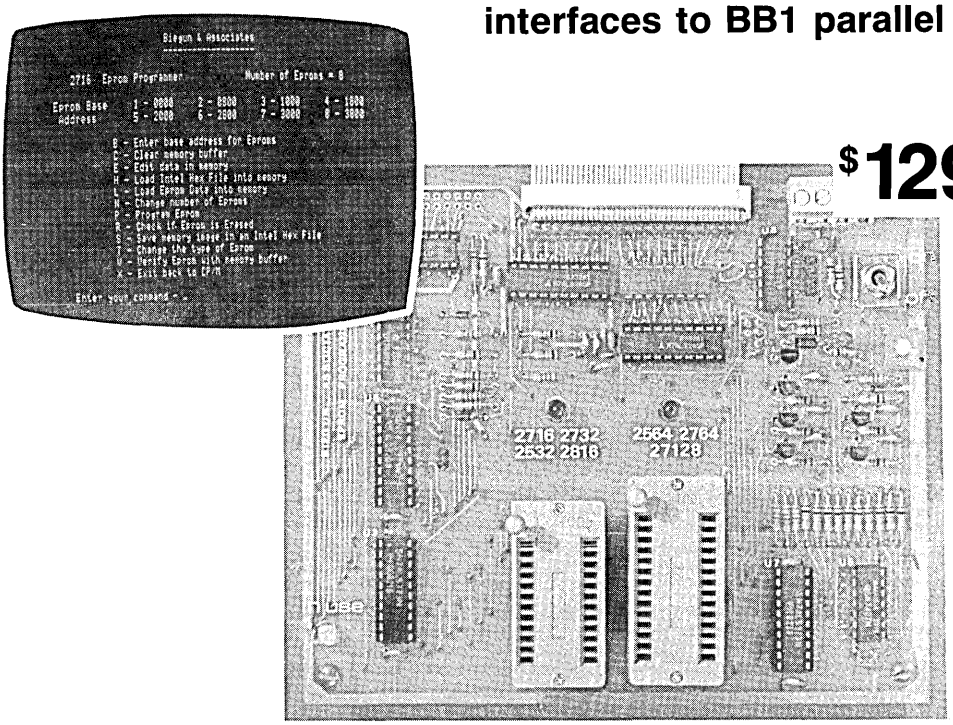
SLICER COMPUTER INC. 2543 Marshall St. N.E.
Minneapolis, MN 55418 phone (612) 788-9481

Mastercard, Visa, check, money order or C.O.D. orders accepted. Please allow 4-6 weeks for delivery.

***CP/M a trademark of Digital Research Inc.**

CP/M EPROM PROGRAMMER

interfaces to BB1 parallel port



$129.95

SPECIAL INTRODUCTORY PRICE

(offer expires Sept. 30, 1983)

features are:

- Program, Verify, Load, and check for Erased
  Intel Eproms 2716, 2732(A), 2764, 27128
  T.I. Eproms 2516, 2532, 2564
  Xicor EEprom 2816A
- 16 k byte memory buffer allows you to work with up to 8-2716 EPROMS at the same time
- Upload and Download Intel Hex Files with the memory buffer
- Edit the data in the memory buffer
- Define your own addresses for the memory buffer so you never calculate where your data is in the memory buffer
- Complete screen error messages
- Software source included
- Menu driven
- Interfaces to most Z80 CP/M systems with parallel ports and a TPA = 100H

options (available later)

- EPROM Emulator
- Adapters for single chip processors

Requires +5 v. @ 300 ma., +25 V. @ 100 ma., and interface cable

Software is delivered on a standard 8 inch SS SD floppy disk.

1. Software and schematic	29.95
2. Bareboard and schematic	39.95
3. Software and bareboard	64.95
4. Software and kit (less ZIFS)	89.95
5. Software and full kit	114.95
6. Programmer A + T	129.95
plus shipping	5.00

Still available
BB11 software and source
(uses the BB11 for programming) 29.95

all prices shown are in US Funds

Allow 4-6 weeks for delivery

Send order to:

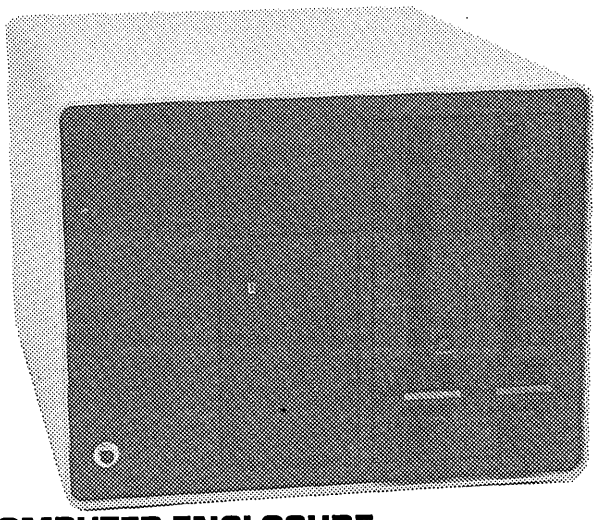
Biegun and Associates
P.O. Box 4071
Station "B"
Winnipeg, Manitoba
Canada R2W 5K8

Biegun & Associates

CP/M is a trademark of Digital Research, Inc.

FERGUSON ENGINEERING

817-640-0207



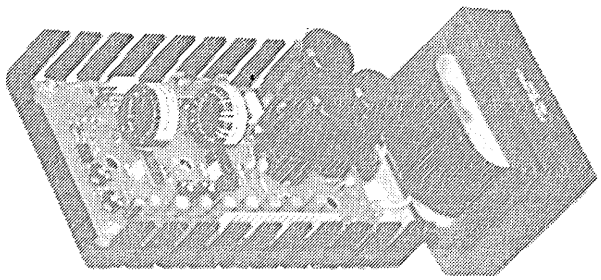
COMPUTER ENCLOSURE CE - I $195.00

All aluminum disk and computer enclosure for BIGBOARD I or II and others. Made of 1/8 inch material. An attractive chassis for single-board computers. Comes with fan, switches, power cord, and fuse holder. IBM precision tan W/black front & rear panels. 10 H x 13 W x 19 D.

Enclosure and power supply combo.

CE - I/PS $375.00

Includes UPS - 5 - 200



UNINTERRUPTABLE POWER SUPPLY

UPS 5-200 $195.00

+ 5 V @ 8.5 AMPS + 12 V @ 6.5 AMPS – 12 V @ 500 ma

+ 24 V @ 1.5 AMP + 24 V @ 1.5 AMP

This supply is designed to power entire computer systems with Winchester and floppy disk subsystems, as well as providing emergency battery backup capability. Designed to meet the needs of single-board Winchester systems. Call for pricing on the BP-1 battery pack.

FEATURES:

- High Reliability
- High Efficiency (75% TYP.)
- High Power - 200 Watts
- 25 Khz Switching Frequency (inaudible)
- Soft Start
- Built-In Quick-Response O.V.P.
- Cycle-By-Cycle Fold-Back Current Limiting
- Drop-Out Immunity for up to 250 Ms (without battery)
- Uninterruptable (with addition of BP-1 battery pack)
- 5 Output Voltages

XEROX 820 COMPUTER BOARDS

UNTESTED

$175.00

These boards are in like-new condition. Two out of three will run right out of the box. 820 is VERY similar to BIGBOARD I. Limited supply.

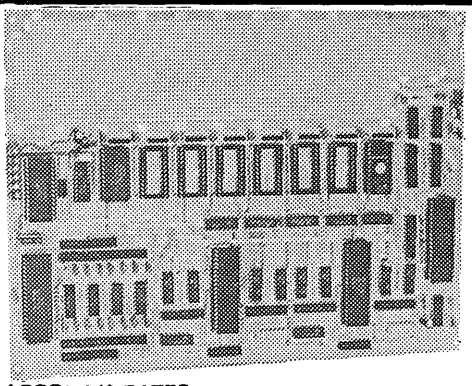
Complete 820 Documentation Package	$15.00
820 Video Cable-10 Pin Header to CRT Edge Card-1'	$ 5.00
820 CBIOS & Rom Monitor Source	$35.00
CP/M Primer (Xerox CP/M Handbook)	$10.00

BIGBOARD CABLE SETS

8" Dual Drive Cable 17"	$ 45.00
10 Pin Header to CRT Edge Card. 12"	$ 5.00
DB25S to 26 Pin Header. 18" Long	$ 25.00
AMP-8" Drive AC Power Connector W/Pins	$ 1.00
AMP-8" Drive DC Power Connector W/Pins	$ 2.00
AMP-5 1/4" Drive Power Connector W/Pins	$ 1.50
XEBEC to Hard Disk Data Cable	$ 15.00
AMP Mini-Jumpers for easy configuration strapping	$ .20
XEBEC to Hard Disk Mounting Brackets & Hardware	$ 6.00
WD-1000 Hard Disk Controller Board W/Doc. New	$250.00
TM503 15M Winchester Disk Drive. New	$750.00
ST506 5M Winchester Disk Drive, 50 Hours	$450.00
Hard to get BIGBOARD I & II Parts	CALL

If you are a user of the BIGBOARD I or II, the XEROX 820, or of the KAY-PRO, and you wish to be included in my mailing list, please send a post card with your name and address. Jim B. Ferguson

GENERAL PURPOSE INDUSTRIAL CONTROLLER



LB-I

$95.00

Bare Board & Doc

ACCOMMODATES

- Z80, Z80A, OR Z80B Series Parts
- Dual RS232 Serial I/O W/Current Loop Option on "A" Channel (both strappable as terminal or modem in RS232 Mode)
- 16 Programmable Baud Rates on each serial channel
- 32 Bits 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
- All I/O Fully Mode II Interrupt Compatible
- Watchdog Timer for Long-Term Unattended Operation

Rom monitor and source listing $35.00

FERGUSON ENGINEERING

817-640-0207

P.O. BOX 300085 ARLINGTON, TEXAS 76010

TERMS: MasterCard/VISA add 3%, money order, certified check, allow 3 weeks for personal or company checks to clear. Texas order add 5% tax. No COD. Minimum order $25.00.

Kaypro Column

By David Thompson

I got my first good look at the insides of the Kaypro 10 during the SOG. There were a couple here, and they both ran like tops.

Many of the visiting Big Board folks suspected that there might be more than casual similarity between the Kaypro 10 and the Big Board II. There is quite a bit of similarity but not nearly as much as between the Kaypro II and the BB I.

Inside the 10

The Kaypro 10 has two Z80 SIO chips, so there are 4 serial interfaces available. One interface is used for the keyboard, just like the II and 4. Two are brought out to the back panel, one marked "modem" and the other marked "printer." That totals three. The fourth port is left as pads E5 through E17 on the board.

The parallel ports have changed radically from the earlier Kaypros. The Z80 PIO is gone, and a couple of 74LS373 octal latches (U18 and U11) have taken its place. It appears that they are using a 74LS244 (U5) to handle the strobe output. The system and character ROMs are 2732As.

The video is generated by a Signetics 6545. Two 4016s make up the 4K video RAM which hold the characters and the character attributes. Each character can be underlined, reversed, blinking, reduced intensity, or underlined.

We've had problems with jitter on the output of the BB II's 6845 video controller, so getting a really good screen image has been a fight for us. I noticed that Non Linear picked a very slow phosphor for the 10—which means that the new block cursor leaves a trail as it bounces about the screen. However, a slow phosphor really cleans up a jittery image so I'm definitely not complaining. As far as I'm concerned, the ghosting of past letters is a small price to pay for a very solid image.

Graphics are generated in two-wide by four-high blocks.

The processor runs at 4 MHz and the 4 MHz is generated from a 16 MHz crystal.

The cabinet is classic Kaypro, which is a tribute to the amount of room that was left over in the original system. The winchester is quite silent but there is now a small—but fairly noisy—fan just behind it. I really appreciate the fan, it keeps things much cooler, but some very popular person is going to come out with a quieter version. There is something very nice about a really silent system when you work in as quiet a place as I do.

We will start work on a Kaypro 10 schematic and theory of operation shortly.

The Winchester

The winchester appears to go to the safety area a second or two after a disk access. So it may not matter whether you run the safety program. After all, having the head parked in a non-data zone means that power glitches can't accidentally write nasties onto the system or directory tracks (a definite no-no when the system boots off the winchester).

There is a phone connector on the back panel of the 10. It is not the handset connector but rather a standard wall type connector. Perhaps for a built-in modem connection?

More on drives

I've received a number of calls about Tandon drive alignment, two of them from Tandon itself.

According to Tandon, the drives get only a rough alignment before going to OEMs (original equipment manufacturers) like Non Linear. The OEMs are responsible for the final tweaking before shipment.

So, with that in mind, Dana and I bundled both of Micro C's Kaypros into the Civic and went tooling down to the local dealer.

All four drives were in very good alignment; we adjusted alignment on one but it was within spec. before the adjustment. The spindle speed was slightly fast on two, the other two were right on.

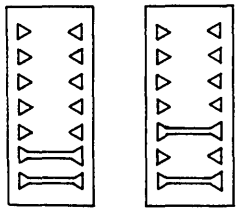
Removing the drives

On the newer Kaypros with the A: drive on top of the B: drive, you begin by removing the main circuit board. Then remove the allen screws from both sides of the metal drive shield. On the older version you don't need to remove the circuit board.

After you have disconnected the ribbon cable, the 3-wire power connector, and the ground clip from the back of each drive, you can pull each unit forward out the front of the system.

Now reconnect your A: drive to the system (it's easier to have just one drive connected at a time). Look at location 1E on the circuit board (the socket which contains the shunt). The shunt for drive A: has the bottom two connections closed and the rest broken. The shunt for drive B: has the bottom trace closed, the next open, and then the third one closed. All the rest are open. See Figure 1 for shunt information.

Figure 1 - Kaypro Drive Shunts



With drive A: connected to the ribbon cable (the red edge of the ribbon cable is pin one) and to the power connector (and ground clip), turn on the Kaypro and boot up your system disk. If the disk does not boot, recheck all your connections.

Now you can use one of the disk utilities such as DU77 on user disk K2 to step the head out to the proper track. Just load DU77, then remove the system disk, and replace it with your alignment disk (Dysan 224/2A).

Now have DU77 read sector 1 on whichever track you need. Because the alignment disk doesn't have normal information, the system should hang up while trying to read the track, occasionally homing and then retrying the track.

The instruction sheet with the alignment disk will tell you which tracks you need to select.

After you have finished aligning the first drive, remove the drive from the power and ribbon cables and swap shunts between the first drive and the second drive. That way the second drive (drive B) will look like drive A to the system. Now connect the second drive to the cables and do the alignment. Before reinstalling the drives in the system, swap the shunts again. That's so the ter-

minator (in socket 2F) will be on the last drive on the ribbon cable.

The Tandon manual does a commendable job of explaining the alignment procedure. Section 1 contains a description of the alignment process and section 2 gives you a blow by blow account of what to do. The manual is P/N 179:22-001 and covers both single-sided and double-sided drives. It is available for $25.00 from Tandon Corporation, 20320 Prairie St., Chatsworth, CA. 213-493-5965.

Broken Drive Door latches

I've heard numerous stories about drive latches that fall off in users' hands. (I've watched it happen.) I've also received one report about why it happens.

According to the report, tightening down the drives in their housing causes them to pull against the Kaypro's front panel. This causes the drive bezel to bend outwards, stressing the door latch.

Some dealers have been using hot soldering irons to "fix" the little plastic hook which breaks. They don't even pull out the drives. However, it looks like a more permanent fix is to check incoming units for slightly bowed bezels, and then adjust affected drives forward in the case to reduce the pressure.

Anyone with more information about this should get in touch.

Speeding up the II and 4

I've received a number of questions about the speed-up mods in issue 12. First, if you have a Kaypro II, you have to change the monitor ROM.

That ROM can be any 350 ns or faster part and it will be designated as a 2716-1 or 2716-H. To get the part programmed you'll need access to a ROM burner, or you can send us the part and an article or software contribution and we'll burn it for you (with our zingy new monitor besides). Otherwise, you can send us $29.95 and we'll send you our pro monitor ROM, all ready to go 5 MHz.

If you have a Kaypro 4, the monitor is already fast enough to run 4 or 5 MHz.

You might not have to change the CAS and MUXC to get the system to run, but the change even improves the RAM timing for 2.5 MHz so it wouldn't hurt everyone to make that change.

As for the Z80, all you need is a Z80B CPU. It should cost between 10 and 20 bucks at your local parts outlet.

If the system doesn't come right up, you'll probably need to locate someone with an oscilloscope, to check the clock signal you are getting to the processor.

Kaypro reverse video
Two more parallel ports

Dear Editor,

I have made a simple hardware mod to my Kaypro II which gives you black characters on a green background.

I added a 74LS27 gate (I used only 1 section) and used the spare flip flop from U32. You could use a combination of gates fom the 74LS27 but it didn't seem worth the effort just to avoid using any extra parts. See Figure 2 for schematic of changes.

I also brought out the unused PORT B of U54 and U72 to connectors on the back and use them for real-time I/O (ham radio applications mostly). I've gotten a CW sender to work using a bit from one of these ports (not too difficult).

I'm working on a simple mod for bit-mapped graphics using the TI 4416 (16K X 4) chips. Two of them will do for graphics memory. I intend to put them in the second memory bank but software to drive it may get tricky.

Mike Chepponis
3606 Mintwood St
Pittsurgh PA 15201

■ ■ ■

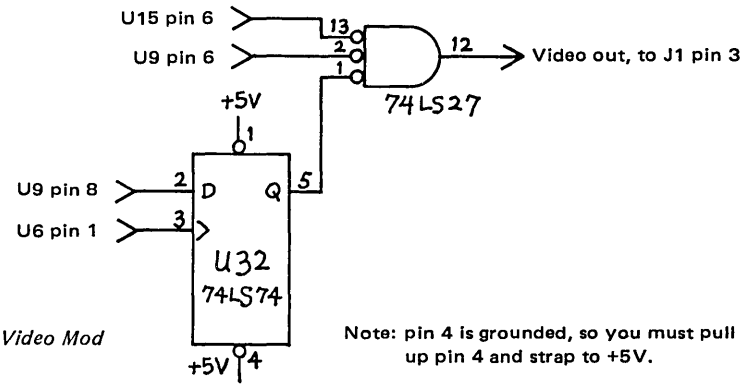


Figure 2 - Reverse Video Mod

Note: pin 4 is grounded, so you must pull up pin 4 and strap to +5V.

The Search for a Cheap, Fast Modem

By Glen Widener

270 SE 15th #5
Hillsboro, OR 97123

A year and a half ago I started looking for a modem to use with my Big Board. I wanted to do some of my work at home, and I needed to transfer letters and programs to my father in Texas. 300 baud was simply too slow to do really productive work. Screen editors are intolerable at that speed and long distance charges would run at least $.93 for a 4 page letter or $25 for a full 8" SSSD disk.

So I started looking for a 1200 baud modem so I could use the screen editor on Tek's system, and cut long distance to $.25 a letter. (You would not believe how much mail had been lost between Oregon and Texas in the last two years. Oh, so you would believe it!)

Too Cheap . . .

The first thing I ran across was a 202 compatible modem card for $250. The 202 standard is very old, uses Frequency Shift Keying (FSK) like the 300 baud Bell 103 standard, and is half duplex only.

FSK is a modulation method which uses two transmission frequencies—one to represent a space (1) and the other a mark (0). In 202 the frequencies are 1300 and 2100 Hz, so 1200 baud is the limit because it is difficult to detect a frequency in less than one cycle. (The bandwidth of the telephone network is 3000 Hz, so you can't raise the frequencies much).

Half duplex means that data can travel only one direction at a time and at least 200 milliseconds are required to reverse the data direction. This makes terminal operation a lot like shouting across a canyon unless you use local echo, and even then you have no feedback on data errors. Half duplex also slows down computer-to-computer transmission with block error checking because of the line turnaround required to acknowledge the block.

Because of this and because the modems on the computers at Tek use the newer Bell 212A and Vadic 3400 protocols, I decided to keep looking. (Incidently, AMD's new $90 1200 baud modem chip is a 202 unit.)

Vadic had a Better Idea

212A and 3400 are very similar protocols. They both are full duplex at 1200 baud. They use phase shift keying, in which a constant frequency and amplitude carrier changes phase by 0, 90, 180, or 270 degrees for each dibit. A dibit is a pair of successive bits which are encoded into one signal element time (in this case, one phase shift). PSK is about twice as complicated to implement as FSK, so the modems are a good bit more expensive and it will probably be at least a couple of years before a one-chip 212 modem is available.

Vadic invented this technique with their 3400 modem, but unfortunately when Bell decided to compete, they made their 212 incompatible with 3400, either because of Vadic patents or because they believed they could improve data transfer reliability over long distance lines. Bell chose a different set of frequencies (1200 originate/xmit, 2400 originate/receive vs. 2250 and 1150 for 3400) and a different dibit to phase shift mapping (00=90, 01=0, 10=180, 11=270 vs. 90, 270, 0, and 180). Having a non-zero phase shift on unchanging data gives the PSK modem its characteristic frying sound on an idle line.

Some believe that Vadic's protocol has higher reliability primarily because the carrier frequencies are not integral multiples of each other. In my testing I haven't seen this.

Regardless, both are more reliable than 202, Vadic is almost the only manufacturer supporting their protocol, and Vadic's most popular modem supports both. So it appears that the 212 standard is the best option.

Why all the Standards?

A comment about the nature of standards might be in order here. Most standards are de-facto standards which came about because the industry was dominated by one or two major manufacturers, in this case Bell and Vadic.

212A has emerged as the standard instead of 3400 because of Bell's clout, not because 212 is better. Thus standards bodies (EIA, ANSI, ISO, CCITT, etc.) often act as referee, declaring the winner and formally defining the result after the dust has settled.

More recently, standards bodies have taken a more active role in defining standards in order to clean up problems with existing standards or for an application which is not yet in widespread use. Examples are RS-449, the proposed replacement for RS-232, and X-21/X-25 for packet-switching technology. It is not yet clear how successful the committees will be in this effort - RS-449, for example, remains an almost unused standard after 5 years of existence.

Hayes Smartmodem II

Last June, Hayes introduced its Smartmodem II, which does both Bell 103 and 212A along with auto-dial and auto-answer. It is controlled by characters from the DTE (terminal or computer) rather than using the out-of-band signalling technique of Bell and Vadic which requires additional wires from the DTE.

At power-up the modem is in auto-answer mode. It is also looking for a command sequence from the DTE which begins with 'AT' (ATtention). The AT is used to set baud rate, parity, and stop bits automatically. Commands include functions like: dialing, redialing, disconnect, and modes such as: duplex, carrier wait time, and answer on nth ring. The dialing command includes a string of digits to dial (pulse or touch-tone). After dialing, the modem goes into originate mode and waits for a carrier.

The smartmodem connects directly to the telephone line using the standard modular phone jack (RJ11C) and to the DTE with an RS-232 connector. It has an internal speaker so that you can hear the progress of the call until a carrier is received (or until someone answers the phone becaused you dialed the wrong number).

Hayes was swamped with orders (10,000) when they first announced the new modem, even though the announcement came 2 months before they began production. Other manufacturers quickly jumped on the bandwagon. All of the units carry a list price of around $700, too steep for my blood. So, I waited to see if anyone could do it cheaper.

US Robotics

After chasing some dead ends I discovered a company called U.S. Robotics with a unit which duplicates the functionality of the Hayes with half the parts and a list price $100 lower. In addition, their product line includes three less ex-

pensive Bell 212A compatible units.

They have been around about five years but they haven't had a strong distribution network. They are now aggressively pursuing dealers and distributors with very good discounts and terms.

I ordered a unit to test and found it to be very well constructed using 3 microprocessors, 10 other I.C's, and a handful of discrete parts.

Unfortunately, I ran into line hit (error) problems. It turned out that USR was still figuring out how to make the demodulator insensitive to line noise, and our GTE lines around Beaverton are terrible.

After six weeks and two revisions to the 8049 demodulator code, I had a modem with performance equal to or better than Vadic. I found a connection on which both modems consistently got about 2 errors per screen of text. However, the Vadic went into fits when data was being transmitted in both directions. Through all the testing, USR was very helpful, promptly delivering both modems and new micros.

Finally

Because I had a lot of friends interested in 212 modems and because USR was looking for a Northwest distributor, I signed up. See my ad in this issue.

In the future I'll cover the most common problems users have with RS-232C. The Hayes Smartmodem is reviewed in *Byte* (March 83 pg 282), and the USR modem is reviewed in *Microcomputing* (March 83 pg 20).

■ ■ ■

The New Nevada COBOL

Review by Sgt David Burgess

Box 5921 Ramstein AB
APO NY, NY 09012

I know that the new Nevada COBOL isn't really new, but the price is. At $29.95, I couldn't see any good reason to pass it up. I'm now very glad I didn't. This has been a very impressive bit of software. Granted, it won't soon replace anything by Microsoft but it should put a major dent in their little monopoly. I will first explain the application, and then I'll cover my feelings about the program and the documentation.

I am a computer programmer for the U.S. Air Force stationed in West Germany. As part of my job, I am writing the first system for U.S. Air Force Europe on our new Sperry 1100 Base Level Computer.

The program I am writing is relatively simple. When finished, it will control inventory in the warehouses used by the Housing Supply people. The old system was made up of 10 programs and was a major headache to run (batch processing all the way).

My new program is very different because it doesn't rely on cards. And, this is where I used Nevada COBOL. I needed to prove to the folks upstairs that I could build a screen-oriented entry-mode processor for the system.

Using COBOL, I was able to build a front-end processor for our new Sperry 1100 computer. I used a Big Board as a pre-processor for the Sperry to show that the logic is not to complicated. Of course, the program does run!

The Compiler

CC.COM is the COBOL compiler, and all this program does is generate an object code file which can run under RUN.COM. This way, the compiler can be much smaller and the programs can be much larger. The overlay structure is also simplified.

According to Ellis Computing, this two-part format also makes it possible for the user to install his/her own I/O functions.

Weaknesses

I have only found one bug with the compiler. That is, any time the sentence:

SAME AREA FOR filename-1>,
filename-2>.

appears, the computer will hang and I have to re-boot. Since this particular option is really only used for files which have been defined strangely, I don't run into it very often.

However, that's not the only problem with this product. Nevada COBOL does not support the following options of the standard COBOL-74 definitions:

- Multiple GIVING clauses.
- SORT.
- VARYING FROM A BY B on a PERFORM.
- Nested IF's will generate incorrect object code if any is generated at all.
- Lower case letters will not be accepted in a literal prompt in a DISPLAY or ACCEPT message.
- 66 and 88 level data-items are not supported. This particular limitation is one which I find almost unacceptable.
- The verb GO TO is fully supported. (Yes that's bad news.)

Strengths

Nevada COBOL does support all the data descriptions most COBOL programmers use. At least I couldn't find any missing.

It also has the most complete documentation package I have ever seen. The manual is 152 pages and is filled with information. It includes examples of how to use every verb in the language and how to run the programs. This system was really a bargain at $299.95. It is unbeatable at $29.95.

Conclusion

If you are looking for a complete COBOL system and can live with the limitations placed on the code by Ellis Computing, then Nevada COBOL could be the programming language for you.

COBOL is still being used in the business community on some of the larger/older mainframe systems for production of payroll, etc. systems.

This package is an excellent tool for learning COBOL because your code will also run on any COBOL-74 compiler on the market.

The only real changes I would like to see are a VARYING clause on the PERFORM verb and the implementation of 88 level data-descriptors. The nesting of IF's would be nice but isn't necessary.

Nevada COBOL
Price $29.95
ELLIS COMPUTING
3917 Noriega Street
San Francisco, CA 94122

■ ■ ■

Interfacing YE Data Drives

By James T Reid

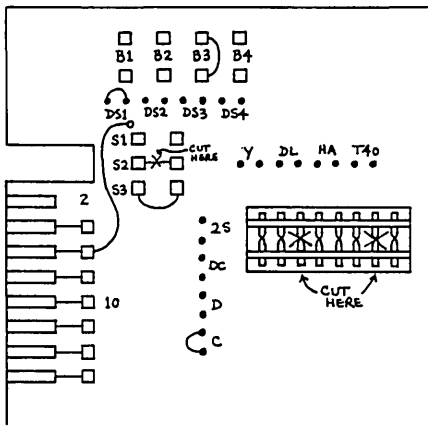
81 Pattiwi Rd
Mosman 2088 Australia

I recently took delivery of two of the new YE Data YD180 double-sided 8" D.C. Drives. I chose these drives mainly because two fit in the same space as one A.C. drive and they require less power.

They were hot off the production line and arrived without any users manual but the supplier said they were electrically compatible with the older YD174 A.C. drives so I did not anticipate any difficulty in getting them up and running. Silly me.

These two drives appear to my system as 4 drives. The necessary jumpers are relatively self explanatory. However, the side select signal is varied by jumpers on pads S1-S3. On the YD180 there is a strap across S2 on the back side of the PCB and this must be cut before soldering jumper S3 into position or attempts to select drives C or D will give unpredictable results. The PCB must be removed to cut the track.

The sketch shows the necessary jumpers for the YD180 to configure it as drives A and C.



■ ■ ■

LETTERS

Dear Editor,

It seems that there is some confusion over the 8002 video attributes controller on the BB II. According to SMC there are several versions of this part indicated by the letter A, B, or C. These suffixes indicate the maximum dot rate that the part can handle. It is not an indication of the dimensions of the character font.

Rather, the -003 suffix indicates the 5X7 font, while the part without the -003 generates a 7X9 font.

If you want a 7X9 character display you should use a CRT8002A with a 16 MHz crystal. A CRT8002B might work but it is only spec'd to run at 15 MHz.

If you want to run the 5X7 display, you may use the CRT8002A-003 or a CRT 8002B-003. In either case you'll use an 11 MHz crystal.

Also, I've found a small bug in the BB II monitor. I am using the internal keyboard and video for a console and have encountered a problem while using the polled mode SIO routines.

The SIOOUT routine expect to have the control and data ports for the selected SIO port plugged into a fixed place by initialization.

This only happens when one of the SIO ports is being used as the console. An easy fix that makes the B SIO port available via the monitor jump table (when using internal keyboard and video) is to change 0FF6DH to be 83H and 0FF6EH to be 82H.

A more general solution would make polled mode access to SIO B available even when using SIO A as the console. This would require adding a second SIO output routine (call it SIOBOUT). The existing SIOOUT routine would be used only when the console is connected to one of the SIO prots. SIOBOUT would be the routine that is accessed via the jump at 0F018H.

```
SIOBOUT:  PUSH   A          F
SB2:      IN     A,(SIOCPB)
          AND    00000100B
          JR     Z , S B 2
          POP    A          F
          OUT    (SIODPB),A
          RET
```

Cole Chevalier
17862 Fitch
Irvine, CA 92714

Dear Editor,

On page 13 of the BB I theory of operation, the pin assignments on the disk connector should be: WRITE—pin 40, TRACK0—pin 42, WRITE PROTECT—pin 44 (NOT 42, 44, 46 as shown).

Kevin Giehl
Soft System Designs
3174 N Booth
Milwaukee, WI 53212

Dear Editor,

Would anyone be interested in an I/O expansion board? I will produce one if enough folks contact me and say they are interested. They should let me know whether they prefer the versatility of SIOs and PIOs or the ease of use of UARTS and parallel latches.

Bruce Halco
1434 Mayfield Ridge
Mayfield Hts, OH 44124

Dear Editor,

I very much enjoyed your column in the debut of Pro-Files magazine. As one who spent 15 years hacking around as a tramp journalist, I felt a strong kinship with you and your escape from the corporate treadmill. Trouble with my choice of professions is that even though I am now "self-employed" in the eyes of the IRS, I sometimes feel that I'm just on another treadmill. However, now that I have a Kaypro II, I feel I'm just beginning to see the vast number of alternatives available.

I'm not much of an engineer, but pretty fair as a technician. I used to have a Ham ticket and built a Heathkit transmitter. I purchased a 5MHz hop up for the Kaypro for $99.95, but I can't switch back to 2.5 MHz without opening the case and sticking in the old CPU. Is there a reason I'd want to go back to 2.5 MHz?

How about a way to put the reset button on the front panel so you don't have to grope around the back? Also, do you know why the assembly language routines are written in 8080 instead of Z80? Is there a program that will allow me to use autodial on a modem? Is there a way I can look at memory and poke around to see what is stored where?

Donald Marshall
13 Lisa Dr
Greenville, SC 29615

Editor's note:

You can move the reset simply by drilling a hole above your drives and moving the switch to the new hole. Drill from the outside of the cabinet and be very careful about filings! Some programs generate software timing by putting the processor in a loop. When you speed up the processor, it changes the timing. The early copy programs don't work at 4 or 5 MHz but they work fine at 2.5.

Some of the speed up mods add wait states when accessing memory so they don't have to change the monitor ROM and they say they go back to 2.5 MHz when accessing the drives. This means that those speedups probably won't improve system throughput very much.

CP/M was originally written for the 8080 processor so much of the code remains in 8080 mnemonics. Modem7 lets you to autodial with the PMMI modem, and we'll have a new modem program shortly that will let you do autodial with the Hayes Smartmodem (or any Smartmodem compatible modem).

DDT.COM which you received on your Kaypro CP/M disk will let you look at memory, write short assembly language programs, disassemble programs, and modify memory, among other things.

Dear Editor,

I have finaly completed my Big Board and it works very well.

I've installed my BB in an Apple case (Ed. note, apparently not an apple crate) with an Apple switching supply (the Big Apple?). To this I've added a separate National JA-751 drive with power supply, a BMC monitor, and a Microline 80 parallel printer.

Incidentally, if any of your readers are interested, we can sell them the power supply for $80.00 (5V at 3A, 12V at 2.5A, plus -5V and -12V). We also have the cabinet for $75. Prices include air parcel.

I have rewritten the CBIOS that came with the BB I using 8080 mnemonics so I can use the CP/M assembler. I had a little trouble incorporating John Jones parallel print driver from issue #2 until I noticed that he used the interrupt vector for port A instead of B. I also had to eliminate the 'CPL' because my printer didn't need it. If you think anyone would be interested in this CBIOS, I would be happy to send you a copy.

Thanks for the great job you're doing! I have all issues of Micro C and I am looking forward to receiving my next.

Lionel B Yotoko
PO Box 772 MCC Makati
Metro Manila, Philippines

Editor's note:

Yes, we certainly would like to get a copy of your CBIOS. Please include the step-by-step instructions for assembly and incorporation into CP/M. Don't forget to tell us which user disk you'd like in return.

Dear Editor,

I need some help with a problem. Many times, momentary power problems have crashed my Big Board out of CP/M into the PFM monitor. When I examine memory with the PFM dump command, I have found my text file intact, but I don't know how to save it onto a disk.

When I use the CP/M SAVE command, neither the TYPE command nor my text editor seem to be able to interpret the file. A friend who is an ace assembly language programmer gave me little hope, but I find it hard to believe that it can't be done.

Anyone who can solve this is going to save me a lot of time when the neighbor downstairs turns on his arc welder.

Also, the XDIR program works fine on drive A but sometimes crashes the system when used elsewhere. It happens consistantly with some disks. Is this a problem with my BB or with XDIR?

John S. Allen
40 Rugg Rd
Allston, MA 02134.

Dear Editor,

This letter is in response to your plea for a source of low cost disk drives.

I recently purchased a pair of new Shugart 801's from Bob Scott of Epic Sales, PO Box 401471, Garland, TX 75040. 214-494-3800. They were $150 each. They were manufactured for Europe so they have 230V 50 Hz motors. Bob changes the motor pulley for 60 Hz and he sells a 110V to 230V transformer.

Each drive is exercised for several hours on one of four Big Board computers.

Tom Lipe
7815 Apache Rd
Little Rock, AR 72205

Dear Editor,

We've gotten a number of BB user disks. The 820s run just fine using all the programs even with the original Xerox monitor.

Raymond Pettis
6713 Tung Ave W
Theodore AL 36582

Dear Editor,

In issue #12, pg 38 my zip code should be 19130, not 19138. (Just in case someone writes to you with problems.)

Gary Kaufman
2001 Hamilton St. Box 87
Philadelphia, PA 19130

Dear Editor,

I'm especially interested in the "On Your Own" feature and would be interested in seeing a future issue dedicated to starting your own mail order business.

Some topics I'm interested in are:

1. Mail order laws and regulations.
2. Software copyrighting process.
3. Licensing fees and royalties.
4. Taxes associated with mail order.
5. Advertising.
6. Marketing a software application.
7. Anything else.

Since you are mail order, I'm sure you are an excellent source of information. Also, I'm sure there are a lot of other software types who need this kind of information.

John Couchman
3829 Heather Drive
Eagan, MN 55122

Editor's note:

Good suggestions, John. In one sense we are experienced in this field, in another, we are just beginning. The SpellSys project is really helping us get our feet wet in the software marketing field and we are fumbling along like everyone else.

Anyone with additional information (or suggestions for information sources) should let us know. As for item #7, I'm sure we have covered that already.

Dear Editor,

Thanks for the rapid delivery of the back issues. It is very interesting to note the passage of two years time in a few days reading. You started on a high note and have managed to maintain that level. I am very impressed with the quality and content.

A local wholesaler (junk warehouse) called Space Age Metal Products, 1490 W Artesia Blvd, Gardena, CA 90247, has a huge quantity of Siemans 8" drives. I bought one for $200, but asked where the break point was on quantity discount. The fellow hesitated and bit and replied "In a quantity of 20 or more, you can have them for $150 each."

Mine looks good, and came sealed in a form-fitting foam box. It had inspector's stamps in several places.

Also, is there any possibility of a short recap of those user disks which run on Big Board IIs without problems? (Or the patches needed if there are problems?)

Does anyone care to offer a service in selling or renting expendable disks so that when people try their drives for the first time they won't blow lots of bucks before they know what happened?

Will Tiny Basic fit in one 2732 for the BB II?

Please let us in on the staff joke (if it is one) "Typography, Irish Setter."

I suggest that you solicit advertisers for hard disk drives and controllers. Tell 'em that you have a bunch of drooling customers on your list!

Bob Hall
801 Pacific Ave
Hanhattan Beach, CA 90266

Editor's note:

Thanks for the comments about Micro C. About the Siemen's drives, they were dumped because the stepper motors die after about 2,000 hours of stepping (actually not too bad). On systems such as the Slicer, the BB II, and the Kaypro with an 8" mod (which use fancy data separators rather than the ttl version used on the BB I), the Siemans drives actually work better than the Shugarts.

Rob Devoe volunteered to check on the compatibilities (and incompatibilities) between the user disk software and the BB II. So, I sent him all the disks. I haven't heard from him. Are you there Rob?

Try some Elephant disks for testing if you can't find anything used. They are just rebranded Wabash media. Then get some Dysans or BASFs for the important stuff. Stay away from Maxells, they are too abrasive.

Tiny Basic will fit into one 2732, but it probably won't run on the BB II.

Patty and Martin do our typsetting. Their typsetting business is called the "Irish Setter" which is pretty outrageous, but that's the type of people we have to set with. (They are also very good typesetters because they have a real feel for the art form. Sandy and I feel very fortunate to have them as friends.)

Dear Editor,

I just got the back issues and want to thank you for the fine job with the magazine. I only wish I had had the foresight to purchase Micro C before buying a Big Board. I would have saved time and money.

One thing I'd like to pass along. As long as the 1771 disk controller wasn't in its socket, the board would come up just fine. When it was installed, I got alternating garbage on the screen. It must have been a timing problem on the RDB line because a 1K pullup on U82 pin 5 solved the problem. I had tried three 74LS241s so it evidently was not a chip problem.

Dick Barnett
604 Robinson St
West Lafayette, IN 47906

■ ■ ■

On Your Own

By Guest Columnist
Leroy Searle

6273 19th NE
Seattle, WA 98115

The following is the story of a journey. A few of you are making this journey already, and can write your own travelogues. For the rest of you, read on. Perhaps you'll find you've already joined us. The Editor.

A little more than two years ago I took a sabbatical leave to finish a book I had been working on for nearly ten years. The book is a study of the history of literary criticism and theory, and its premise is that philosophy and science started from a grand Platonic puzzlement about the imagination. Thus, the book is about imagination, but it is also about philosophy and science.

When I started working on this book, it was with the conviction that what happened in good poetry was very like what happened in good science—making discoveries and connections. As a teacher in the English department, I am not expected to know much about science, but as an instructor of mine once said, the great thing about studying English is that anything in print (even on a screen) is fair game.

Besides, the very length of my manuscript gave me a powerful reason for getting a computer.

Therein lies a story

Having more time than money, I decided to study computers in the most practical way—by building one. The project absorbed a great deal of my time (and too much money), but it had more of an effect than I had bargained for. I ended up with a word processor, but more importantly, I ended up with a computer education that can be had in no other way.

When people ask me now if I would recommend building a computer, I ask whether or not they are interested in rebuilding their imagination. This may seem to be an odd question, but that is what is at stake with this technology. The closer you get to it, the more transformative and liberating it becomes. When I started out, I had no engineering experience (unless one quarter of civil engineering labs 20 years ago counts) and no useful knowledge of computing. In fact, one of my reasons for doing this was a certain uneasiness I felt about computers—thinking of HAL in 2001, for example, or the computer that always seems to screw up your utility bill.

What I found as I began building my first computer (an S-100 system made from bare boards, surplus, and junk) is that you need to have one computer to get another one running. Upon discovering this paradoxical rule, I set about a gloriously time-wasting venture: I wire-wrapped a version of an IMSAI front panel, using a keypad instead of toggle switches. My thought was that if I could get this device to operate, I stood a better chance of understanding what really happened in a real computer.

My front panel worked, and it worked well enough that I found out how to do elementary troubleshooting without it and even ventured into designing and wire-wrapping other boards (including a 64K dynamic memory board). Then, however, I found out that once you have built a computer, you need a troupe of people for support—if for nothing else, to help you get your own handiwork up and running, purged of its bugs and defects.

In my case, having so little money, I ventured into areas where I was so far out of my element that the people I encountered had no idea of my ignorance. They would assume, naturally enough, that someone who was rewriting the device drivers for his system in assembly language (and evidently succeeding at it) MUST know what he is about.

In a way I did, but all I knew was what I had been obliged to learn when my hardware refused to work.

Community

The most important discovery, however, was that a community of remarkably generous and thoroughly knowledgeable people was quickly emerging at the fringe of computer mania, and this community was not mad at all.

In the midst of advertising, salespeople by the gross, and much hysteria over fortunes to be made out of micros, the people I met in my desperate search for help appeared to have grasped the elementary fact that the computer revolution is not just a revolution in technology but a revolution in ways of life.

I found it striking that I heard so few names in discussions of computers. This is partly because the complexity of the work computers present requires cooperation. I could see this clearly when I set about rewriting the software for a couple of video controllers.

I was building on the work of dozens, perhaps hundreds of people who I would never meet, and when I was able to get a board worth a couple of hundred dollars to do the work of a terminal worth about a thousand, the surplus value reflected a largely anonymous community passing on what they had done. I in turn passed on my work with hardly a mark.

By stages, my year on leave had led me to an interesting condition. I had met a large number of remarkable people and had learned enough to be able to talk to them in a non-trivial way.

When I returned to teaching, I found myself in the middle of a familiar situation but seeing it in a new way. Information technology is rapidly changing the nature of education and not just in the technological or scientific fields. More importantly, it is changing what we think is possible, and therein lies the rebuilding of imaginations that seems central to computing.

Since I teach English for a living, it seems not at all odd that I should do it in a more richly augmented way.

Starting the Computer Collaborative

By a complex series of events, I found myself in the middle of a group of people—a top-flight computer production engineer, a systems programmer, a marketing specialist, and a couple of clever entrepreneurs—and decided to organize a company to put the ideas of these people into concrete form.

Early this year we formed Computer Collaborative around an idea and a product. The idea was that a company could proceed on the same basis that a cooperative community proceeds—by collaboration where many people with many skills share the work and multiply its effects. The product (also an idea) was a very simple and elegant implementation of hard disk technology that made it possible to add a very high-quality hard disk subsystem to almost any 8-bit micro computer.

To put this project together, we started a small group, Delphi Development. At the start the group consisted of Tom Ngan, Henry Takeuchi, and myself. We worked with the generous help of other people in Seattle—most notably, Charles Thurber of Puterparts (a unique business supplying computers by parts and wholes, along with good advice and encouragement), and Jim Mitchell of Anchor Computer Systems, a business that is growing so fast it takes my breath away.

At first we just wanted to see if it was feasible to interface a hard disk to small computers like the Big Board, the Kaypro, and S-100 systems using a very simple host adapter consisting of only 8 ICs.

Once it was clear that it would work and once we had worked out the software to match this elegant little piece of hardware, it seemed obvious that the next thing to do was produce it.

Putting It Together

Before we knew it, we were in business. Computer Collaborative came together within a matter of weeks. Collectively, we worked out the design for our winchester drive systems, and before we even had time to complete the planning for a production facility, we had orders.

It is exciting to see something go out the door, but it is at least as exciting, from my point of view, to see a way of doing business take shape not from a desire for wealth but from a desire to make a liberating technology more readily available to people.

Throughout, what we have aimed at is quality and functional simplicity. We think we have achieved that and in doing so have put together an organization that reduces the distance between ideas and their realization in products. We have done it on our own, and it is a great pleasure.

■ ■ ■

TWO WAYS TO ENHANCE YOUR BIGBOARD'S CAPABILITIES:

#1 DUAL DENSITY

HARDWARE

- A daughter board that plugs into the 1771 socket. With this board the system employs automatic density select.
- You can run 5¼" drives by following the simple steps outlined in the manual. A 50-34 pin disk drive adapter board is included with 5¼" orders.

SOFTWARE

- Choose 2.5 MHz or 4 MHz software, for 5¼" or 8" drives. Also select software for single or for double-sided drives.
- 8" users have up to 674k bytes of user storage per disk (per side). 5¼" users have up to 185k bytes of user storage per disk (per side).
- Dual Density software includes:

- DDINIT. COM: a double density disk initialization and verification program. Options:
  - 8 formats.
  - Format an entire disk or just system tracks.
  - Selection of sector skew.
  - Option to verify.
  - Choice of drive to be used.
  - Has a default which chooses the format that gives the most disk space.
- DDSYSGEN. COM: a double density sysgen program with three options:
  1) Read double density system tracks into memory.
  2) Write double density system tracks from memory to a double density disk.
  3) Generate a double density system disk complete with printer driver. This process uses your single density CP/M disk, the SWP distribution disk, and a blank disk. Five serial printer drivers and a parallel driver are included, and there is an option to install a user-written driver. All drivers can be modified.
- DDCOPY. COM: a double density copy program that copies all files from a source disk to a destination disk.

- Being a dual density system, the computer automatically distinguishes between single and double density disks. Densities may be mixed.

~~$199.95~~ SPECIAL OFFER: $149.95

#2 CO-POWER-88+

HARDWARE

- A powerful 16-bit 8088 coprocessor.
- Available in three RAM sizes: 64k, 128k and 256k.
- Consists of two main boards, the Z80 adapter board and the main processor board. The Z80 adapter board plugs into the Bigboard Z80 socket. A ribbon cable connects the Z80 adapter board to the main processor board. The main processor board holds 128k of RAM. An additional 128k RAM is available using an add-on RAM card.

SOFTWARE

- Runs CP/M-86 or MSDOS. CP/M-86 is compatible with CP/M 2.2. Its command files have .CMD as the extent, making it possible for CP/M-86 and CP/M 2.2 files to co-exist on the same disk (CP/M 2.2 command files have .COM as the extent). MSDOS is the operating system of the IBM-PC. IBM-PC MSDOS programs are compatible with the CO-POWER-88 MSDOS.
- Simple commands move the user between the Z80|CP/M2.2 system and the 8088|CP/M-86, MSDOS system.
- The RAM of CO-POWER-88 can be used as a "memory" ("M") disk drive for CP/M 2.2. Programs can be compiled in M, then saved to disk, decreasing job time by avoiding disk access time.

PRICING:

128K CO-POWER-88 with MSDOS and CP/M-86	$799.95
128K ADD-ON RAM CARD	$300.00
256K CO-POWER-88 with MSDOS and CP/M-86	$999.95

+CO-POWER-88 is available for nearly all Z80 or 8080 computers using CP/M 2.2.

SOFTWARE PUBLISHERS, INC.
2500 E. Randol Mill Rd., Suite 125 Arlington, TX 76011 (817) 469-1181

Packet Radio, A Closer Look—Part I

By Peter J. Eaton WB9FLW

35 Norspur, Route 4
Edwardsville, IL 62025

Now that you've been introduced to packet radio in Micro C Issue #12, I thought you might be interested in looking a little more closely at this new technology.

It turns out that amateurs are not the only ones looking a little more closely at packet radio. Recently, several commercial packet radio applications have been announced. They include Genstar, developed by Gerard O'Neil of Princeton University; and the Portable Communications Terminal developed by Motorola. And I believe that there will be many more such commercial digital packet systems in the coming years.

In the meantime, amateurs are blazing many new trails—from the TNC (terminal node controller) to the PACSAT orbiting bulletin board system.

Doug Lockhart and friends in Vancouver BC lead the way with their development of the first TNC. Its popularity spread quickly after they began supplying the PC board and parts kit.

Pockets of packets (Editor's note: Not to be confused with packets of pockets.) rapidly developed in Washington DC, Tucson, San Francisco, St. Louis, New Jersey, and southern California. The first ARRL (American Radio Relay League) Packet Radio Conference was held in October 1981 in Gaithersburg, Maryland.

Following this, the Tucson Amateur Packet Radio (TAPR) group began designing a completely new TNC. They took the best of the ideas from the Vancouver group and combined those ideas with the latest hardware.

While the Tucson group was working on hardware, another group lead by Tom Clark W3IWI, went to work on a protocol standard. The original Vancouver protocol worked fine for small geographic areas but clearly would not support any kind of national networking. The Phase IIIB amateur satellite with its packet radio repeater made Clark's work particularly important.

The new standard, called AX.25, was finished just in time to be incorporated into the first of the Tucson TNC boards. About 200 of those boards have been shipped around the country for testing, and they have been a real hit. The test phase is almost over and the parts kit should be available later this year.

The TAPR TNC

As we discussed in Issue #12, the terminal node controller interfaces your terminal (or computer) to your transmitter and receiver. The interface with the terminal can be either serial RS-232 or parallel, although there is software support only for the serial interface.

The interface with the transmitter and receiver is via the speaker, the microphone, and the push-to-talk.

The TNC is like a very smart modem, since it will allow any computer or terminal to talk to any other computer or terminal, but as you will see, it is much more since it deals with the problems of supporting the protocol.

The Tucson TNC is based on a 6809, with 24K of ROM and 6K of RAM. It includes a 1200 baud modem (bell 202 half duplex) and the power supply on-board.

The 24K ROM supports the AX.25 protocol, which means it formats and sends packets, receives and deciphers packets, and filters out information not destined for your station.

The beta test version of this unit sold for only $200. We'll take a more detailed look at both the hardware and software in the next installment.

You can get a complete package (packet?) of information about the Tucson TNC if you send them $15.

The following people have put in long hours on the Tucson TNC. Mark Baker, Marc Chamberlin WA7PXW, Chuck Green N0ADI, Dave Henderson KD4NL, Lyle Johnson WA7GXd, Dave McClain N7AIG, Dan Morrison KV7B, Margaret Morrison KV7D, Harold Price NK6K, Den Connors KD2S, and (of course) myself.

Editor's note:

Clayton Schmitt of Estacada OR sent along the following addresses after my appeal in issue #12.

Tucson Amateur Packet Radio Corp
PO Box 22888
Tucson, AZ 85734

V.A.D.C.G.
818 Rondeau St.
Coquitlam, BC Canada V3J 5Z3 ■ ■ ■

Especially For The Big Board

(503) 382-8048

USERS DISK #1
1-Two fast disk copiers
2-The manual for Small C+
3-Crowe Z80 Assembler
4-Two disk formatters
5-Modem7
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 I
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-Modem7 for Port A
3-Modem7 for Port B
4-PACMAN, the arcade game
5-FAST, buffers the disk to speed up assemblies
6-NOLOCK, removes BB I shift lock
7-VERIFY, cleanup & verify a flaky disk
8-DUMPX, enhanced for BB I
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 I (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 transfer 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
11-Extended CRC checker, creates file & checks file
12-Super disk formatter program for BB I

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 I 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, Biorythm 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

All Users Disks .....$15.00 each (US,Can,Mex) .....$20.00 each (other foreign)
All The Users Disks Contain Documentation On Disk In .DOC Files.

OTHER GOODIES

Screen Editor in Small C $39.00 $44.00
A simple but full-function screen text editor plus a text formatter, all written in Small C by Edward Ream. This package includes the editor and formatter .COM files setup for the Big Board, Small C itself, and source code for all. With the documentation this is over 400K on a flippy disk. Edward is selling this package for $50, you can buy it from us for $39 (and Ed gets a royalty). Where else can you get an editor, a formatter, a C compiler, and source for all, for under $40?

FREE Your choice of a user's disk or the deluxe character ROM free if you send an article or software and a ROM or extra disk.

US,CAN,MEX Other Foreign

Your Fortune in the Microcomputer Business $26.45 $36.45
This is the best, most complete collection of "working for yourself" information I've found (and I've heard nothing but good comments from those who have received it). This two-volume set is a perfect for those blustery fall evenings when you snuggle up in front of the fire and dream of great riches.

MORE ROMS
Fast monitor ROMs for speed freaks and our famous 'better than Texas' character ROM (V2.3) for screen freaks.
Fast Monitor ROM $25.00 $30.00
Version 2.3 Char ROM $25.00 $30.00

- Send Big Board number with ROM orders.
- Monitor & char. ROMs $5.00 each if you send a fast ROM and a stamped, self-addressed return envelope.

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

From Micro Cornucopia

USERS DISK #13 - General Utilities, BB I

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 which almost looks identical 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 ...

REMEMBER
FREE Users Disks in exchange
for submitted software or articles

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-MODEM7+, Modem7 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.

BB II DRIVE INTERFACE
For 5¼" and 8" Drives

Andy Bakkers is making this special software package available through Micro C. Complete source, HEX, & documentation files on an 8" SS SD disk. Also outlines on disk the hardware changes needed.

$ 29.95 NEW!

All Users Disks .....$15.00 each (US,Can,Mex) .....$20.00 each (other foreign)
All The Users Disks Contain Documentation On Disk In .DOC Files.

BACK ISSUES

$3.00 each US,CAN,MEX
$5.00 each Other Foreign

ISSUE NO. 1 (8/81)
Power Supply
RAM Protection
Video Wiggle
½ PFM.PRN
16 pages

ISSUE NO. 2((10/81)
Parallel Print Driver
Drive Motor Control
Shugart Jumpers
Program Storage Above PFM
½ PFM.PRN
16 pages

ISSUE NO. 3 (12/81)
4 MHz Mods
Configuring Modem 7
Safer Formatter
Reverse Video Cursor
FORTHwords 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

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709

TECHNICAL TIPS

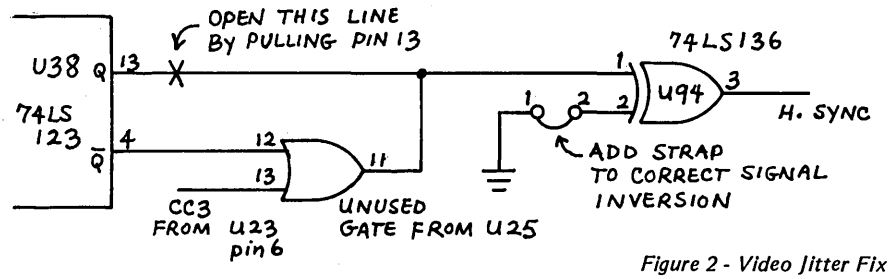


Figure 2 - Video Jitter Fix

Tabling your keyboard

When I wanted to rearrange the output of some of the keys on the Kaypro, I took a look in the monitor to see if there was room for a 127-byte translation table so I could do Daryl Coulthart's translation table (see pg 6 in issue #4).

I found that there definitely wasn't room. However, I found that whoever had written the Kaypro monitor was already doing some character translating using a Z80 compare instruction (cpir) that I hadn't used before, so I simply added a few bytes. See Figure 1 for complete routine.

I looked through both my Z80 manuals and neither had a clear explanation of cpir. However, after a couple minutes, the code in kbdmap: made perfect sense. It's obvious that cpir is ideal for someone who is doing a short or medium size table and doesn't have to have instantaneous response. (Coulthart's routine should be faster.)

David Thompson

Figure 1 - Keyboard Translation Routine

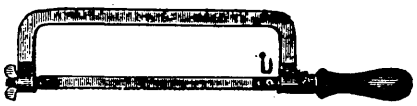
```
kbdmap: ld      hl,mapin        ; input map table
        ld      bc,mapout-mapin ; table length
        cpir                    ; search table
        ret     nz              ; not found
        ld      de,mapin        ; make hl=table index
        or      a               ; hl-mapin=index
        sbc     hl,de
        ld      de,mapout-1     ; index
        add     hl,de
        ld      a,(hl)          ; get char from mapout
        ret

mapin:  defb    0F1H, 0F2H, 0F3H, 0F4H  ; up, down, left, right arrows
        defb    0B1H, 0C0H, 0C1H, 0C2H  ; 0,1,2,3
        defb    0D0H, 0D1H, 0D2H, 0E1H  ; 4,5,6,7
        defb    0E2H, 0E3H, 0E4H, 0D3H  ; 8,9, '-', ','
        defb    0C3H, 0B2H, 7BH, 5BH    ; return, '.', {, [
        defb    5DH, 22H, 27H, 5FH      ; ], ", ', _
        defb    2DH, 7CH, 5CH, 7EH      ; -, |, \, ~
        defb    60H, 40H                ; `, @
        defb    0FFH                    ; end of mapin table
mapout: defb    80H, 81H, 82H, 83H      ; vector pad, xlate in bios
        defb    84H, 85H, 86H, 87H
        defb    88H, 89H, 8AH, 8BH
        defb    8CH, 8DH, 8EH, 8FH
        defb    90H, 91H, 7CH, 5CH      ; , , |, \
        defb    7BH, 7EH, 40H, 22H      ; {, ~, @, "
        defb    27H, 5DH, 5BH, 5FH      ; ', ], [, _
        defb    2DH, 60H                ; -, @
```

Another video jitter fix

The BB I video problem is caused by jitter in the horizontal sync from U38 and U51. Adding a 33 uF tantalum between +5V and ground at U38 and U51 helped, but was not a complete cure.

The video became rock solid when I gated the horizontal sync from U38 with CC3 from U23 using an unused OR gate from U25. This mod reduced the sync pulse width from 6.5 micro seconds to 4 micro seconds, but didn't seem to cause any problems for the video monitor. See Figure 2 for schematic.

Darrell Collins
8638 E Solano
Scottsdale AZ 85253



BB II pull up transistor

The BB II uses the standard waveshaping circuit to drive the clock input (pin 6) of the Z80A. Part of that circuit is a PNP transistor pull-up. However, they specify (and install) a 2N2907, which is an NPN transistor. If you replace the 2907 with a 2N3906 (a real PNP) you'll find that rise time drops from about 45 ns to about 25 ns.

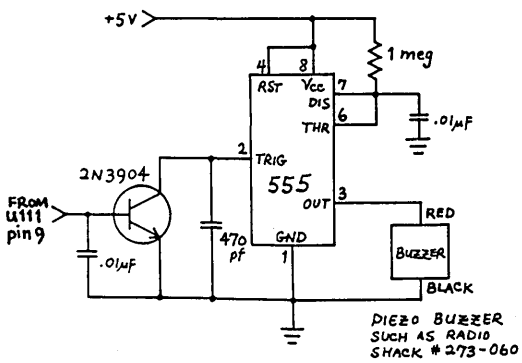
More Bells (but still no whistles)

The following circuit employs the workhorse of the timer world, the 555. The timer is configured as a one-shot, triggered by a negative-going pulse. Since the pulse from U111 is positive-goin, it needs to be inverted (hold this schematic upside down to see what I mean). The inversion is done cheerfully by the NPN transistor. The capacitors on the base and collector prevent false triggering.

The output of the 555 triggers the buzzer. The length of time the buzzer stays on is set by the formula TIME = 1.1 X R X C. The values specified give a .1 second beep. See the schematic, Figure 3.

Danny Stone
2409C Fairway Dr
Raleigh, NC 27605

Figure 3 - Bell Circuit



(Technical Tips continued)

Driving the Prowriter, C Itoh 8510, or the NEC 8032 from WordStar

I have finally managed to get my Prowriter to handle superscripts, alternate pitches and more under WordStar. Here's how:

First, select A: ADM-3 from the terminal menu and I: 'Half-Line-Feed' printers from the printer menu.

Select 'No' communications protocol, since your BIOS should handle the interface to the printer. On the driver menu, select CP/M list device.

Note that if you have a BB I or II and have not installed a printer driver into your BIOS this will not work. Either use the CBIOS on BB user disk #9 or check the WordStar manual for directions on installing an internal printer driver.

When WordStar asks 'are the modifications to WordStar complete?' answer no. This will invoke the patcher so that you can enter the values shown in Figure 4.

Gary Kaufman
2001 Hamilton St Box 87
Philadelphia, PA 19130

LOCATION	Old value	New value	
DELCUS:	10	0	;the BB doesn't need delays!
DELMIS:	5	0	
PSCRLF:	02	03	;set up cr/lf routine to send 2 lf's
PSCRLF+3	00	0A	;to compenstate for 1/2 line spacing.
PSHALF:	00	02	;Set up 1/2 line feed routine to
PSHALF+1	00	0D	;issue 1 line feed.
PSHALF+2	00	0A	
PALT:	00	02	;alternate char font
PALT+1	00	1B	
PALT+2	00	50	;go to proportional
PSTD:	00	02	
PSTD+1	00	1B	
PSTD+2	00	4E	;back to Pica
USER1:	00	02	
USER1+1	00	1B	
USER1+2	00	45	;select elite pitch
USER2:	00	02	
USER2+1	00	1B	
USER2+2	00	51	;select compressed pitch
USER3:	00	01	
USER3+1	00	0E	;select elongated font
USER4:	00	01	
USER4+1	00	0F	;reset elongated font
PSINIT:	01	07	;initialization routine:
PSINIT+1	0D	1B	
PSINIT+2	00	4E	;first reset to pica
PSINIT+3	00	1B	;next set line spacing to
PSINIT+4	00	54	;1/2 line feed per lf to
PSINIT+5	00	31	;allow for 1/2 line spacing
PSINIT+6	00	32	;for superscripts,subscripts.
PSINIT+7	00	0D	;CR to reset carriage
PSFINI:	00	06	;reset printer to STD settings
PSFINI+1	00	1B	;at conclusion of WS
PSFINI+2	00	4E	
PSFINI+3	00	1B	
PSFINI+4	00	41	
PSFINI+5	00	1B	
PSFINI+6	00	66	

Thats it.

Figure 4 - WordStar Modifications

changes direction quickly). It's a tricky sort of thing, and Dorcas (our soggy staff assistant) got dumped into the river. Someone held out a paddle and she pulled herself back into her raft.

Of course, after we were finally wet enough and experienced enough to take on anything, we came to the end of the trip.

Immediately below the take-out place was a falls rated 6. Some of us hiked along the bank to check it out. There was no question at all that it deserved a 6.

Dinner

After the rafting, they took us to the dinner site, a quiet place along the river where there were picnic tables, good guitar music, and some very good food for some very relieved (and very wet) people.

The slack water group joined us at the picnic site, and after sizing up our victorious expressions, mentioned they'd be taking the white water trip next year. (Maybe it should be an all-day excursion instead of 2½ hours.)

Great Weather

The weather was beautiful, not a cloud in the sky, except Saturday evening. Just as Jim Ferguson and Russell Smith began fielding questions on the Big Board II design philosophy, great claps of thunder began rolling across the hills. (I didn't think to ask Jim and Russell if they had arranged it.)

We got just a few drops of rain along with the drama, but they sent folks scurrying about, removing precious equipment from outdoor display tables.

Nearly the entire SOG was held outdoors. The possibility of snotty weather was a real concern since we wanted to have it here at the house, but we hardly have room for ourselves, much less over 100 people and their equipment.

Organization

People mentioned how organized we were. We certainly didn't feel organized. We just sort of threw open our house and waited with trepidation to see what happened.

One of the things that happened was that people volunteered to give talks. I only got to sit in on bits and pieces of many of the talks (very frustrating—there was some great material for articles), but what I saw was very interesting and people came up toward the end of the SOG and asked how we'd arranged such a varied and talented group of speakers. Thank you, speakers, there is no question that you are all talented (and varied).

The Deschutes River

Otto Baade leading a discussion about the Slicer

Ron Anderson called a week before the SOG and asked if he could bring a power distribution network, normally used for stage lighting. After he hooked the network into the house's 220V electric heating circuit, it was obvious we no longer had any worries about power. (Boy did we look organized when all the equipment tables had full-length power strips and unlimited power.) Thanks, Ron.

The following folks made the following presentations:

Saturday

10 am: Roger Pryor—Aztec C, Video Tex, BB I monitor mods.

11 am: Huntington Data—BB I hard disk interface.

12 pm: Hampton Miller—On Your Own

1 pm: Andy Bakker—BB I SASI, BB II 5 and 8 inch interface.

2 pm: Glen Widener—BB I character ROM, BB I port extensions, modems.

3 pm: Otto Baade—The Slicer.

4 pm: Rex Buddenberg—Pascal.

5 pm: Don Brittain—Color graphics on the BB I and Kaypro.

6 pm: Ray Buvel—Iforth and a 68000 cross compiler.

7 pm: Jim Ferguson and Russell Smith—General discussion.

Sunday:

10 am: Leroy Searle—Delphi hard disk system.

11 am: Tony Ozrelic—Dyna Disk.

12 pm: Trevor Marshall—BB I extensions, bulletin boards, CCP in ROM.

1 pm: David Mitchell—Unix, and bulletin board design.

Hang on for the 3½!

Russell Smith and Jim Ferguson

The Editor is all wet!

David Mitchell tries out Micro C's accomodations

2 pm: Meeting of BB II users.

3 pm: Bob Carol—BB I disk controller based on the 2795.

4 pm: (Me)—Micro C, the growing connection, and a thanks to all who came.

Bend

A couple of people said that they expected Bend to be a small backwater community consisting of a general store and gas station. They said that they were surprised how prosperous Bend looks.

Maybe I've left the wrong impression. Bend has about 17,000 people. It is the center of commerce for most of central and eastern Oregon, so it has three shopping malls (small compared with California but malls all the same), half a dozen good-size grocery stores (one stays open 24 hours), a private sports club, a golf course, dozens of restaurants (three or four of them competitive with the best in San Francisco), more motels than you can count on your hands and toes, and a dozen sports equipment stores.

Bend also has the highest percentage of professional people of any community in Oregon—this includes doctors, lawyers, dentists, CPAs, you name it. It is the hardest place in the state to begin a practice but many still try. You see, once they've set up a practice, they are pretty much locked in for the rest of their working years.

And, of course there is the community college. It was the first community college in Oregon and it now offers fully accredited bachelors and masters degrees in conjunction with Oregon State University and the Oregon Graduate Center. So you can stay right here and take all your course work for your BS or masters in computer science, for instance.

Bend is not exactly a backwater community.

Micro C

We finally ran an ad for Micro C. It was a full page in *Pro Files*, a new publication for Kaypro users (since the Kaypro is 95% identical to the Big Board I it was a natural).

As a result, we are getting nearly 100 new subscribers a week (compared with 80 per month before the ad ran).

The phone has gone crazy. Fortunately (or unfortunately) the darned thing can't ring while someone is using it. We now have two really neat people doing their best to answer both the phone and the mail (Dorcas Dsenis and Alice Holbrow). Plus, we have Dana Cotant puttering along with the Slicer and the Kaypro and the BB II . . .

Technical Level

I've gotten some comments from folks on both sides of the technical fence. Some think we should cut out all frills such as Bugsbury and references to hijinks and frivolity. There is serious business at hand and we're wasting space.

(continued next page)

Others think we should lighten up. They'd like a centerfold of cartoons (what other kind of centerfold would there be), and they'd like us to go a little easier on the technical stuff so they have at least half a chance to figure out what we're doing. (They have a point.)

On the other hand, the large majority of people tell us to keep doing exactly what we are doing (whatever that is).

So, for those of you who find laughing disconcerting, we'll try not to be too funny (if we fail miserably, be sure to let us know immediately—we might be on to something).

For those of you who are afraid that you're not going to understand anything, we've made a staff decision not to print the street addresses or PO Box numbers in HEX. (At least not anytime soon.) It appears that the post office hasn't really mastered the decimal system, so it will be a while before we can use HEX anyway.

Bulk Rate Blues

Most of you good folks are still bulk rate, and most of you are receiving your magazines slowly but surely. That's great, because at about 11 cents per copy, bulk rate is definitely the least expensive way to mail a bunch of anything, especially Micro Cs.

However, if you move (I've said this before, so just bear with me) you have to let us know or Micro C simply won't get to you. Period!

First class mail will follow you to the far reaches, free. Second class mail (most large-circulation magazines) will also follow you, and that is what the post office is referring to when they ask you if you want them to forward your magazines. You pay for the forwarding.

Bulk mail, however, will not follow you across the street, even if you threaten to rub German shepherd musk on your carrier's legs.

Some people have asked us why we don't mail second class. We may someday. But until Micro C gets a little heavier, there will be only a few cents difference between first and second class (second class is not cheap). So we've stayed with first and bulk.

We've also had a lot of questions about the time difference between bulk mail and first class. First class takes between 2 and 4 days, anywhere in the U.S. Bulk rate mail takes from 1 day to 3 weeks on the west coast and from 2 weeks to 4 weeks on the east coast.

In fact, the policy at your local post office and the attitude of your carrier often make more difference than the distance you are from here. We mail all the bulk rate issues the same day, but they certainly don't arrive the same day.

Practically Stealing

I'm almost afraid to tell you about these because I know there'll be a run on them, but . . . Jim Ferguson has a bunch of Xerox 820 boards (with documentation) for $175 each.

I've known about this for quite a while, but it wasn't until I saw one at the SOG that I realized what a steal they are. They are beautiful! I haven't put mine in a system yet but the word is that they are completely compatible with BB I software.

You might check with Cascade Electronics (507-645-7997) and see if they have received any more 8" drives. If so, you'd only have about $400 in the board and the drives for a two-drive system. Incredible.

Wanted, A Few Good Programs

MicroGroup is going to be gearing up for a national advertising campaign in the not too distant future. We're looking for a couple of really zingy packages, so that we can afford full-page ads in some of the big mags.

SpellSys has really matured in the past few months and we're getting responses from those users that they'd like to see more packages in the same price range, particularly a full screen editor. However, a full screen editor would be a nightmare to support unless it were aimed at a specific set of terminals.

Hopefully, we'll have a super assembler shortly (Murphy willing), and possibly even a spreadsheet (what, another spreadsheet?). I'd really like someone to come forward with a cardfile type data base handler, plus a query type data filer. Then, of course, we'd like to sell a small ADA compiler for $29.95!

Happy Computing

David Thompson
Editor & Publisher

Electronic Buyers Club

Review by Don Brittain

4200 Spruce Street, Apt. 208
Philadelphia, PA 19104

If you are into hardware (by choice or otherwise), I strongly recommend joining the Electronics Buyers Club. The dues are $35 per year, which may seem a bit steep till you hear what you get. First of all, EBC sends you a personalized high-quality 3-ring binder which contains manufacturer's literature and catalogs, as well as price lists. The prices are unbelievably low (some examples will follow) and the service is excellent.

Orders can be mailed in or telephoned in through an 800 number. If you request UPS Blue Label you are virtually guaranteed to have the merchandise in 3 working days. Plus, EBC will be happy to check stock on the items you ordered, so you will immediately know if the items you ordered are in stock. Out-of-stock items are rare. (Editor's note: that's probably why they are out of stock.)

You always have the option of back-ordering or cancelling out-of-stock items. The handling charges are quite reasonable—you pay $2.00 per order plus shipping charges.

You pay no sales tax (unless you live in Missouri) and shipping charges are usually less than than the tax would have been. Catalog updates are sent out periodically (they are sheets which you add to the 3-ring binder) so you're always up to date about what's avalable.

All merchandise is new, fully guaranteed, and of the highest quality. For example, all Z80 components are direct from Zilog—no second sources or black-market copies. Manufacturer literature is available for almost all items—including the complete Z80 line and all logic circuitry—and is absolutely FREE.

Prices

Z80A-CPU	3.20 (29.04 for 10)
Z80B-CPU	9.19 (83.49 for 10)
Z80A-SIO/x	10.53 (x=0,1,2,or 9)
Z80A-PIO	3.63
Z80B-PIO	8.96
Z8002	28.90
Z8671	17.36
Z6132	14.38
4116(200ns)	14.52 for 10 (not 8!)
4116(150ns)	15.18 for 10
74LS14	0.35
74LS240	0.63
74LS245	0.63
2114L-2	16.50 for 10
2114L-3	13.80 for 10
uA78XX	0.60
RESISTORS	0.57 for 100 (1/4 W, 5%)

These prices are meant only to whet your appetite. The 3-ring binder is virtually full of electronics components (including complete lines of 74__, 74L__, 74F__, 74LS__, 74S__, 4000 chips, Zilog products, 6800 series of microprocessors, sockets, voltage regulators, plus tools from OK Machine and Tool, Vector, and AP Products.

They also carry Hitachi oscilloscopes, rectifiers and switching diodes, small signal transistors, LEDs and LCDs.

Unfortunately EBC does not carry the Intel line but what they do carry is unbelievably low priced.

If you're interested in joining EBC, the address is:

Electronics Buyers Club, Inc.
PO Box 617
Columbia, MO 65205

You can also call them at 800-325-0102 if you're sure you want to join (I don't think this number is intended to be an info line, but when I joined I just called up, gave them my credit card number, and even placed my first order right then and there.)

By the way, I don't want to sound too pushy, but if you do decide to join and you mention my name and membership number (Don Brittain, 101689) then they will credit $5.00 to my account.

■ ■ ■

WHITEWATER RAFTING PICTURES

Whitewater rafters who would like color prints of themselves at their dripping best should contact Dorcas at Micro C.

4" x 6"	$ 5.00 each
8" x 10"	10.00 each

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.

5 Mbyte Winchester hard disk drive, new, $375, guaranteed. Tandon 602S. ST-506, SASI Compatible. Works great on BB II with Xebek S-1410 controller. Roman-Data 612-722-4594.

Protect your schematics. Heavy duty vinyl sheet protector, 11X17, 3-hole punched, perfect for BB schematics! Six for $6.00 postpaid. Tony Ozrelic, LA Software, 6708 Melrose, Los Angeles CA 90038.

Dyna Disk assembled and tested, for sale with software. Ready to run on BB1. $400. Rex Buddenberg, 1910 Ash Street, North Bend, OR 97459. 503-756-2180 evenings and weekends.

Kaypro Owners Unite! The Tampa Bay Kaypro Users Group (TBKUG) is looking for frustrated Kaypro owners. Tired of looking for software made for you Kaypro? Look no farther, we have it. We offer an RCP/M (Kaypro 10) that's online 24 hours a day. The present catalog has over 500 files ready for downloading and it's growing weekly. We also have a wealth of Kaypro information and special discount purchase priveledges for members. The Tampa Bay Kaypro Users Group, 14 Cypress Drive, Palm Harbor FL 33563.

EPROM Programmer for the Big Board and Xerox 820. Supports 2716, 2732A, 2732, and 2764 type EPROMS. NEW extensive software included (with source) for easy operation. Software permits read, verify, program, bidirectional PROM to disk transfer and more. All voltages are controlled via software. Designed for easy installation with only two connections (PIO and +5V). Fully socketed, including zero insertion force EPROM socket. All include manual and software on 8 in disk. EZPROMMER A&T $125, kit $90, bare board $50. Check, M.O. or COD. Available from Optronics Technology, PO Box 81, Pittsford NY 14534. Add $3 postage and handling. 716-377-0369.

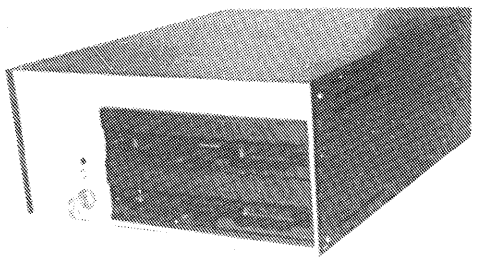
THE BOX The home for your BIG BOARD that you will be proud of. With a POWER SUPPLY that will run the BIG BOARD and two standard or slim-line eight inch drives. It comes fully wired with all connectors and is pre-wired for disk expansion. The BIG BOARD mounts on the inside of the top cover allowing all cables to dress neatly to the rear of the cabinet and to allow ease of access for repair.

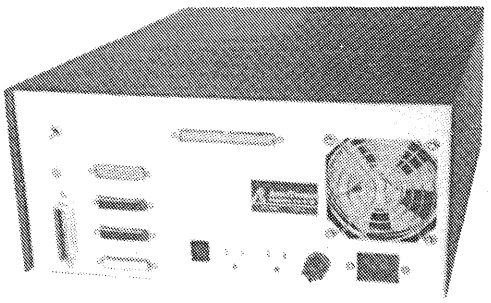
The enclosure comes in single or double wide. The double wide will fit both standard drives or (with the adaptor, $10) the over-sized Shugart SA 800-2, from Cascade Electronics, Inc.

Available without connectors and un-wired but with power supply and primary wiring as a drive cabinet or as a do-it-yourself enclosure for the BIG BOARD, BIG BOARD II, or other SBC.

STANDARD FEATURES INCLUDE:

- Power Supply
  +5.0V @ 4.0A w/OVP, +2.5A, ±12V @200mA
  all voltages have over current protection.
- Fan
- Key lock power switch
- AC outlets, one switched
- Composite video jacket
- Disk drive expansion pre-wired (50 pin + DC + AC)
- Color - beige and chocolate
- 6″H x 13″W x 16″D or 24″W for the double wide.
- Reset switch
- Bell ckt. and PIEZO speaker
- Solid state AC relay
- Reverse video switch
- Optional - adapters & plates





THE BOX (La Caja)

Single wide, wired.....$379
Double wide, wired.....$399
Single wide, un-wired.....$279
Double wide, un-wired.....$299

Shipping & Handling.....$ 10

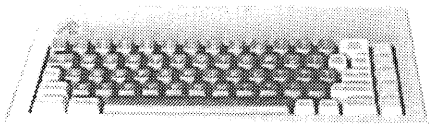
ADAPTORS:

SA800-2.....$ 10
BIG BOARD II.....$ 10
Dummy Plate.....$ 10

KEYBOARD The keyboard was designed to complement the ''La Caja'' enclosure in color, design and function and to be 100% compatible with the BIG BOARD.

FEATURES:

- 66 keys
- ASCII 8, positive logic
- Sculptured key caps
- Power requirements: +5.0V @ 150mA, −12V @ 20 mA
- Color, beige and chocolate
- Two-key roll over
- Delayed negative strobe
- Five user defined keys
- 1.5″-2.5″H x 13″W x 8″D



KEYBOARD w/o cable.....$159
w/cable for wired unit.....$179

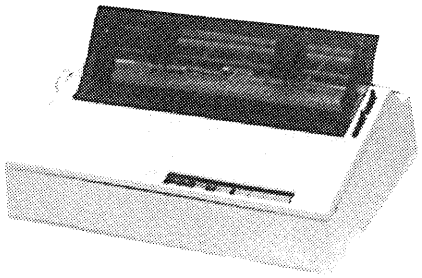
PRINTER The BROTHER HR-15 daisy wheel printer is a compact printer that will handle paper up to 13.5 inches.

A variety of operations have been added to increase clarity, precision and to emphasize important points. The HR-15 prints super-script and sub-script characters with the ability to adjust character spaces prportionally and provides automatic underlining with either bold or red print. For ease of operation both daisy wheel and ribbon are enclosed in cassettes, making changing trouble-free.

In closing, the BROTHER HR-15 makes an excellent choice for word processing and general printing.

FEATURES:

- 2K byte buffing
- Bold printing
- 13 cps
- RS-232C or CENTRONICS parallel
- Bi-directional logic seeking head, 1/120, 15, 15, 10 positions per inch
- Bi-directional friction platen, 1/48, 6, 4, 3 positions per inch
- Options - Tractor feed, Auto cut sheet feed
- Color - beige
- Graphic printing
- Proportional
- Cassette type daisy wheel
- 6″H x 19.5″W x 13″D



PRINTER Brother HR-15.....$595

Shipping & Handling.....$ 5

POWER SUPPLY for the BIG BOARD (+5.0V @ 4A w/OVP, +24V @ 2.5A, ±12V @ 200mA). All supplies have over current protection (4″H x 3″W x 11″D, 6.5 lb.)

POWER SUPPLY.....$95 +$5 S&H

TRANSFORMER for the BIG BOARD as above (3″H x 3″W x 4″D, 5.5 lb.)

TRANSFORMER.....$29 +$3 S&H

ASTROTRONICS ALSO SELLS DISKS, RIBBONS, DAISY WHEELS AND PAPER.

AstroTronics MICROSYSTEMS

California Residents
Add 6% Tax
(714) 734-6006

1137 TOPAZ ST.
CORONA, CA 91720

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 kinds of information do you need right now?

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Quantity	Description	Price Each			Total
		U.S.	Can & Mex	Other Foreign	
	USER'S DISKS—8" SSSD, CP/M #'s	$15	$15	$20	
	BACK ISSUES #'s	$3	$3	$5	
	SUBSCRIPTION (1 year—6 issues) ☐ New ☐ Renewal	☐ $16 (Bulk) ☐ $20 (1st Class)	☐ $20 (Air Mail)	☐ $26 (Air Mail)	
	OTHER ITEMS:				

Prices include media, package, 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

NAME _____ **PHONE (?)** _____

ADDRESS _____

CITY _____ **STATE** _____ **ZIP** _____

MICRO CORNUCOPIA • P.O. Box 223 • Bend, Oregon • 97709
(503) 382-8048

BIG BOARD USERS DISKS

$15.00 each (US, Can, Mex)

$20.00 each (Other Foreign)

USERS DISK #1
1-Two fast disk copiers
2-The manual for Small C+
3-Crowe Z80 Assembler
4-Two disk formatters
5-Modem7
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 I
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-Modem7 for Port A
3-Modem7 for Port B
4-PACMAN, the arcade game
5-FAST, buffers the disk to speed up assemblies
6-NOLOCK, removes BB I shift lock
7-VERIFY, cleanup & verify a flaky disk
8-DUMPX, enhanced for BB I
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 I (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 transfer 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
11-Extended CRC checker, creates file & checks file
12-Super disk formatter program for BB I

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 I 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, Biorythm 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 I
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 which almost looks identical 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-MODEM7+, Modem7 plus BBPAT, lets you talk to anything from port A

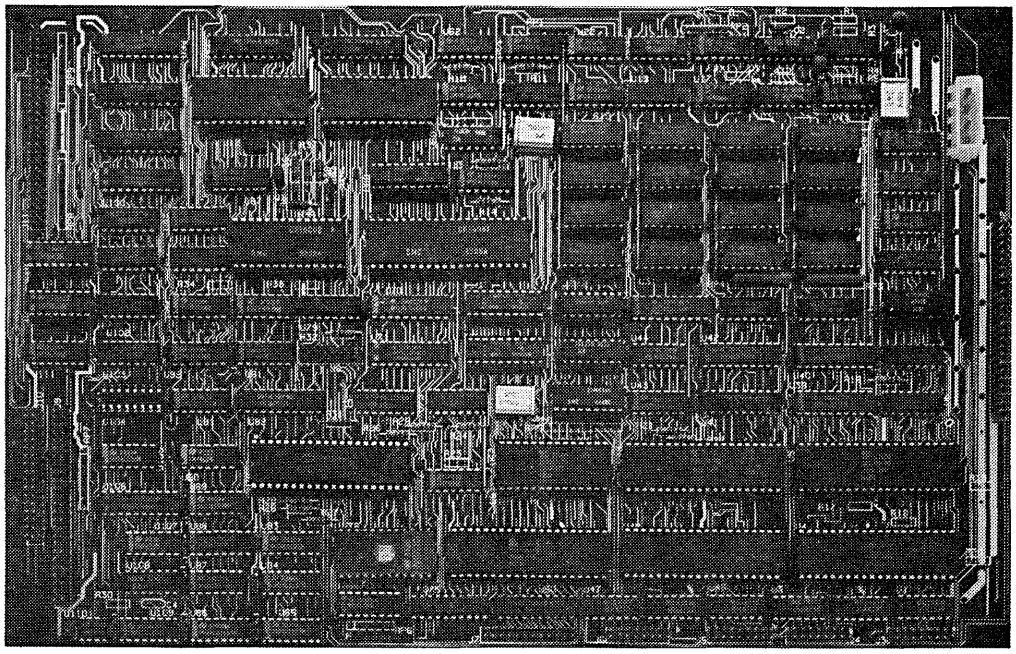
USERS DISK #17 - Small C version 2
SMALLC2, this substantially expanded version of Small C now includes for, goto, label, switch (case); external declarations; new preprocessor commands; expanded I/O includes redirection; initializers; plus 12 new expressions. The I/O and runtime libraries have been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

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.

ATTENTION BIG BOARD USERS!
WHY USE OUTDATED TECHNOLOGY?

INSIGHT ENTERPRISES IS NOW DELIVERING A NEW STATE-OF-THE-ART CP/M Z80-A SINGLE BOARD COMPUTER



On board video • 128K of Dynamic RAM and 4K of STATIC (video memory) • DMA • SASI (hard disk interface) • Floppy disk controller (4 drives any combination of 5-1/4" or 8") • 4 serial ports • Full Centronics printer port • Expansion bus • Wide line and thin line graphics • Extended track buffer • Printer buffer • Fully interrupt driven • Parallel or Serial keyboard • Compact Size • (8x14 inches)

$750.00 single quantity OEM pricing 2-4 weeks delivery $10.00 shipping

IEC INSIGHT ENTERPRISES CORPORATION
373 N. Western Ave., Suite 12 Los Angeles, CA 90004 (213) 461-3262

Dealer, OEM, International Inquiries Welcome
MANUFACTURING LICENSES ARE AVAILABLE WORLDWIDE UPON REQUEST

MICRO CORNUCOPIA
The Single Board Systems Journal

P.O. BOX 223
BEND, OREGON 97709

BULK RATE
U.S. POSTAGE
PAID
Bend, OR
Permit No. 447