THE ADDRESS BUS

PORT TO PORTAL.....Below
Extending Zenith's CP/M BIOS Beyond the
Three Soft-Sector Drive Limit.....1, 3
by Kirk L Thompson
SOFTWARE LIST.....2
VENDOR.UPDATE.....8
Troubleshooting the '89, Pt. 2:
The Keyboard Module (Concluded).....8
by Dan Jerome, with Kirk L Thompson
R/O MEDIA.....11
MISCELLANY.....12

PORT TO PORTAL -- Editorial

This issue will be something of a change of pace. Mainly, that's due to my "usurping" more than my usual number of pages to report on the successful conclusion of a year-long project. That enterprise was to modify Zenith's CP/M BIOS to work with that fourth soft-sector drive I installed a year ago. If you're running soft-sector as well, this is a **significant** enhancement since it can increase your on-line storage by a third or more. If nothing else, it presents the opportunity, as mine was, to add a different style drive to improve compatibility with existing systems. The details are given below, including comments on related topics that you may well find of interest. If you only run hard-sector, I **still** recommend you peruse it since it contains information not covered elsewhere to my knowledge.

Despite my **large** contribution, some of the regular features are still here. (The rest of them will return with the next issue.) But you'll also find a new column. Shrinking support for our systems has been praying on my mind of late. And I think one way I can help you locate other information resources is to present them up front, so to speak. So this issue inaugurates a new column, called "R/O MEDIA". You'll find necessarily brief coverage of what's appearing in other 8-bit-oriented publications. Information is out there; the problem is **finding** it amongst disparate publications. I hope this helps you locate what you need. And if you know of relevant magazines or newsletters not listed here, by all means let me know about them.

To conclude, I'm pleased to announce that the problem with my Diablo 630 daisywheel printer has been resolved. I'll describe what happened and have some notes on maintenance of the old printer in the next issue.

Kirk L Thompson

Extending Zenith's CP/M BIOS Beyond Its Three Soft-Sector Drive Limitation

By Kirk L. Thompson

As noted in last year's first issue (#22/23, pp. 1 and 4), I installed a 96-tpi (80-track) soft-sector drive to better support those of you who **only** run that configuration. Bringing up the new drive wasn't much of a hassle under either HDOS (2.0 or 3.0x) or Magnolia Microsystems' CP/M 2.24. For instance, the HDOS soft-sector support package from HUG (p/n 885-1127[-37] and which I recommended way, way back in issue #3, p. 3) comes with a utility that can increase the number of drives supported by the device driver from three to four. And the H-37 driver that accompanies HDOS 3.0x **already** supports all four that Zenith's controller board can handle.

Magnolia's CP/M requires a little more work. You have to add **both** logical and physical drives to the system image preserved in MOVCPM. But you can do this with the SETUP.COM utility and the procedure is essentially the same as the one I described for hard-sector in issue #15 (p. 12). That's how I brought up the drive for CP/M last year.

Doing the same to Zenith's CP/M 2.2.03, though, is a **lot** more work. What's required are **alterations** to the BIOS source code. I became interested in this specific topic when I installed that fourth drive a year ago and discovered that 2.2.03 wouldn't recognize it. So I began "pestering" Pete Shkabara for information, since I'm not an assembly language programmer, much less a systems programmer! He made several **very** helpful suggestions and those serve as the basis of this BIOS mod. In the process, I also learned a few of the whys and wherefores of the BIOS and CONFIGUR, so I'll pass these along to you during my exposition. Further, I have some notes on MOVCPM and SYSGEN usage you might find of interest.

Pros, Cons, and Set-up. Fortunately, the additions that I'll discuss below are fairly simple. And you don't even lose any transient program area (TPA) because they don't change the size of the loaded BIOS module beyond what it already is. But they require a rather high-powered text editor **and** some facility with that dreaded MAKEBIOS procedure! For the editing, I used PeachText; Magic Wand, WordStar, TXTPRO, or a similar editor will do the job. Just bear in mind that BIOS.ASM is a **large** (55K) file, so the likes of PIE or plain-vanilla VDE won't cut the mustard. As for performing MAKEBIOS, my space here is somewhat limited, so I recommend you read the section in your CP/M manual devoted to it. I've discussed it myself in some detail in issue #2 and **SEBHC Journal** most recently ran an article on it in Vol. 5, #10 (May, 1991), pp. 5-6.

I should add that this mod could be made to Zenith's ver. 2.2.04 **and** Livingston Logic Lab's BIOS-80 .ASM source, too. The labels we'll be looking for and the code in the areas we'll be supplementing are **identical** to those for 2.2.03. I haven't **tested** this mod with either 2.2.04 or BIOS-80, but don't foresee any problems if you undertake them.

However, this mod has some distinct **limitations**. CONFIGUR.COM **only** recognizes three soft-sector drives. So there's no way to use it to set drive step rate and track density on the new drive. Of course, a custom utility could be written for the task or CONFIGUR even disassembled and modified. But I decided that, for my own purposes, I would **hard-code** the step rate and track density directly into the BIOS. I'll discuss the available options below. But if you **do** hard-code these and

[Continued on p. 3]

SOFTWARE LISTING

>----------<

General Software Catalog

A catalog of **Staunch** software is available. Initially prepared by Ralph Money, the disk files include listings for **both** HDOS and CP/M. If you wish to combine the catalog with other offering(s) on the same disk, the files will be "squeezed" to conserve disk space and an unsqueezer provided to recover them. This catalog now requires two (2) disks in standard hard-sector, one (1) disk in any other format. See the "Placing an Order" section, below, for information on formats.

Source Code in this Issue

If this issue includes any source code, be it BASIC, Pascal, C, assembler, or whatever, you may obtain it **at no charge**! Merely send a **formatted** disk with a **postage-prepaid** mailer and I'll tranfer it for you. Please clearly indicated the format you are supplying. See below for supported formats.

FOR CP/M ONLY

These CP/M packages have just been released by Generic Computer Products, Inc.
(My thanks to David Powers of GCPI)

CATALOG-MASTER
A Disk Volume Cataloging System
By GCPI (P.O. Box 790, Marquette, MI 49855)

CATALOG-MASTER is a utility for creating and maintaining a master file directory for multiple disk volumes. Since CATALOG-MASTER uses the I/O functions of the operating system to obtain volume directory information, any type of disk volume can be processed. The program is menu-driven for ease of operation. CATALOG-MASTER has its own file description editor, called FDEDIT. This unique editor allows you to create and maintain file description files. File descriptions provide added information about a file in addition to the filename. At your option, file descriptions can be included in master directories. CATALOG-MASTER provides master file directories with file selection options and sorting. CATALOG-MASTER also allows merging of previously-created description files into the master catalog output. Catalog output can be directed to display screen, printer, or a disk file. CATALOG-MASTER requires the following system hardware: a 80x24 display screen with direct cursor addressing, clear screen, and clear to end of line functions; 2 floppy disk drives; 48K RAM; and CP/M V2.2. Three terminal configuration files are supplied: Epson QX-10, Heath H-19, and Kaypro. However, these files may be edited with a text editor for custom use with other systems; instructions are included. If you have a PC-compatible, a comparable program is still available from GCPI for $40. This package requires 142K of disk storage.

FOOTBALL PICKS
An NFL Football Game Prediction Program
By John S. Mays

FOOTBALL PICKS allows the user to make predictions on National Football League games for office pools, resolving debates with friends and colleagues, or just plain fun. FOOTBALL PICKS makes predictions with an assist from a personal computer. The program uses information easily found in most daily newspapers, and, unlike some other prediction programs, does not require the user to predict each and every game of the entire season in order to keep the database current and the predictions viable. With FOOTBALL PICKS, the user can predict as many or as few games as he desires. The minimum requirements to use the program are at least one disk, 48K, and CP/M-80 V2.2. If you have a PC-compatible, a comparable program is still available from GCPI for $25. This package requires 126K of disk storage.*

INVESTMENT-MASTER
Lump Sum and Annuity Investment Program
By David J. Powers

INVESTMENT-MASTER is a program used for lump sum and annuity investment calculations. It uses a data entry screen form to allow for convenient input and verification of investment parameters. Your input is checked for syntax and range errors in order to minimize errors during processing. INVESTMENT-MASTER is able to solve for any unknown investment parameter. And it can handle both a deposit or withdrawal annuity. It also provides an investment summary which lists all the input and calculated parameters for the investment. Investment summaries can be output to display screen, printer, or a disk file. INVESTMENT-MASTER requires the following: a Z-80 computer using CP/M-80 V2.2 with 48K, one disk drive, and an 80x24 display terminal with direct cursor addressing. If you have a PC-compatible, a comparable program is still available from GCPI for $50. This package requires 140K of disk storage.*

LOAN-MASTER: Loan Analysis Program
By David J. Powers

LOAN-MASTER is a program used to generate amortization schedules for any type of fixed-rate loan. It uses a data entry screen form to allow for convenient input and verification of loan parameters. Your input is checked for syntax and range errors in order to minimize errors during processing. LOAN-MASTER is able to solve for any unknown loan parameter by using sophisticated iteration techniques. Zero-interest loans and "balloon" contracts can also be handled. It provides the capability to output either a periodic or annual amortization schedule. A loan summary is provided which lists all the input and calculated parameters for the loan. Amortization schedules and loan summaries can be output to display screen, printer, or a disk file. LOAN-MASTER requires the following: a Z-80 computer using CP/M-80 V2.2 with 48K RAM, one disk drive, and an 80x24 display terminal with direct cursor addressing. If you have a PC-compatible, a comparable program is still available from GCPI for $50. This package requires 156K of disk storage.*

FORMS-LIB for Microsoft BASIC or Turbo Pascal
By David J. Powers

FORMS-LIB is a set of high-level source routines which can be included with your application program to control the flow of information from your keyboard to your display screen. You can use FORMS-LIB to simplify data entry operations. The FORMS-LIB libraries require the following system software and language support: a text editor or the Forms EDitor (FED, included) to create the data entry screen forms used by the FORMS-LIB and **either** the Microsoft Basic Interpreter or Compiler **or** Borland International's Turbo Pascal compiler. The FED program is a flexible screen-oriented editor used to design data entry screen formats. FED also generates a data entry parameter file that defines the locations of data entry fields. This data entry parameter file can be used by your application programs to simplify data entry. FED, together with the appropriate high-level FORMS-LIB library routines, can greatly enhance any data entry or database program. FED requires the following: a Z-80 computer using CP/M-80 V2.2 and 48K RAM, one disk drive, and an 80x24 display terminal with direct cursor addressing. The program comes configured for the H-19/89 terminal. However, the terminal parameter file is ASCII, so may be edited for non-Heath systems. These libraries could be edited for use with other BASIC or Pascal dialects and for use under HDOS. If you have a PC-compatible, comparable libraries are still available from GCPI for $50 each. FORMS-LIB for Microsoft BASIC requires 126K of disk storage; FORMS-LIB for Turbo Pascal requires 138K of disk storage. **Note**: the BASIC and Pascal versions of this product are **separate** items.

* **Note**: A terminal installation program for these packages (TCONFIG) supports the following models: ADDS 25, ADM-20 and -31, Beehive 150, DEC VT-52/-Heath H-19, Epson QX-10, Hazeltine 1500, IBM-PC/TI-PROF, Kaypro II/4/10, Xerox 820, Osborne 1, Televideo 912/920, and Zenith Z-100. If you have another make, you will have to manually install terminal codes and data-entry functions with TCONFIG.

Placing an Order

Your cost for this software depends on what you supply:

Formatted disk(s) and self-addressed, stamped return mailer ..... $2.00 per disk
Formatted disk(s) without mailer .... $4.00 per disk
No disk(s) or mailer ..... $6.00 per disk

Disk formats available are standard (SS/SD) or double-sided (DS/SD), 48-tpi hard-sector and single- or double-sided, 48- or 96-tpi soft-sector for both HDOS and CP/M. (**Staunch** now supports 96-tpi soft-sector whether you supply a **formatted** disk or not; see the lead article in this issue.) If you do **not** have an H/Z system available, **Staunch** also supports **many** CP/M-80 formats, but **only** in 48-tpi (40-track), such as Osborne, Kaypro, and Xerox, as well as PC-XT; let me know what your requirements are. Please **clearly** indicate the format you are supplying or require. If you desire DS hard- or any soft-sector format, I will pack multiple items onto one disk. I will **not** subdivide a disk. Send mailorders to:

Kirk L Thompson / **The Staunch 8/89'er** / P.O. Box 548 / West Branch, IA 52358

>----------<

Extending Zenith's BIOS [Continued from p. 1]

later decide to **replace** the drive with a different type, you'll have to alter the source and rerun MAKEBIOS. This mod also presumes you're adding a double-sided drive; I haven't attempted it with single-sided.

Further, if you want to increase the number of drives running off the **hard-sector** controller beyond three, changes to the system software similar to those discussed here won't do the job. The standard H-17 controller supports a **maximum** of three (3) drives. To add more, you must **modify** the controller board! For details on how to do that, you'll need Chaos Computer's "Six-Drives Mod" package described on the SOFTWARE LIST in issue #22/23. The easiest way to expand hard-sector drive capacity beyond standard 40-track/single-sided is to buy Livingston Logic Labs' BIOS-80 from Quikdata.

To take full advantage of this mod for soft-sector, of course, you'll have to install a fourth drive running off the H-37 controller. This may entail crimping an additional cardedge connector to your interface ribbon cable and adding a "Y" to the power supply cabling. Don't forget to program the drive as hardware unit "3" and move the terminating resistor pack to it from your old hardware unit "2". (On some drives, such as the Mitsubishi I used, installing the terminating resistor wasn't necessary.)

However, you may **not** mix an old Siemens/Wangco Model 82 single-sided/48-tpi drive with other types. The H-37 controller has a jumper at J3 were "precompensation" is set. This jumper is set to different positions depending on whether your drives are that old make/model or newer types. If you do mix them, you could suffer decreased reliability. You **may** mix 48- and 96-tpi types provided they aren't the Model 82, however. These could even be the later Siemens FDD100-5 that externally appears to be **identical** with the Model 82; check the nameplate on your drives to be sure of what you have. If you have questions about this process, by all means contact me. While the hardware is certainly important, my interest in this article is to describe for you those modifications required by the system **software.**

The next thing to do is to **isolate** the BIOS source and the utilities you'll need for editing and running MAKEBIOS. So copy the **entire** contents of disk III of your distribution disk set to a formatted disk. Be sure you have **at least** 65K of free space on this disk for any .BAK file created by your editor and for the assembled BIOS.SYS.

On the bootable disk you use for this operation, you should have ASM.COM, SUBMIT.COM, PIP.COM, and your editor, at minimum. And while you're at it, you should also prepare **another** bootable disk on which you later can run the mod through its paces as I outline below. As I noted in #22/23, I have a Mitsubishi (96-tpi, double-sided, half-height) in the fourth drive slot. But if you prefer another style (such as 48-tpi), that won't make much difference in terms of the BIOS.ASM mods we'll be

doing. There are a couple of places where you **can** exercise some "options" as I noted above, depending on your exact configuration. But I'll expand on those below. And if you want to follow my discussion in the printed BIOS code that Heath provided, by all means do so; I'll give page numbers for the 2.2.03 source.

Finally, before I describe the modifications, I should make two observations. The first is that Zenith's BIOS contains an inherent limit on the **number** of drives it can handle. This limit is eight (8) drives. If you'll turn to p. 13 of the BIOS listing, you'll find a short section beginning with

```
NDISKS  EQU     H17ND+H37ND+H47ND+H67ND
```

This EQUate totals the number of drives from the one or two different types you've selected with MAKEBIOS. Then this count is compared with "8" to see if it exceeds it. This limit won't affect the modification described here on a reasonably **standard** system. For example, my H-89 now has four H-37 and three H-17 drives. But if you've already extended hard-sector capability beyond the standard maximum of three, you **will** run into trouble installing this mod. The second thing I should note is that **mixing** drive types when running MAKEBIOS will not effect performance of this modification. Though originally developed for only H-37, I reran MAKEBIOS for a combined H-17/H-37 system and had no problem. But you should note my cautionary remark below about combining system images with different BIOS's.

Editing BIOS.ASM. Once you've installed that fourth drive, boot your working disk and put the disk with the files from distribution disk III in drive B:. Since many of you have absolutely **no** experience with assembler, the remainder of my discussion will be at a novice level and in a stepwise fashion. So those of you who are assembler hunks, please bear with me! Now invoke your editor to alter B:BIOS.ASM. Use its "find" function to locate the label "H37ND" (without the quotes, of course). This label is part of an "EQUate" that tells the BIOS the **maximum** number of soft-sector drives there can be on the system. (This section of the code is on p. 3 of the printed source.) We're going to increase that from three to four. The relevant area you'll see on screen will look something like this:

```
BRKKEY  EQU     FALSE AND INTINP
H37ED   EQU     TRUE AND H37T
H47ED   EQU     TRUE AND H47T
H17ND   EQU     3*H17T
H37ND   EQU     4*H37T          ;FROM '3' TO '4'
H47ND   EQU     2*H47T
H67ND   EQU     2*H67T+H67PART2
```

Notice that I've boldfaced the number four (4) further along the same line that the "H37ND" label you just found is on. You'll actually see the number three ("3") here; change it to four and TAB beyond the normal end of the line to add the comment ";FROM '3' TO '4'" (as illustrated above). This is a reminder of what you've changed. Be sure you put the semicolon (;) in front of your comment. If you don't, the assembler will think the comment is **part** of the code!

The other two mods are further down in the .ASM code; the last is actually near the end. So if you're using PeachText, Magic Wand, or TXTPRO, you'll have to do one or more "write" and "read" commands to find the labels. I should also note that all **three** of these modifications must be in place for the BIOS and that fourth drive to work correctly. If they aren't, the results will be highly unpredictable! One place where you can check the "quality" of your mod is in the console messages from the assembler. I also have a series of tests to check performance. But I'll discuss these at the appropriate places.

The next label to look for with the "find" function of your editor is "DPE37$2". What we're in the middle of now (once you've found it) is a set of tables that **define** drive hardware. (You'll find this section on p. 22 of the printed 2.2.03 BIOS source.) We're going to add space in this table for that new H-37 drive. The label the cursor is now presently at is the start of the last entry in that table and it extends down to the line just above "ENDIF". So **replicate** this last entry in the table with your editor's **copy** command. What you end up with should look like this:

```
DPE37$2 DW      0000H,0000H
        DW      0000H,0000H
        DW      DIRBUF,DPB37$2
        DW      CSV37$2,ALV37$2
        DB      DPEH37+DPEDD
        DB      CONDS2
        DB      2
        DB      8
        DB      DPEUNK
        DB      FDFS30
        DB      0
        DB      0
; NEW TABLE FOR 4TH H-37 DRIVE
DPE37$3 DW      0000H,0000H
        DW      0000H,0000H
        DW      DIRBUF,DPB37$3
        DW      CSV37$3,ALV37$3
        DB      DPEH37+DPEDD
        DB      CONDS3
        DB      2
        DB      8
        DB      DPEUNK
        DB      FDFS30
        DB      0
        DB      0
        ENDIF
```

Again, I've boldfaced the new material. (The boldface print somewhat distorts column line-up here and below; just be sure the columns are separated by TABs.) And notice that I've inserted a comment at the start of the new table entry for future reference. (Don't forget the semicolon at the beginning of the line!) Other changes you must make to the new section are to alter all occurrences of "$2" to "$3" and "CONDS2" to "CONDS3". You'll see those above as they should be when you're finished.

Finally, in this section, you may exercise the two options I mentioned above. The first of these is to indicate to the system how many tracks-per-inch

(tpi) the drive you've installed has. If your new drive is 40-track (48-tpi), you can ignore this first option. I discuss why a bit further below. But if that drive is 80-track (96-tpi), you **must** add some additional code to the fifth line in the **new** table entry. To properly set up an 80-track drive, that line should now read:

```
DB      DPEH37+DPEDD+DPE96T+DPE48RO
```

As above, I've boldfaced the addition you should make. I'd better also observe that my additions use **preexisting** EQUates in the source code; you'll find these on p. 16. I'm using them because there's considerably less likelihood of your or I making a keyboard error than if we tried directly entering the binary values they represent. (EQUates are to assembler what constants are to BASIC, Pascal, etc.) Now what have we done here?

This line contains the drive hardware configuration and is stored using bit-wise values. That is, each **bit** in the byte means something **different** to the system. So let's look at the actual bit values for these four as taken from p. 16 of the source code:

Label	Binary Value	Dec.	ASCII
DPEH37	0 1 1 0 0 0 0 0	96	`
DPEDD	0 0 0 0 0 0 1 0	2	STX
DPE96T	0 0 0 0 1 0 0 0	8	BS
DPE48RO	0 0 0 1 0 0 0 0	16	DLE
Final value	0 1 1 1 1 0 1 0	122	z

Notice that the on-bits (1's) in each of the four bytes is in a different location. By adding the four EQUates together as I do, I get an ASCII character ("z"). But the system doesn't care what the specific character is since it only reads the **bits** to figure out the drive's configuration.

Now DPEH37 designates that the drive we're adding is to run off the soft-sector controller; if you look at the EQUates on p. 16 of the source, you'll notice that some of the "Heath extensions" given there include **different** bit-wise values for each drive type (H-17, H-37, H-47, and H-67) supported by the BIOS. And if you scan the "Disk Parameter Entry Tables" on pages 21-23 of the source, you'll see that this usage is consistent for each of the four types.

The DPEDD label is the **assumed** density of the drive at cold-boot and means double-density. Actually, this could probably be anything (single-, double-, or extended), but I'm keeping with the default used for the other three drives. When this mod is complete, if you log a disk in the new drive that has been formatted at a **different** density than double-, the drive behaves just like the other three. It automatically "adjusts" itself to the format of the disk.

The next two labels, DPE96T and DPE48RO **must** be present, as I wrote above, if you install a 96-tpi drive. The default configuration for the drives is 48-tpi. A glance at the comment on p. 16 immediately following the DPE96T EQUate tells the story; if that bit is "0" rather than "1" (as it would be if DPE96T weren't there) the drive is assumed to be 40-track.

With this bit equal to 0, FORMAT.COM **always** comes up with the message that the disk will be formatted at 48-tpi! (Perversely, this isn't true if booting is done from **other** than hardware drive 0; I discuss this quirk below.) Indeed, this was one of my frustrations as I worked on this modification. The break in my dilemma came when I paused to consider the meaning of those bit-wise values.

But just adding DPE96T to the line isn't the whole story. If you want the option of at least **reading** 48-tpi disks in a 96-tpi drive, you also need DPE48RO. This EQUate sets the drive to do that. If you **don't** include it, you'll get a BDOS error every time you try logging a 40-track disk in that drive! (I've done that!)

The second option you have available here is to set the drive step rate. But the values you have to choose from are limited. These are given in the source code on p. 6. The BIOS **only** recognizes four (4) values: 6, 12, 20, and 30 ms. If you want other values, you're out of luck since these are coded, like the drive configuration, as bit-wise values. If you've installed a Siemens FDD100-5, you should probably use 20 ms. All other drives can be set for six ms. So, as above, since I'm using an EQUate label, for most drives you can change line 11 in the new table to:

```
        DB      FDFS6   ;HARD-CODE 6 MS STEP RATE
```

For a Siemens, use the EQUate, "FDFS20". And add a comment with the appropriate value to the right of the code.

Before I leave this table, I have one further comment. This area appears to be where CONFIGUR patches the BIOS for the two parameters (track density and step rate) you can input in option B of the program's menu. The related section for setting step rate on hard-sector is a little earlier in the source listing. So now we both know!

Continuing, the last additions to BIOS.ASM are near the end of the file. In the printed code, you'll find this on p. 140. And, again, we have to extend an existing table. So use your editor's "find" function to locate "DPB37$0". Once there, all you need do is add three lines as follows:

```
        IF      H37T
DPB37$0 DS      DPBL
ALV37$0 DS      50
CSV37$0 DS      64
DPB37$1 DS      DPBL
ALV37$1 DS      50
CSV37$1 DS      64
DPB37$2 DS      DPBL
ALV37$2 DS      50
CSV37$2 DS      64
DPB37$3 DS      DPBL    ;4th drive entry
ALV37$3 DS      50
CSV37$3 DS      64
        ENDIF
```

Don't forget the comment. Once you've inserted these lines, you can save the amended BIOS source file.

MAKEBIOS. The next step in this process of adding a fourth drive to your soft-sector BIOS is to run MAKEBIOS. If you've followed my instructions above,

you already have all the files you require for this procedure. One thing I recommend, though, is that you give the BIOS you're about to assemble a distinct name. That way, if you fluff the destination for it and accidentally key drive A:, you won't **overwrite** the BIOS on your boot drive!

The command line I used was:

```
SUBMIT B:MAKEBIOS B:BIOS4D37.SYS B:
```

But the name itself is up to you. After you've selected the drive combination you want from the menu that appears, assembly will take upwards of eight minutes. So you have time to get up, stretch, and even get a cup of coffee. When done, you'll see that the BIOS has gone through the assembler (ASM) twice; this is normal.

What I want you to specifically look at now is the part of the screen directly **after** the assembler was invoked each time. Both of these should look something like:

```
CP/M ASSEMBLER - VER 2.0
13D4
02DH USE FACTOR
END OF ASSEMBLY
```

The hexadecimal numbers you see on the second and third lines will vary somewhat because these reflect the size of the BIOS you assembled and is to be expected. The second assembly is even somewhat different from the first. What you **don't** want to see is **anything else** between the assembler sign-on and "END OF ASSEMBLY" messages. If you do, go back and recheck your mods to BIOS.ASM! This is the first of the quality checks you **must** make **before** you put the new BIOS into service. Of course, if you see something amiss here and can't figure out what's wrong after reviewing the BIOS source, let me know. And if you can't get a "clean" assembly, **don't go any further**! Something is definitely wrong and it must be fixed before it damages hardware or causes corruption of your data!

System Images and the BIOS. To digress a minute, I don't plan to use this four-drive BIOS on all my bootable media. I keep a small number of **special-purpose** systems and BIOS's on one of my general-purpose system disks. These provide CP/M systems with special capabilities or limitations which I can install with SYSGEN if the need arises. I also keep a custom BIOS in reserve; my standard BIOS includes both H-17 and -37 drive types, but the custom one is exclusively H-37. Indeed, I used the latter when bringing up WordStar 4.0 before I began work on the patches to that word processor back in December. The four-drive H-37 BIOS described here will be another of the custom type.

Further, it isn't obvious from either the 2.2.03 or 2.2.04 manuals that SYSGEN will accept a **filename parameter** on the command line. But this is handy for quickly preparing bootable disks with special configurations. As examples of how this works, I have one system image file prepared with MOVCPM37 named CPM64H37.COM for a 64K, H-37-only system. This one was set up explicitly to maximize TPA on a soft-sector system. Another system image is named CPM48.COM and is for compiling Lucidata and Turbo Pascal with limited memory (48K) and both H-17 and H-37 drive types. The syntax of this poorly-documented SYSGEN command is:

```
SYSGEN filename.ext
```

You **must** include the extension of the system image file name; I use ".COM" as recommended in Magnolia's CP/M manual. When you do this, instead of pulling the system from your boot disk's boot track(s), the **file** is copied to the destination's boot tracks. You must PIP over the BIOS yourself, but this is handy when setting up custom configurations.

But care must be exercised when pairing system image files with BIOS's. You **can't** mix as you wish. The BIOS file actually tranferred to a disk **must** match the file (called "BIOS descriptor" in the 2.2.03 manual) specified when MOVECPMxx was run to create the system image file or, at least, be the same size. (While testing this BIOS mod, I had one cold-boot system sign-on that said it only had **01K** of memory! This was because I was "illegally" mixing system images and BIOS's.) So you should set up a system for matching system image files and BIOS.SYS files. I also strongly recommend that you review the section on MOVCPM in the system manual.

Testing the New BIOS. Testing of the modified BIOS is actually pretty straight-forward. You first may want to set up a custom system image with MOVCPM as discussed above and in the system manual. Then the thing to do is to prepare a bootable disk for testing (if you haven't done so already) and PIP the new BIOS to it. For this, I would recommend the command-line:

```
PIP d:BIOS.SYS=B:BIOS4D37.SYS[RV0
```

where "d:" is the destination drive, probably C: if you're adding a fourth drive to your system. The "R" switch indicates that you're tranferring a ".SYS" file. If the BIOS on your destination happens to be write-protected, you'll be prompted whether you want to delete it before the copy operation begins. Answer with a "Y" and carriage return. Once you have the test disk prepared and PIP'd some files to it (CONFIGUR, FORMAT, PIP, and some ASCII files, in particular), remove all disks from your system's drives.

A formal matrix of the tests I performed is presented in Figure 1 [bottom of the facing page]. You may recall that I introduced the matrix as a means of organizing and documenting tests in issue #25 in my erratic series on software testing. But the idea can be used for other things as well. In the figure you'll find almost a dozen tests I've outlined. These were specifically designed to ensure that the 96-tpi Mitsubishi drive I installed last year functioned just like any **other** drive on the system. So, if you've set up a 40-track drive, you won't need either the references to the higher track density or tests 7 and 8 (where I'm checking that a 48-tpi disk is accessible but read-only in the 96-tpi drive). But other than that, you should perform the tests described. And I probably needn't mention it, but the files on the data disk you use for your tests had better be expendable in case something goes awry.

Commentary on some of these tests is worthwhile. First, insert the test media into your normal boot drive, and boot as usual. During loading and system sign-on, you shouldn't see anything unusual on the screen. If you see anything out of the ordinary (such as unusual sign-on text), there's a problem. If you hear the drive begin an epileptic fit during the boot process, press SHIFT-RESET immediately! Something is wrong with your mods even though no errors appeared during assembly. (As all programmers know, that does happen!)

Second, if you have CONFIGUR on the test disk, run it if you like, but as I observed above, the program **won't** recognize the new drive. The remainder of the tests are straightforward and can be performed as given in the matrix, except for the last two.

Things change when cold-booting from a drive other than hardware unit 0, that is, with "B1" or "B2" at the monitor prompt. Under both Zenith's CP/M and HDOS, the drives do something of a round-robin. Drive A: or SY0: is **always** the boot drive, so B:, C:, SY1:, SY2: etc., rotate through the drives recognized by the BIOS or driver. But **only** if the drive is recognized. If it isn't, that drive is skipped! For example, before making this BIOS mod, even though the fourth drive was installed, when booting from hardware 1 or 2 that new drive was **skipped** over by the system. After this mod, the fourth drive is **included** in the round-robin drive assignment. Further, before this mod was installed, if you try cold-booting from the fourth drive (with "B3"), the boot process hangs. After the mod, booting proceeds normally.

And there was one thing that surprised even me during testing of cold-booting from a drive other than hardware unit 0. This is test 10 on the matrix. Running FORMAT on the new drive produced the message that it would be done at **96**-tpi, even though CONFIGUR displayed a track density of 48 for the drive. So the BIOS continues to retain some of its secrets!

I have one more thing to add before I close. And this is about Magnolia's CP/M 2.24 when cold-booting from hardware units **other** than 0. You can't do it! Though the boot process starts in the drive you've selected, it **reverts** to hardware unit 0 (that is, the ready light on the boot drive goes out and the one on unit 0 lights) and boot-up hangs. Here, Zenith's implementation has an advantage if hardware unit 0 is acting up.

Wrap Up. Anyway, I'm quite pleased with the result of this BIOS enhancement project. It helps me serve your needs a bit more easily. And it brings Zenith's CP/M up to the same level of compatibility with my installed hardware that HDOS 2.0 and 3.0x and Magnolia's CP/M 2.24 already have. It also demonstrates, I think, that you don't **necessarily** have to be an assembly language or systems programmer to make improvements to your operating system software. (Finishing a project just takes longer!)

Finally, I must thank Peter Shkabara for his invaluable assistance and encouragement. I **also** have to thank him for not giving me all the answers! Though I've added a major enhancement to my system, more important, certainly, is the knowledge I've gained about how parts of the BIOS work and the experience in assembly language. Both are indeed **inestimable.**

=====

Figure 1: Test Matrix for Four-Drive H-37 BIOS Mod.

Tst #	Test Description	Expected Result	Error Recovery (if any)	OK? Y	OK? N
1	Cold-boot from hardware drive 0	Normal sign-on & no excessive drive clatter	SHIFT-RESET	x	
2	CONFIGUR.COM	Doesn't recognize new drive	None	x	
3	DIR w/ no disk in drive	Ready-light lites and get BDOS select error	CTRL-C	x	
4	FORMAT disk	Formats at 96-tpi	SHIFT-RESET	x	
5	PIP files to drive w/ [V switch	Copied with no errors	CTRL-C if BDOS error	x	
6	Read ASCII files on drive	No errors	CTRL-C if BDOS error	x	
7	Warm-boot and access 48-tpi disk w/ files	Display disk directory and ASCII file	CTRL-C if BDOS error	x	
8	Copy file to 48-tpi disk	BDOS error: R/O	CTRL-C	x	
9	Cold-boot from drive other than hardware 0 (B1 or B2) and DIR w/ no disk in drive	System recognizes new drive, BDOS select error	CTRL-C	x	
10	FORMAT disk in the configuration of test #9	Formats at 96-tpi!	SHIFT-RESET	x	

VENDOR.UPDATE

HEATH/ZENITH MAIL-ORDER LIBRARY CATALOG. [From Lee Hart, 323 W 19th St., Holland, MI 49423] "A vast amount of information has been published on H/Z computers. The solution to your problem has probably already been discovered and published; you just have to find it!

"Your local library isn't much help; they only have popular periodicals, and their selection of computer books is quite limited. You may also need a manual for a new piece of hardware, or would like to look at the manual for a particular software program to see if it's worth buying.

"TMSI has an excellent collection of books, manuals, and periodicals relating to microcomputers in general, and Heath/Zenith systems in particular. I would therefore like to announce formation of the Heath/Zenith Mail-Order Library.

"You can borrow any item for 4 weeks for a handling fee plus postage. A deposit is listed below for each category, depending on its value. Send the deposit plus enough for postage, and the requested items will be sent to you by first class priority mail. When you return them, the deposit and any leftover postage minus the handling fee will be refunded.

"Example: For 4 issues of REMark, send $4 + $1 = $5. They weigh about 2 pounds, so postage is $2.90. When you return them, the refund will be $5 - $2.90 - (4 * $.25) = $1.10.

"Some items may also be borrowed with software on disk. If you want the disks, include an extra $1 and specify the disk format desired. Unless otherwise specified, software is CP/M format. Note: the disks will contain DEMO versions of the program, not the full package! Print, configure, and save to disk options are typically disabled or removed. If you like a program, ask about price and availability.

"The price is deliberately kept low to encourage people to learn more about their computers. Inevitably, some items will be lost in the mail or not returned. If you have extra copies of these publications or items you think others would find useful, donations would be greatly appreciated!

"To use the H/Z Periodical Lending Library, contact:

H/Z Library c/o Lee Hart / 323 West 19th Street / Holland, MI 49423

Periodicals ($1 deposit, $.25 handling fee)
Books ($5 deposit, $1 handling fee)
Hardware Manuals ($2 deposit, $1 handling fee)
Software Manuals ($5 deposit, $2 handling fee)
[Lee's list is much too long to reproduce here; contact either him or **Staunch** for the entire thing. -Ed.]

"P.S. - Indexing! **Z-100 Lifeline** offers an invaluable index to every article on the Z-100. It's available on disk from Paul Herman, 9317 Amazon Drive, New Port Richey, FL 34655. It makes it easy to search for information on any topic with a word processor, regardless of the date or publication in which it appeared.

"This is so handy that I hope it will encourage others to do the same for other H/Z computers. If anyone can help compile an index on disk of publications for the H8, H89, or other H/Z computer, please let me know." [**Staunch** has sent Lee the files for the indices from issue #'s 10 and 16. Others will be provided as they become available. -Ed.]

=====

Troubleshooting the '89, Pt. 2
The H-89 Keyboard Module

By Dan Jerome, with Kirk L. Thompson
[Concluded from the last issue. -Ed.]

(2) PROBLEM: DUST AND JUNK ACCUMULATES BETWEEN THE KEYS Even if your keys are not stuttering, it doesn't take long for a nasty collection of dust and junk to become visible between the keys and if you shine a flashlight down between them, you may see additional particles of junk on the topside of the keyboard circuit card.

FIX: KEEP THE COMPUTER AND KEYBOARD COVERED If you do not have a standard Heath H-89 cover for the computer and keyboard, plus any disk drives that you may have on the system, you may be able to locate something suitable in an office supply store. If not, cover the keyboard with an appropriately sized and tucked-in piece of common plastic sheeting.

It is highly recommended that you put some kind of dust-proof cover over the computer and keyboard whenever the computer system is not in use, so as to prevent dust and junk from accumulating. If you acquire too much dust or junk particles, they can definitely interfere with keyboard operation and cause you grief.

If you notice that your keyboard is covered with dust and junk between the keys, not to worry! The fix for this problem is to first sketch the location of each key on the keyboard and keypad on paper. This technique is sure to help keep Murphy's law from interfering. (The main corollary of Murphy's law is: if a thing can go wrong, it will.)

Then remove the keycaps from all of the keys. I found it quicker, when putting it all back together later, to carefully arrange the keycaps in the proper sequence. Once the keycaps are removed, take a 1-1/2-inch, soft, angled paintbrush of "fine" quality. (The one I use has a legend that says, "DUPONT, IDEAL FOR LUCITE PAINTS.") and carefully brush between the key innards and also each key.

After all the keycaps have been removed, **without disassembling** the component parts, apply a tank type vacuum cleaner **carefully**, and vacuum as much of the dirt as possible. There is always a residue of dirt remaining. Now use your paintbrush to sweep up the residue into piles so that it can be vacuumed.

A tiny, dedicated computer vacuum which would do a great job in helping you to keep your computer clean is a stock item at most mail order businesses and in many of your local local computer stores. For example: Lyben Computer Systems, 1150 Maplelawn, P.O. Box 1237, Troy, Michigan 48099, phone: 1-313-649-4500. One example is "Metro Vac 'N' Go," which sells for $37.95 - ID NO. LCS3818. Extra dust bags: $2.95 for 5 - ID NO. LCS0524. Shipping and handling cost is a flat $3.00. I can recommend this place, as

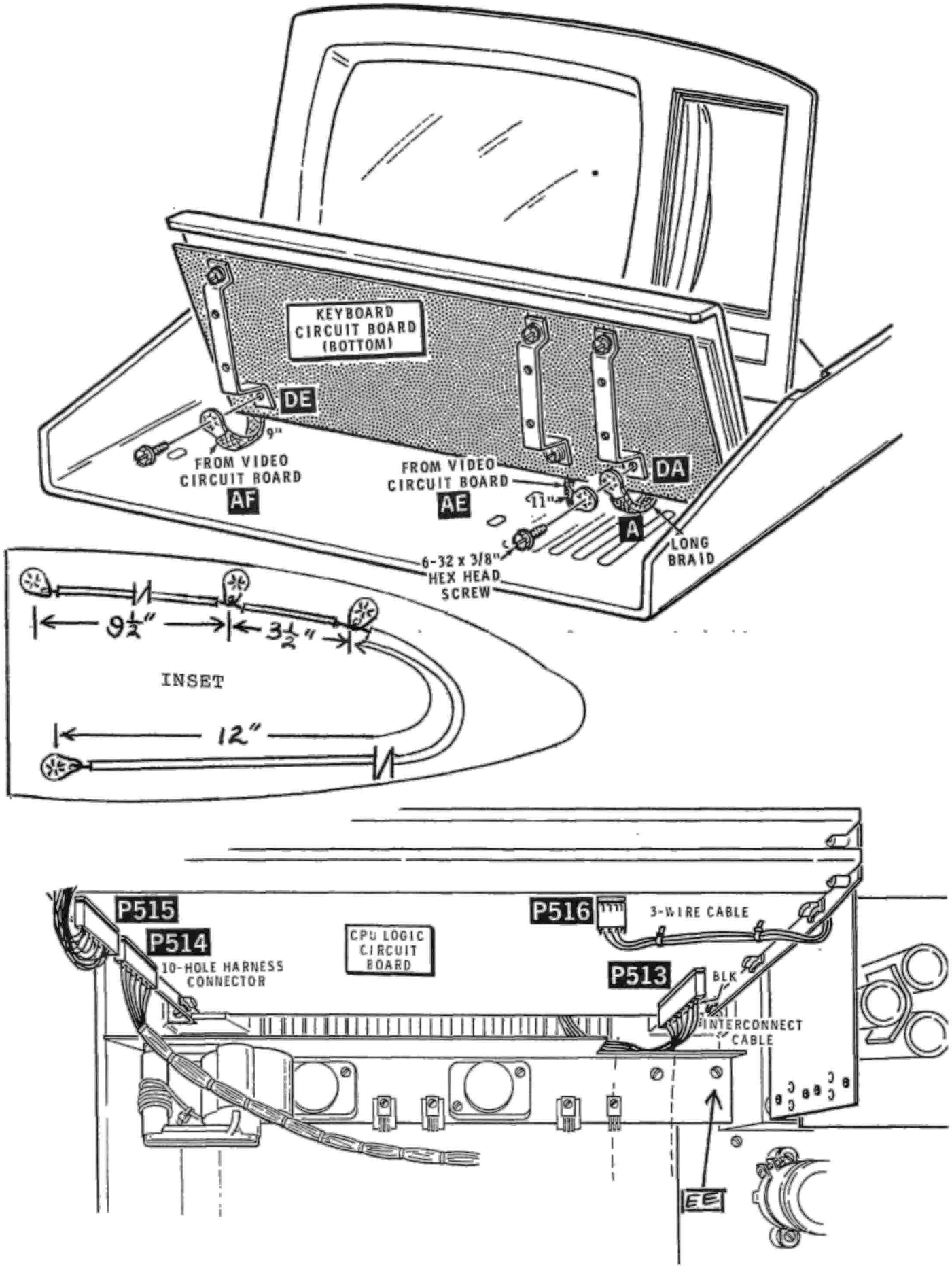
KEYBOARD CIRCUIT BOARD (BOTTOM)
DE
9"
FROM VIDEO CIRCUIT BOARD
AF
FROM VIDEO CIRCUIT BOARD
AE
11"
DA
A
LONG BRAID
6-32 x 3/8" HEX HEAD SCREW
9½"
3½"
INSET
12"
P515
P514
10-HOLE HARNESS CONNECTOR
CPU LOGIC CIRCUIT BOARD
P516
3-WIRE CABLE
P513
BLK
INTERCONNECT CABLE
EE

FIGURE 2.

they have always given me quick and reliable service - they even carry hard-sector disks.

(3) PROBLEM: KEYBOARD IS INTERMITTENT Common symptoms of this computer "disease" often occur when you press one or more keys, and there are no letters printing on the screen. This condition has been getting progressively worse until now you are stymied as to how to continue using the computer at all.

SIMPLE FIX: FIGHT CORROSION Always keep in mind that corrosion is the biggest problem for anything electronic. The older the unit gets, the more corrosion accumulates. Finally, resistance grows so high that some connectors or switches can no longer pass the current or signal.

If you have any problems getting your computer on line after you have done the above fixes, not to worry. Your next line of defense to combat corrosion - and often the most successful - is to unplug the connectors and reseat them twice. For example the pins and sockets on the 34-pin flat cable that runs from the keyboard to the TLB board. Believe me, it is worth unhooking and lifting the CPU board to clean the pins of the TLB cable connector.

If the problem persists, try using a common pencil eraser to "erase" the corrosion. Use few and gentle strokes. If gentle doesn't work, remove the boards or cables and try spraying them with TV tuner cleaner as described above.

Malfunctioning or down computers can be as stubborn as donkeys when they" decide" to go down. The object is to "convince" them that what you are doing is absolutely essential! Therefore, if the first technique doesn't work, try another. Persistence often pays off!

Refer to PROBLEM 1: KEYS STICKING OR STUTTERING (in the last issue) for the exact technique to clean and lubricate all the keys on the keyboard. Then cross your fingers and try again.

If that doesn't clear the problem, it is a good sign that some part (or parts) somewhere has "dishonorably surrendered in the line of duty." In this case, after having done all of the above techniques, start first level troubleshooting procedures with checking the power supply voltages [covered in issue #26/27, p. 16].

THOROUGH FIX: REPLACE KEYBOARD Sometimes the cause for an intermittant key on the keyboard is more serious than corrosion. If the computer has been well-used, the contact tines of one or more keys may be on the verge of breaking off deep inside the honeycomb base! Complete breakage could **actually happen** when performing the operations described above under (1) PROBLEM: KEYS STICK OR STUTTER. The most likely candidate key for breakage, in our experience, is the space-bar.

The only repair possible is to completely **replace** the keyboard unit. Simply replacing a contact tine is impossible because the keyboard honeycomb base is soldered, as a unit, to the underlying printed circuit board. To replace the keyboard, turn off the computer and unplug it from its AC socket. Unfasten the keyboard assembly from the base by removing from underneath the cabinet the six long screws that hold it in place. These screws are all forward of the front rubber feet on the cabinet base so you can do this with the keyboard overhanging your table-top. As you tip up the keyboard assembly from the front, as illustrated in the upper drawing of Figure 2 [on the facing page], be careful of the connecting ribbon cable that extends out the rear.

Now note down on a slip of paper the orientation of that connecting ribbon cable, such as whether the darker stripe on the cable is to the left or right edge or whether the cable, as it goes into the connector, is flush with or in the air above the circuit board. Carefully disconnect the cable. Also check to see if there are any wires or copper braid straps connected to locations DE through DA as shown in the figure. If so, remove the screws at these locations to completely free the keyboard unit.

Place the keyboard unit upside down on your work surface and remove any remaining screws holding the three support brackets in place. Lift the old keyboard unit free of the cabinet piece and set it aside; you can use it as a source of keycaps and other key innards. Set the new keyboard over the cabinet piece and refasten the support brackets, but only lightly tighten the screws. Flip the assembly over and check the action of the keys around the periphery of the keyboard. Some may be rubbing on the cabinet piece. If so, gently push the keyboard around on its mountings until **all** of the peripheral keys move freely. Now tighten the screws and reassemble the keyboard in the reverse order in which you just disassembled it. Be sure you have plugged the ribbon cable connector in correctly.

(4) PROBLEM: CABLE BECOMES DISCONNECTED With just a little stress, such as when the user is attempting to install an Anapro key repeat accessory that requires that you detach the keyboard from the computer case and unplug the 34-pin flat cable from the keyboard, sometimes you may accidentally pull the cable out of the TLB board connector.

FIX: Open the computer cover, unplug and detach the CPU board by releasing the screws holding it to the frame. Then gently lift the CPU board up so that you have enough room to slip your hand under it and plug in the keyboard cable.

(5) PROBLEM: STATIC DISCHARGE LOCKS UP SYSTEM On an older H-89, one of Heath's design flaws was improper grounding of the keyboard unit. When the computer was redesigned in 1980/81 to comply with FCC RFI-emission requirements, ground straps were added, as illustrated at locations AF, AE, and A in the upper drawing of Figure 2. But on the older system, that lack of appropriate grounding may cause the system to lock up. This can occur from a discharge of static electricity when you merely touch the keyboard on cold, dry winter days. Only a SHIFT-RESET will clear this condition and in the process, of course, you lose whatever you were working on.

FIX: First, check to be sure that a ground wire hasn't already been installed. Turn off the computer and unplug it from its AC socket. Unfasten the keyboard assembly from the base by removing from underneath the cabinet the six long screws that hold

it in place. These screws are all forward of the front rubber feet on the cabinet base so you can do this with the keyboard overhanging your table-top. As you tip up the keyboard assembly from the front, as illustrated in Figure 2, be careful of the connecting ribbon cable that extends out the rear. Now look for a wire connected to each of the keyboard mounting brackets and extending further back into the cabinet on the right side. Usually, this wire will be attached to points DE through DA in the illustration. If one is already there, well and good; you need proceed no further, so refasten the assembly to the base.

If one **isn't** there, prepare a length of wire as shown in the inset in Figure 2. Use a fairly heavy-gauge stranded wire for this, #18 or #16. Ideally, the solder lugs you use should be the type that integrates lock-washer tines into its structure, as illustrated on the left. However, you can also use spade- or ring-tongue lugs if these are the only kind available to you. However, avoid the type of terminal lug that you **crimp** to the wire; only a **soldered** connection will ensure long-life to this ground wire. The lengths of wire between lugs is approximate; a little extra won't hurt anything.

Now remove the screws and fasten the lugs of the ground wire at points DE through DA, using the corresponding labels on the inset as your guide. Thread the **remainder** of the wire under the keyboard assembly and front bezel parallel to flat cable. With this part of the installation complete, refasten the keyboard to the cabinet base. As you do so, carefully reroute the ribbon cable and your new ground wire.

The last lug on this new wire must be attached to the heatsink of the video board. The most "convenient" place to do this is at the right-rear mounting for the board, location EE in the bottom illustration in Figure 2. To gain access to this part of the computer, unlatch, lift, and remove the top shell of the cabinet. Don't forget to disconnect the power cable to the ventilation fan. Now remove the support bracket for the I/O cards on the right side of the CPU board and all of the cards. Be sure you mark on a sheet of paper which cards and cables go where, particularly the two or three cables that connect to the serial I/O board. Locate your new ground wire, remove the screw at EE, and fasten the wire to the video board's heatsink.

This completes installation of the keyboard ground wire. Reassemble the computer, being particularly careful as you plug the interface boards back into the CPU board and the interconnecting cables to these boards.

=====

R/O MEDIA

(A Summary of Resources about 8080/Z80 Systems)

The Computer Journal (P.O. Box 12, Southfield, NJ 07080-0012; bimonthly, $18/yr.): The Jan/Feb., '92, issue celebrates the 10th anniversary of ZCPR. Columnist Jay Sage, in the lead article, explores that history and the changes from ZCPR1 through ZCPR34. Regular "Advanced CP/M" columnist Bridger Mitchell discusses I/O redirection for CP/M Plus, with asides on ZEX and BackGrounder ii. David Goodenough explores interrupts on the Z80. George Warner discusses doubling the clock speed of the Ampro Little Board. Paul Chidley introduces a new column titled "Hardware Heaven"; in his first installment he discusses the Dallas Semiconductor DS1216E "smart" IC socket and where to find data books. Guy Cousineau summarizes the development of TDOS, a p.d. operating system for the Z80. Other articles discuss a FORTH assembler for the 6808, LANs, ZCPR on the 8088, the conclusion of a three-part series on a software-controlled waveform generator, and miscellaneous topics.

COMPUTER MONTHLY (P.O. Box 7062, Atlanta, GA 30357-0062; monthly, $15.95/yr.): Last summer and fall, material the might relate to our equipment dropped off to virtually nothing. The reason was because of the reorganization and move of FOG International (the group that provided the material) to a new address. The dust from those changes has now settled, the "FOG NOTEBOOK" section begun to grow again, and relevant material to reappear. In the March issue, a reprint from **TOGGLE** (newsletter of the Tacoma Open Group for Microcomputers), presents a five-segment briefing on some of CP/M's BDOS calls. And W.H. Friedman reviews VSPELL, a spelling checker for CP/M and MSDOS from Greenview, formerly CompuView, famous for VEDIT. FOG's BBS listing is absent from this issue, but the user group listing has been completely rewritten; **Staunch** is even there after a prolonged absence. Nancy Black's "Fearless Computing" discusses printer problems with the CP/M version of WordStar 4.0, though from a Commodore angle. The remainder of the issue is, as I described is #22/23 (p. 12) mainly for the PC, though **CM**'s regular columns for the Adam, Amiga, Atari, Commodore, Sanyo, TI-99, TRS-80, and Timex/Sinclair are here.

Eight Bits and Change (Small Computer Support, 24 E. Cedar St., Newington, CT 0611; bimonthly, $15/yr.): The Feb/Mar. issue continues a discussion of fractals and the Mandelbrot set by Chip Bradley and Larry Schnitger (see Lee Hart's excursion of the same subject for the '19/89 in **Staunch** #11) and has some nicely-detailed printouts. Tom Viele has a humorous discourse with "Hardhacker" about a revolutionary programming breakthrough called BLASTDOS. Lee Bradley reviews ZSIM, a Z-System emulator for PCs. The issue also includes an update on the YASBEC (including song lyrics by Jeff Duntemann and Lee Hart), and notes on STAT and other CP/M and Z-System utilities.

H-SCOOP (2618 Penn Circle, Sheboygan, WI 53081-4250; monthly, $28/yr.): In issue #144, Henry Fale announced the dropping of a number of 8-bit products from his catalog. The largest block of these is software from Software Toolworks for CP/M or HDOS - COMPUTER CHEF ($19) (and two add-ons, WHAT'S FOR DINNER [$12] and BEST OF WOK TALK [$15]), ELIZA ($15), LISP-80 ($15), PACK/CRYPT ($12), and REACH modem package ($19). These are supplied as

soft-sector; add $5 for conversion to hard-. Also in limited supply are the ACCESS modem package for CP/M ($35 on soft-sector and highly recommended) and Electrokonsult's CP/M utility package ($25 and also recommended). On the hardware side, write and ask him specifically about his Magnolia 77320 hard-drive/serial card deal for $95 or $120, depending on options. Also being cleared are chips for the H-8's WH-64 RAM board; price is $3 apiece. Henry notes that he plans to continue some 8-bit items in his catalog for the next year, but that may be it!

SEBHC Journal (895 Starwick Drive, Ann Arbor, MI 48105; monthly, $24/yr.): The January issue concludes a BASIC listing for a CAI program by Oscar Yohai. As well, this issue includes notes on the text editor, TXTPRO, and care of your printer and ribbons. In the February issue, Rick Swenton briefly describes use of a PostScript printer. Editor Lenny Geisler also announces the release of software by Rick to drive an X10 appliance controller through the CP-290 interface. And Lenny has a tutorial on using Skycastle's CALLIGRAPHY-II printer graphics software.

Smaller is Better (CCP/M, c/o Tom Viele, 26 Slater Ave., Norwich, CT 06360; monthly, $15/yr.): This newsletter is published by the Connecticut CP/M Users Group. Being the club's newsletter, part of each issue is the minutes of the most recent meeting. However, the February issue (the first I'd seen) included a challenge by Tom Viele to the CCP/M membership to prepare an article about the new field of "genetic algorithms," a merging of digital and genetic technologies for optimizing. A gossip column also predicted that "CP/M will replace UNIX"! The feature article in the March issue, by Gary Stagliano with Lee Bradley, described how to "decode" file attributes in the directories of CP/M Plus and MSDOS.

The Z-Letter (Lambda Software Publishing, 720 S. Second St., San Jose, CA 95112-5820; monthly, $15/yr.): In his February issue, Editor Dave McGlone announced the expansion of his (Lambda's) font and kerning table selection because of a clearance sale on HP LaserJet laser printer fonts from DIGI-FONTS, Inc. Joe Wright of Alpha Systems is also moving east, so Dave will be selling his products under a royalty arrangement, including NZ-COM and ZCPR 3.4 source code. Dave has also finalized arrangements with Digital Research to distribute CP/M-80 and asked his readership for provide him with system information. Further, he included a list of utilities shipped in the most recent mailing to subscribers to the Z-System Software-Update-Service (Z-SUS). The feature article, by Jay Sage in his now-regular "Script of the Month Club" column, is a poor man's ZEX script to park the head on a hard-drive. Besides the usual plethora of ads (including one from yours truly), the regular Eagle users group section closed out the issue.

=====

MISCELLANY

The Bookshelf. I began discussing books I'd recently seen a few issues ago. Let's pick that up again and I'll cover those I have room for here.

C. If your programming interest lies with this up-and-coming, "middle"-level language, here are two, somewhat expensive, volumes to consider. The first is an introduction to the language for the person who already knows one or more other languages, Kenneth Pugh's **C Language for Programmers** (QED Information Sciences, 1989, hardcover). Because it's written for the experienced programmer, Pugh's discussion is concise. And the principal difference between this and similar books I've seen is that illustrative code from **other** languages (BASIC, FORTRAN, Pascal, PL/1, and COBOL) is included for comparison with how C does the same thing. This book is only available from the publisher: QED Information Sciences / P.O. Box 82-181 / Wellesley, MA 02181 / 617-237-5656. Cost is $29.95 plus $3 shipping.

One topic that's given short shrift by Pugh is that of pointers; he only devotes 10 pages to it. In my experience with books on other languages (BASIC and Pascal), this is also typically the case for them, mainly because of the complexity of the subject. Hence, I found the mere **existence** of Robert Traister's **Mastering C Pointers: Tools for Programming Power** (Academic Press, 1990, hardcover) of interest. Now I know almost "zip" about C, so won't even try to evaluate this book. But it can be mail-ordered from the publisher: Academic Press / Orders-Inquiries / Troy, MO 63379 / 1-800-321-5068; price is $34.95 and shipping is included if you prepay.

ASM. "Advanced" books on assembly language programming are something of a rare breed. But I picked up one through a book mail-order firm since I "eventually" hope to delve deeper into the subject. The book is Gary Elfring's **Microcomputer Assembly Language Programming** (Van Nostrand Reinhold, 1984, hardcover). But it now seems to be out of print. Anyway, it features 8080 assembler and discusses top-down design, structured programming, assembler use and techniques, and writing routines for use with high-level languages.

THE STAUNCH 8/89'er, created by Hank Lotz, is a bimonthly newsletter on 8-bit H/Z computers. The editor is Kirk L. Thompson; P.O. Box 548; West Branch, IA 52358; home: 319/643-7136. Subscriptions always start and end with the calendar year. Rate: $12.00/year. (Overseas, add $4.) Single copies: $2.50. Make checks payable to "Kirk L. Thompson". **Staunch** pays authors for their articles; write for an author's guide. It also accepts commercial ads for a modest fee; contact the editor. Neither this newsletter nor its editor is responsible for damages or losses resulting from use of any information presented herein. Info from **THE STAUNCH 8/89'er** may be reprinted only if this publication's name and address is included. Credit should also be given to authors and other sources of said material, if known. This publication is archived by the University of Iowa Libraries. CP/M is a registered trademark of Digital Research, Inc. **REMark** is a registered trademark of Zenith Users' Group. EOF