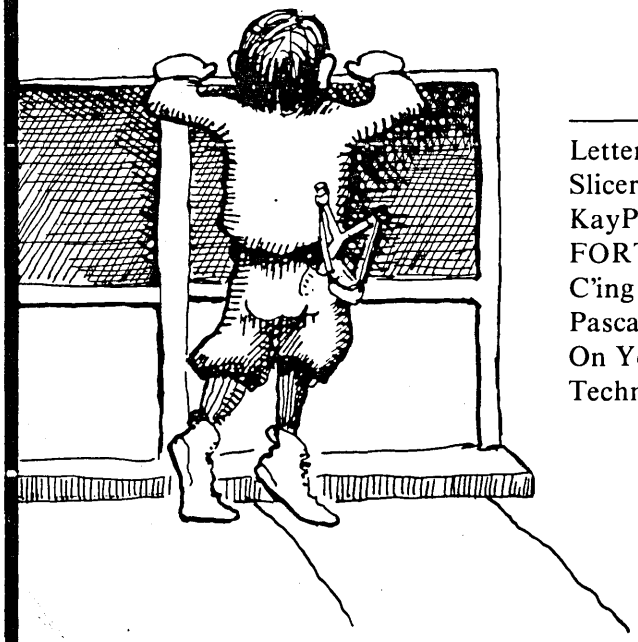
$3.00

MICRO CORNUCOPIA

THE SINGLE BOARD SYSTEMS JOURNAL

October 1983 No. 14

TABLE OF CONTENTS



REGULAR FEATURES

NEW LOWER PRICES

"THE ORIGINAL BIG BOARD"

OEM - INDUSTRIAL - BUSINESS - SCIENTIFIC

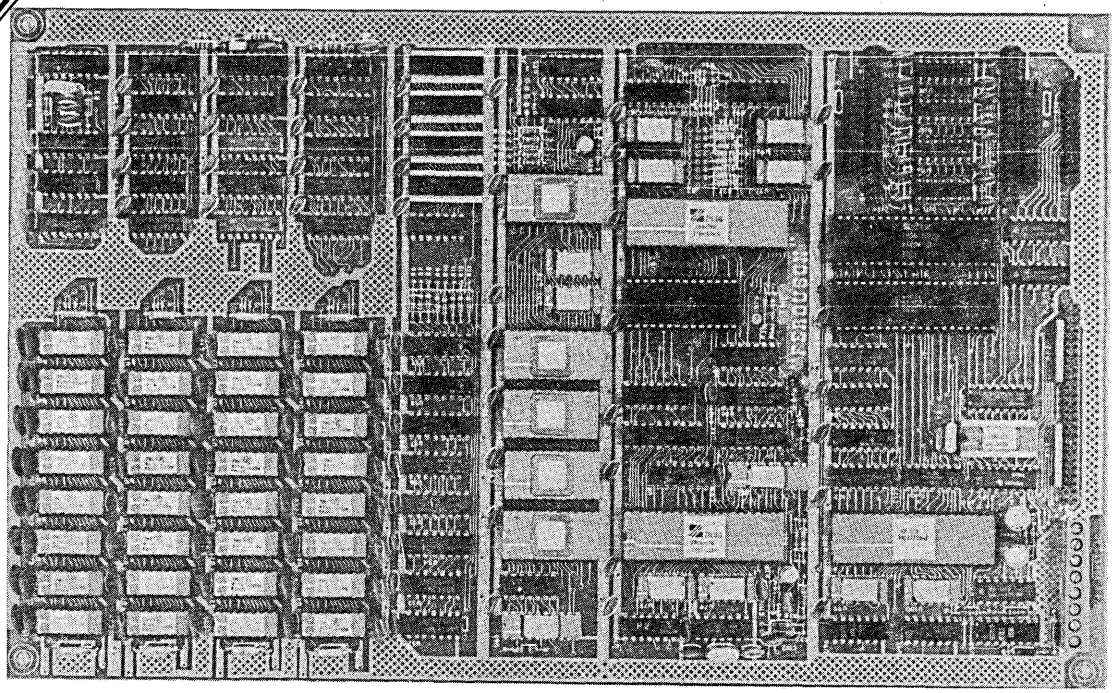
SINGLE BOARD COMPUTER KIT!

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, Big Board II, Xerox 820, Kaypro, and Slicer.

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

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

ADVERTISING RATES: Available on request.

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

SOFTWARE, HARDWARE, AND BOOK VENDORS: We would very much like to review your CP/M & MSDOS compatible products. Send materials to the Review Department, Micro Cornucopia.

MICRO CORNUCOPIA

October 1983 **The Single Board Systems Journal** **No. 14**

Oh, Bother

Reach Out and Bother Someone

Dana and I have received a number of calls and letters from folks who are using software on the user disks or who are reading our columns and articles and have questions about the articles or the programs. And, we've heard from folks who just say thanks for the great software and information. Of course we appreciate the thanks and we try our best to answer the questions.

However, when I suggest that a reader contact an author, the reaction usually is, "I'm sure the author would be too busy to respond to me."

Meanwhile we have gotten calls and letters from authors wondering whether anyone has tried their software or read their articles. (See FORTHwords in this issue for Arne's sentiments.)

One author lamented, "Even a complaint about my writing style would be better than complete silence."

So there you have it. You readers are afraid to bother the writers and the writers are waiting anxiously to be bothered. What can I say?

S-BASIC

Those of you who know me well, know I spent a good deal of time programming in BASIC (during my past life at Tektronix). It was pure and simple Dartmouth BASIC and I hated it. If you want to write good readable, structured code, then you have to write in something other than this classic BASIC. (I really enjoy working in Pascal, C, and dBASE II.)

So, when I got my first Kaypro, I ignored its single BASIC offering. After all, every system comes with one BASIC or another doesn't it? And a BASIC by any other name smells.

Well, one of the BASICs that came with my Kaypro is called S-BASIC and it's really exciting. You can write readable structured source code in this BASIC and then compile the source into a real .COM file. Jerry Pournelle of Byte lumped S-BASIC in with CBASIC when he reviewed the Kaypro. That tells me that he didn't check out S-BASIC any better than I had. (Kaypro is now shipping S-BASIC, CBASIC and MBASIC.)

My excitement is not without two reservations. First, S-BASIC is not well documented, so you will spend a lot of time writing and running little programs to test the syntax and structure of some of the more obscure (but necessary) features. Second, S-BASIC has some definite bugs (sequential file access problems, for instance).

S-BASIC was written by Gilbert Ohnysty under the name Topaz. Kaypro bought S-BASIC lock, stock, and name—and I understand that Gilbert is now an employee. Little wonder that Kaypro 4s and 10s include it. (You can purchase S-BASIC for only $75 directly from Kaypro.)

An informal poll has led me to conclude that less than 5 percent of Kaypro owners have tried S-BASIC. Part of the problem might be that it is a compiler. But the main problem, I'm sure, is the documentation.

I think this inexpensive but powerful language deserves a place in Micro C, as a regular column.

Another BG Bargain

BG Micro has come up with another winner. Billy Gage is selling bare Xerox 820-1 boards for $39.95 each (he has thousands of them and I hear they are selling very well). He is including documentation and a monitor ROM in the package.

This is another genuine steal from BG. (Note that all the BB users disks run on the 820.) You can reach BG Micro at PO Box 280298, Dallas Texas, 75228, 214-271-5546. Tell Billy you heard about him in Micro C. Who knows, maybe he'll advertise with us again so you can read about all the great bargains first-hand.

(continued on page 43)

LETTERS

Dear Editor,

I just read issue #12 and noted your remarks about broken drive door latches on the Kaypro II. I believe the problem is sloppy operating rather than bowed bezels. People don't seat the disk in the slot well enough and the two metal arms sometimes contact the front edge of the envelope.

Anyway, the parts needed for repair are available from Tandon. The TM100-1, 2, 3, or 4 units take latch assembly 171140-001 (costs $2.14). The B drive repair is easy, I don't know how you get to the A drive.

Your publication is most welcome and several of us have talked it up at our local (300+) Kaypro users group. One thing you might consider—publishing a list of the various modifications by serial number. I own #4868, an early machine, and would like to know how mine differs from later models. I have been particularly impressed by its ability to run for hours with no apparent problems due to heat.

A local guru says that it will cost me several hundred dollars to convert my machine to the fast version as I have an old "A" board which must be replaced. I think I may be a mark! It will be people like you and publications like MC that save us from a debt worse than fate! (P.S. A piece of half-inch thick wood with a three-inch door stop screwed into each end—a little farther apart than the width of the keyboard—makes a fine prop for the KP and frees the keyboard.

Col Malcolm A Hormats
10401 Brosvenor Place Apt 1602
Rockville, MA 20852

Editor's note:

I'm indebted to you for all the great information, Malcolm (but then I suppose that's fate). Anyway, we speeded up both of our Kaypros. The old one (#5005) should be identical to yours. U67 is a 74LS04 instead of a 74HC04, and Q1, the transistor pull-up is in the Z80 clock circuit, but it speeded up just as easily as the later one (number over 30,000). In fact, your old one does faster disk accesses than the newer ones. That's one reason why we started with the old code when we designed our PRO-MONITOR ROM. (Be sure to do the simple "drive timing fix." from issue #11 to your machine, it'll make it even more solid.)

You've got a very good machine. In fact, Kaypro is now sorry that they went from the vertical drives to the horizontal stack because the change caused many heat problems.

Your machine's most obvious problem is its character set. If you have access to the character ROM on one of the new machines, you can copy the ROM into a 2716-1 (350 ns). Use a Data-IO or other ROM programmer. Hopefully we'll have a ROM programmer construction article for the Kaypro within the next few issues, as well as software to run it.

Anyway, copying the ROM into a fast part will clear up the white flashing dots or fleckies on your screen in addition to giving you a better looking character set.

If you want to go all the way up to really neat characters, check out our character ROM. (Someday I'll tell you the history of the Kaypro character set and how I created those original characters one afternoon at Tektronix. I wouldn't admit it now, of course.)

If your board ever goes belly up, your dealer can exchange it with Kaypro for a brand new one. The exchange costs your dealer $100 (no matter how old your board is).

Also, I watched a drive latch break (with no disk in the drive) just minutes after the Kaypro had been removed from the box. Meanwhile, we bang disks and doors around with great abandon (here in the office) with no problems.

Dear Editor,

I have a Radio Shack "Plug'n power controller." In order to connect it to the Big Board and program it, I need information on how it works. I contacted Tandy Corp in Forth Worth and they would not give (or sell) me the information. Can anyone help?

Raymond P Shea
914 W Center St
Duncanville TX 75116

Dear Editor,

There is a bug in the Kaypro's cursor routine. Supposedly non-destructive cursor movements actually destroy underlines.

Is this fixed in your PRO-MONITOR?

David Hillman
2006 NE Davis
Portland, OR 97232

Editor's note:

Yes David, Our ROM with the non-blinking block cursor will not eat your underlines because we've put it on a strict diet of solid blocks. (Someone else might eat your underlines, but that's beyond the scope of Micro C.)

Dear Editor,

I'm using a BB I with two 8" drives. Now I am going through the book *Soul of CP/M* by Mitchell Waite and Robert Lafore, which seems to be really helping me understand how CP/M works. The book is published by Howard W. Sams and is $18.95. Keep up the good work.

Gordon P. Batey
Rt 5 Box 164
Staunton VA 24401

Dear Editor,

I was pleased to see my BB II printer interface article in print (August, 1983 issue, page 8). Unfortunately, there is a typographical error in one of the listings (my fault, not yours). Line 50 of figure 3 should read "JR Z,LIST" rather than "JR Z, LISTST." The corresponding opcode becomes: E76D: 28 FA

In the original form, the driver will simply return to the calling program if the printer is not ready to accept a character. With the correction, the driver will wait until the printer is ready.

John Taylor
Taylor Electric Company
PO Drawer 11 N
Milwaukee, WI 53201

Dear Editor,

I recently converted my Big Board to 4MHz, using the modifications outlined in Issue 3 of Micro C, but some problems have arisen. When I reset the computer and press return, 50 percent of the time I get a fifteen character string of garbage on my H/19 terminal instead of the PFM monitor prompt. I have tried setting the terminal at lower baud rates, but that has only made things worse. I have also tried changing the monitor PROM to a faster version (350ns) but have not seen any change. The computer runs perfectly (no reset problems) when I use the on-board video and a separate keyboard. I would appreciate any suggestions that would help rectify this problem.

L. C. Chen
654 Vernon St. #6
Oakland, CA 94610

Editor's note:

There is a baud rate timing loop called BAUD2 located at F081 in RAM (see issue #1 page 11). You'll have to increase the time it takes to go through that loop by a factor of 4 divided by 2.5 (the amount you speeded up the processor) or you'll need to change the baud rate table values. One way to increase the loop time is to call a routine just beyond the monitor and then return.

Dear Editor,

Congratulations to Henry Holcolm (Letters, Micro C., #12) for solving my video jitter problem.

When I tried a slight variation, I was quite impressed by the rock solid display. The only minor problem was the missing 1/3 of the display on the right side. My monitor, a Ball Brothers TTL120, has an adjustable horizontal position but was unable to bring the display into proper position. By alternately eliminating U51 and 1/2 of U38 and triggering the display on the trailing edge of the pulse, it became apparent that both one-shots were contributing to the jitter.

Obviously a more stable time delay was needed, and the unused gate in U10 at the top of Schematic Page 2 was available. Removing U51, bending out U38 pin 1, and jumpering from U10 pin 12 to U38 pin 1 (on IC chip) gave a stable display, which, when properly positioned, was right at the end of my monitor horizontal position adjustment. This amount of delay may, however, be optimum for some monitors.

Additionally, cutting the etch between U10 pin 13 and U11 pin 2 (bottom of board) and jumpering U10 pin 13 to U23 pin 11 gives a pulse that occurs slightly later and is well within the range of my monitor.

Keep up the good work with Micro C. My issues of it become quite worn, while my Byte barely gets the cover turned.

Ron Scott
133 Malvern St.
Scarborough, Ont.
CANADA M1B 2H1

Dear Editor,

Sorry I took so long to renew. I was going to a clinic for recovery from a strange disease—Micro Cornucopiitis (copee-eye-tiss), and I had just gone cold turkey. Then my dreams of a normal life were shattered when I received No. 12 without asking. Now I am fully addicted once again, you louses!

I can't get my Crowe Assembler to work with my double density system. What do I do???

Stuart Russell
Pragmatronics, Inc.
2015 10th St.
Boulder, CO 80302

Editor's note:

Dana has just finished reworking the Crowe Assembler so that it will work on any CP/M machine (single, double, double double, Kaypro, Osborne, Northstar, BB . . .). The new version is on user disk #20.

Dear Editor,

I have a nice, new serial keyboard that I would like to use on my BB II. This would seem to pose no problem, since the source code for the CBIOS supplied by Cal-Tex Computers has an EQU (SERIAL) which can be set to TRUE, enabling an SIO/CRT combination. The problem comes when I try to assemble and link the modified source code (I have also added some code to make header J11 a parallel printer port for LST:). M-80 works fine, but L-80 gives an 'OUT OF MEMORY' error message, regardless of what offset I use and even if I make no changes in the source code! LINKMT, although able to handle the linking, generates an incompatible .HEX file. Maybe this is why Cal-Tex supplies both .MAC and .HEX files for the standard CBIOS variants. I can, of course, get around the printer code problem easily enough with an OVERLAY.COM file, but the keyboard problem requires assembly and linking. Any ideas?

For those like me who are learning but are less than 'computer whizzes,' I recommend USER'S GUIDE TO CP/M, a bimonthly publication featuring articles and tutorials on CP/M and the more popular application programs. ($18/year, PO Box 3050, Stanford, CA 94305).

Gary Cooper
1943 Palo Alto Way
Menlo Park, CA 94025

Dear Editor,

I am writing to let you know of my frustration with JRT Systems.

On April 25th I placed a Visa order for JRT PASCAL. (My account was billed on May 2nd.) When I placed the order, they said there would be a 6-8 week delay. Unfortunately, the 6-8 weeks became 5 months, and several phone inquiries did nothing to speed my order.

Today marked the end of my patience—I called and demanded a refund. I hope my experience is an isolated one. If not, perhaps the FTC needs to increase its case workload by one.

David M. Bauscher
3034 Rosedale Blvd.
Louisville, KY 40220

Dear Editor,

Do have any info on interfacing to amateur radio for CW/RTTY, ASCII, etc.? I have a Kaypro II.

Bud Atkinson
6708 San Louis Obispo, NE
Albuquerque, NM 87109

Dear Editor,

I recently changed from my BB I to a BB II. The assembly instructions permitted me to get the board up and running without too many hassles. The only problem I have is the system's occasional tendency to leave a cursor here and there on the page, or sometimes a stripe of reverse video—especially when using Wordstar.

Also, when I'm transferring files from a single-sided to a double-sided floppy, all is well when I'm on Side 1 of the double-sided drive. Once I get onto Side 2, the BIOS apparently forgets to switch back to Side 1 when accessing the single-sided floppy. The effect is to send the system wandering off into limbo. Maybe someone else has noticed this and will publish a fix in Micro C. Otherwise, I am delighted with the BB II.

John F. Dalstead
Lot 7, Mt. Gisborne Road
Gisborne, Victoria
AUSTRALIA 3437

Editor's note:

There is a new monitor ROM available for the BB II which should fix most of those problems (I'm not sure about the double sided mess). Anyway, the ROM should be available from Cal-Tex now. It probably wouldn't hurt to send Bill a 2732 when you ask for it.

Dear Editor,

My Big Board has been plagued with a problem since its conception. When first turned on, it will not respond to the serial port. A reset does not correct the problem. After a moment of sitting, the CRT displays the PFM sign-on very slowly and prints '?'s after the prompt. The '?'s continue for a minute or two and pick up speed. After the question marks stop, I can hit the return on my terminal and the system will reset and respond. Apparently, the problem is not the PIOs because I swapped them, and there was no change. If I fan cool the board, I cannot boot the system. What's Wrong?

Christopher Farrar
75 North Street
Saco, ME 04072

Editor's note:

You say you are using a terminal so I wouldn't suspect the PIOs. The SIO (or 1489s) might be generating garbage, or your terminal might be strange. If it is not in the serial interface (a good way to check is to try the built-in monitor) then I'd look at the monitor ROM, the clocks, and the Z80.

(continued on page 36)

Installing The BB II

By William L. King

8222 Sprenger N.E.
Albuquerque, NM 87109

I am very happy with my BB II and would not trade it for anything. The motor turn-on is long enough so I don't get "disk not ready" messages, and I really like the display.

I purchased the "UNKIT," which came with the ICs installed at no extra cost (thanks Cal-Tex). *(Editor's note, Bill stuffed and tested the first 30 unkits to verify that there weren't going to be problems. 28 of the 30 came up immediately.)*

So I only needed to connect it up and turn on power. That is what I thought at first. I would like to see a manual at least as good as BB I. I have read everything I could in Micro C on the BB II. Fortunately the BB II monitor is almost identical to the BB I monitor. So, this is how I installed my BB II.

Disk AC control

This control is known as "motor" on the mini-floppy or "SSR" (on the power connector) for the 8 in. My Silicon motor control switch requires a TTL signal, so I connected a 1k ohm resistor between plus 5V and the positive input to the switch. I then connected the minus input to the silicon switch to SSR on the BB II.

Connecting the CRT

The connectors for the BB I and BB II for the CRT are the same so I just plugged it in and turned it on. After adjusting the horizontal, I found I had a picture but it was way off center, and I could only see columns 1 thru 59.

So, I set aside the BB II until I located a data sheet for the 6845. It turned out that the 6845 set up in the ROM was incorrect. R2 must be larger than R0. My ROM listing shows R2 smaller, and this turned out to be the real problem. (R2 is the horizontal position.) Nor did I like the vertical position R7. So, instead of using Taylor's suggested change, I now use:

ODC,2
ODD,52
ODC,7
ODD,19

Connecting the Disk Drives

I chose the SS60E8 CBIOS because I have 2 DSDD drives which I wanted to run as 4 SSDD units. I followed the instructions supplied on the disk (they were adequate) and saved the new system by typing SAVE 43 CPMII.COM.

However, I got the shock of my life. The disk keeps the heads loaded. In my book this is unthinkable. I use the door lock function of the DT8 so I had to wait for the motors to shut down before I could remove a disk.

I called Cal-Tex about this, but all I got was a justification of the way they did it.

So it was up to me. The schematic shows the HLD (head load) signal from the 1797 connected to pin 8 of U-8 (the driver) and pin 10 of U-14 HLT going to pin 23 of the 1797. It turns out, however, that the HLD and HLT signals of the 1797 are not used. Jim Ferguson probably hoped to save an IC by doing the HLT signal via software. *(Editor's note: Bill Siegmund just sent me a new set of schematics for the BB II; you all should get them in the near future.)*

When this didn't work he changed the signal HLT supplied by U-14 to HLD and connected it directly to U-8 pin 8. It turns out that it is difficult to do a HLD for each read, write or seek. So, the heads are loaded when the motor is turned on and unloaded when the motor is turned off. I changed this by using a one-shot to generate HLT from HLD as recommended

Figure 1 - Big Board II Printer Cable

```
Data port (out2)[DA]
Status port (in1)[D9] ...Bit 7  0 = busy, 1 = not busy
```

Port connector Pin	Name	Printer connector Pin	Name
1	020	2	DATA 1
3	021	3	DATA 2
5	022	4	DATA 3
7	023	5	DATA 4
9	024	6	DATA 5
11	025	7	DATA 6
13	026	8	DATA 7
15	027	9	DATA 8
17	ODAV2	1	DATA STROBE*
		19	GND
19	TAKE2*	10	ACK*
20	OE2*	29	GND
even 2-18	GND	19-28	GND

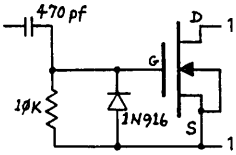
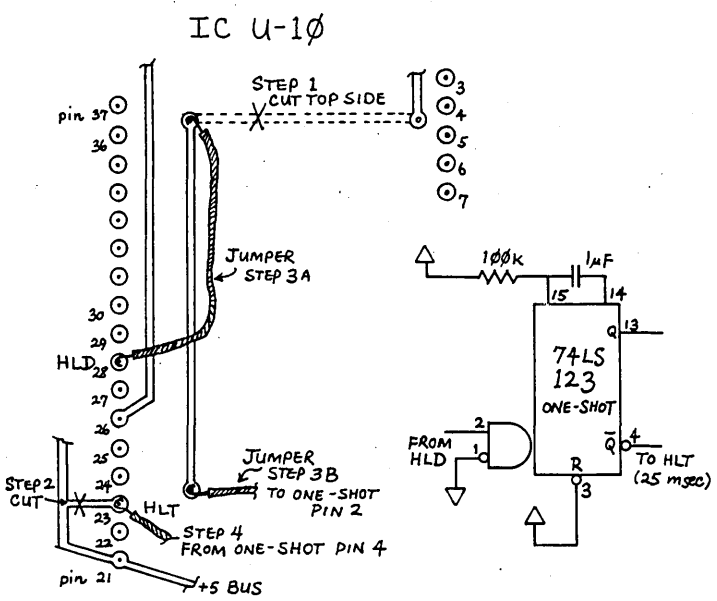


Figure 2 - Big Board II Floppy Interface Modification



by the manufacturer. I did this as follows (also see Figure 1).

1. Remove U-10. Locate the run coming from U-14 pin 10 going to U-8 pin 8 which is under U-10 close to pin 36. Cut this run (see drawing). Check that the run is open with an ohm meter. Then replace the IC.
2. Turn the board over and cut the run between U-10 pin 23 and the +5 bus. Again check with an ohm meter.
3. Now jumper between pin 28 and the run going to U-8 pin 8. Also jumper to the input of the one-shot (see Figure 1).
4. Connect the output of the one-shot to pin 23 of U-10.

Adding a Line Printer

I have a parallel input printer (standard Centronics type interface) so I set up a parallel printer output.

Had I known that I would need a 74LS123 for the drives I would have used the other half here. Instead, I used a trick I have used before to drive printers from a level output. I just converted the true signal to a false signal with an FET (field effect transistor). This gave me the correct strobe edge for my printer. The connector at the printer is big enough to hold the extra components, so I didn't need to modify my board. See Figure 2 for the cable wiring diagram.

Modifying the CBIOS

I like the way Russell Smith used the software switches to select the different options for the CBIOS. Because I might use a serial printer someday, I elected not to remove that code. I just added one more switch for the parallel output.

Below are the changes I made to the CBIOS. The changes are noted by a per cent sign (%) in column 1. DO NOT type the %. See Figure 3.

■ ■ ■

Figure 3 - Big Board II Custom Bios for Parallel Printer

```
;*****************************************************
;*                                                   *
;*     --  CUSTOM BIOS FOR CP/M VERSION 2.2  --      *
;*                                                   *
;*     DOUBLE DENSITY CBIOS FOR BIGBOARD-II          *
;*                                                   *
;*     COPYRIGHT (C) 1982 BY RUSSELL SMITH           *
;*                           818 WEST HICKORY        *
;*                           DENTON, TEXAS  76201    *
;*                                                   *
%;*     ORIG ISSUE DATE:  21-AUGUST-82                *
%;*                                                   *
%;*     PARALLEL PRINTER ADDED  BY WLK                *
%;*     LAST EDIT DATE:   05-SEPTEMBER-82             *
;*                                                   *
;*****************************************************
;
        ASEG
;       .SFCOND                 ;DON'T LIST FALSE CONDITIONALS

^  Note the comment was added to allow me to use my old M80

        .Z80                    ;USE ZILOG MNEMONICS
;
TRUE    EQU     1
FALSE   EQU     0
;
IOBYTE  EQU     0003H           ;INTEL CONFIGURATION BYTE
USRDRV  EQU     00H             ;USER/DRIVE TO LOG ON AFTER COLDSTART
SERIAL  EQU     FALSE           ;DISABLE SIO/CRT CONSOLE COMBINATION
%SERPRN EQU     FALSE           ;DISABLE SIO PRINTER
%PARPRN EQU     TRUE            ;ENABLE PARALLEL PRINTER
;
MSIZE   EQU     60              ;MEMORY SIZE IN KBYTES (RANGE 20..60)

        *

        *

        *


;
;
;       ... EQUATES FOR PRINTER DRIVER PORTS/HANDSHAKE MODE ....
;
%       IF      SERPRN
LSTDAT  EQU     80H             ;SIO CHANNEL A DATA PORT#
LSTCTL  EQU     81H             ;SIO CHANNEL A CONTROL PORT#
LSTBAUD         EQU     89H             ;SIO CHANNEL A BAUDRATE CTC PORT#

XONXOFF         EQU     TRUE            ;ENABLE XON-XOFF HANDSHAKE

CTS     EQU     00100000B       ;CLEAR-TO-SEND BIT POSITION IN SIO RR0
SYNC    EQU     00010000B       ;SYNC/HUNT BIT
DCD     EQU     00001000B       ;DATA-CARRIER-DETECT BIT
TBE     EQU     00000100B       ;TRANSMIT-BUFFER-EMPTY BIT

TXMASK  EQU     DCD+TBE         ;LOOP UNTIL DCD AND TBE BITS ARE TRUE
%       ENDIF
%;
%IF     PARPRN
%LSTDAT EQU     0DAH            ;PIO PRINTER DATA PORT#
%LSTCTL EQU     0D9H            ;PIO PRINTER STATIS PORT#
%       ENDIF
;
;
;
;
;
LISTST:
%       IF      SERPRN
%
        LD      A,00010000B
        OUT     (LSTCTL),A      ;DO SIO CLEAR STATUS/INTERRUPT CMD
        IN      A,(LSTCTL)
        AND     TXMASK          ;DISCARD BITS NOT BEING TESTED
        CP      TXMASK
        LD      A,255
        RET     Z               ;RETURN WITH A=255 IF PRINTER READY

        XOR     A
        RET                     ;ELSE RETURN WITH A=0
```

(Listing continued next page)

(Big Board II Custom Bios for Parallel Printer continued)

```
%       ENDIF
%;
%       IF      PARPRN
%
%       IN      A,(LSTCTL)
%       BIT     7,A             ;DISCARD BITS NOT BEING TESTED
%       LD      A,255
%       RET     NZ              ;RETURN WITH A=255 IF PRINTER READY
%
%       XOR     A
%       RET                     ;ELSE RETURN WITH A=0
%
%       ENDIF
%;
 ;
 ;
LIST:
%       IF      SERPRN
%
        CALL    LISTST
        OR      A
        JR      Z,LIST          ;LOOP TILL SIO CAN SEND A CHARACTER

        LD      A,C
        OUT     (LSTDAT),A      ;OUTPUT ASCII TO SIO

        IF      XONXOFF
        IN      A,(LSTCTL)
        BIT     0,A             ;TEST SIO RX DATA AVAILABLE
        RET     Z               ;EXIT IF NO INPUT FROM PRINTER

        IN      A,(LSTDAT)      ;ELSE GET CHARACTER AND TEST FOR
        RES     7,A             ; 'XOFF' CHARACTER (CONTROL-S)
        CP      'S'-64
        RET     NZ

LIST2:  IN      A,(LSTCTL)
        BIT     0,A
        JR      Z,LIST2         ;LOOP TILL NEXT HANDSHAKE CHARACTER

        IN      A,(LSTDAT)      ;INPUT AND DISCARD 'XON' CHARACTER
        ENDIF

        RET

%       ENDIF
%;
%       IF      PARPRN
%
%       CALL    LISTST
%       OR      A
%       JR      Z,LIST          ;LOOP TILL PIO CAN SEND A CHARACTER
%
%       LD      A,C
%       OUT     (LSTDAT),A      ;OUTPUT ASCII TO PIO
%       RET
%
%       ENDIF
 ;
 ;
 ;
```

****

For the file ONESECT.MAC make the following changes:

```
        LD      (IOBYTE+1),A    ;SET DEFAULT DRIVE/USER FOR COLDSTART
        LD      HL,SCRATCH
        LD      (HL),0
        LD      DE,SCRATCH+1
        LD      BC,SCRLEN-1
        LDIR                    ;ZERO-OUT SCRATCH MEMORY FOR BIOS
%
%       IF      SERPRN
%
        LD      HL,LSTINIT
        LD      B,6
        LD      C,LSTCTL
        OTIR                    ;SET PRINTER PARITY/LENGTH/STOP BITS
        LD      B,2
        LD      C,LSTBAUD
        OTIR                    ;SET PRINTER BAUDRATE
```

(Listing continued next page)

In Search of the Perfect Terminal

Review By David Thompson

Those of you moving into the Slicer world, or who are interested in moving up to something fancier than the ADM-3A for your Big Board I, have probably noticed that there are more than a few terminals to choose from.

I've been looking too, and the choice is overwhelming. A few things have stood out during my initial search. (I'm treading on fairly unfamiliar ground so the following is definitely open to attack from the ranks.)

A lot of non-graphic terminals are DEC VT-52 compatible (including the Heath/Zenith H-19). A lot of graphic terminals are DEC VT-100 compatible (at least as an option).

Most terminals have hardware or software controls that allow them to emulate one or more less-powerful terminals. For this reason almost every terminal worth its salt (and a few that aren't) will emulate the ADM-3A (which tells you something about the 3A).

However, the Xerox, the Big Board, and the Kaypro all say that they look like the ADM-3A, but they all have slightly different screen control characters and they are each different than the 3A. In some cases, the differences don't matter, but the screen driver built into dBASE II will do strange things on the Kaypro II if it thinks it's talking to an ADM-3A. See Figure 1, for a stroke-by-stroke account.

Before you choose a terminal, you need to figure out what it needs to do (graphics display, character translator, printer buffer . . .) and then you need to find what fits those needs and see if you and it are compatible. (Similar to marriage, though the initial cost of a terminal can be higher.)

If you are a touch typist, you definitely need to try out the keyboard for feel and layout. In fact, you should try the one you'll be buying rather than a display model. Manufacturers often have two or three different brands of keyboards that they install into a single terminal model. (But then most single models are terminal.)

Look closely at the monitor. Is every character on the screen sharp? Is the image rock solid? Can you select a blinking or non-blinking cursor? Do you like the color (green or amber)? Is the keyboard movable? Will it really run 9600 baud with your computer?

Also, I've received the following terminal evaluation that was distributed via the Unix Net. I have no way of identifying the author, but I know that Tektronix folks used this information when deciding which terminals to purchase. This information is about two years old (the notes in parenthesis and the conclusion are mine).

Ann Arbor Ambassador

Wonderful features including reverse video. Has a nice keyboard with all the keys in the right places. However has a slow phosphore. (Has been popular around Tek.)

Visual 200

Excellent character font but a rotten keyboard. No extra memory for multiple pages. (The new Visual 50 looks very interesting however.)

(continued next page)

(Big Board II Custom Bios for Parallel Printer continued)

```
%
%       ENDIF
%
        LD      A,12
        LD      (NREVS),A       ;MAKE DRIVE READY WAIT 12 REVOLUTIONS
        LD      A,1
        LD      (TRKTAB+4),A    ;SET DRIVE ZERO TO DOUBLE DENSITY
        JP      CBIOS           ;EXECUTE BIOS COLDSTART ROUTINE
 ;
 ;
 ;
%       IF      SERPRN
 LSTINIT:                       ;SERIAL PRINTER INIT CONSTANTS
        DEFB    4
        DEFB    01000101B       ;16X CLOCK / 1 STOP BIT / ODD PARITY
        DEFB    3
        DEFB    01000001B       ;RX 7 BIT CHARACTERS
        DEFB    5
        DEFB    10101010B       ;TX 7 BIT CHARACTERS / DTR AND RTS HIGH

        DEFB    01000111B       ;PROGRAM CTC AND LOAD TIME CONST
        DEFB    4               ;DIVIDE BY 4 GIVES 9600 BAUD
%       ENDIF
 ;
 ;
 ;
 BTERR: LD      A,'?'
```

END

(continued)

Ampex Dialogue 80

Pretty Box but a rotten keyboard. It has a nice display font, good features, multiple pages of memory but no reverse of forward scroll to view it. (You have to page up and down.)

Teleray 100

VT100 compatible. Problem with VT100s is that their idea of insert/delete line is so bizzare that they are almost unusable. Ugly box, OK screen font. Too expensive.

ADM 32

Nice box, rotten keyboard, weird features (sorry no details on this) and an OK screen font.

Televideo 950

Excellent font, weird features (again, no details), no local scrolling to view multiple pages. Video tubes go out of whack and flyback squeal is bad. Sticky keyboard.

HP-2621

Many features but also many problems. Lots of characters in a cursor move sequence so most of the time programs use the up, down, left, and right sequences to move.

Datamedia 40

Nice but large. Televideo 950, ADM 31 compatible. Nice keyboard, pretty good features and has local scroll. I didn't see this one myself, so I don't know why we didn't pick it.

Adds Viewpoint

Real loser. Has a rotten keyboard, excruciatingly loud bell and no features. Ugly.

Televideo 920

Nice keyboard as long as you have a hammer in your hand. All the features of the 950.

ADM-3A

Everyone knows what these do, and don't do.

Concept 108

This terminal has more features than any other I have seen, including the Ambassador. It has the best keyboard for me, because all the keys are in the right place except for the caps lock. It has local scroll mode—96 to 192 lines of text in 80 column mode, fewer in 132 column mode. Bindable function keys and local editing. Can be programmed to send cntl-s and cntl-q when it gets behind, so fill characters aren't needed. The manual is half-an-inch thick. There is a 25th status line which can be programmed in 17,000 different ways. We didn't buy one of these because it is a bit too expensive.

Falco Data Products TS-1

This is the winner. It has a very nice keyboard, lots of features, and choice of green, white, or amber monitor. Also, local editing, scrolling, bindable function keys, no padding needed, setup mode like VT-100 but nicer. Terminal has an internal time clock which it displays on the status (25th) line, battery backup of the status/time information, and a detachable keyboard. The price is right too.

Conclusion

Now, I know you are all going to ask how to locate the TS-1. I don't have the slightest idea, I don't even know if Falco is still in business. However, I think the above listing should give you a start in your pursuit of the perfect terminal. If you find it, drop me a postcard. (Or, just tell your local dealer that you want all the information on the Falco TS-1, that should keep him off the streets for a week to two.)

Meanwhile: The first person who suggests that we hunt up one of the ever-elusive Snipe 13s is risking early expiration. You see, I located the last existing Snipe while attending Boy Scout camp many, many years ago—which of course explains why none have been seen since. (Good luck on your terminal hunt.)

■ ■ ■

Figure 1 - Four Versions of the ADM - 3A

Command	ADM-3A	BB I & II	Kaypro (ALL)	Xerox 820
Bell	07H 7	07H 7	07H 7	07H 7
Backspace	08H 8	08H 8	08H 8	08H 8
Cursor Down	0AH 10	0AH 10	0AH 10	0AH 10
Cursor Up	0BH 11	0BH 11	0BH 11	0BH 11
Cursor Right	0CH 12	0CH 12	0CH 12	0CH 12
Clr to End of Scrn	12H 18	11H 17	17H 23	12H 18
Clr to End of Line	11H 17	18H 24	18H 24	11H 17
Clr screen	1AH 26	1AH 26	1AH 26	1AH 26
Home cursor	0EH 14	1EH 30	1EH 30	0EH 14
Insert Line	13H 19	.	1BH,45H 27,69	14H 20
Delete Line	04H 4	.	1BH,52H 27,82	19H 25

Cursor Positioning (Same for all)

Use the 4 byte sequence -- ESC, Equals Symbol, Row + 20H, Column + 20H.
Which for the Home position (row 0, column 0) would be - 1BH,3DH,20H,20H.

Note, hexadecimal (base 16) numbers are followed by an H. All other numbers are decimal. Commas are just for separation of numbers.

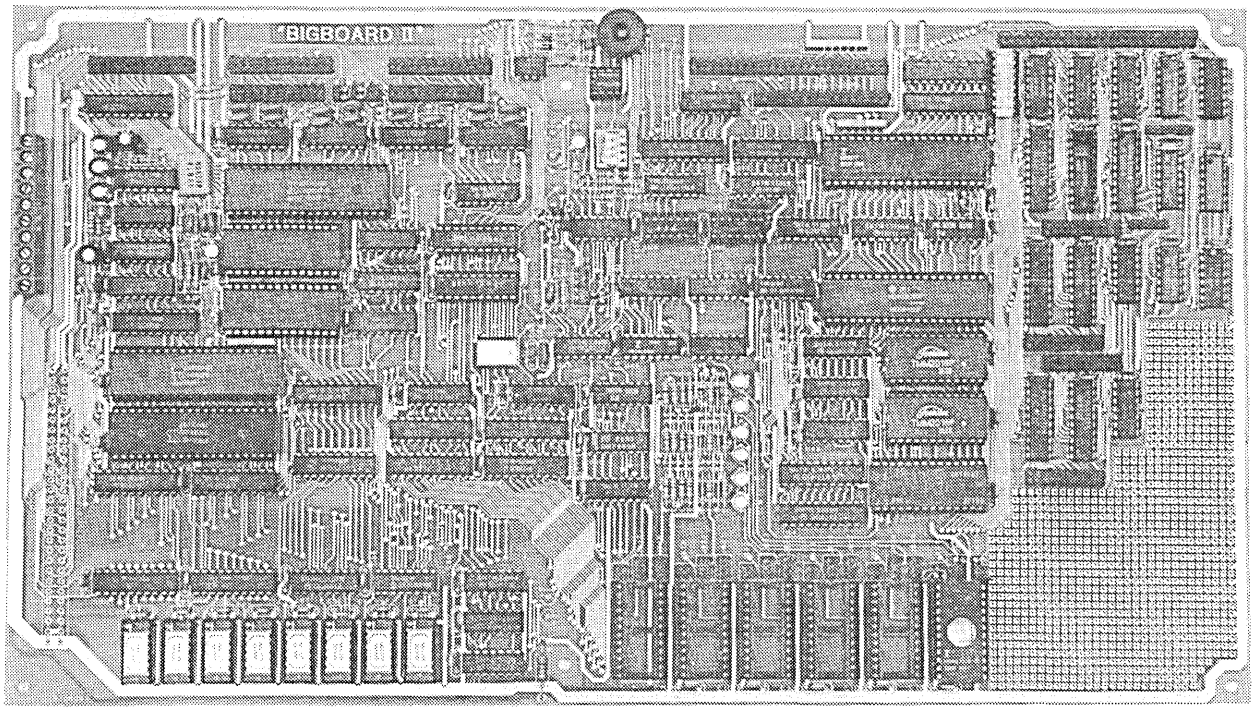
A "." in a column means that the function is not supported.

The Clear to end of screen (CLEOS) character is incorrectly listed as ^W (17H) in the BB I documentation. It is really ^Q (11H) as noted above. The Kaypro designers thought they were making the Kaypro monitor compatible with the BB I when they made ^W (17H) the CLEOS character.

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.

Video Jitter Fix

By Art Boehm

2000 29th Ave NW
NewBrighton, Mn 55112

If your video display is not quite perfect even though you have eliminated the slow swimming wiggle with the right (13.9776 MHz) crystal—have eliminated the tearing by hand-picking U11 or using an S part, and replacing C24 with a 10 pf capacitor—and have added filters to the video section (including tantalum capacitors), then take heart.

We have the real fix for the jitters. The problem is that U38 and U51 (the horizontal/vertical sync and horizontal sync delay one shots) are 74LS123's, and the output pulse width of these parts is both temperature and voltage sensitive (they are not fully compensated).

The solution is to replace both U38 and U51 with 74LS221's, which are fully compensated and rock stable. Incidentally, if you doubt that it matters, note that according to the 123's specs, a 1 percent change in +5V (like from 5.00 to 4.95) will reduce the horiz. sync pulse width by about 0.4 percent (or 30 ns) and that is almost half a dot of jitter.

The good news is that the 221 is pin compatable with the 123. The bad news is that it doesn't quite operate the same way. The two differences are:

First, the Cext pins cannot be tied to ground (pins 6 and 14).

Second, the constant in the delay formula is 0.7 not 0.45.

So to make this change, cut the following three traces on the component side. They come from under the sockets and go to ground.

Cut the run from U38 pin 6 to the ground grid (look under R4).

Cut the run from U38 pin 14 to the ground grid (also under R4).

Cut the run from U51 pin 14 to the ground grid (look under R5).

Change the timing components as follows:

New RC values for 74LS221s

Leave	C54	4700 pf	Make	R4	68K	T=223 us (V sync)
Make	C53	100 pf	Make	R3	75K	T=5.25 us (H sync)
Make	C69	180 pf	Make	R5	68K	T=8.57 us (H delay)

Note that we reused C53 and R3 so you only need three new parts besides the two 221s.

Finally, C69 is tied from U51 pin 15 (Rext/Cext) to ground. The grounded lead must be isolated from the ground grid (cut around the pad) and then the free lead must be connected to U51 pin 14.

Now, replace the 74LS123s in U38 and U51 with 74LS221s.

■■■

Cheap Daisy Blossoms

By Gary Kaufman

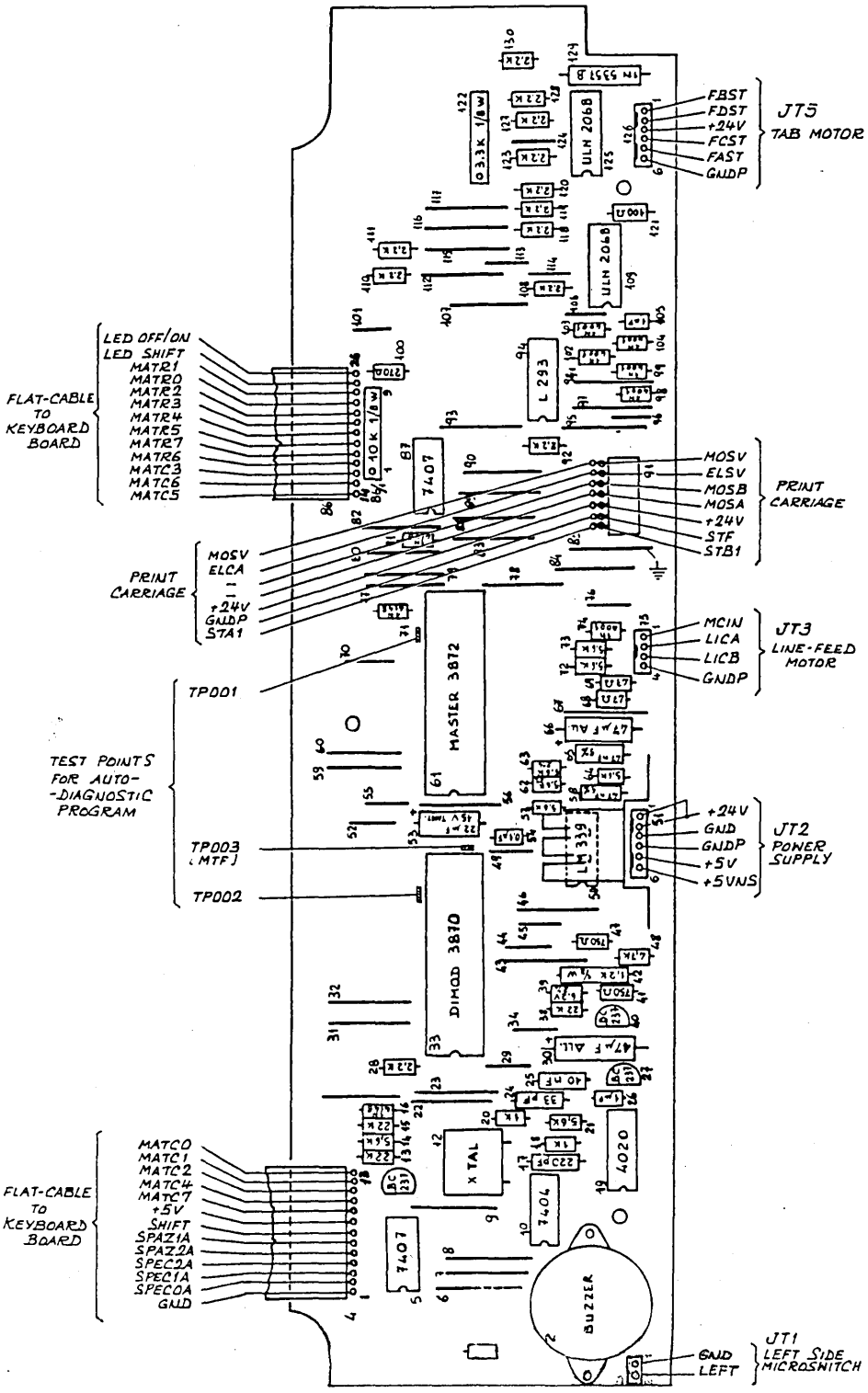
2001 Hamilton St. Box 87
Philadelphia, PA 19130
215-496-0687 (Evenings Before 9 pm)

The typewriter I received for college graduation has turned out to be a double purpose gift. You see, the Praxis 35 electronic typewriter is really an inexpensive daisy wheel printer which prints about eight characters per second.

So I modified it to work with my Big Board, and though it is slow and prone to misalignment (it's needed repair twice in the last six months) it has served my needs well. It prints the complete ASCII character set except for brackets, greater-than/less-than, braces, and the up-carat.

Though it took me only two hours to build and check out, I would recommend the following interface only to those of you who are confident in your electronic and mechanical repair skills. You void your warranty when you open the unit.

Figure 1 - Praxis Logic Board: Layout and Connections



About the Praxis

The Praxis' electronics are all mounted on two PC boards inside the case. In the left rear is the power supply board (supplies 5V and 24V). The other PC board is mounted below the keyboard. This board contains 2 Mostek single-chip processors. The first, called DIMOD in the Olivetti Service Manual, controls the character selection motor. The second processor (called MASTER) controls the keyboard, the line feed motor, the tab motor, and the buzzer.

MASTER scans the keyboard through two 8-bit ports. All the keys except the shift, keyboard selection, and pitch are part of this 8-by-8 matrix. When you press a key you short one of the 8 rows to one of the 8 columns. Master decodes the short and carries out the instruction.

So it's pretty easy to talk to the Praxis. I used two CMOS 4051 3-to-8 multiplexers so one 3-bit code could select the column and the other could select the row. Thus, six bits (0-5) can short the proper row to the proper column. I use bit 6 for the shift line (it is pulled low for upper case) and bit 7 to disable the multiplexers when there is no input from the computer.

Interface

To send a character, output a byte (to the PIO) with bit 7 low (to enable the

(continued next page)

(continued)

4051s), bit 6 high (unless the character is upper case), and bits 0 - 5 set to short the proper row and column on the Praxis keyboard.

Next, wait so that the Praxis has time to decode the "keypress" and then output an FFh (all bits high) to unpress the key. (Editor's note: this is probably easier than trying to find 4051s with key-return springs.)

Meanwhile a key pressed on the Praxis keyboard will also be received, so you

(continued next page)

Figure 2 - Praxis 30/35 Electronic Typewriter Interface

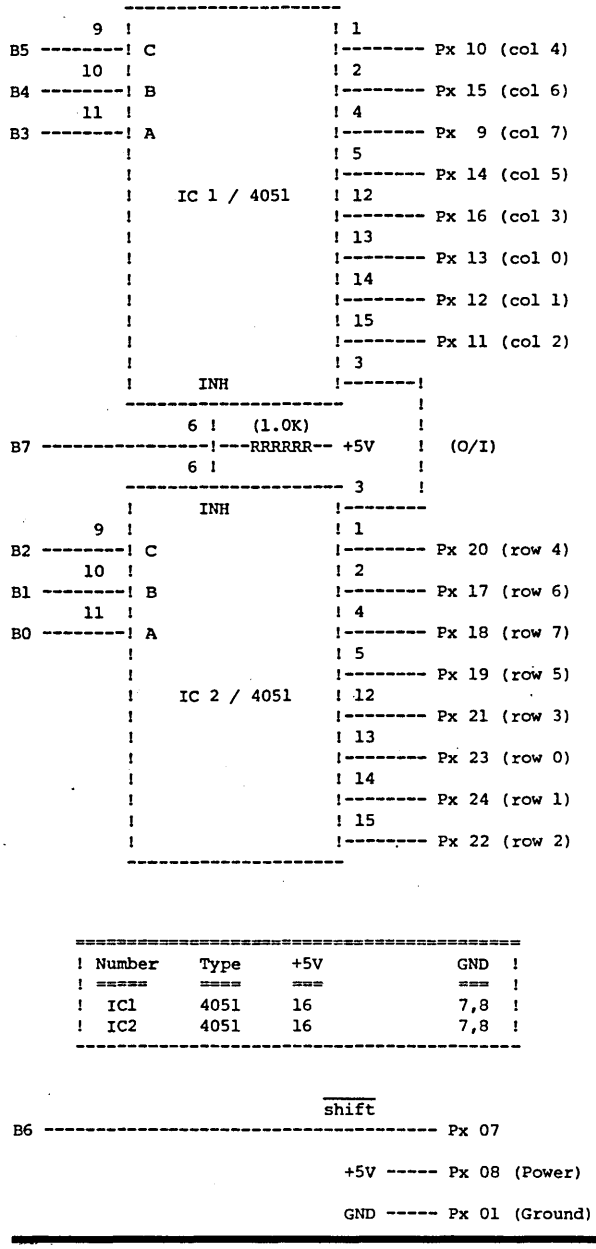


Figure 3 - Praxis Printer Routine in JRT Pascal

```
PROGRAM FILEOUT;                (* JRT Pascal Version *)

CONST
    PORTASTATUS = 9;            (* PIO/A Status port *)
    PORTADATA = 8;              (* PIO/A Data port *)
    CR = 13;
    BACKSPACE = 8;
    TAB = 9;
    CLEARSCREEN = 26;           (* Clearscreen Character *)

TYPE
    ASCII = 0..127;
    BYTE = 0..255;
    BLOCK = ARRAY (1..16) OF CHAR;
    NAME = ARRAY (1..14) OF CHAR;

VAR
    INFILE:FILE OF BLOCK;
    FILENAME : NAME;
    CHARACTER:CHAR;
    CHARCOUNT,P,IO:INTEGER;
    C : ARRAY(0..127) OF BYTE;
    CH : ASCII;

PROCEDURE INITPORT;
    BEGIN
    PORTOUT(PORTASTATUS,CHR(15));  ( SET UP PORT FOR OUTPUT )
    PORTOUT(PORTADATA,CHR(255));   ( DISABLE PRINTER )
    END;

PROCEDURE INITARRAY;

    PROCEDURE INIT1;
    BEGIN
    C(  0) := 255;    C(  1) := 255;    C(  2) := 255;    C(  3) := 255;
    C(  4) := 255;    C(  5) := 255;    C(  6) := 255;    C(  7) := 255;
    C(  8) :=  53;    C(  9) :=  56;    C( 10) := 126;    C( 11) := 255;
    C( 12) := 255;    C( 13) :=  54;    C( 14) := 255;    C( 15) := 255;
    C( 16) := 255;    C( 17) := 255;    C( 18) := 255;    C( 19) := 255;
    C( 20) := 255;    C( 21) := 255;    C( 22) := 255;    C( 23) := 255;
    C( 24) := 255;    C( 25) := 255;    C( 26) := 255;    C( 27) := 255;
    C( 28) := 255;    C( 29) := 255;    C( 30) := 255;    C( 31) := 255;
    C( 32) := 119;    C( 33) :=   8;    C( 34) :=   9;    C( 35) :=  24;
    C( 36) :=  32;    C( 37) :=  48;    C( 38) :=  51;    C( 39) :=  73;
    C( 40) :=  38;    C( 41) :=  30;    C( 42) :=  46;    C( 43) :=  14;
    C( 44) :=  90;    C( 45) :=  86;    C( 46) :=  18;    C( 47) :=  74;
    C( 48) :=  94;    C( 49) :=  72;    C( 50) :=  80;    C( 51) :=  88;
    C( 52) :=  96;    C( 53) := 112;    C( 54) := 104;    C( 55) := 115;
    C( 56) := 110;    C( 57) := 102;    C( 58) :=  17;    C( 59) :=  81;
    END;

    PROCEDURE INIT2;
    BEGIN
    C( 60) :=  38;    C( 61) :=  78;    C( 62) :=  30;    C( 63) :=  10;
    C( 64) :=  16;    C( 65) :=  12;    C( 66) :=  50;    C( 67) :=  31;
    C( 68) :=  28;    C( 69) :=  27;    C( 70) :=  36;    C( 71) :=  47;
    C( 72) :=  49;    C( 73) :=  37;    C( 74) :=  41;    C( 75) :=  33;
    C( 76) :=  25;    C( 77) :=  34;    C( 78) :=  42;    C( 79) :=  29;
    C( 80) :=  21;    C( 81) :=  11;    C( 82) :=  35;    C( 83) :=  20;
    C( 84) :=  43;    C( 85) :=  45;    C( 86) :=  39;    C( 87) :=  19;
    C( 88) :=  23;    C( 89) :=  44;    C( 90) :=  15;    C( 91) :=  38;
    C( 92) :=  74;    C( 93) :=  30;    C( 94) :=   8;    C( 95) :=  22;
    C( 96) :=  86;    C( 97) :=  76;    C( 98) := 114;    C( 99) :=  95;
    C(100) :=  92;    C(101) :=  91;    C(102) := 100;    C(103) := 111;
    C(104) := 113;    C(105) := 101;    C(106) := 105;    C(107) :=  97;
    C(108) :=  89;    C(109) :=  98;    C(110) := 106;    C(111) :=  93;
    C(112) :=  85;    C(113) :=  75;    C(114) :=  99;    C(115) :=  84;
    C(116) := 107;    C(117) := 109;    C(118) := 103;    C(119) :=  83;
    C(120) :=  87;    C(121) := 108;    C(122) :=  79;    C(123) :=  38;
    C(124) :=  86;    C(125) :=  30;    C(126) :=  40;    C(127) :=  53;
   END;

    BEGIN
       INIT1;
       INIT2;
    END;

PROCEDURE DELAY(DURATION : INTEGER);
VAR D:INTEGER;
BEGIN
FOR D := 1 TO DURATION DO BEGIN END;
END;
```

(continued next column)

(Praxis Printer Routine in Pascal continued)

```
PROCEDURE SEND(CH:ASCII);
   VAR   A : INTEGER;
      BEGIN                                (* Delays are for 4 mhz *)
      PORTOUT(PORTADATA,CHR(CH));
      DELAY(700);
      PORTOUT(PORTADATA,CHR(255));
      DELAY(400);
      END;

PROCEDURE SETTABS(P:INTEGER);
VAR I,K:INTEGER;
BEGIN
   SEND(62);     ( EXPR )
   SEND(67);     ( MARGIN BYPASS )
   SEND(62);     ( EXPR )
    FOR K := 1 TO (P * 2) DO  BEGIN  SEND(55);    END; ( SPACE )
    SEND(59);     ( MARGIN LEFT )
    FOR I := 1 TO 13 DO
      BEGIN
        SEND(56);   ( TAB )
        SEND(63);   ( TAB CLEAR )
      END;
    SEND(62);     ( EXPR )
   FOR I := 1 TO (P - 2) DO
       BEGIN
      FOR K := 1 TO 8 DO  BEGIN  SEND(55);    END; ( SPACE )
      SEND(61);     ( TAB SET )
      END;
    SEND(56);     ( TAB )
    SEND(57);     ( MARGIN BYPASS )
   SEND(56);     ( TAB )
   FOR K := 1 TO (P-1) DO  BEGIN  SEND(53);   END; ( BACKSPACE )
   SEND(60);     ( MARGIN RIGHT )
   SEND(62);     ( EXPR )
END;

PROCEDURE PRINT(CH : CHAR);
   VAR   A : INTEGER;
      BEGIN
      A := ORD(CH);
      PORTOUT(PORTADATA,CHR(C(A)));
      DELAY(600);                          (* Delays are for 4 mhz *)
      PORTOUT(PORTADATA,CHR(255));
      DELAY(450);
      END;

BEGIN
INITPORT;
INITARRAY;
   WRITELN(CHR(CLEARSCREEN));
   WRITELN(' THIS PROGRAM WILL PRINT A LISTING OF A TEXT FILE WHICH IS ON');
   WRITELN(' EITHER DISK.  THE LISTING WILL APPEAR BOTH ON THE VIDEO SCREEN');
   WRITELN(' AND ON THE PRAXIS 35.  MAKE SURE THAT THE PRAXIS 35 TYPEWRITER' );
   WRITELN(' IS PLUGGED IN AND IS TURNED ON');
   WRITELN;
   WRITE('SET TABS ? ');
   READLN(CHARACTER);
   IF (CHARACTER = 'Y') OR (CHARACTER = 'y') THEN
      BEGIN
        REPEAT
        WRITE('PITCH (10,12,15) ? ');
        READLN(P);
        UNTIL ((P = 10) OR (P = 12) OR (P = 15));
      SETTABS(P);
      END;
     WRITELN;
     WRITELN(' TYPE THE FILE NAME THEN RETURN');
     WRITE(' OF THE FILE NAME  ---)');
     READLN(FILENAME);
   RESET(INFILE,FILENAME,BINARY,256);
   CHARCOUNT := 0;
   WHILE NOT EOF(INFILE) DO
        BEGIN
          READ(INFILE;CHARACTER);
          CHARCOUNT := CHARCOUNT + 1;
          WRITE(CHARACTER);
          PRINT(CHARACTER);
           IF (ORD(CHARACTER) = TAB) THEN CHARCOUNT := CHARCOUNT + 7;
          IF (ORD(CHARACTER) = BACKSPACE) THEN CHARCOUNT := CHARCOUNT - 1;
          IF (ORD(CHARACTER) = CR) THEN
               BEGIN
                 DELAY((CHARCOUNT * 50) + 2500);
                 CHARCOUNT := 0;
               END;
         END;
END.
```

END

CHEAP DAISY BLOSSOMS

(continued)

can easily enter names into form letters right from the Praxis.

If I had had another port available I would have used it to control the pitch selection and the keyboard selector switch.

Power

I stole the +5V for the 4051s from the Praxis keyboard connector. One reason was laziness and the other was that the chips needed to be powered whenever the Praxis was running to prevent them from generating random garbage.

Even though the CMOS 4051s don't draw much current, I have mounted a small Sprite fan behind the power supply and suggest that you do the same thing even if you don't build this interface.

Disassembling the Praxis

Make certain that your typewriter is running perfectly before opening the case. To open the case (voids your warranty), do the following:

1. Remove the 4 outermost screws on the bottom of the case.
2. Unscrew the platten knobs by turning them counter-clockwise.
3. Gently lift off the top cover, deforming it slightly to fit around the platen.

Next, remove the keyboard and expose the controller board.

1. Remove the 2 screws holding down the keyboard and unplug the cables running to it.
2. Remove the 3 screws on the bottom of the keyboard which hold the processor board in place.
3. Gently bend the processor board aside leaving the 2 ribbon cables intact.

The connection points on the processor board are now exposed.

Connecting the Ribbon

Carefully tack-solder the ribbon cable from the 4051's to the processor board at the same 2 places that the keyboard ribbon cables are attached. See Figures 1 and 2 for details.

I used ribbon cable because it was easy to bring out of the typewriter case. Now you can reassemble the case backwards (which can be interesting).

(continued next page)

(continued)

Assembling the Board

I wired the circuit board point to point on a small piece of perf-board. It took about 2 hrs to wire, mount in a small box, and check out. I ran ribbon cable to both the BB and to the Praxis. I added several .01 capacitors between +5V and ground to reduce noise.

Software

To talk to the printer, you need a lookup table and some delays so that you don't overflow the unit's 12-character buffer. I have written programs in several languages including MBASIC, UCSD Pascal, and JRT Pascal. The MBASIC and JRT Pascal versions follow this article (see Figures 3 and 4).

The delay routines are set up for a 4 MHz clock. You can adjust these for your clock. If you have UCSD Pascal, you can obtain a much more advanced version of the program from me if you send an 8" disk and a self-addressed stamped return envelope.

Final notes

The Praxis 30 is the same as the 35, both electrically and mechanically. The only differences are the color and the addition of the pitch selector switch to the 35. If you add the switch yourself then you have a 35 for the price of a 30.

■ ■ ■

Figure 4 - Praxis Printer Routine in MBASIC

```
1 REM ============================================================
2 REM Gary Kaufman  2001 Hamilton St. Box 87  Philadelphia, PA  19130
3 REM               (215) 496-0687 (evenings before 9 pm)
4 REM ============================================================
5 P1 = 9 : REM PIO/A Status Port
7 P2 = 8 : REM PIO/A Data Port
10 GOSUB 2000 : REM Initialize character array and PIO
15 INPUT "Set tabs ";T$
16 IF LEFT$(T$,1) = "Y" THEN GOSUB 6000
20 LINE INPUT "File name ";F$
25 INPUT "Title Listing ";T$
26 IF LEFT$(T$,1) () "Y" THEN 30
27 A$ = "Listing  File ----) " + F$ + CHR$(10) : GOSUB 3000
28 GOSUB 5000 : REM Underline
29 A$ = CHR$(13) : GOSUB 3000
30 OPEN "I",1,F$
31 LC = 0
40 IF EOF(1) THEN 1000
41 LC = LC + 1 : IF LC ( 66 THEN 50
42 LC = 0
43 PRINT CHR$(7);"Change paper, type C to continue"
44 INPUT A$
50 LINE INPUT #1,A$
55 A$ = A$ + CHR$(13)
60 GOSUB 3000
80 GOTO 40
1000 PRINT : PRINT"File done ..."
1999 END
2000 REM OUTPUT CHAR ROUTINE INITIALIZATION
2005 PRINT "Initializing PIO ";
2010 DIM A(128) : WIDTH 79
2020 FOR I = 0 TO 128 : READ X : A(I) = X : PRINT "."; : NEXT I
2030 DATA 255,255,255,255,255,255,255,255
2040 DATA 53,56,126,255,255,54,255,255
2050 DATA 255,255,255,255,255,255,255,255
2060 DATA 255,255,255,255,255,255,255,255
2070 DATA 119,8,9,24,32,48,51,73
2080 DATA 38,30,46,14,90,86,18,74
2090 DATA 94,72,80,88,96,112,104,115
2100 DATA 110,102,17,81,38,78,30,10
2110 DATA 16,12,50,31,28,27,36,47
2120 DATA 49,37,41,33,25,34,42,29
2130 DATA 21,11,35,20,43,45,39,19
2140 DATA 23,44,15,38,74,30,8,22
2150 DATA 86,76,114,95,92,91,100,111
2160 DATA 113,101,105,97,89,98,106,93
2170 DATA 85,75,99,84,107,109,103,83
2180 DATA 87,108,79,38,86,30,40,53
2190 DATA 255
2195 REM .......PWHEW !
2200 OUT P1,15:OUT P2,255
2210 PRINT : RETURN
3000 REM ***          OUTPUT A$              ***
3010 FOR I = 1 TO LEN(A$)
3015 OUT P1,15:OUT P2,255
3020 A = ASC(MID$(A$,I,1))
3025 IF A)127 THEN A = A - 128
3030 OUT P2,A(A)
3040 GOSUB 4000
3060 NEXT I
3070 PRINT A$ : RETURN
4000 REM ***          DELAY ROUTINE          ***
4001 REM delays are 80.50,200 for MBASIC / 160,100,750 for BASCOM
4002 REM all are for 4 mhz.
4010 FOR J = 1 TO 80: NEXT J : OUT P2,255
4020 FOR J = 1 TO 50 : NEXT J
4025 IF A = 13 THEN FOR J = 1 TO 750 : NEXT J : REM (CR) delay
4026 IF A =  9 THEN FOR J = 1 TO 200 : NEXT J : REM (TAB)delay
4030 RETURN
5000 REM ***          UNDERLINE              ***
5010 FOR I = 1 TO LEN(A$) - 1
5020 OUT P2,A(95) : GOSUB 4000
5030 NEXT I : OUT P2,A(13) : GOSUB 4000 : RETURN
6000 REM ***          SET TABS               ***
6010 OUT P2,62 : GOSUB 4000
6020 FOR I = 1 TO 7 : OUT P2,56 : GOSUB 4000 : OUT P2,63 : GOSUB 4000 : NEXT I
6030 OUT P2,62 : GOSUB 4000
6040 OUT P2,67 : GOSUB 4000
6050 FOR I = 1 TO 7 : OUT P2,53 : GOSUB 4000 : NEXT I
6060 OUT P2,59 : GOSUB 4000
6070 FOR I = 1 TO 7
6080 FOR K = 1 TO 8 : OUT P2,55 : GOSUB 4000 : NEXT K
6090 OUT P2,61 : GOSUB 4000 : NEXT I
6100 OUT P2,56 : GOSUB 4000
6110 OUT P2,57 : GOSUB 4000
6120 FOR I = 1 TO 9 : OUT P2,55: GOSUB 4000 : NEXT I
6130 OUT P2,60 : GOSUB 4000
6140 OUT P2,54 : GOSUB 4000
6150 RETURN
```

Slicer Column

By David Thompson

Otto Baade called in with some interesting news from Slicer.

80186s?

They've had trouble getting 8 MHz 80186s. Intel has had yield problems. They're getting lots of 186s that run 6 MHz but not many able to go 8. Otto says that the problem has to do with some kind of delay that was designed into the chip rather than a state-of-the-art speed problem. He's confident that Intel will have it straightened out shortly (it turns out that their most recent tweak made the situation worse, not better).

The Slicer company has ordered 50 of the 6 MHz parts to supplement their reduced allotment of the 8 MHz 186s.

Memory bug

Otto said that the ground plane in the RAM area needs a bit of help. The information is included in their newsletter and the mod is very simple to do. Just add a couple of wires. Everyone who owns a Slicer should be getting the newsletter, if you aren't, let them know.

Operating Systems

The Slicer folks are now selling CPM/86 for $85. They are purchasing IBM PC compatible CPM/86 and then modifying it to run on the Slicer. Such a deal—no expensive contracts for Slicer, and we get CPM/86 for less than the $250 that I paid for an early copy. (Otto hasn't decided about MPM or concurrent CPM/86 yet.)

MSDOS is the next challenge. Microsoft doesn't want to bother with small companies like Slicer, so Otto and company are looking at purchasing PCDOS ($40) and going from there. However, they need an install program which will make PCDOS run on the Slicer. If you can help, get in touch with Otto.

Once they've got the installation figured out, we hope to put it on an 86 user disk—complete with source.

86 User Disks (Actually 3)

Speaking of User disks, we now have three disks of 86 compatible public domain software. Some of the software showed up as magical gifts during the SOG. (There are some definite advantages to inviting a whole scad of folks over for the weekend.) This initial batch has been excellent material because the best of the 8080 stuff is the first software being brought over to the 86 world. Note that these disks are 8" single side single density and they are $15 each.

Disk 86-1—Disk Utilities

D.CMD/A86, SD.CMD/A86, XDIR.CMD/A86 Three extended directory programs. Each does it differently, so we included all three.

FILE-EXT.CMD/A86 Disk status program with good display format.

PAGE.CMD/A86 A text paging program. Displays 24 lines at a time.

PRINT.CMD/A86 File printing routine. Puts a header at the top of each page along with page number and file name.

MUCHTEXT.CMD/A86 Counts words and lines in a text file.

ERQ.CMD/A86 Selective file erase program. Displays all selected files and then asks you one at a time for a Y/N.

INUSE.CMD/A86 Prints "In Use" on your terminal and asks for a password. It will not release the console until you enter the password.

FINDBAD.CMD/A86 Finds and collects bad sectors on a disk. If there are no bad sectors, information on the disk is unaltered.

Disk 86-2—DU and Modem Programs

DU-V75.CMD/A86/DOC This is the popular disk utility from CP/M 80. It lets you read, write, and modify disk sectors.

MODEM4.CMD/A86 This is a modem program set up for the Slicer. This program includes a built-in help file.

MODEM7SL.CMD/A86/DOC No modem disk would be complete without this standard. This is modem7 set up for the Slicer. It displays a menu when it is called.

Disk 86-3—Small C

C86.CMD This is the original Small C compiler which appeared in Dr Dobbs Journal in 1980. It runs under CPM-86 and generates 8086 source for the ASM86 assembler.

C86.COM This is the C86 compiler which runs under CPM-80. This 8080 program produces 8086 assembly language. In fact, the compiler itself can be compiled into 8086 executable code on an 8080 system by using this C86.COM followed by ASM86.COM and GENCMD.COM (supplied with CPM-86).

C86LIB.A86 This is the C86 I/O library.

SMALLC86.DOC Documentation on Small C.

C?????.C Source of the C86 compiler.

Plus, there are a number of demonstration files and ENTAB (insert tabs in place of spaces) and DETAB (replace tabs with spaces) programs all written in Small C.

New Monitor and BIOS

Otto is just finishing up a new monitor and BIOS for the slicer. The monitor will have routines for 5", 8" and winchester drives (simultaneously). The BIOS will be smart enough to take advantage of the new monitor. He plans to have them available by the end of October.

Otto said, "The new software will have an automode feature which will assume 9600 baud terminal 8 bits, transmit 2 stop bits, and receive one stop bit. This should allow it to talk to just about any kind of terminal."

"It autoboots CPM 86. Otherwise it says 'No CPM' and goes to the monitor."

He went on to add that the system will be able to boot from any type of disk including a winchester and you can have it automatically boot from any drive (A, B, C . . .).

The upgrade package will be only $25 if you agree to return the old EPROMS and disks after you receive the upgrade.

Trivia

Otto felt the following were worth mentioning. JDR Microdevices is selling double sided, 5" 80-track drives for $150 each (approximately 800 K per drive).

He says that the Amdek 3" drive runs on the Slicer right out of the box. It looks just like a 5" drive to the system (but it doesn't look like a 5" drive in your cabinet). This could be the start of the notebook size slicer.

(continued next page)

(Slicer Column continued)

Otto has been checking out some of the new terminal boards so that folks can put together a complete system without a separate terminal. He was not impressed with the terminal board by John Bell engineering because: (1) they brought out signals through an edge connector, (2) they have signal timing problems, (3) they haven't cleaned up the flickering display, and (4) it runs only at 600 baud with the slicer without losing characters.

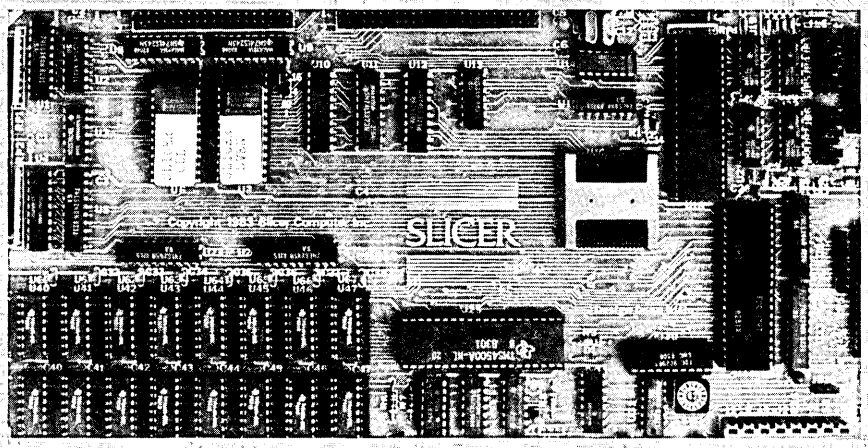
Otto likes the cheap ($150-$200) surplus Siemens drives. He feels that the bad publicity is unjustified.

"I have six running here, and friends of mine have many more, and they run very well both single and double density. I have been running them heavily since last September and they have been running fine. Many of them come jumpered for hard sector, make sure that the SS (soft sector) pins are jumpered and the HS (hard sector) pins aren't."

And finally, he mentioned, that the latest Zenith monitor (which ever that one is) will run fine with the BB II.

■ ■ ■

GET REAL 16 BIT POWER



THE SLICER

A SINGLE BOARD COMPUTER FEATURING THE INTEL 80186

(see Byte magazine April '83)

- Full 8MHz 16-bit microprocessor having complete software compatibility with the 8086 and 8088
- 256K Bytes of RAM plus 32K Bytes of EPROM memory capacity on board
- 90 line expansion interface for more memory and peripherals consist of 16 bit data bus, 20 bit latched address bus and all important 80186 control signals
- Floppy disk controller can run the combination of 8", 5-1/4", 3-1/2" drives simultaneously
- SASI port for hard disk controller
- Two full function RS232C serial ports with individually controlled baud rates from 50 to 38.4K baud
- Baud rate for console port acquired automatically
- 8K of EPROM contains drivers for peripherals, commands for hardware checkout and software testing
- Source for monitor and bios included on disk
- Bios for CP/M 86* supports 8", 5-1/4", 3-1/2" drives and the Xebec 1410 controller for hard disks
- Board size 6" x 12" power requirements +5 @ 3A, +12V @ 60mA, −12V @ 50mA
- Complete documentation included.

Sold in various forms:

Assembled and tested	$1,075.00
Full Kit	895.00
NEW Easy Kit (hard to get parts)	450.00
Bare Kit (board, Rom, doc, and disk)	150.00
CP/M 86* available	85.00

Regular U.P.S. shipping within continental U.S. is included.

SLICER™

Available Now!

Choose 8 MHz CPU for speed
6 MHz CPU for speedy delivery

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

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

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

BB I Video Size

By Warren H. Buske

(USAFS Berlin)
APO, NY 09742

Most people have solved the video wiggle problem by making the Big Board's video output frequency exactly equal to the power line frequency. If you are using 50 HZ power, however, more modification is required than for 60 Hz (see Micro Cornucopia issue #7, page 19, 'Viewing 50 Hertz' by T. Hameenaho). Also, 50 HZ verticle means more flicker.

The other way to solve the wiggle problem is to keep the power line frequency from interfering with the monitor. This has been straight forward on the two monitors that I have used (Zenith D12-PF-1 and a USI Pi-3 amber monitor).

I removed the power transformer, spliced in longer leads, and located it a foot or so away from the CRT. These were well-made transformers, complete with steel cases. However, there was still enough magnetic flux leakage to wiggle the electron beam in the CRT.

If moving the transformer does not completely cure the problem, then add filtering to the monitor's power supply output (see Micro Cornucopia issue #1, page 6).

Purists who object to the thought of an outboard power transformer could try wrapping the transformer in layers of Mu-metal magnetic shielding material.

Wider Display

On both my monitors, there is not enough adjustment range to expand the display horizontally to the outer edges of the monitor. Fortunately, it is possible to make some simple modifications to the board to increase the width. It turns out, that the character dot output frequency is reduced by about 22% while maintaining the standard 60 Hz TV format. This is especially important if you are trying to use a converted Television set or a low-bandwidth monitor, since the characters will appear sharper.

How It Works

The scan clock generator (U23) determines the width of the display. The input to U23 is the character clock, which generates one pulse for each character position in the horizontal line. U23 divides this character clock by 128 to generate the scan clock, which is one pulse for each horizontal scan line. With this arrangement, there would be sufficient time to write 128 characters on each line, if you didn't count the time it takes for the monitor to move the electron beam from the right side of the screen to the left side (retrace time) in preparation for starting a new scan line.

During this horizontal retrace, the beam cannot write any information on the screen. With most monitors there is sufficient time left to display about 108 characters on the line, which is more time than is needed for an 80 character display. Therefore, some of the space on the screen is wasted and the characters appear narrow. To make matters worse, the monitor must process the characters as part of a 108 character line, which requires a higher bandwidth monitor than one which handles a true 80 character line.

"Why did they design it that way, then?" you ask. Actually, it appears that the board was originally designed with U23 set to divide by 96 instead of 128. My schematic indicates that U11 pin 3 was originally connected to U10 pin 12 and not to the +5V as it is now. In addition, the board I received was cut and jumpered in the same way the schematic was changed.

If you reconnect it as it was, then U23 will divide by 96, but you also have to slow down the video clock by the same amount. I tried this, and it worked fine on my Zenith monitor, but did not allow sufficient time for the horizontal retrace on the USI monitor. That is probably why U23 was changed to divide by 128.

Mods

The following changes will cause U23 to divide by 100 (works with both monitors) and slows down the video clock by 22 percent. The resulting video output frequency will be (7/9)(128/100), or .9956 times what it was originally. The dot frequency is 78 percent of what it was, so the characters will be sharper on a low-bandwith monitor.

The only disadvantage to this modification is that the video output frequency works out to be 59.733 Hz (with a 13.9776 MHz crystal for Y1). This means the you will have a 0.277 Hz wiggle unless you get a 14.040 MHz crystal (from who knows where) or move your transformer outdoors.

For those of you who did not have the wiggle problem (or learned to live with it) and still use the original 14.31818 MHz crystal, your output frequency will drop from 61.46 Hz to 61.19 Hz.

Step by Step Instructions:

1. Isolate U11 pin 3, then connect it to U10 pin 12.
2. Isolate U10 pin 13, then connect it to U23 pin 5.
3. Isolate U24 pins 4,5,&6. Connect pins 4 & 5 to the +5V supply. Connect pin 6 to ground. (This preloads a 0111 binary into U24 causing it to divide by 9, since 16 - 7 = 9.)
4. Reduce the horizontal delay to re-center the image on the monitor. If you cannot use the controls on the monitor to do it, change the horizontal delay by reducing the value of C69 or increasing the value of R5. (I find that simply removing C69 works well.)

Another combination which would produce a smaller increase in width but would maintain the original output frequency, would be to set U23 to divide by 112 and U24 to divide by 8. The output frequency is then (7/8)(128/112), or 1 times the original output frequency. This will increase the width by 14%. (Since I do not live around 60 cycle power, I cannot verify this; however, it should work.) To make this modification, follow the preceding procedure except replace steps 2 & 3 as follows:

2. Isolate U10 pin 13, then connect it to U23 pin 11.
3. Isolate U24 pin 3, then connect it to ground.

Note: If you are like me and cringe at the thought of carving up a circuit board just to test an idea, try this instead: Pull the IC out of its socket and put it in a spare socket.

Now you can isolate IC pins by bending up the equivalent pins on the new socket. Then reinsert the IC, socket and all, back into the old socket and use clip leads to make your new connections.

Video Output Thoughts

The output impedance of the composite video driver on the Big Board is approximately 14 ohms (depends on the beta of Q1). It is better to use a driver with a 75 ohm output impedance, coupled with a 75 ohm cable because a good impedance match minimizes reflections at the ends of the video cable which could degrade the image.

If you use a short cable, don't worry about impedance matching, you won't notice a difference. The Big Board video output impedance is low because the impedance at the emitter of Q1 is approximately equal to the impedance seen by the base, divided by the beta of the transistor. The base sees R21, R25, and R26 in parallel, (in addition to the output impedance of U94 at pins 3 & 6 when those outputs are low). If the beta of Q1 is 50, then the impedance at the emitter of Q1 is 863 ohms divided by 50, or 17.3 ohms. This impedance is in parallel with R20, which is 75 ohms. The resulting output impedance is approximately 14 ohms.

One way to design a video driver with a 75 ohm output impedance is to put the 75 ohm output resistor in the collector leg and drive the load from the collector—that gives an inverted output. Another way is to put a 58 ohm resistor in series with the output which, when combined with the 17 ohms there already, will give you 75 ohms. Since that reduces your output voltage into the cable (the cable has a 75 ohm impedance too), you would want to increase the gain of Q1.

■ ■ ■

Kaypro Column

By David Thompson

Latest developments on the Kaypro front.

As of the first of October, Kaypro has been soldering in all the small ICs. That's nice for reliability but it means that you definitely need to have some experience working with PC boards before you do the speed up mod on a new board. Unsoldering IC pins is not for novices.

At the same time, Kaypro began shipping the Kaypro 4 ROM (a 2732A) on both the Kaypro II and Kaypro 4 so you shouldn't need a new monitor ROM to go 5 MHz. Also, all it should take to turn your Kaypro II into a Kaypro 4 are two double sided drives and a copy of someone's 4 system disk. Whooppee! (We're looking into this very closely.)

IC Pin Numbers

Many folks aren't sure which IC pins are which, so here goes. If you are looking at top of the IC (legs pointing away from you) and the notch end is at 12 O'Clock, then pin 1 is toward 11 O'Clock. The pins go in a circle counterclockwise from pin 1. So, if it is a 16-pin chip, pin 16 will lie toward 1 O'Clock.

If you are looking at the underside of the IC (or at the underside of the PC board), then the pins count up clockwise starting from the notch. Pin 16 is on the opposite side of the notch from pin 1.

Some ICs also have a tiny circle next to pin 1, but many don't. All ICs have the notch.

Speed

We've received a good deal of feedback about the 4 and 5 MHz mods on the Kaypro II and 4. In fact, some of the feedback is from manufacturers of other Kaypro add-ons. They have found that many of their customers have already done our speed-up mods. Most systems come up flying and never quit. A few have troubles. Only one person has called to say he'd screwed up his board.

Many people commented on my lack of specific directions on the speed up mod in issue #12. In my own fumbling way, I was trying to make sure that people had some knowledge of hardware before digging into their own systems.

The best option if you aren't already familiar with ICs (but want speed right away) is to locate (or start) a Kaypro user group (your dealer may know if there is one in your area) and then propose that the group hire an experienced technician (or lean on a heavyweight member) to make the mods all at once.

The difference between a fast Kaypro and a slow Kaypro is incredible. Those who make the change won't go back without a fight. It's like ZCPR (disk K9), you find something that makes things work better and you don't want to go back.

The Monitor ROM

The Kaypro 4 (and the newest Kaypro II) contain a 2732A ROM, which should run any speed you wish. However, if you have one of the older IIs you'll have to replace the monitor ROM.

The 2716 monitor ROM (U47) supplied on the older IIs will not run at 4 or 5 MHz. Even if you get one that will run cold, it will no doubt die as the system warms up. As chips get warmer, they get slower.

A 2716-1 (350 ns) will run at 4 or 5 MHz. You can copy the contents of your present ROM into a 2716-1 using a standard ROM programmer like a DATA-IO, or you can order our fancy new PRO-MONITOR ROM for the Kaypro. Besides being able to run fast, PRO-MONITOR does some other nice things: it ignores null characters, gives you six retries on a disk read error, gives you a non-blinking block cursor (like the big fancy terminals) and it gives you faster disk access.

You see, Kaypro had some drive problems (controller timing) which they initially tried to correct by adding additional writes and verifies. Rather than just getting a 512 byte block of data, using it, and then writing the modified data back after finishing with all 512 bytes, they write each 128 byte "pretend sector" back to the disk. Since disk reads and writes are slow, the more you do, the slower you get. They did a hardware correction (the timing fix in issue #11) for the drives, but they still have the slow code in the ROM.

The Processor

Another critical part is the Z80 processor. You need to replace the Z80 with a Z80B. The difference between the Z80, Z80A, and Z80B is speed. All three can come off the same silicon wafer, and they aren't separated until the final testing. If the chip will run at least 2.5 MHz, it is a Z80, if it will run at least 4 MHz, it is a Z80A, and if it will run at least 6 MHz, then it is a Z80B.

If you buy parts from a standard dealer, then you can be pretty sure that the Z80B you get will run at least 6 MHz. If the parts are from an outfit that handles surplus then it may not really be a Z80B. But, usually the part will be fine. It is safest to use a Z80B when doing speed ups.

If you have done the modification properly, you should have no problem booting right up. If you are one of about 5% of the folks who can't even get the thing to run at 4 MHz, then you need to read on.

The Clock

The final critical part of this mess is the system clock. You have modified this circuit so that the Z80B is receiving 4 or 5 MHz instead of 2.5 MHz. It sounds simple, and usually is.

ICs expect to receive signals that look like rectangles with nice square corners. The signal is either high (almost 5V) or low (ground) and doesn't loiter when switching from low to high (rising edge) or when switching from high to low (falling edge).

Of course it always takes a little time to make the change, and the Z80B expects to have the signal go from low to high (rise time) in 20 ns or less.

On a few systems, I've seen rise times as long as 50 ns. If you connect an oscilloscope (with at least a 50 MHz bandwidth) to pin 6 on the Z80 (use a X10 probe) you can see what your clock is doing.

Pull-Up Circuit

If the rise time is too long (the usual problem) then check to see if the 2N3906 (Q1) pull-up transistor is in place. If you have a newer Kaypro, it probably isn't.

So, purchase a plastic 2N3906 (50 cents) and a 50 pf (50 cents) ceramic capacitor (C6) and just solder them into the board. The transistor and capacitor are located at the pin-40 corner of the Z80B. The transistor's center pin (base) goes to the pad closest to R26's rear pad. The flat side of the transistor points toward the

drives. The collector and emitter leads drop right into their appropriate holes. C6 mounts parallel to R26, on the CRT side.

Finally,

If the clock is OK and you have the fast ROM and Z80B, but it still won't run dependably (even at 4MHz) then I'd seriously consider trying another Z80B and/or ROM. So far, we haven't had any problem with the Z80 PIOs. Remember, the Kaypro is not the best place to test old surplus parts.

More Drive Info

After reading issue 13 of Micro C, Bill McDonald, chief engineer at Kaypro, called with the latest info on drives (and other things).

He mentioned that Kaypro purchases its drives from Tandon fully aligned, and though they test them before installation, Kaypro doesn't do any alignment. It's Tandon's job to do the complete alignment process.

It also turns out that Kaypro is looking for other sources of hard and soft drives. They will be purchasing drives from three companies (including Tandon) so that they can compare the product from three makers over a large sample before settling on one or two manufacturers. (That's the way they selected their monitor and power supply.)

Meanwhile, I have received a release from Microscience stating that they have an $8,000,000 order from Kaypro for half-wide 5" winchesters. The first of the 3 lb drives were to be shipped September 26 with the rest of the order to be delivered over 10 months. Bill McDonald said he was quite impressed with Microscience's little drives. I guess he really was.

There is a little interface board mounted on the winchester housing (under the PC board). That board can be changed so that Kaypro can use the Xebec hard disk controller card as well as the Western Digital card. Nothing like keeping your options open.

Kaypro 10

Bill mentioned that the extra "telephone" connector on the back of the 10 is for a light pen. The video controller has a light pen input so they simply brought it out to the back panel. There doesn't appear to be any software that takes advantage of that facility. Yet. (What a neat idea for a super text editor, you'd just erase words, or move things around with the touch of a pen.)

He also affirmed my guess that the head on the winchester goes back to the safety area if the disk isn't accessed for a couple of seconds. In fact, any time the red "in use" light is not on, the head is in the safety area. So, there is no need to run the safety program—just wait for the access light to go out and you can turn off the machine.

Power Supplies

The Kaypro 10s have a 75 watt switching power supply. The IIs and 4s have had a smaller one, but since Kaypro is purchasing supplies in such quantity, the 75 watt units are now as cheap as the littler ones. Now there should be no more screen twitching when the drives fire up.

Morrow Goes After the 10 Market

An old-timer in the micro computer business, George Morrow is going after the Kaypro 10 market with his Morrow MD11. It is a 4 MHz Z80 based system with CP/M 3, 128K of RAM, and an 11 Mbyte winchester. Their products have been pretty good but I'm still waiting for them to come up with a decent terminal.

I received a release from them (I don't usually read releases for reasons that should soon be obvious) discussing their new controlled memory access (CMA) which improves data access speed.

It states, "The CMA controller differs from conventional controllers in that it uses the CPU to control memory access during disk operations."

Gee whiz. It sounds like they are bragging because they don't have a DMA (direct memory access) chip. It may also mean that they don't have a winchester controller card, which doesn't make sense. They also take exclusive credit for all the standard features of CP/M 3, so I shouldn't expect too much.

No doubt this same PR person would be able to explain to us why it's much better to run translated Z80 software on an 8088 even though it runs half as fast and takes twice as much memory space (but that's a different story).

Color Graphics

I have seen some really super full-color sprite graphics running on the Kaypro II and 4 (you 10s will have to wait). A group of engineers got together and

(continued next page)

formed a company called MicroSphere Inc. (They did it right here in Bend. Oh joy! oh joy!) Don Brittain did most of the hardware and software design, and he had a prototype running here during the SOG. When Bill McDonald saw it, he was absolutely delighted. (They are also doing a B&W composite video board which is advertised in this issue.)

Please assume that I'm biased about this company, both because I've been involved in helping it get started, and because I'm tickled with the way the color board works. Graphics, especially animated color graphics (have I got a PAC-MAN for you) are really exciting!

The color graphics board uses the 9918 and generates standard composite video. It just plugs into one of the PIO sockets, that's it. We plan to put together disks of color graphics software for this system as folks send us their applications.

Anyway, the assembled and tested board, complete with a menu driven screen editor (select your color, draw the shapes and go), a step-by-step course in writing your own software from scratch (with examples in SBASIC), and application programs will be $299.95.

It should be available in mid November (Murphy willing).

MicroSphere
PO Box 1221
Bend, Oregon 97709
503-388-1194
9-5 Pacific Time

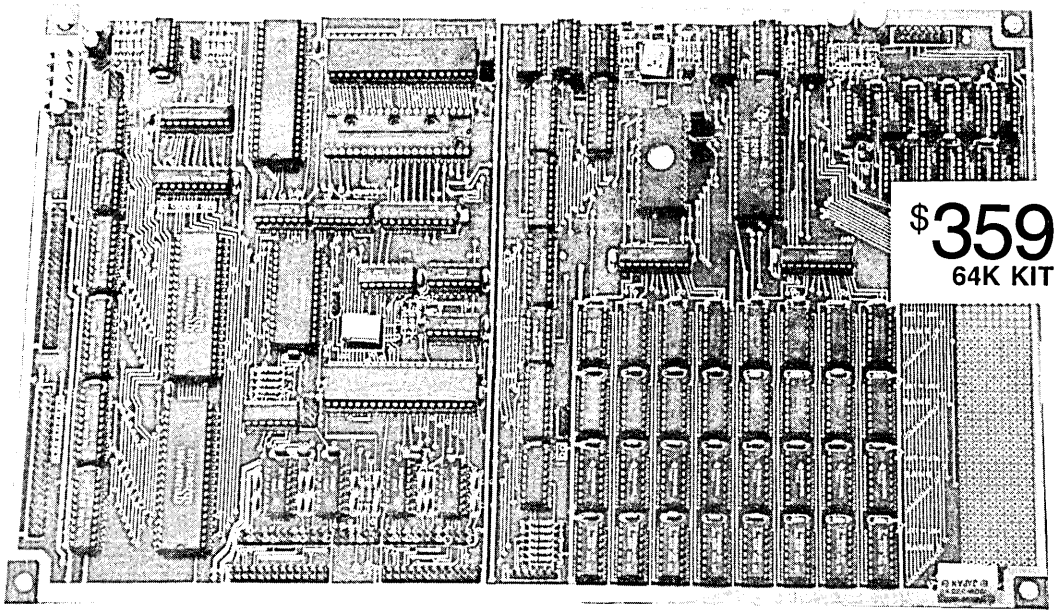
8" Drives For The Kaypro

This is an interesting case of "now you see them, now you don't." I have just been notified by a disappointed subscriber that the Auburn Computer Center, in Auburn CA, wouldn't take his order for their 8" drive interface for the Kaypro. It appears that there are a lot of older Kaypros that don't match the one they used as a model. So, they have gone back to the drawing board. (I mentioned them in the issue #12 Kaypro column.)

■ ■ ■

WADE — I

Universal Interface Expander



$359.00
64K KIT

Use the Universal Interface Expander for Intelligent Communications

Printer Buffer	Parallel or Serial driven Ram disk
System Multiplexer	Par/Ser to Ser/Par converter

features:

- 4 MHz. Z80 cpu
- 256K bytes of 250 nsec. dynamic ram with DMA capability, configured as 4 banks of 64K bytes with each bank split into a 48K lower bank and a 16K upper bank
- Eprom socket configured as a separate bank, supports an Intel 2716, 2732, 2764, 27128 or 27256
- 4 fully buffered parallel ports
- 2 synchronous/asynchrnous configurable serial ports
- Complete documentation included
- Power requirements:
  +5 volts @ 3 amps.
  +/– 12 volts @ .1 amps.

1. Bareboard with software to configure the WADE I as a printer buffer with Ser/Par I/P, Ser/Par O/P	99.95
2. kit with 64K Ram	359.00
3. A+T with 64K Ram	459.00
- for each additional 64K bank of Ram add	79.00
S/H within N.A.	15.00
S/H foreign orders	20.00

all prices shown are in US Funds

allow 4-6 weeks for delivery

Send cheque or money 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.

FORTHwords

Column by Arne A. Henden

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

This is going to be another one of those hodge-podge columns. National Semiconductor has me running ragged trying to bring UNIFORTH up on the 16032 and that leaves little free time!

Get Off The Dime

Before I get started, however, I would like to complain a little. This column was started to encourage programmers to try FORTH, and to present techniques of FORTH usage. I really need reader involvement! If you don't suggest topics, contribute short applications for publication, or write to me, then I sit here in a vacuum. Help! (Editor's note: Sitting in a vacuum sucks! Let Arne know what you are doing with FORTH!)

Vendor News

Quickview Systems (Los Altos, CA) is working on a Rolodex-like package for notebook computers (HX-20, HP-75, etc.). It is written in FORTH (surprise).

Hemenway Corporation now markets a multiuser/multitasking FORTH for their operating system (68000). It is a full 32-bit FORTH, and the specs look good on paper.

8086/8088 Notes

Since we now have many Slicer fans subscribing to Micro C, I am expanding FORTH coverage to include the 8086.

A review of two leading 8086/8088 FORTHs (Quest's FORTH-32 and FORTH Inc's PolyFORTH II Level 2) can be found in the October issue of PC Tech Journal. Both are good, but I suggest looking at UNIFORTH and PC/FORTH before making a decision.

One of the reasons for moving up to a 16-bit processor like the 8088 is the increased address space. Unfortunately, the segmented address structure of the 8088 makes it difficult for programs to grow beyond 64K bytes.

Almost every FORTH uses a single 64K byte segment for FORTH program space, though some, like UNIFORTH, allow data storage in the remainder of the 1M byte address space.

Two new contenders have resolved the addressing problem.

Quest shifts to 32-bit addressing when a program exceeds 64K bytes. PC/FORTH has a new 32-bit version for the 8086, in addition to their 16-bit address version. I thought of creating a 32-bit version for UNIFORTH but gave it up because it looked too slow. If someone has either of these two versions running on their computer, I would be interested in a review and a speed comparison!

8087 Math Chip

While I'm on the subject of the Intel products, I might mention my experience with the 8087. It took me about three weeks to fully exploit all of the 8087 features (warning: there are errors in AP-113) but the work was worth it. My IBM PC running at 4.77 MHz performs the basic floating-point operations at about the same speed as an LSI-11/23 with the FP-11 option, but executes high-level functions like sine and exponentiation about ten times faster than the 11.

Not only that, all operations are performed in double-precision, compared to single-precision FP-11 times! An 8MHz 80186 machine like the Slicer, running an 80187 at 8MHz, would be roughly comparable to a PDP-11/44! Talk about genius!

The current version of UNIFORTH uses the same stack for floating-point numbers and for integers. This really slows down the 8087 because it must convert numbers before moving them off its internal stack and storing them in RAM. A more efficient method would be to have separate stacks for floating-point and integer numbers, and to have the floating-point stack be the 8087's hardware stack.

However, the 8087 can only be 8 elements deep, and some of these are used whenever the high-level functions are calculated, making an effective depth around 5 elements. In addition, you must now keep track of two stacks, and realize that the number converter will automatically select which stack to place the number. If you can live with these restrictions, then I recommend the stack split. (A split stack?)

A new book on the 8087 looks interesting. *8087 Applications and Programming for the IBM PC and Other PCs* by Richart Startz (Brady Publishing, 1983, $20) gives many examples of how to program in 8087 assembly language.

You can now buy a board that plugs into an 8088 socket that will add the 8087 to your system. You might look into that if you have the Co-Power 88 addition for your Big Board.

```
SCR # 10
  0 ( editor) ' INDEX FENCE ! FORGET TRIAD BASE @ HEX
  1 : TEXT HERE C/L 1+ BLANKS WORD HERE PAD C/L 1+ CMOVE ;
  2 : LINE DUP FFF0 AND 17 ?ERROR SCR @ (LINE) DROP ;
  3 : WHERE DUP B/SCR / DUP SCR ! ." SCR " . SWAP C/L /MOD C/L * ROT
  4       BLOCK + CR C/L TYPE CR HERE C@ - SPACES 5E EMIT QUIT ;
  5 : #LOCATE R# @ C/L /MOD ;  : -MOVE LINE C/L CMOVE UPDATE ;
  6 : #LEAD #LOCATE LINE SWAP ;  : #LAG #LEAD DUP )R + C/L R) - ;
  7 : H LINE PAD 1+ C/L DUP PAD C! CMOVE ;
  8 : E LINE C/L BLANKS UPDATE ;  : CLEAR SCR ! 10 0 DO I E LOOP ;
  9 : S DUP 1 - 0E DO I LINE I 1+ -MOVE -1 +LOOP E ;
 10 : D DUP H 0F DUP ROT DO I 1+ LINE I -MOVE LOOP E ;
 11 : M R# +! CR SPACE #LEAD TYPE 5F EMIT #LAG TYPE #LOCATE . DROP ;
 12 : T DUP C/L * R# ! DUP H 0 M ;  : L SCR @ LIST 0 M ;
 13 : RE PAD 1+ SWAP -MOVE ;  : P 1 TEXT RE ;  : IS DUP S RE ;
 14 : COPY B/SCR * SWAP B/SCR * B/SCR OVER + SWAP DO DUP I BLOCK 2
 15       - ! 1+ UPDATE LOOP DROP FLUSH ; BASE !


SCR # 11
  0 ( editor, one screen, from WFR-79MAY01, for figFORTH, 8B ROM)
  1 ( editor explainations)
  2 ( WHERE = when error occurs, WHERE shows screen # and line.)
  3 ( H = Hold line # on stack at pad.)
  4 ( E = Erase line # on stack.)
  5 ( CLEAR = enter with screen # on stack to clear a screen.)
  6 ( S = Spread line # on stack.)
  7 ( D = Delete line # on stack and hold in pad.)
  8 ( M = Move cursor by signed amount and print its line)
  9 ( T = Type line # on stack and hold in pad.)
 10 ( L = reList screen # on stack.)
 11 ( RE = REplace line # on stack with text in pad.)
 12 ( P = Put following text on line.  Proceed with T command.)
 13 ( IS = InSert text from pad to line)
 14 ( COPY = COPY screen #1 to screen #2--#1 #2 COPY.)
 15 : ANOTHER 0 T FLUSH SCR @ 1 + DUP LIST CLEAR 0 RE L ;
```

The 9511 Math Chip

The 9511 is also a stack-oriented floating-point chip. You might try the same stack split trick for it also, but you would be limited to a four-element stack. (A short stack.)

MicroSpeed FORTH for the Apple has gone this way and there is a programmer in Houston who markets a 9511 library for MicroSoft FORTRAN.

Mini FORTH Book Reviews

Tom Mason was kind enough to send me a couple of new FORTH books to review, and I've broken down and purchased several others. Devoting an entire page to each for a review would be fair to the authors, but expensive for Dave Thompson!

Introduction to FORTH

This 142-page minibook (by Ken Knecht, Blacksburg Continuing Education Series, 1982, $10) is based on MMS-FORTH for the TRS-80. It has many good examples, with equivalent BASIC programs for each. Knecht assumes you have no knowledge of FORTH, and only minimal computer experience. Several words not included in FORTH-79 systems are presented, including string manipulators.

However, the book has some weaknesses. The English is stilted, it only covers MMS-FORTH, and it does not indicate where that dialect differs from the FORTH-79 Standard. No system-level words are included (such as INTERPRET or CREATE).

Discover FORTH

Tom Hogan is not one of the FORTH gurus who typically author FORTH texts. He decided that the best way to learn FORTH was to write a book about it, and proceeded to learn the language. His 120 pages of text (plus 25 pages of appendices) are clear and straightforward but lack depth. No system-level words are presented, and no examples longer than a single line are given. My feeling is that the book is way overpriced for what you get. (Thom Hogan, Osborne Press, 1982, $15)

And So FORTH

On first glance, this hefty 370-page book looks good. It covers both high-level and system-level words, and has many examples (including a primitive data-base system!). There are even exercises for each chapter (without solutions). This book is obviously aimed at the college textbook market.

I recommend this book with two reservations.

First, only the first 180 pages are really applicable to every reader; the last half of the book is devoted to Haung's Victor 9000 FORTH (really a User's Manual for that system). A college text cannot be specific to any computer, much less one that few universities would have.

Second, the first half of the text contains nearly 50 pages of material by other authors (first printed in FORTH Dimensions). The reprint selection is good, but extensive reprinting bothers me. (Timothy Huang, 1983, $25, available through MVP)

FORTH Fundamentals, Volume 1

Volume 1 is a 190-page (plus 46 pages of Appendices and index) text covering the usage of the most important FORTH-79 and FIG-FORTH words. Volume 2 is a combined glossary and detailed explanation of the FORTH core (kernel) words. The first volume has many excellent examples though the writing tends to be dry.

The first 6 chapters cover the fundamentals of FORTH, from dealing with the stack, arithmetic, data storage, defining words, conditional branching, through the actual FORTH memory organization. The remaining 6 chapters cover advanced topics such as the address and text interpreters, the header structure of words, terminal and disk I/O, and basic block editing and loading. The appendices include typical system error messages, a history of FORTH, a vocabulary list, and an index.

If *Starting FORTH* didn't exist, I would recommend Volume 1 as the text-of-choice for beginners. The two-volume set should give even *Starting FORTH* a run for the money, but I reserve judgement until I can get my hands on a copy of the second volume. (C. Kevin McCabe, Dilithium Press, 1983, $16)

I've found that B. Dalton Booksellers carry most of the FORTH titles at one time or another. The other source is Mountain View Press, which advertises in every issue of Byte.

Next Month

We've got some FORTH applications, plus more details of the FORTH-83 Standard coming up. I'm attending the FIG National Convention in October, and I'll report on happenings at the meeting. Have a happy holiday season!

FORTH Screen Editor

Editor's note: Charles Johnsen sent in the following one-screen text editor.

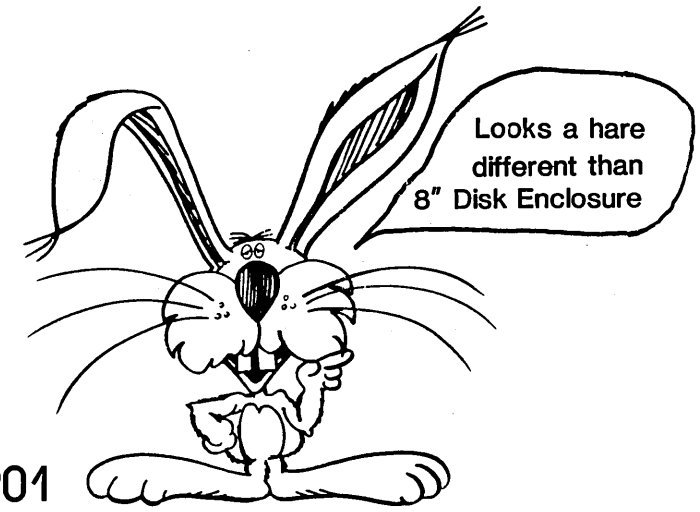
This editor is nothing fancy, and that's on purpose. All it is meant to do is give a beginner with nothing but ROM FORTH (or other standard fig FORTH like IFORTH) a way to enter screens. If you hand-enter the screens, you can forget the first line. I never used those last few functions in the ROM so I used that line to "forget" them so I'd have a little more space.

Notice that these words are added to the FORTH vocabulary so a special editor vocabulary is not created. You might want to do that now, but I suggest waiting for a more complete editor.

Charles Johnsen
19704 E Loyola Circle
Aurora, CO 80013

■ ■ ■

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

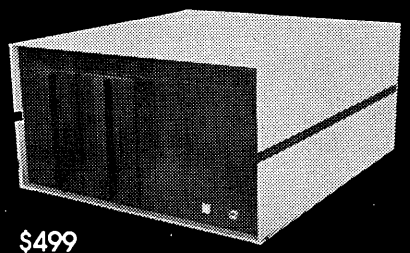
8620 Roosevelt Ave.
Visalia, CA 93291
209-651-1203

Main/Frames

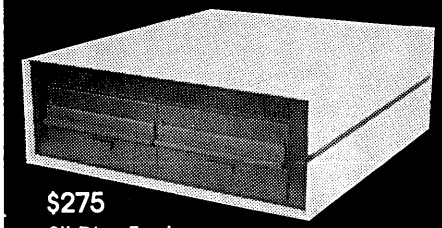
Main/Frames

from
$175

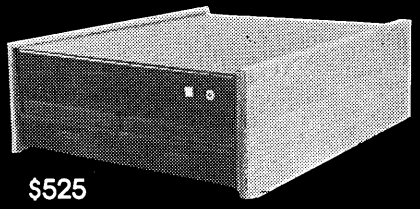
- 70 Models of Enclosures
- Assembled and tested
- Quasi-Coax Motherboards
- Power Supply
- Card cage and guides
- Fan, line, cord, fuse, power & reset switches



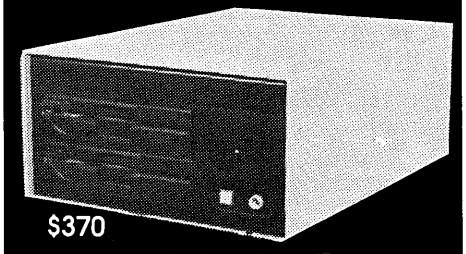
$499
8" Floppy Main/Frame



$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

C'ing Clearly

By Gary L. Hylton

1007 Macfarlane
Aberdeen, WA 98520
206-533-7924

Pointers in C can be a problem. My first attempt to use pointers sent the processor in my Osborne off to Peoria from where it rang the bell and wrote wonderful patterns all over the display. Pointers can, however, be mastered and they are powerful and elegant tools.

I've been hacking in BDS C for nearly two years. Several months ago, a program I've been working on got out of hand. The sorts could no longer be handled in memory, so I decided to write some virtual memory routines. After some preliminary cuts I decided that it would be cleaner to use a compiler that treated files in the standard C manner, one byte at a time.

So in one fell swoop I subscribed to Micro C (I had purchased it from the newsstand on occasional trips to Portland), ordered the Aztec C compiler (a $50 discount for Micro C subscribers) and since I'd just saved the $50, I ordered the Big Board User Disk #14. It all arrived on Friday. What luck.

The first program I ran was setclk.com. After several tries I was still getting an error message, 'ERROR CM at 0000H', but, print2.com verified that the clock had indeed been set. An examination of setclk.asm led me to the conclusion that I'd rather program in C than assembler.

Since I had a shiny new compiler, why not just knock out a program to set and read the clock and maybe even modify my favorite print utility to include automatic time and date? See Figures 1 and 2 for date.c and setdate.c. Note that anything within slash-asterisks, i.e. /* comment */ and anything that is typeset is just a comment.

```
/* date.c */
#include "libc.h"
   /* this is required  */

main()
{
```

Since we'll be putting bytes somewhere in memory to hold the current value of the year, month, day, hrs, etc., let's declare a pointer for each one. Since we're pointing to a byte, the pointer is of the type "character."

```
char *year, *month, *day,
     *hrs, *mins, *secs;
```

In this program we initialize the pointers to point to the memory locations where the current values are stored. The following code initializes, or assigns a value to the pointers so they point to the proper memory locations.

```
year = 0xff81;
month = 0xff80;
day = 0xff7f;
hrs = 0xff7e;
mins = 0xff7d;
secs = 0xff7c;
```

You have to initialize pointers, otherwise you don't know where they are pointing. (Editor's note: They might be pointing at you.) If they are pointing inside your operating system or program, a write can do interesting things (Peoria here we come.) Needless to say, problem pointers can lead to some interesting debugging.

The usual way to initialize a pointer is to declare a storage buffer, then assign its address to the pointer. For example, the following statements declare ap and bp to be pointers to characters and storbuf a 10 element array:

```
char *ap, *bp;
char storbuf[10];
```

Of course, ap and bp need somewhere to point, so let's assign addresses from storbuf:

```
ap = &storbuf[0];
bp = &storbuf[1];
```

Now we can do some assignments.

```
*ap = 'a';
*bp = 'b';
```

Thus, ap points to the ascii value 'a' and bp points to the ascii value 'b'. Since the address that ap points to is the address of storbuf[0] (we assigned that address to the pointer), storbuf[0] now contains 'a'.

The address of storbuf[0] is assigned by the compiler. If storbuf is static or external, the assignment is made at compile time. The address is assigned dynamically during runtime if storbuf is automatic.

(continued next page)

Figure 1 - Date and Time Read Program for the Big Board II

```
/*  date.c
    a program to read the date and time on the Bigboard II computer.
    copywrite (c) 1983 by Gary L. Hylton
    release to the public domain with the provision that it not be
    sold for profit.  9/18/83.

    I would prefer that you leave my name on the program and add yours
    if you make modifications.  Thanks. GLH.
*/


#include "libc.h"

main()
{
        char *year, *month, *day, *hrs, *mins, *secs;
        year = 0xff81;
        month = 0xff80;
        day = 0xff7f;
        hrs = 0xff7e;
        mins = 0xff7d;
        secs = 0xff7c;
        printf("%02x/%02x/%02x  %02x:%02x:%02x", *month, *day,
                *year, *hrs, *mins, *secs);
}
```

Meanwhile, back in our original program, we assign specific memory locations to our pointers. You can verify what's in those locations via the BB II monitor dump command.

```
dff7c,ff81 cr
```

In fact, you can even watch the values change as time passes. You will discover that the hex characters of the hex dump are the values we would like printed out and that the order of the information is mixed up, at least for our purposes.

What's important here is if we convert the binary data in the memory location 0xff7c and display it on our console, we'll have seconds displayed. We don't have to use a hex address, since we already assigned those addresses to our variables—month, day, etc. C has an easy way of making the conversion and doing the display. It's called the "printf" statement.

```
printf
("%02x/%02x/%02x  %02x:%02x:%02x",
 *month, *day, *year, *hrs,
 *mins, *secs);
```

This statement converts to hex and prints the values pointed to by our variables. The conversion is accomplished by %02x. The '%' sign introduces the conversion specification. The '2' specifies a minimum field width. The '0' causes the field to be padded with zeros. The 'x' converts the binary value to hex.

Finished. A simple, clean C program (see Figure 1). Type "date" and the date and time appear. I like that.

Setting The Date

Now, all we need is a program to set the date. (See the code in Figure 2 as you read the following comments.)

Again, I used pointers. argv[1] contains a pointer to a string which contains (for example) "09/17/83." The program is expecting a string in this format. The conversion is accomplished by a function called hextoc().

We could use a standard library function, scanf, but that would require some kind of query (a message to the screen asking for the date and time). That's just too messy for a little utility to set the date and time on our own computer. Our program should be direct and get the job done with as little fuss as possible.

The functions hextoc() expects a pointer to a character string. It treats the two characters as hex and returns the binary value. Since the contents of argv[1] point to our example string "09/17/83," we can pass the pointer to hextoc() and it will return the binary value "09." This is the value we want assigned to the memory location we call "month." Here is the code:

```
*month = hextoc(argv[1]);
```

the pointer in argv[1] points to the zeroth element of our example string, "09/17/83." We can point to the "17" which is the third element, if we add 3 to the pointer.

```
*day = hextoc(argv[1] + 3);
```

The year, "83" starts with the sixth element, so add 6 to the pointer.

```
*year = hextoc(argv[1] + 6);
```

Now, let's do an error check on the time string.

```
if (strlen(argv[2]) != 8)
{
puts("can't use that time\n");
exit(0);
}
```

Hours, minutes, and seconds are handled in exactly the same manner as the date. Now use the second argument of the command line.

```
*hrs = hextoc(argv[2]);
*mins = hextoc(argv[2] + 3);
*secs = hextoc(argv[2] + 6);
```

And finally, a word to let us know that the program did something.

```
puts("done\n");
```

The pair, date.c and setdate.c are finished. All that is needed are the two functions hextoc() and hex().

After I finished using the Aztec C compiler, I ran the source through the BDS C. A couple of observations:

First, BDS C compiled and linked in the twinkling of an eye. Aztec took forever. I finally made a submit file to handle all the operations.

Second, I had used the function hextoc() in another program only I had called it hextoi() and had left it undeclared, (by default an integer). Aztec C took its types seriously and generated 33 error messages. BDS C which doesn't have casts as such so it lets you stuff an integer into a character, which is according to K&R. I should have no complaint; one of the reasons I purchased the Aztec C was to generate standard code that can be transported.

Third, the com file generated by Aztec CII is over 5K longer than the BDS C code. That's the price for having long and float and other standard features. For a simple utilities like these I'd use BDS C.

New Release

Software Toolworks has sent in a short sheet on their new version of C80. They say they've improved code generation and have expanded the library in version 3.0.

They also have a separate "Mathpak" that supports 32 bit longs and floats and includes C source for transcendental functions.

The integer-only C80 vrs 3.0 sells for $49.95 and the add-on mathpak sells for $29.95. C80 generates assembly language output for Macro-80, RMAC, and for the assembler they include with the compiler.

Present owners of C80 may upgrade to vrs 3.0 for $10.00.

Contact the Software Toolworks, 15233 Ventura Blvd, Suite 1118, Sherman Oaks, CA 91403.

■ ■ ■

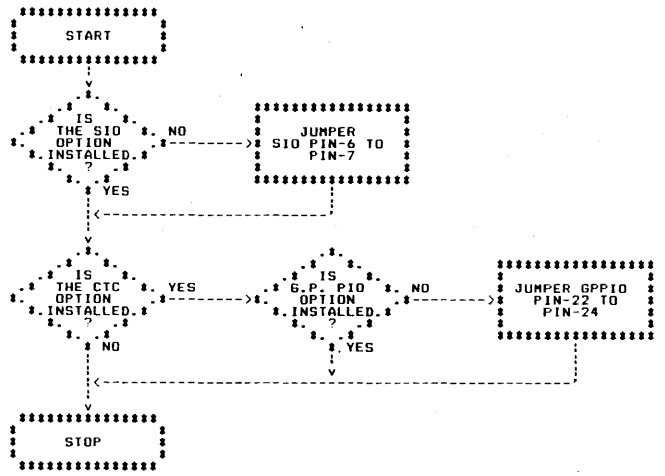
Figure 2 - Date and Time Set Program for the Big Board II

```
/*  setdate.c
    a program to set the date and time on the Bigboard II computer.
    copywrite (c) 1983 by Gary L. Hylton
    release to the public domain with the provision that it not be
    sold for profit.  9/18/83.

    I would prefer that you leave my name on the program and add yours
    if you make modifications.  Thanks. GLH.
*/

#include "libc.h"

main(argc, argv)
char *argc, *argv[];
{
        char *year, *month, *day, *hrs, *mins, *secs;
        year = 0xff81;
        month = 0xff80;
        day = 0xff7f;
        hrs = 0xff7e;
        mins = 0xff7d;
        secs = 0xff7c;
        if(argc != 3)
        {
                puts("\nsetdate.c v 1.0\nCopyright (c) by
                                        Gary L. Hylton, 1983\n");

                printf("sets date and time on Bigboard II\n\n");
                printf("usage:   A>setdate <date> <time><cr>\n");
                puts("example: A>setdate 09/17/83 14:23:15<cr>\n");
                exit(0);
        }
/* first do the date                                                */
        if (strlen(argv[1]) != 8)
        {
                puts("can't use that date\n");
                exit(0);
        }
        *month = hextoc(argv[1]);
        *day = hextoc(argv[1] + 3);
        *year = hextoc(argv[1] + 6);
        if (strlen(argv[2]) != 8)
        {
                puts("can't use that time\n");
                exit(0);
        }
        *hrs = hextoc(argv[2]);
        *mins = hextoc(argv[2] + 3);
        *secs = hextoc(argv[2] + 6);
        puts("done\n");
}


/*  given a pointer, chrptr, to the first of two
ascii hex characters, hextoc() returns the binary value
*/

char hextoc(chrptr)
char *chrptr;
{
        return (16 * hex(*chrptr) + hex(*(chrptr + 1)));
}

hex(nibble)
char nibble;
{
        return((nibble > '9') ? nibble - '7' : nibble - '0' );
}
```

FLOWCHARTER

★ ENGINEERS
★ TEACHERS
★ EXECUTIVES
★ SYSTEMS ANALYSTS
★ DRAFTSMEN
★ OFFICE MANAGERS
★ BANKERS
★ SERVICE PEOPLE
★ TECHNICAL WRITERS
★ CONSULTANTS
★ PROGRAMMERS
★ YOU

```
/* Flowchart showing which Big-Board I pins to
   jumper depending on the installed I/O options */

        T       START;
        D       IS SIO OPTION INSTALLED ?;
        IF      NO GOTO C; ELSE YES;
%A:     D       IS CTC OPTION INSTALLED ?;
        IF      YES GOTO D; ELSE NO;
%B:     T       STOP;
        SKIP    1C;
%C:     P       JUMPER SIO PIN-6 TO PIN-7;
        GOTO    A %B ET;
%D:     D       IS G.P. PIO OPTION INSTALLED ?;
        IF      NO GOTO E;
        IF      YES GOTO B %B ET;
        SKIP    2C;
%E:     P       JUMPER GPPIO PIN-22 TO PIN-24;
        GOTO    B %B ET;
        END;
```



Two years in the making! Doing what they thought couldn't be done on a micro! EasyFlow is a high-level language for producing neat, accurate flowcharts. You describe the flowchart using a simple command language; EasyFlow then produces a complete flowchart.

Not just for programmers! EasyFlow is for everyone needing flow type charts: engineers, executives, systems analysts, draftsmen, office managers, bankers, service people, technical writers, you name it - anyone who needs to be able to explain things in clear and understandable manner.

People have been using flow charts for a long time, for both programming and other uses. The problem is producing them: if you do them quickly by hand, they look awful; if you take your time they look better, but they take forever and are impossible to update or revise without starting over. Really good looking flowcharts require a typesetter, a draftsman and a lot of patience. With EasyFlow you can produce excellent flowcharts quickly, and then modify, correct and update them with very little effort.

EasyFlow is a lot like a high-level language; it accepts source statements, processes them and outputs an object file, but the object file is a flowchart instead of a program. To produce a flowchart you create a source file (using ED or whatever) that describes the flowchart. EasyFlow reads the source file, builds the flowchart in memory and then outputs it. The flowchart is normally output to disk, but it can be redirected to the console, punch or printer.

The EasyFlow command language has facilities for describing what shapes are to go where, what text is to go inside the shapes (centered in the shape automatically by EasyFlow), and the lines that are to connect the shapes. Input to EasyFlow is free-format in much the same manner as C programs, and each statement is terminated with a semicolon. Comments are indicated by "/*" and "*/"; comments can be nested, allowing you to "comment out" an entire section of source, even if it contains other comments.

The example above shows both the source text describing a simple flowchart and the actual flowchart produced. This particular flowchart explains how to setup the interrupt daisy chain jumpers on a Big-Board-I, for all possible combination of installed options. This IS explained in the BB-I documentation, but the flowchart makes it a lot more clear.

EasyFlow works with just about any printer; some are better than others for producing flowcharts, but even a teletype can produce useful flowcharts. A "full sized" flowchart is five shapes wide, eleven shapes high, and requires a printer capable of printing at least 132 columns wide. Printers capable of 80 columns can print flowcharts three shapes wide. A "full sized" flowchart printed at ten characters per inch and eight lines per inch is thirteen inches wide and twenty inches high; reducing such a flowchart by 50% yields a standard-page sized chart. The example chart shown here was a full-sized chart that was reduced. Special printer programs are available for printers such as the MX80, MX100 and u92 which allow flowcharts to be produced on standard 8.5" x 11" paper directly.

EasyFlow comes equipped with twenty standard flowcharting shapes. User defined shapes can be easily added to cover just about any application - it is as simple as editing the EasyFlow configuration file which defines the shapes.

In addition to user definable shapes, EasyFlow allows user selected line drawing characters. The characters for "horizontal line", "vertical line", "upper left corner" and so on are user selectable. You can make you own choice and even take advantage of printers that have special line drawing characters available.

EasyFlow is written in Z80 assembly language and is fast and efficient. A rather large flowchart typically takes about ten seconds of computing time; the example chart shown here is very small, and took less than three seconds. Execution time will vary depending on CPU speed and the type of disk drive, but in any case it is quite fast. Since the source programs tend to be small (one hundred lines of source gets you a very full flowchart), and since EasyFlow needs only a single pass, disk I/O time tends to be modest. EasyFlow was developed on a 2.5MHz system with floppies, and it produces charts at a very respectable speed.

In addition to producing great flowcharts from perfect instructions, EasyFlow also produces meaningful error messages from imperfect instructions. A typical EasyFlow error message is:

```
046  INVALID EXIT-DIRECTION FOUND WHILE PROCESSING CELL B2.
       IF NO GOTO BAFFY XV ET
                   ?
```

The first line is an english-language error report that outlines, as clearly as is possible in one line, what the problem is and where we were in the flowchart when the problem was detected. The second line is the source text that EasyFlow was processing at the time. The third line (the question-mark) points to the exact word that was being processed. Every EasyFlow error message has a number (046 in this case) so that you can quickly look up a detailed explanation of an error in the Error Description Appendix of the manual. This explains, in detail, what the error is, why it might have occurred, and how to go about correcting it.

The EasyFlow manual is complete, and is designed for both novices and experienced users. For the novice it explains (in the correct order, starting from the beginning of the manual) how to install EasyFlow on your system, how to back up the distribution diskette, and how to produce your first flowchart using one of the included demo charts. This is followed by a tutorial that explains the use of EasyFlow through the use of examples. The source text of all the examples in the manual is included on the EasyFlow distribution diskette, giving you a starting point for experimentation. For more experienced users there is a detailed syntax and operation description of all the EasyFlow commands, a chapter on configuring EasyFlow to produce the best charts with your particular printer, the error appendix and a chapter on defining your own shapes.

EasyFlow runs on Z80 CP/M machines that have a TPA of 38K or more. EasyFlow is available on SSSD 8" diskettes and KayPro format 5" diskettes. Other five inch formats: call.

$49.95 ($59.95 in Canada; Ontario residents add 7% PST).

Send check or money order today to:

HavenTree Software Limited
R.R. #1, Seeley's Bay, Ontario, Canada
K0H 2N0 (613) 542-7270

Pascal Procedures

Column by John P. Jones

6245 Columbia Ave.
St Louis, MO 63139

There's a full-page advertisement in the Sep/Oct issue of *The Journal of PASCAL and ADA* for JRT PASCAL V4.0.

The ad notes the following changes from vrs 3.0: There is a two-fold increase in compiler speed, system requirements are now 60K (Would 59.5K work?), a company called Blue Earth sells it (and guarantees shipment in 48 hours), and the price is $100.

Since no other changes were listed, I think the increase in price is excessive and I plan to continue using my version 3.0.

I called JRT Systems twice but couldn't reach anyone who would comment on either 4.0 or their new Modula 2 compiler. They were however, willing to send information by mail.

The information turned out to be photocopies of two magazine advertisements. One of the ads is for the three JRT products—PASCAL 3, PASCAL 4 and Modula 2—which could be ordered directly from JRT Systems for $39.95, $69.95 and $99.95 respectively. (Blue Earth must feel that prompt shipment is worth $30 extra to a lot of potential customers.)

The other ad from JRT Systems covered only Modula 2. Unfortunately, most of the ad concerned Modula 2 as a language, not JRT's implementation of it. If any of you have additional information along this line, please let me know.

Meanwhile let's take a look at a very controversial portion of Pascal.

Sequential Files

File handling in standard PASCAL is perhaps one of the language's weakest features. As defined by Jacques Tiberghien in The Pascal Handbook (Sybex, 1981), "A file is a SEQUENCE of data items, all of the same type, physically stored in the peripheral equipment of the computer." (Caps mine) The sequential nature of data access make the standard PASCAL file look like a magnetic tape file.

Let's first look at "standard" PASCAL file I/O and then discuss some of the extensions which most PASCAL compilers have implemented.

Files must be declared just like any other variable in PASCAL, and for some implementations, must be specified as parameters in the PROGRAM statement (see Figure 1).

When a file is declared, an additional variable and associated pointer are automatically created. The window or buffer variable, which is of the same type as the file, is the only means of transferring data to and from the file.

The RESET statement opens a standard file for input, and in most implementations associates the file variable with a system file name.

```
RESET (infile, 'B:MAGIC.SPL');
```

This statement, opens the file and reads the first item from the file into the buffer variable. The buffer value may then be assigned to another variable of the same type.

```
this_one := infile^;
```

The GET statement advances to the next item in the file and transfers it to the buffer variable.

```
GET (infile);
```

The function EOF becomes true when the last item has been recovered from the file.

```
REWRITE (outfile, 'A:TEMP.DAT');
```

This creates (if necessary) and opens a file for output. Again, data is transferred through the window variable. The PUT statement is the converse of the GET statement.

```
outfile^ := number;
PUT (outfile);
```

Within the standard, there is NO mechanism for both input and output in the same file. To alter an item in a file requires reading the input file (while writing to a temporary output file) until the desired item is reached. Then you write the altered item to the temporary file and continuing the GET/PUT's (reading and writing) for the remainder of the file. If a RENAME function is not available, the temporary file must then be copied back to the original file to complete the update.

Random Access Files

Most PASCAL compiler authors are well aware of the cumbersome nature of the above procedure and have extended file I/O to include facilities for random access. Every implementation of random file I/O is different but all have a way to locate a particular item in a file and a way to read and write that item.

PASCAL/M adds a positioning statement (SETNEXT), PASCAL/MT+ combines them in new statements (SEEKREAD,SEEKWRITE), and JRT PASCAL has extended the READ and WRITE statements. (JRT requires a random file be OPENed, not RESET.)

Of course, it isn't practical to go into the idiosyncrasies of random file I/O for even a few of the available PASCAL compilers so you'll have to consult your own manual for details. With random

(continued next page)

Figure 1 - Pascal File Declaration Example

```
PROGRAM file_demo (infile, outfile);

  TYPE  word = array [1..15] of char;

        equipment = (rod, robe, hat, brazier, talisman,
                     powder, familiar);

        magic_spell = record
        chant : array [1..80] of word;
        needed : array[1..5] of equipment;
        end;

  VAR   infile : file of magic_spell;
        outfile : file of real;
        this_one : magic_spell;
        number : real;
        .
        .
```

file I/O, PASCAL becomes much more useful. (Definite understatement!)

The pre-defined type TEXT is available for most PASCALs (though not for JRT PASCAL). With TEXT files, all I/O is ASCII. Other types of files transfer data in internal binary format.

TEXT files are often split into segments (lines) separated by end-of-line characters (for CP/M it's CR, LF). The pre-defined files INPUT and OUTPUT are of type TEXT. All console I/O is via these two files.

Because of the special characteristics of TEXT files, a set of I/O statements has been dedicated to them.

```
READ (filevar,var1 {,var2,...});

{ filevar can be omitted
                    for console I/O }
{ also, the above is equivalent to: }

var1 := filevar^;
GET (filevar);
```

Note: console input does the GET before the assignment (in the above example).

READLN functions in the same way as READ except that after satisfying the input list, the file pointer is advanced to the next end-of-line character, the function EOLN will be set FALSE, and the next READ will be from the following line.

WRITE and WRITELN are the equivalent procedures for TEXT file output.

PASCAL/MT+ requires the file variable and system file name be linked to one another with the ASSIGN statement before the file can be opened. The RESET and REWRITE statements above thus follow this form:

```
ASSIGN (filevar, 'A:FILE.NAM');
RESET (filevar);
```

Versions of JRT PASCAL earlier than 3.0 did not have GET and PUT, all file I/O was handled through extensions to READ and WRITE. Both methods are available with version 3.0. A pre-defined type TEXT is not supported in JRT.

Instead, a text file must be declared as a 'file of char' and opened (RESET, REWRITE) in TEXT format. For example:

```
VAR  text_file : file of char;
       .
       .
RESET(text_file, 'A:INP.FIL', TEXT,
      buffsize);
```

Output to the system list device (printer) is generally handled by declaring it as a TEXT file, after which it can be accessed as any other output text file. (JRT PASCAL uses the special SYSTEM statement to activate output to the printer.)

```
VAR  printer : TEXT;
       .
       .
RESET (printer, 'LST:');
WRITELN (printer, var1,var2);
```

Finally, most PASCAL's have an explicit CLOSE statement in addition to the implicit close which is executed when exiting a program. This allows more flexibility when many files are being handled.

Figure 2 - Print Routine for Pascal/MT+

```
program print_doc (input, output, manual, printer);

var
        printer,manual : text;
        line_counter : integer;
        reply : char;
        result : integer;
        line : string[82];
        name : string[20];


procedure print_page;
begin
        line_counter := 0;
        readln(manual,line);
        if not (eof(manual)) then
         repeat
           writeln(printer,line);
           readln(manual,line);
           line_counter := line_counter + 1;
         until (line_counter = 65) or (eof (manual));
        if not(eof(manual)) then writeln(printer, line);
end;

procedure skip_page;
begin
        line_counter := 0;
        repeat
           readln(manual,line);
           line_counter := line_counter + 1;
        until (line_counter = 66) or (eof(manual));
end;

begin   { program }
        write ('Enter filename to print : ');
        readln (name);

        assign (printer, 'LST:');
        rewrite (printer);
        assign (manual, name);
        reset (manual);

        repeat
           print_page;
           skip_page;
        until eof(manual);

        close (manual, result);
              { result is an integer error code }


        write ('Flip paper, press any key to continue ');
        read (reply);

        assign (manual, name);
        reset (manual);

        repeat
           skip_page;
           print_page;
        until eof(manual);

end.
```

Figure 2 - Print Routine for JRT Pascal

```
program print_doc;
  const  lf = 10;         {line feed value}
         endfile = 26;  { ctl-Z = end of file}

  var
       manual : file of char;
       name : string[20];
       ch, reply : char;
       line_counter : integer;

procedure do_page;
begin
   if not (ord(ch) = endfile) then  {if file not empty}
   begin
   line_counter := 0;
   repeat
     repeat
          write(ch);                          { send to printer }
          read (manual; ch);
     until ord(ch) in [lf,endfile];           { continue til line feed or eof}
     line_counter := line_counter + 1;
   until (line_counter = 66) or (ord(ch) = endfile); { 66 lines per page }
   writeln;                                    { allow for discarded lf }
   end; { if }
end;

procedure skip_page;
begin
 if not (ord(ch) = endfile) then
   begin
   line_counter := 0;
   repeat
      read (manual; ch);
      if ord(ch) = lf then line_counter := line_counter + 1;
   until (line_counter = 66) or (ord(ch) = endfile);
  end; { if }
end;


begin
   write ('Input file name : ');
   readln (name);                 { get file name }

   reset (manual, name, binary, 2048);
{          filevar, filename, mode, buffer size }
{ use binary mode so can handle <LF>'s }

   system (list);                 { activate printer }
   system (nocons);               { kill console }
   read (manual; ch);             { get first char }

   repeat
     do_page;
     skip_page;
   until ord(ch) = endfile;

   system (nolist);               { kill printer }
   system (cons);                 { enable console }
   close (manual);

   write ('Turn paper over, enter any character to continue ');
   readln (reply);

   reset (manual, name, binary, 2048);

   system (list);
   system (nocons);

   read (manual; ch);              { opposite sequence for even pages }

   repeat
     skip_page;
     do_page;
   until ord(ch) = endfile;

   close (manual);

end.
```

Two Example Programs

The following two programs (see Figure 2) perform the same function. One was written for PASCAL/MT+, the other for JRT PASCAL. They take a TEXT file which is separated internally into 66-line pages, print the odd pages, prompt the user to remove and reverse the paper, and then print the even pages.

The programs assume that there are no form feeds in the file and that any style parameters such as blank lines at the top of the page, page numbers, etc. have been included in the file. If you're printing documentation on tractor feed paper rather than cut sheets these programs are quite useful.

Of course, they also point out some of the differences in file I/O between these two implementations of PASCAL.

The JRT version does NOT expand tabs. (Editor's note, the tabs are probably large enough as they are.) If you added the getstring function presented in Micro Cornucopia issue #12 to the JRT version, the two programs would operate virtually identically.

Both versions will run a 120 CPS printer full tilt with only a slightly longer pause when skipping pages for JRT (4 MHz Kaypro 4).

Obvious expansions to the programs could take care of embedded form feeds and do some rudimentary (or fancy) page formatting. Try it!

JRT Systems Inc.
45 Camino Alto
Mill Valley, CA 94941
415-388-0530

Blue Earth
1891 23rd Ave
San Francisco, CA 94122

■ ■ ■

On Your Own

By David Thompson

I've always felt that if you are going to start something, then go for it. Anyway, the following is the first saga of a story I'm sure will be unfolding for quite a while. If it goes, it'll be one of the biggest goers going. If it doesn't, it'll go down in glory.

Business Computer Network

Well, Business Computer Network, a small (and new) company is obviously thinking very big, and obviously not tied to the dock. It's sink or swim time for them and I thought you'd probably enjoy a few details.

Their idea was to purchase large blocks of time from the commercial bulletin boards (like the Source) and then resell the time to end users like you and me.

The Carrot

In order to make everything very easy, BCN spent 8 months developing a software package that is user friendly. You don't even need to know anything about CP/M. (The BCN disk has system tracks, and it autoboots right into the program called BCN.COM.)

After you answer some questions about your system (what it takes to dial out) and yourself (especially your Visa/Master Card number), the program automatically dials a toll-free number at BCN headquarters where your Visa number is traded for a password into the system and the phone number of the closest dialup port. The password and phone number get implanted in your copy of BCN software, and from then on, access is a simple matter of entering "BCN" and a carriage return. (It would probably be best not to let someone else copy your BCN modem disk—your Visa bill could rise dramatically.) The BCN software also lets you communicate with other individuals who have the BCN modem software.

Of course, there is the obvious question. Why would someone purchase time from BCN rather than purchasing directly from the supplier?

Assuming users are interested in getting involved in the commercial data banks (a critical assumption in this case), there are a number of advantages to a BCN connection. For instance, BCN membership gives you access to a number of commercial services rather than just one, the initial fee is lower ($5 instead of hundreds), and a third is, of course, their software package (Kaypro and Smartmodem compatible).

Mass Marketing

The most dramatic thing that BCN did however, was not the purchase of large blocks of time or the creation of zingy new software. You see, BCN put a disk of their modem software (and a small manual) in all 65,000 copies of Profiles Magazine (September/October issue). In addition, they are sending 10,000 disks to Kaypro dealers and are putting 25,000 disks in boxes with new Kaypros. That's 100,000 disks of their fancy autoboot, menu-driven software turned loose on the public. They have really stuck their idea in front of a lot of folks.

The time frame is also interesting. From the time they got the idea of distributing their software free to virtually every recent Kaypro purchaser (plus any other Profiles readers), they had just three weeks to put the whole package together, purchase the disks, have them copied, and have them hand-bagged with the magazine.

I talked to Morris Camp, Vice President of BCN about 2 weeks after Profiles hit the streets. He said they didn't know yet whether it was going to be boom or bust, but he was glad those first three weeks were over.

He indicated that they were getting 300 to 400 enquiries per day on their toll-free voice lines but that people weren't really signing up in any numbers yet.

The package looks very good. They coordinated the disk and manual with a one-page article in Profiles written by Morris. There is no question that if they have really come up with what people want, they should be very busy.

Two Big Needs

Morris said that he sees two basic needs that aren't being met. The commercial networks haven't really caught on so they need customers. Meanwhile, many potential customers are intimidated by data communications. They don't understand it. He hopes that they have found the solution to both these needs.

If it's easy enough for folks, he assumes they'll take advantage of these services. BCN can then sell small chunks of network time to large numbers of people, charging retail prices for the time plus a $5.00 per month (or more depending on number of database accesses) membership fee.

He may well be right, and BCN is certainly giving it its best shot. However, there are some things that I'll be watching.

Three Problems

First, not all the disks are going to reach new users' hands. A lot of people are paying more than $3.00 each for double density 5" disks, so it's cheaper to purchase extra copies of Profiles magazine from their local Kaypro dealer than to purchase new disks.

Second, many people are not willing to give someone carte blanche with their MasterCard. Plus, the idea of monthly surprises (even legitimate surprises) sometimes doesn't go over too well.

Third, no one is really sure that large numbers of people are waiting with bated breath (not pleasant sounding is it?) for a chance to access the data networks.

One Potential Problem

It turns out that BCN is also releasing 100,000 free copies of CP/M (that's how they autoboot their program) with no agreements or restrictions. How long did it take Digital Research Inc (DRI) to distribute the first 100,000 copies of CP/M? How would you feel if someone released 100,000 copies of your bread and butter?

Plus, all those copies have the same serial number. The only place they'll be traced, is to BCN. (The grapevine has been full of stories about Digital Research's heavyweight treatment of dealers caught making a few "extra" copies of CP/M.)

I asked Morris if he had checked with DRI before shipping his product and he said he hadn't—though he didn't believe that DRI would be damaged because the software was intended for Kaypro users, all of whom already have CP/M.

I hope he is right about Digital Research, but this is an area where Murphy can stick his finger. (I have visions of a forced recall of all copies of the disks.) Anyway, it's obvious that the most valuable part of BCN's offering isn't the disk, it's the system tracks (onto which folks can add their own boot and cbios).

(On Your Own continued)

Finally

I've talked about finding a need, writing user friendly software, testing it, and marketing it well. BCN has done all of this in grand style. Now we'll sit back and see what happens. Perhaps Murphy is vacationing in the Bahamas (they don't need him there either) and BCN will be rewarded for its innovation. And perhaps, there'll soon be a whole new market—free software (you have to enter your charge card number before you can run it, and late some moonless night it calls "home").

Technical note: The way I checked the system tracks on the BCN disk was to use Inspect (DU also works) to look at the system tracks. In the first 16 bytes of track 0 sector 1, there is a string, 03 42 43 54 00. (This says: "3-byte command, B C N" followed by a "00" to terminate the command.) So CP/M on this disk automatically runs BCN at boot up. I changed the 5 bytes to 00 00 20 20 20 and the disk booted up displaying the normal 64K CP/M prompt.

■ ■ ■

LETTERS

Dear Editor,

I heard your cry for software to run on your CO-POWER-88 under CP/M-86 in your July editorial. You also mentioned the 100-volume CPMUG library that you had sitting in the corner. Have you ever thought about using an 8080 or Z80 emulator program to allow you to run that CPMUG library material in the C-P-88 during the present public domain 8086 software drought? An emulator may be worthwhile, even with its attendant increased run-time penalties, to tide you over the 8086 software gap. There are several commercial ones already out, e.g., EM-80/86 by Dynamic Microprocessor Associates, 545 5th Ave., NY, NY 10017, (212) 687-7115, which lists for $95.

As you already noted, most of the 8086 software that has been or is being written is for the IBM PC and so-called compatible look-alikes and is not in the public domain. It will be interesting to see how many "mad 8086 programmers" come out of the woodworks and respond to your plea for input to your new library.

Sy Lieberman
4706 Tyrone Ave.
Sherman Oaks, CA 91423

Editor's note:
They're coming out! They're coming out!

Dear Editor,

Hack placidly amidst the noisy printers and remember what prizes there may be in Science. As fast as possible, get a good terminal on a good system. Enter your data clearly, but always encrypt your results. Listen to others, even the dull and ignorant, for they may be your customers. Avoid loud and aggressive persons, for they are sales reps. If you compare your outputs with those of others, you may be surprised, for always there will be greater and lesser numbers than you have crunched. Keep others interested in your career and try not to fumble; it can be a real hassle and could change your fortunes in time.

Exercise system control in your experiments, for the world is full of bugs. But let this not blind you to what virtue there is; many persons strive for linearity and everywhere papers are full of approximations. Strive for proportionality. Especially, do not faint when it occurs. Neither be cynical about results, for in the face of all data analysis, it is sure to be noticed. Take with a grain of salt the anomalous data points. Gracefully pass them on to the youth at the next desk. Nurture some mutual funds to shield you in times of sudden layoffs. But do not distress yourself with imaginings—the real bugs are enough to screw you badly. Murphy's Law runs the Universe—and whether or not it is clear to you, no doubt

0 B*n ds = O

Therefore, grab for a piece of the pie with whatever proposals you can conceive. With all the crashed disks, skewed data, and broken line printers, you can still have a beautiful secretary. Be linear. Strive to stay employed.

Submitted by Text Writter
(Unemployed technical writer)

Dear Editor,

I bought a Co-Power-88 from SWP Inc., and I have some good news and some bad news.

The good news is that the board runs well and looks like a nice piece of work. The price is no bargain but it's not exorbitant either.

The bad news is the documentation, what there is of it.

The instructions included a page and a half of description of the floppy contents and enough information to get the system booted. No schematics. No CPM-86 manuals.

I called SWP and talked at length. I also went to CPM 83 and talked to the reps. Everyone was polite and pleasant, but firm. No manuals. Meanwhile, SWP says eventually they will send MSDOS manuals because they cost less than CPM-86.

Jim Martin
98 Foster St
Brighton MA 02135

Editor's note:
I called Fred Helms at SWP (I tried talking to Russell Smith too but Fred is more coherent). Anyway, it sounds like you got one of the early boards. SWP is now shipping manuals with their CPM 86. They used to pay $100 per copy for CPM 86 without manuals. Now they get CPM 86 with manuals for under $60 by purchasing the IBM PC version. They tried dealing directly with Digital Research without any luck. (That is the story I'm hearing from all over.) Anyway, Digital Research wouldn't even negotiate with Kaypro for CPM 86. (I suppose they were waiting for some high volume customer to come along.) I don't know what Digital Research is thinking but they shouldn't complain about being hung if insist on dragging around a rope. (An obvious gag.) Get back in touch with Fred about the manuals.

I also got some good news on prices for the Co-Power. They are now charging $699.95 for the 256K board with MSDOS plus either CPM 86 or MSBASIC. It's $550 for the 128K version with the same software. These prices were effective 10/17/83.

Kaypro has also announced the Kaypro II (called the II-88) with a 256K Co-Power, MSDOS, and MSBASIC for $1995. It will include the RAM disk and be IBM disk compatible. The Kaypro 4-88 will be $2295.

Dear Editor,

In response to the letter in the August issue from John S. Allen regarding saving a text file after a system problem, there are three requirements which must be met. The editor must not encode the text in memory (making it impossible to find), the entire document must fit in the available memory space, and the entire document must have been loaded into memory (if doing a re-edit).

Using Wordstar as an example, do the following: Reset the system to be certain that all of the monitor is intact. Use the Dnnnn (dump address hexadecimal) command within the monitor to display the contents of memory and locate the document. Wordstar's text buffer begins around 07800H. Write down the exact address of the last character of the document. Use the Cnnnn xxxx 0100 (copy start-address end-address destination-address) command to move the characters down to address 100H.

This will position the document so that CP/M will be able to SAVE it. Subtract nnnn from xxxx (in the last step) and divide by 0100H to obtain the number (yyy) of 256 byte pages in the file. Boot the system and type the following line after the CP/M prompt: SAVE yyy filename.typ. Replace yyy with the calculated number of pages, and replace filename and typ with the actual (unambiguous) file name and file type wanted.

John G. Ruff
5636 Rebecca Lane
Minnetonka, MN 55343

Dear Editor,

The 5MHz conversion works great and is really an improvement. I did some benchmark work on the conversion and found some interesting things. When I ran it on another machine that used your Pro-Monitor ROM, it ran faster than my copy of Kaypro's rev C monitor chip.

To investigate this further, I took the liberty of disassembling your monitor and found that you did not include the new section of code in the WRITE routine. That new section is the only difference between the Tinker Kit monitor and the rev C monitor.

Do you think this will have any effect on validity or security of data? I know yours runs faster. Apparently the new section must write out the 512 byte sector every time a 128 byte sector is updated, regardless of any intervening operations.

Sven Pretorious
1302 N. 14th St.
Coeur d'Alene, ID 83814

Editor's note:

See the Kaypro Column in this issue for more information.

Dear Editor,

I ordered user disk #10, and when it came as fast as it did I thought "Gee that was quick, doesn't anyone like #10?" Then I got cocky and thought that maybe the reason I got it so quickly was that it was still blank.

After dinner I sat down at the BB with its new dual density upgrade and prepared for a nice evening exploring new software.

NO FILE

Certainly you jest, knave processor. I know, let's boot up the single density system.

NO FILE

What! How about reading the directory tracks using the monitor. Let's see, R1,2,1.

There is something about seeing a screenful of E5's that makes a grown man want to sit on his system. Well, by now you should have gotten my drift. The disk enclosed is eagerly awaiting to be filled, and my drives spin in anticipation (got to invest in some of those AC control units).

By the way, the problem I had with the BB going off into deep space has gone away since I disconnected a rather long keyboard cable. One must learn not to attach an antenna onto one's home computer.

Good bye for now, and may all your bugs be 6-legged.

Joseph Ayala
715 Linden St
Rochester, NY 14620

Editor's note:

I looked up the order. You asked for user disk #10, which is what we sent. Now if you had wanted the FILLED user disk #10 . . .

Dear Editor,

Micro C. is so interesting that I and my students are glad it comes out only once a month and is relatively short. Some thoughts/remarks I had on issue #12:

Contrary to what Uwe Pitz said in the April issue, p. 33, there is no noticeable jitter down under 240 v, 50 hz—at least on mine. I am using the Kaypro's internal transformer, having shifted the lead to the 240v position.

I have found a way to format long footnotes with Perfect Writer. FOOTNOTE (by Pro/Tem, 814 Tolman Drive, Stanford, CA 94305) can be used with PW by using FOOTNOTE's option to alter the fn call character. Then replace the WORDSTAR commands generated by FOOTNOTE with PW commands.

Michael Stocker
La Trobe University
Bundoora, Victoria
Australia 3083

Editor's note:

If you were glad that Micro C came out only once a month, then you'll be absolutely tickled to know that we really only come out once every two months. Thanks for the information.

Dear Editor,

I've just started subscribing to Micro Cornucopia, but already I'm crying for help. When printing with Perfect writer or Perfect Filer and printing 2 or more pages, sometimes I lose parts of a line of print. Also, when I'm printing multiple copies of a document I'll get 20 or 30 extra form feeds in a row, wasting 20 or 30 pieces of paper. Then it will resume printing but the page positioning will be wrong for subsequent copies.

I don't know whether these problems are the fault of the printer, the computer, of the software. The problems are very intermittent.

Daniel Wiener
4250 Yukon Ave
Simi Valley, CA 93063

Editor's note:

One version of perfect writer that I received directly from the Perfect people (version 1.033) did strange things on my Epson printer when I asked for multiple copies. It double printed on some lines and left out others. The page positioning got messed up too. I haven't tried the perfect filer.

However, that same computer and printer combination has worked flawlessly with dozens of other text editors and filing packages.

Dear Editor,

Thank you for a great time at the SOG. Has the Thompson household returned to normal? It takes a lot of courage to invite over 100 deranged souls into your home.

Brian N. Kibler
Huntington Data Systems
307 6th Street
Huntington Beach, CA 92648

Editor's note:

Yes, the household has returned to about as normal as it gets around here. The trick is that we invited deranged souls instead of deranged heels.

WANT ADS

The following folks are reaching you for only 20 cents per word. If you wuld like to reach the same audience, send your words and 20 cents for each to Micro Cornucopia.

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.

Wanted: Experienced hardware type in San Diego or LA area to make the 5MHz mod to my Kaypro 4 in exchange for software or $$. Bruce (619) 224-1177.

For Sale, two SASI 5¼" hard disk controllers. New, tested. Each handles two drives. Xebec/ST506 compatable. OEM Documentation. $150 each. Also BB I, assembled, runs great. Wired for SIO, Video IC's included, with P.S., $425. O.B.O. 215-384-9439.

For Sale, one used, one new 256K Co-Power-88 with CP/M-86, $550 and $625. One used Xerox 820 motherboard $100. David C Worth, Suite 150 - 10451 Shellbridge Way, Rochmond, B.C. V6X 2W8. 604-270-4813.

The Tampa Bay Bandit Board RCP/M is on-line 24 hours with 1200/300 baud and 8MB of files for download. SIG/M and CP/MUG files as well as specific Kaypro programs available. Membership is $15/yr which includes unlimited use of the RCP/M, a 50 per cent discount on library copies ($5 per disk), and special group discount purchases. The current catalog has over 1300 files on 70 volumes. Call (813) 937-3608 data or (813) 937-7249 voice for full details on how to join. The Tampa Bay Kaypro Users Group, 14 Cypress Drive, Palm Harbor, Fl 33563.

Educators: Would like to locate and correspond with instructors using the Bigboard, Kaypro, Xerox 820, etc., in Electronic Technician Training Programs. Terry Owen, Associate Professor Electronics, Central Oregon Community College, N.W. College Way, Bend, Or 97701. (503) 382-6112 Ext 359.

Big Board II system. Includes BB II built by Bill Siegmund, 8" DSDD Shugart floppy, 10Mb Tandon Winchester, Zenith monitor, Keytronics keyboard and enclosure, Prowriter printer, Boschert 1033 power supply. Includes all cables, cabinet and CP/M. $3700. The perfect system for a small business. Liberty Drug, PO Box 308, Liberty, NC 27298 (919) 622-4241.

Hard Goods and Soft Goods

Your Fortune in the Microcomputer Business ..... $26.45 (US, Can, Mex)
..... $36.45 (Other foreign)
This is the best, most complete collection of "working for yourself" information I've found (and I've heard nothing but good comments from those who 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.

Micro Cornucopia
P.O. Box 223
Bend, OR 97709
(503) 382-8048

MasterCard VISA

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

BB I, BB II, and XEROX 820 USERS DISKS

The following are full 8" disks of software. Each program has a .DOC (documentation) file and many come with source.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

USERS DISK #21 - Winchester Utilities NEW!
BACKUP: Helps you back-up the winchester onto multiple floppies. Creates a catalog of the files on each disk and includes the date of the latest backup. Will not back-up an unchanged file more than once. Plus many more super features.
FLOPCOPY: Lets you make floppy copies (with only one floppy drive) by using the winchester as a buffer.
BIGBURST: Backs up a very large winchester file onto multiple floppies. Joins the copies to recreate the original file.
MULTCOPY: Use this like PIP but it prompts you to change disks. Accepts ambiguous file names.
MDIR: Displays files in all user areas on selected drive. Many features.
MAKE, MOVE: PIP-like utilities that make it easy to move files between user areas.
SWEEP: The famous disk cleanup and transfer routine that does just about everything you can do with TYPE, ERA, DIR, and PIP.
UNSQ: This is the latest, greatest file unsqueezer. Enter UNSQ *.* and it will check every file on the disk. All squeezed files will be unsqueezed.

8" Users Disks

$15.00 each (US, Can, Mex) ..... **$20.00 each** (other foreign)

OTHER GOODIES

Screen Editor in Small C ..... $39.00 (US, Can, Mex)
..... $44.00 (Other foreign)
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?

More ROMS: Fast monitor ROMs for speed freaks and our famous 'better than Texas' character ROM (V2.3) for screen freaks.
Fast Monitor ROM BB1 ..... $29.95
Deluxe Character ROM BB1 or KayPro $29.95

From Micro Cornucopia

CP/M-86 DISKS

NEW!

DISK 86-1 — Disk Utilities

D.CMD/A86, SD.CMD/A86, XDIR.CMD/A86: Three extended directory programs. Each does it differently, so we included all three.
FILE-EXT.CMD/A86: Disk status program with good display format.
PAGE. CMD/A86: A text paging program. Displays 24 lines at a time.
PRINT.CMD/A86: File printing routine. Puts a header at the top of each page along with page number and file name.
MUCHTEXT.CMD/A86: Counts words and lines in a text file.
ERQ.CMD/A86: Selective file erase program. Displays all selected files and then asks you one at a time for a Y/N.
INUSE.CMD/A86: Prints "In Use" on your terminal and asks for a password. It will not release the console until you enter the password.
FINDBAD.CMD/A867. Finds and collects bad sectors on a disk. If there are no bad sectors, information on the disk is unaltered.

NEW!

Disk 86-2 — DU and Modem Programs

DU-V75.CMD/A86/DOC: This is the popular disk utility from CP/M 80. It lets you read, write, and modify disk sectors.
MODEM4.CMD/A86: This is a modem program set up for the Slicer. This program includes a built-in help file.
MODEM7SL.CMD/A86/DOC: No modem disk would be complete without this standard. This is modem7 set up for the Slicer. It displays a menu when it is called.

NEW!

Disk 86-3 — Small C

C86.CMD: This is the original Small C compiler which appeared in Dr Dobbs Journal in 1980. It runs under CPM-86 and generates 8086 source for the ASM86 assembler.
C86. COM: This is the C86 compiler which runs under CPM-80. This 8080 program produces 8086 assembly language.
C86LIB.A86: This is the C86 I/O library.
SMALLC86.DOC: Documentation on Small C.
C?????.C: Source of the C86 compiler.

Plus, there are a number of demonstration files and ENTAB (insert tabs in place of spaces) and DETAB (replace tabs with spaces) programs all written in Small C.

KayPro Schematic

This is a complete schematic of the KayPro, logically laid out on a single 24" x 36" sheet — no more searching to see where a signal goes or comes from. Even the unused gates are shown.

It's drawn in positive logic, lines are labeled, and we've tossed in hours and hours of careful checking for accuracy. Then we added a **Theory of Operation** that's keyed to the schematic.

KayPro Schematic Package $20.00

8" CP/M-86 Disk

$15.00 each (US, Can, Mex) ..... **$20.00 each** (other foreign)

KAYPRO II USERS DISKS

The following are full disks of software assembled specifically for the KayPro II. Each program has a .DOC (documentation) file and many come with source.

KayPro Disk K1 - Modem software

This disk is absolutely priceless if you will be using a modem to communicate with bulletin boards, other micros or mainframes.
MODEMPAT. COM: Menu selection of baud rate, bits/character, stop bits, & parity for serial port.
MODEM7.COM: Very popular MODEM 7 configured for KayPro.
MODEM7+.COM: This is MODEM7 & MODEMPAT combined - you can communicate with anything!
KMDM795.COM: Super-version of MODEM7 set up for KayPro.
TERM.MAC: Commented disassembly of the TERM program you get with your KayPro so you can configure it for any interface.
SQ/USQ.COM: Programs to squeeze and unsqueeze files for faster transfer.

KayPro Disk K2 - Utilities

Really oodles of spiffy little (and big) programs to help you get full use of your KayPro.
ZESOURCE.COM: A true Zilog format disassembler for 8080 and Z80 object(.COM) files. Now you can turn .COM files into . MAC files.
UNERA.COM: Simply enter "UNERA" followed by the name of the file you just erased and presto, the erased file is back! A lifesaver.
FINDBD54.COM: Checks an entire disk, reports bad sectors, and then creates a special file containing those sectors. You save a bundle on disks.
CAT2: This a group of programs which create and maintain a single directory of all the programs you have on all your disks. Even keeps track of which programs are backed up and which aren't.
UNSPOOL.COM: Use your KayPro II and print files at the same time. Doesn't slow down system response!
DUMPX, DU-77, COMPARE, SUPERSUB, FORMFEED, DIR-DUMP, . . . and all have documentation on disk.

KayPro Disk K3 - Games

PACMAN.COM: Despite the KayPro's lack of graphics, this one looks and plays amazingly like the real thing! Keep it hidden.
ZCHESS.COM: Chess with a 1-6 level look ahead.
OTHELLO.COM: You learn it in minutes, master it in years.
BIO.COM: Generates custom graphic biorhythm charts.
MM.COM: Master Mind.
WUMPUS.COM: Classic wumpus hunter's game.

KayPro Disk K4 - Adventure

This disk contains one 191K game, Adventure.
ADV.COM: This is the latest, greatest, most cussed adventure ever devised by half-mortals. This is the 550-point version so the cave is greatly expanded and the creatures are much smarter.

KayPro Disk K5 - MX-80 Graphics

A complete MX-80 graphics package including example files.

KayPro Disk K6
Word Processing Utilities

A powerful line oriented text editor that looks like Unix's EX, plus a scad of text utilities written in C which handles pretty printing, shortening a file, multiple space output, add tabs, remove trailing whitespace, and more.

5" KayPro Disks

$12.00 each (US, Can, Mex) ..... **$16.00 each** (other foreign)

KayPro Disk K7
Small C Version 2 Compiler

This is a greatly extended version of Ron Cain's original C compiler. Version 2 includes many more expressions, a substantially extended library, and much more. This disk contains the compiler, documentation, and library.

KayPro Disk K8 - Small C Version 2 Source

More of Small C Version 2. This disk contains the compiler, documentation, and the source of Small C version 2. It compiles itself.

KayPro Disk K9 - ZCPR

ZCPR: The big news on this disk is the self-installing version ZCPR available only from Micro C. Once you have ZCPR in your CP/M, you'll never go back to straight CP/M! For instance, ZCPR searches drive A for any program not found on drive B, so, even an empty disk in drive B appears to contain every program on A. It's great for text editors, compilers, etc. Plus many more new features to make CP/M easier to live with. In fact, Digital Research incorporated many features of ZCPR into CP/M 3.0.
PASWORD: Lets you encrypt and decrypt your precious files. Includes source.
EX14: a super replacement for SUBMIT and XSUB. Plus many more: **TREK, FIX, FIND, SNOOPY ALIENS and DIF2.**

NEW!

KayPro Disk K10 - Assemblers

We've received a lot of requests for a Z80 assembler. So Dana put in some long hours getting the Crowe Z80 assembler to run on the KayPro (and every other Z80 machine).
CROWECPM: This is a first class Z80 assembler. We use this assembler daily (and we included its source). Takes standard Zilog mnemonics.
LASM: This is a more powerful version of the ASM assembler you received with the KayPro. This will link multiple programs together at assembly time.
PRINTPRN: This program makes it easy to print the listing files generated by the Crowe assembler.

NEW!

KayPro Disk K11
Library & Checkbook Programs

We've had excellent response to both these programs from Big Boarders and numerous requests from KayPro folks.
CHECKS: This has been a very popular group of programs. Categorizes checks so you can keep track which are tax deductible and which get charged to which projects. Includes source and excellent example check files. Very powerful.
LIBR: This is a complete set of library routines which let you group files into a single file called a library. Then CP/M sees them as a single program, but with the library routines, you can list them out separately, run them separately, or divide them up again. Almost like a unix environment.
DISPLAY, VLIST, PGLST: Additional screen and print utilities.

NEW!

KayPro Disk K12 - FORTH

Yep, this is FORTH, one of the most unique, most extendable languages know, and for a paltry $12.00. This disk contains not just one FORTH, but two, along with an editor, decompiler and 8080 assembler! The editor even uses the cursor control keys.
FORTH: This is true fig-FORTH.
KFORTH: A very nicely extended version of fig-FORTH.
PLUS, all the rest of the FORTH goodies. (Forth Heaven!)

NEW!

KayPro Disk K13
Source of fig-FORTH

All this disk contains is the 40K ASM source of fig-FORTH with the hooks in place for the KayPro. This disk is for FORTH hackers who just can't leave anything alone. (Look, you probably have faults, too.) The source of FORTH is here because there isn't room on K12. This is the only disk that isn't stuffed.

KayPro Disk K14 - Smartmodem Programs

This is the disk for you if you have a Smartmodem compatible modem. With this disk you can communicate with anything but a recalcitrant spouse. Handles goodies like autodial along with multiple directories.
SMODEMK: Smartmodem program set up for the KayPro (and source).
XMODEM: Lets you remotely control your KayPro from a distant computer.
KAYTERM: This is the information you need to run or write modem software on the KayPro.

NEW!

KayPro Disk K15

BACKUP: Helps you back-up KayPro 10 winchester onto multiple floppies. Creates a catalog of the files on each disk and includes the date of the latest backup. Will not back-up an unchanged file more than once. Plus many more super features.
FLOPCOPY: Lets you make floppy copies (with only one floppy drive) by using the winchester as a buffer.
BIGBURST: Backs up a very large winchester file onto multiple floppies. Joins the copies to recreate the original file.
MULTCOPY: Use this like PIP but it prompts you to change disks. Accepts ambiguous file names.
MDIR: Displays files in all user areas on selected drive. Many features.
MAKE, MOVE: Pip-like utilities that make it easy to move files between user areas.
SWEEP: The famous disk cleanup and transfer routine that does just about everything you can do with TYPE, ERA, DIR, and PIP.
UNSQ: This is the latest, greatest file unsqueezer. Enter UNSQ *.* and it will check every file on the disk. All squeezed files will be unsqueezed.

REMEMBER

FREE Users Disks in exchange for submitted software or articles

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

Packet Radio, Operating Details

By Peter Eaton WB9FLW

35 Norspur, Rt 4
Edwardsville, Il 62025

In this segment of the series, I'll describe how you'd use the Tucson Amateur Packet Radio (TAPR) Terminal Node Controller, referred to hereafter as the controller.

This controller operates in three modes: command, converse, and transparent.

Command mode

Command mode lets me modify system parameters such as my callsign, or which packets I will monitor (just the ones addressed to or from a specific station, or all packets), and command mode lets me select whether my station will operate as a digipeater. (A digipeater repeats packets addressed to another station but containing my callsign as a repeater address.)

It's interesting to note that the default is digipeat "on." It's so automatic that unless I hear my system go on or off, I'm not aware that it is being used to relay information between two other stations.

Many of the alterable parameters are stored in an EAROM (electrically erasable read only memory). Because I can store my operating parameters in the EAROM, restarting the system is very easy.

Example First Time Operation

MYCALL WB9FLW—This sets my callsign into the address field (only necessary once).

PERM—Save my my callsign into the EAROM.

CONVERS—Enter Converse Mode.

HELLO TEST—Sends an unaddressed packet containing the text "HELLO TEST."

CNTL-C—Returns to command mode.

CONNECT K7OMT—Requests that I be connected to K7OMT. The system automatically enters converse mode if K7OMT acknowledges my connect request.

HELLO DAVE—Sends an addressed packet to K7OMT with the text "HELLO DAVE."

CNTL-C—Returns to command mode.

DISCONNECT—Disconnect from K7OMT.

As you can see, operating the controller is really easy. In fact, you can power-up and be on the air in a matter of seconds.

More Command Mode Features

I can also define a number of operating parameters for the communication. I can define the length of a packet (default is 128 characters) and I can define the character that causes the terminal to send a packet (default is a carriage return). I can tell the terminal to automatically send a packet whenever the keyboard (or the channel) has been inactive for so many seconds. I can also have the terminal send a beacon (identification) every so often.

These are just a few of the features available in command mode. The software is extremely well done, which makes using the controller a pleasure.

Converse Mode

Converse mode is similar to terminal mode in a standard modem program, especially if you simply enter the "CONVERSE" command.

In this case, the data packets will be unaddressed and can be copied only by stations which are monitoring all of the packets on the channel. I use this mode for sending CQ (i.e. asking if anyone wants to talk to me). I have created a CQ file on my Kaypro 4 which contains my name, address, callsign, station equipment, phase of the moon, and more. To say CQ, I simply send this packet. If someone else is around, he'll usually reply and our QSO (discussion) will begin.

The controller will automatically enter converse mode whenever another station sends a connection request and my callsign. If K7OMT entered CONNECT TO WB9FLW, I would see the message:

Connected to K7OMT

Now, everything I type will be sent to K7OMT.

I can also send packets via a digipeater by entering:

CONNECT WD0ETZ VIA WA0KGU

WA0KGU is one of several local packet repeaters that's available 24 hours a day. My station sends its packets to WA0KGU which verifies that they are valid and then retransmits them to WD0ETZ. Neat!

Transparent Mode

Now that we've covered the command and converse modes pretty thoroughly, let's look at transparent mode.

This mode turns the controller into a dumb modem. The controller will not locally echo any characters and it will not look for control characters. This mode is used for transferring binary files. With the error detection and correction built in to the X.25 protocol, you can count on error free transmissions.

Transparent mode also lets you remotely operate someone else's station. I can access WD0ETZ's Big Board from my station and run programs, look at directories, and so forth just as though I were sitting at his keyboard. The only difference is the time delay required for transmission of the packets at 1200 baud.

You can exit from the transparent mode by entering a special sequence of characters. The default is three cntl-c's in a row, preceded and followed by a quiet period lasting at least 1 second.

The Software

The controller's code was written in Pascal and 6809 assembly language by Harold Price NK6K, Dave Henderson KD4NL, and Margaret Morrison KV7D. Most of the code is in Pascal with interrupt, data buffering, and device driver portions in assembly language. The over-20K of object code is stored in the controller's ROMs.

In coming issues, we'll cover some of the activities going on around the U.S.

■ ■ ■

(Editorial continued)

Wanted, One Good Person

It is with great trepidation that I announce a job opening here at Micro C. I'm looking for someone who would like to live in Bend and work with me on the editorial end of the magazine.

I don't really know how to describe the job, because it would depend a lot on the individual. However, if you'll bear with me, I'll take a stab at it.

Right now, all the responsibility for filling the magazine rests on one pair of shoulders, mine. As the magazine has gotten larger, and the phone more insistent, the magazine has gotten farther and farther behind (and the shoulders are looking more stooped).

I need a person who is comfortable with this strange language, who is terribly organized, is a little bit crazy, doesn't smoke, is comfortable editing for grammar and style, and can take projects and run with them enthusiastically.

Some important extras would include: a feel for words (i.e. someone who can take the plod of technical jargon and make it dance lightly), and a sense of humor (kinda' goes without saying, I suppose).

Work Environment

Working here is like being in the middle of a giant whirlpool of information: it's absolutely the opposite of the classic engineering life where you disappear into your bench for a year or two before surfacing to see if the rest of the world is still there. Here, it feels like the world is watching everything you do. When you do something well, a lot of people will know, and they'll tell you. When you blow it, you'll hear about that too (usually the feedback is much quicker when you mess up).

Second Thoughts

I'm a little worried about announcing this job for a couple of reasons.

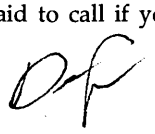
First, in a small business like this, employees are, for all practical purposes, family members. Our present group of five fits together beautifully, and though we're all looking forward to finding someone to ease my load, we will also be careful in our selection of another group member. (You won't find a much funnier, closer, harder working, more excited, and, some days, more exhausted group than this.) So, you can't be just anyone, and it's probably not going to be easy to be sure that you are the someone special who will really fit in and contribute.

Second, Sandy and I chose Bend because it's an absolutely incredible place to live and raise our family. However, there are not a lot of other technical positions here. If the job doesn't work out, you'd probably have to move again. So, we're going to be very cautious about our selection, and you need to understand the risks.

However

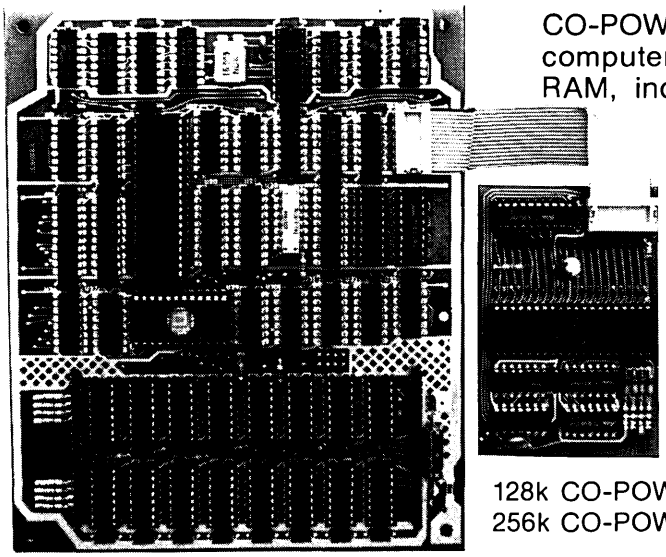
If you spend half your waking hours writing Pascal and the other half writing poetry—if you think you'd be absolutely right for the position and you don't mind stepping off a cliff to try it, then send in your resume (let us know if you want it returned).

Be sure to include long-term goals (plus salary requirements), send along a hand-written note (at least one full page) on standard unlined typing paper and, finally, send a sample article or paper you've done. You might even include a picture of yourself (and family). (Don't be afraid to call if you have any questions.)



David Thompson
Editor & Publisher

SWP's CO-POWER-88 makes Z80, CP/M microcomputers IBM-PC compatible!



CO-POWER-88 is a 16-bit 8088 coprocessor for Z80 CP/M computers. Both versions of CO-POWER-88, 128k and 256k RAM, include both CP/M-86 and MSDOS, complete with IBM-PC compatibility. A facility for transferring data files between CP/M and MSDOS is included.

Simple commands move system control between the Z80 and 8088 processors. CO-POWER-88's RAM can be used in CP/M as a RAM drive! Currently available for Kaypro, Bigboard, Zorba, Xerox, Access Matrix, Televideo, and ATR8000 computers.

128k CO-POWER-88 w/ CP/M-86 and MSDOS $550.00
256k CO-POWER-86 w/ CP/M-88 and MSDOS $699.95

ATR8000: SWP's $499.95 CP/M Computer

SWP's ATR8000 is a 64k RAM, Z80A, 4 MHz computer that includes double density CP/M 2.2. The ATR-8000 runs up to four disk drives that are any mixture of size (5¼" and 8"), type (single-sided and double-sided), and density (single, double and quad). The ATR8000 has an RS-232 port for a modem or serial printer and includes software for both. There's also a parallel port with a parallel printer driver. The ATR8000 inferfaces to an RS-232 terminal or to an ATARI home computer. Software includes a program that allows the ATR8000 to use CP/M disks from other comptuers. The ATR8000 can be upgraded to also run CP/M-86 and MSDOS by adding CO-POWER-88.

Bigboard Dual Density

We've just released a new version!

Hardware

- **A daughter board that plugs into the 1771 socket. With this board the system employs automatic density select.**
- **Instructions tell how to run 5¼" drives. A 50-34 pin disk drive adapter board is included with 5¼" disk orders.**

Software V#061983

- **One 8" version includes the code to make a 60k double density CP/M for:**
  8" SS 2.5 MHz 8" DS 2.5 MHz
  8" SS 4 MHz 8" DS 4 MHz
  5¼" SS 2.5 MHz
- **Printer drivers are built-in, selectable in the IOBYTE.**
- **Easy to change port parameters.**
- **8" SSDD disk storage is 674k; DS is twice as much! 5¼" is 183k.**
- **Includes DDINIT for SD and DD initializing and DDSYSGEN for DD sysgening.**
- **Special features have been added including a deluxe pause, screen print and clock.**
- **Source code is available for $25 after you sign a disclosure agreement.**

For orders: We accept MasterCard, Visa, money orders or checks. Shipping charges and applicable taxes will be added. Call or write for delivery time.

SWP Inc
MICROCOMPUTER PRODUCTS

2500 E. Randol Mill Rd – 125
Arlington, Texas 76011
817/469-1181

(We used to be Software Publishers, Inc.)

TECHNICAL TIPS

TV Monitor

I decided to change the Big Board video section so that I could connect my TV directly. (Through a $10 RF modulator, of course!) Anyhow, the changes were simple and maybe other poor folks like me may be interested.

First, replace R5 (normally 39k) with a 21k resistor. I just tack-soldered a 47k resistor (1/4 watt or larger is fine here) in parallel with the existing 39k resistor so that in the future when I'm rich and famous and can afford a real monitor, I can easily restore the Big Board to its original configuration.

Second, replace R3 (normally 68k) with a 31k resistor. I tacked a 56K resistor across R3 to get the 31K.

Third, remove the jumper connecting pins 10 and 11 on JB1.

These three modifications (along with some fiddling with the horizontal width, vertical width, vertical linearity, and contrast controls of the TV set) were all I needed to do. The display is quite reasonable, and though it's not as nice as a real monitor, the cartoons are better.

Don Brittain
4200 Spruce Street, Apt. 208
Philadelphia, PA 19104

Parallel Okidata on BB I PIO

If you have an Okidata Microline 82A or 83A (or one of the many identical units under another name) the following should help you interface it with your PIO.

First, your CP/M BIOS has to be set up for a printer on the PIO (channel A or B). You next add jumpers on JB-4 of your BIG BOARD as shown in Figure 1.

Figure 1 - BB I Jumpers for JB-4

```
PIO channel A

   Ready      1 - 2
   Strobe     7 - 8
   Upper Dir 15 - 16
   Lower Dir 13 - 14

PIO channel B

   Ready      5 - 6
   Strobe     3 - 4
   Upper Dir  9 - 10
   Lower Dir 11 - 12
```

Now you need to build a printer cable. You will need to get a 36-pin plug, equivalent to 57-30360 Amphenol or Daiichi Electronics, a 40-pin female dual ribbon cable connector for PIO on the BB I, and a 6-foot length of 40-wire ribbon cable.

To build the cable, first install the 40-pin connector on the ribbon cable very carefully. I used my vise since I did not have the proper tool. At the other end of ribbon cable divide the ribbon cable into two 20-wire units. Now connect the 36-pin plug for the printer as shown in Figure 2.

Figure 2 - Cable Wiring for BB I

PIO Channel A			PIO Channel B		
Big Board I - 82A			Big Board I - 82A		
Ready	2	10	Ready	22	10
Strobe	4	1	Strobe	24	1
Bit 0	6	2	Bit 0	26	2
Bit 1	8	3	Bit 1	28	3
Bit 2	10	4	Bit 2	30	4
Bit 3	12	5	Bit 3	32	5
Bit 4	14	6	Bit 4	34	6
Bit 5	16	7	Bit 5	36	7
Bit 6	18	8	Bit 6	38	8
Bit 7	20	9	Bit 7	40	9

All odd pins on BB I are grounded.

Once you have built the cable, you need to remove the printer case to set the DIP switches in the printer. The first DIP switch is on the front of the printer, turn all 8 switches to the off side. Now on the circuit board in the back of the printer is a DIP switch for serial operation of the printer, do not touch it. To the left of it are two jumper plugs, the left jumper plug may need to be on the "B" side, that is how I am running it.

I hope you have better luck than I did, I had a bad PIO and it was not sensing the strobe coming back from the printer.

William W. Barncord
2910 Furneaux Lane
Carrollton, Texas 75007

Shugart 901 Drive Interface

I recently purchased a couple of used Shugart 901 drives for $99 each. They appear to be very similar to the 801 except for the read edge connector which is 22 pin double-sided with 0.156" spacing instead of the 50 pin, 0.1" spacing. See Figure 3.

Someone had removed R13 from my drives. It should be a 150 ohm pull-up.

There is also no gating to activate the head load from the drive select signal. The head load is always active which means that with two drives ready, the heads will load together. To overcome this problem, I cut pin 1 of IC 3F and connected it to pin 1 of IC 3E.

Steve Hawley
757 Dyer Ave
Cranston, RI 02920

Figure 3 - Shugart 901 Interface

Signal	901	801
INDEX*	5	20
STEP	6	36
WRITE GATE	7	40
READY*	8	22
WRITE DATA	10	38
TRACK 0*	12	42
DRIVE SEL 0,1	13	26,28...
HEAD LOAD	18	18
DIRECTION	15	34
READ DATA	16	46
WRITE PROTECT	21	44

* Require pull-ups

The DC power can be connected to the 901's edge connector as follows:

+5V	11 and M
GND	1 and A
-5V	20 and X
+24V	2 and B
24V ret	3 and C

Jumpers:

Jumpered:	A	B	D	E	DS	P	M	X	U1
Open:	C	N	R	T	S	Y	U2		

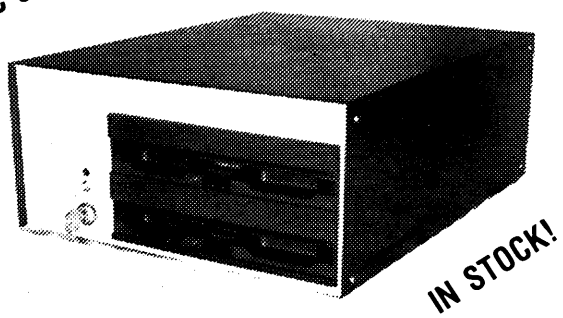
Jumpered on last drive only:

Pull-ups: F G H J K

FOR THE **BIG BOARD** and other SBC's

The "LA CAJA" Enclosure w/Power Supply

$279 FOR ONE DRIVE **$299** FOR TWO DRIVE



IN STOCK!

SINGLE DRIVE ENCLOSURE SHOWN

STANDARD FEATURES INCLUDE

- Power Supply
  5V @ 4A + OVP, 24V @ 2.5A, ± 12V @ .2A
  All voltages are current protected
- Reset Switch
- Fan
- Key-lock Power
- Solid State AC Relay
- Bell Circuit and Piezo Speaker
- 2 AC Outlets, One Switched
- Color, Beige and Chocolate
- 6" H x 12" W x 16" D 24"W for 2 Drive Unit

"La Caja" can be supplied fully wired for the Big Board with all ports disk DC and timed AC for $150.00

CALIF. RES. ADD 6% TAX

AstroTronics MICROSYSTEMS (714) 734-6006 1137 TOPAZ ST. CORONA, CA 91720

POWER SUPPLY

For the BIG BOARD and other SBC's
Assembled, tested and burned-in 48 hours.

- +5V @ 4A+OVP @ 2.5A, ±12V @ .2A
- All supplies have over-current protection
- Bell circuit and Piezo speaker for BB
- Low line warning circuit - brown out
- 4.5"H, 3.8"D, 11"L - 5 lb.

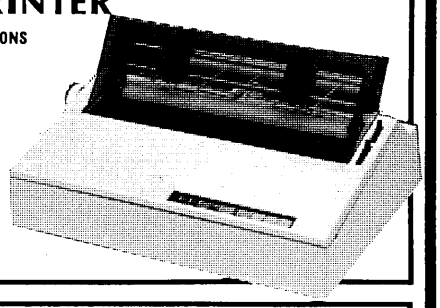
$105

IN STOCK! PLUS $5 SHIPPING & HANDLING

IN STOCK! **brother HR-15**

DAISY WHEEL PRINTER

- CASSETTE TYPE PRINT WHEELS AND RIBBONS
- RS232-C OR CENTRONICS PARALLEL
- BI-DIRECTIONAL PLATEN AND HEAD
- PROPORTIONAL (MICRO ADJUSTMENT)
- GRAPHIC PRINTING
- BOLD PRINTING
- 2K BUFFER
- INSTALLS LIKE DIABLO 620/630



$588 PARALLEL PLUS $12 SHIPPING

AMP CONNECTORS FOR YOUR DISK DRIVE

KIT OF 2 EACH WITH PINS

$5 Plus $1 Shipping & Handling

DOUBLE DENSITY HARDWARE

Cheep!

$19 PCB with DRAWINGS

DAUGHTER BOARD FOR THE WD2797 DSDD CONTROLLER

$79 KIT with DRAWINGS

CALIF. RES. ADD 6% TAX

AstroTronics MICROSYSTEMS (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

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

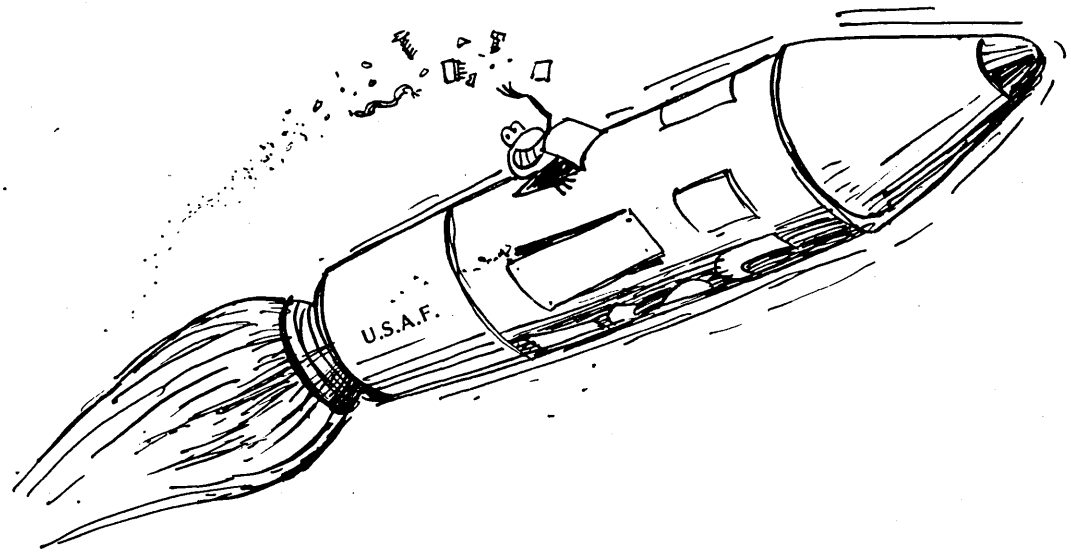
ISSUE NO. 11 (4/83)
BB I Expansions
BB II Details
Dyna, RAM Disk Review
Easier Reverse Video Cursor
PlannerCalc Review
KayPro Column begins
36 pages

ISSUE NO. 12 (6/83)
256K for BB I
Bringing Up BB II
dBase II
Look at WordStar
Double Sided Drives for BB I
Packet Radio
40 pages

ISSUE NO. 13 (8/83)
CP/M Disk Directory
More 256K for BB I
Mini Front Panel
Cheap Fast Modem
Nevada Cobol Review
BB I Printer Interface
44 pages

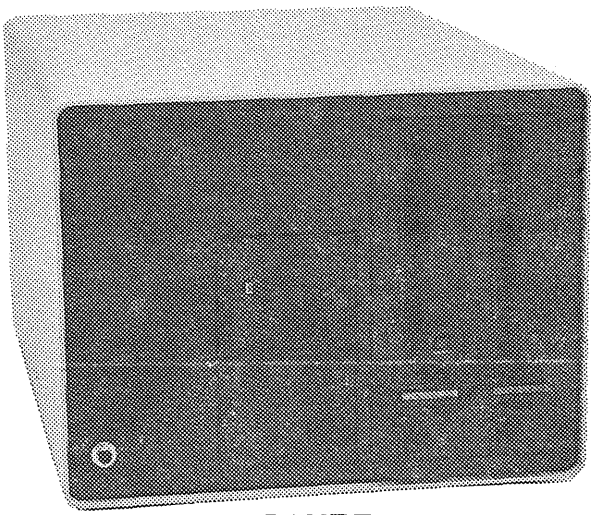
ISSUE NO. 14 (10/83)
BB II Installation
The Perfect Terminal
Interface to Electronic Typewriter
BB I Video Size
Video Jitter Fix
Slicer Column starts
48 pages

BUGSBURY



FERGUSON ENGINEERING

817-640-0207



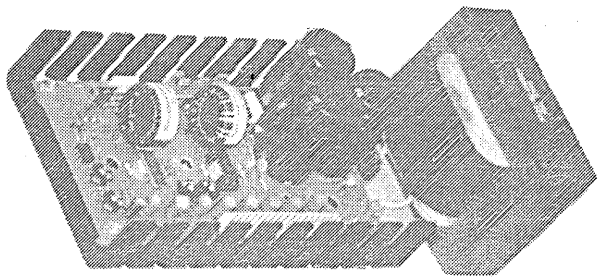
COMPUTER ENCLOSURE CE - I $195.00

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.

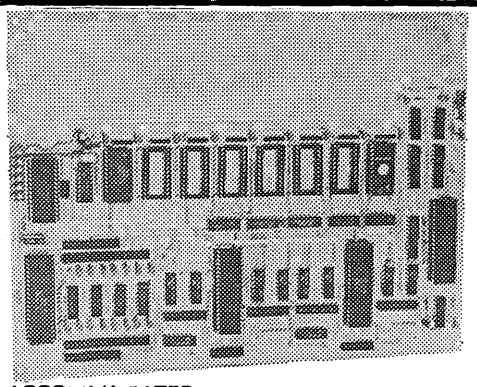
Complete 820 Documentation Package	$15.00
820 Video Cable-10 Pin Header to CRT Edge Card-1'	$ 5.00
820 CBIOS & Rom Monitor Source	$35.00
CP/M Primer (Xerox CP/M Handbook)	$10.00

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.