PEEK (65)

The Unofficial OSI Users Journal

$1.75

JANUARY 1984
VOL.5, NO.1

P.O. Box 347
Owings Mills, Md. 21117
(301) 363-3268

INSIDE

Column One

With just over one month of operation under their belts, the folks at ISOTRON are settling in and charging ahead. Although there are not a lot of concrete announcements to make at this time, there are a number of important irons in the fire and almost ready to pop.

By next month we should be able to report on a triple headed agreement concerning dealer packages, vertical market bundled software packages, and a national advertising campaign. Yes, vertical market packages. A couple of well known and respected software houses are nearing agreement to take over the production and support aspects. Announcement is expected in the first part of January.

More good news. A third party service contract is likely to be announced during the first quarter of '84.

Word also has it that ISOTRON will shortly become a bulk buyer of selected peripherals, with the intent of making these devices available at very competitive prices.

The long awaited Source Book, will have to wait some more. Bob Lewis said that he just was not happy with it and won't release it until it is up to his expectations. But it shouldn't be long in coming.

A number of other agreements for hardware/software collaboration are nearing completion. This approach sounds to us like a good method of getting the job done quickly and at the same time keeping the credibility and responsibility in the hands of the creators who are long time OSI'ers, who know the OSI world from our vantage point and ultimately will have their own reputation at stake.

Now to other matters.

Hackers are getting an undeserved bad name. Ever since the "404 Group" used their home computers to break into systems around the country, most people think of hackers as computer criminals, or vandals at least. Not so!

What is a hacker, really? A hacker is a person who is not satisfied to use his computer in simple ways. A hacker is a person who wants to learn everything he (or she!) can about his machine's hardware and software. A hacker is a perfectionist living and working in an imperfect world.

So when you see the term "hacker," don't think of someone hacking away at a door, trying to break into someone else's property: think instead of an adventurer in a jungle, hacking away at the obstructing vines and branches of ignorance with the machete of his brainpower. Hackers improve the breed. Hackers fight the obfuscation fostered by illegible manuals. Hackers help their fellow computerists. Hackers buy PEEK(65).

This issue of PEEK(65) was largely written by hackers. We have two articles on expanding the SBII and other 6502 computers. We have an article on the inner workings of BASIC, another on modification of the popular WP6502 word processor. Several of the letters describe modifications to keyboards, operating systems, BEXEC* and more.

Has PEEK(65), then, become the hackers' computer journal? Not so! We strive to maintain a balance between hackers and business users, but it's not easy. We don't even have to ask for "hacker" articles describing all sorts of wonderful improvements to personal systems -- they come regularly in the mail with detailed listings, schematic diagrams and lovingly crafted descriptions. We have asked repeatedly for business related articles, and consider ourselves fortunate to get a couple of letters with a little detail and rarely a listing. We know that you are out there and hear about all of your neat things when we talk to you on the phone, but what is it going to take to get you to share your "ditty" with the rest of the OSI world? What really hurts, is to hear of the time you may have wasted recreating the same old wheel again and again.

Now that OSI is back on the track, let us hear from you -- if it is not what you have done, then what you would like to have or feel that is missing. Either way, your contributions will go a long way to making 1984 a Happy New Year.

ae

RELOCATING WP-6502 PART 3

By: John T. Roecker
5141 Thomas Ave., S.
Minneapolis, MN 55410

Up to this point in this series of articles, I have discussed getting WP-6502 running with a non-standard monitor ROM. In this third article, I would like to discuss modifications to WP-6502 to utilize some of the capabilities of the C1S/C2S and C1E/C2E monitor ROMs.

1. I have modified the video output section of my C1P in order to get 32 characters per line. This capability is supported by the C1S and C1E monitor ROMs. Currently, the View command does not utilize this expanded video. It would be ideal to modify the Type and L/Edit commands also, but as I indicated in the first article, those commands require the use of the RUBOUT key for backspacing. The RUBOUT key is masked out by the C1S monitor ROM so these two commands cannot utilize the expanded video. In my changes to WP-6502 up to this point, I have utilized the old video subroutines for the View, Type, Move, Zap, L/Edit, G/Edit, Blk View, and R/Tape commands. The new video subroutines have been used for the W/Tape and the added Print command. What I will describe in this article is using the new video subroutines for the View command also. Memory locations used by the non-standard monitor ROMs will also have to be changed to expand the line to 32 characters. I will modify the warm start code of WP-6502 to perform this. This time I will reproduce most of the warm start code because it has been modified so many times. I will use the non-relocated addresses again as in the previous articles. Both the C1S and C1E provide screen clear subroutines which I will utilize. Added instructions will have a & behind them. Instructions which need to have their address fields modified to suit your relocations will have a * behind them.

```
$ØF3A  2006FE  JSR $FE06 &  Clear Screen C1S
       2059FE  JSR $FE59 &  Clear Screen C1E
$ØF3D  A900    LDA #$00     ----
$ØF3F  8D2906  STA $0629 *  Switch to old
$ØF42  8D3906  STA $0639 *  video routines
$ØF45  A92D    LDA #$2D     C1S only
$ØF47  8D1A02  STA $021A
$ØF4A  A0BF    LDA #$BF     See article 1
$ØF4C  8D1B02  STA $021B    ----
$ØF4F  A0FF    LDY #$FF
$ØF51  20A203  JSR $03A2 *
$ØF54  200D04  JSR $040D *
$ØF57  29DF    AND #$DF
$ØF59  AA      TAX
$ØF5A  20F402  JSR $02F4 *
$ØF5D  E054    CPX #$54     Type ?
$ØF5F  F007    BEQ $0F68
$ØF61  E04D    CPX #$4D     Move ?
$ØF63  D00C    BNE $0F71
$ØF65  4C3A0B  JMP $0B3A *
$ØF68  208A0A  JSR $0A80 *
$ØF6B  203203  JSR $0332 *
$ØF6E  4CAD0B  JMP $0BAD *
$ØF71  E05A    CPX #$5A     Zap ?
$ØF73  D00C    BNE $0F81
$ØF75  203C03  JSR $033C *
$ØF78  204503  JSR $0345 *
$ØF7B  203203  JSR $0332 *
$ØF7E  4C0000  JMP $0000
$ØF81  E04C    CPX #$4C     L/Edit ?
$ØF83  D003    BNE $0F88
$ØF85  4C9C0B  JMP $0B9C *
$ØF88  E047    CPX #$47     G/Edit ?
$ØF8A  D003    BNE $0F8F
$ØF8C  4C030E  JMP $0E03 *
$ØF8F  E042    CPX #$42     B/View ?
$ØF91  D003    BNE $0F96
$ØF93  4C700D  JMP $0D70 *
$ØF96  E41F    CPX $1F      R/Tape ?
$ØF98  D003    BNE $0F9D
$ØF9A  4CB20E  JMP $0EB2 *
$ØF9D  A929    LDA #$29     ----
$ØF9F  8D2906  STA $0629 *
$ØFA2  A980    LDA #$80     Switch to new
$ØFA4  8D3906  STA $0639 *   video routines
$ØFA7  A969    LDA #$69     C1S - only
$ØFA9  8D1A02  STA $021A
$ØFAC  A9FF    LDA #$FF     See article 1
$ØFAE  8D1B02  STA $021B
$ØFB1  A920    LDA #$20
$ØFB3  8D2A02  STA $022A    ----
$ØFB6  E057    CPX #$57     W/Tape ?
$ØFB8  D003    BNE $0FBD
$ØFBA  20F30E  JSR $0EF3 *
$ØFBD  E050    CPX #$50     Print ?
$ØFBF  D003    BNE $0FC4
$ØFC1  4C9807  JMP $0798 *
$ØFC4  A91C    LDA #$1C  &  Increase line
$ØFC6  8533    STA $33   &  length
$ØFC8  A900    LDA #$00  &  ----
$ØFCA  8D2202  STA $0222 &  Increase screen
$ØFCD  A91E    LDA #$1E  &  Width - C1S
$ØFCF  8D2303  STA $0223 &
       A91F    LDA #$1F  &  Screen width - C1E
       8D2202  STA $0222 &  ----
$ØFD2  E056    CPX #$56     View ?
$ØFD4  D003    BNE $0FD9
$ØFD6  4C9507  JMP $0795 *
$ØFD9  4C6504  JMP $0465 *
$ØFDC  5D00
$ØFDE  40      Starting text location
```

2. The cold start code will have to be modified to use the new starting text location from above. The second article in this series discussed this. See that article.

Continued on page 4

published monthly

Editor - Al Peabody
Technical Editor - Brian Hartson
Circulation & Advertising Mgr. - Karin Q. Gieske
Production Dept. - A. Fusselbaugh, Ginny Mays

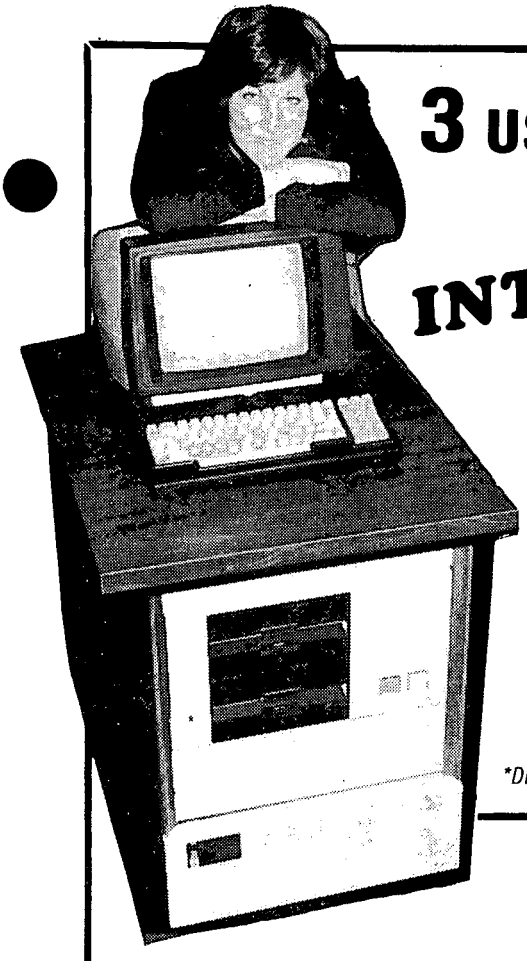
Subscription Rates

US (surface)	$15
Canada & Mexico (1st class)	$23
So. & Cen. America (Air)	$35
Europe (Air)	$35
Other Foreign (Air)	$40

All subscriptions are for 1 year and are payable in advance in US Dollars.

For back issues, subscriptions, change of address or other information, write to:

PEEK (65)
P.O. Box 347
Owings Mills, MD 21117



3 USERS-80 Mega Bytes — $9990.00*

INTRODUCTORY SPECIAL

WITH DUAL FLOPPIES
BRAND NEW —
1 YEAR WARRANTY ON HARD DISK!
REGULAR $12,990.00

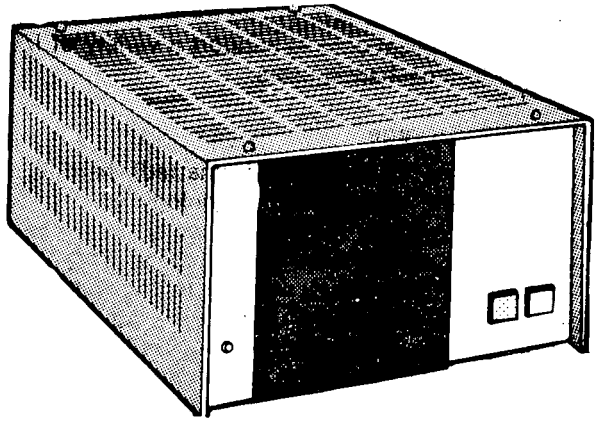
- 90 Days on Power Supply, Floppy Drives — Circuit Boards.
- Configured for Time-Share @ 2 MHZ
- Includes: 2 Serial Printer Ports with Handshake, Improved Cooling, and Ball Bearing Roller Chassis Rails

ALSO AVAILABLE WITH 3 MULTI-PROCESSOR

Denver Boards with 64K each user and Centronics Parallel Printer Port at **$10,990.00**

*DEALER DISCOUNTS AVAILABLE

8" HARD DISK SYSTEMS



SINGLE BOX TABLE TOP WITH IMPROVED COOLING 10 M/B HARD DISK AND 8" FLOPPY DISK **2 USERS AND 2 SERIAL PRINTER PORTS**
$5990.00

AS ABOVE WITH 2 MULTI-PROCESSOR 64K DENVER BOARDS PLUS CENTRONIC PARALLEL INTERFACE **$6990.00**

OR INSTALLED IN CABINET AS ABOVE WITH DUAL FLOPPIES PLUS 10 M/B.

STD. BOARD TYPE	1 USER w/ Centronics Printer Port	$6490.00
	2 USER w/ 2 Serial Printer Ports	$6990.00
DBI MULTI PROC.	2 USER w/Centronics Printer Port	$7790.00
	3 USER w/Centronics & Serial Printer Ports	$8990.00

MULTI-PROCESSOR DEVELOPMENT SYSTEM SPECIAL

SPECIAL
ONLY $5990.00

- 5 M/B Hard Disk-1 8" Floppy
- 1 Centronics Parallel Printer Port
- 1 Serial Printer Port, 1 Modem Port
- 2 DB-1 Multi-Processors
- Complete Programmer Manual and Software Overlays

DEALERS — We have lots of OSI machines and can build virtually any combination you need. Appropriate dealer discounts.

Please Give Us a Call!

WHERE WE STILL LOVE OS-65U — AND SUPPORT IT!

Space-Com International

22991 LA CADENA DRIVE, LAGUNA HILLS, CALIFORNIA 92653

ORDER TODAY (714) **951-4648** SOME QUANTITIES LIMITED

3. WP-6502 ignores leading blanks on new lines. This is inconvenient when working with young children and some text. The following modification allows leading blanks:

$083C F06E BEQ $08AC

4. Because of the video modification I made, I am able to see more lines on my screen. To increase the number of lines displayed on the screen during View, change the value in $0235. The standard value is $18.

5. The page numbering WP-6502 uses is not what I prefer. I would like one page number per page, centered at the bottom of the page. The following modification will change the page numbering to my preference. This code is all new instructions written at the non-relocated addresses indicated.

```
$0815  20E809  JSR $09E8 *&
$0824  4CCE09  JMP $09CE *&
$08CA  4CCE09  JMP $09CE *&
$0977  20A409  JSR $09A4 *&
$099B  20E509  JSR $09E5 *&
$09A4  A532    LDA $32    & ----
$09A6  38      SEC        & Space
$09A7  E51B    SBC $1B    & to
$09A9  E538    SBC $38    & bottom
$09AB  6900    ADC #$00   & of
$09AD  3006    BMI $09B5  & page.
$09AF  F004    BEQ $09B5  &
$09B1  A8      TAY        &
$09B2  20E802  JSR $02E8 *& Print
$09B5  20F402  JSR $02F4 *& page
$09B8  A227    LDX #$27   & number
$09BA  201E04  JSR $041E *& in
$09BD  A64D    LDX $4D    & center
$09BF  F005    BEQ $09C6  &
$09C1  207103  JSR $0371 *&
$09C4  E64D    INC $4D    &
$09C6  20CE09  JSR $09CE *&
$09C9  A007    LDY #$07   &
$09CB  4CE802  JMP $02E8 *& ----
$09CE  204702  JSR $0247 *&
$09D1  C90D    CMP #$0D   & CR ?
$09D3  F0A9    BEQ $097E  &
$09D5  60      RTS        & ----
$09D6  A547    LDA $47    &
$09D8  C902    CMP #$02   & AP Style ?
$09DA  F0BC    BEQ $0998  &
$09DC  20A409  JSR $09A4 *&
$09DF  20E809  JSR $09E8 *&
$09E2  4C1B08  JMP $081B *&
$09E5  20CE09  JSR $09CE *&
$09E8  A900    LDA #$00   &
$09EA  851B    STA $1B    &
$09EC  4CE102  JMP $02E1 *&
$09EF  EA EA EA EA
$09F3  EA EA EA EA
$09F7  EA EA EA EA EA
```

6. The change to the warm start code in step 1 above will cause an insert at the end of the text to fail. The following change should be applied:

$0C46 4C680F JMP $0F68 *

See step 1 above to find the proper instruction to jump to.

This should complete this series on WP-6502. I believe WP-6502 will now work with C1S/C2S and C1E/C2E monitor ROMs. I have also made a few additions to WP-6502 which I hope you will find useful and enjoy. I am currently working on a version of WP-6502 which will run under HEXDOS.

DBPACK REVIEW

By: Fred Schaeffer
84-55 Daniels St. #4f
Jamaica, NY 11435

DBPACK by Compu-Draw Software, NY. FORMAT: IBM3740 CP/M. Sold to me by: Associated/ Consultant Services, NY.

I mentioned the name of the dealer who sold me this relational database package because he (with Compu-Draw's authorization)"translated" the package into OSI format. Compu-Draw apparently doesn't have the OSI equipment to do that.

What sets DBPACK apart from the rest? SIMPLICITY and good documentation and on-line help facilities! How does it compare with OS-DMS? It DOESN'T because it's a completely different set-up; I think it's superior.

Though I haven't got dBASEII (by Ashton-Tate), I've read parts of the documentation. From that I can deduce that DBPACK is somewhat similar but far easier to use. Creating a file is indeed very similar. One invokes DBPAC (part of the DBPACK system), and invokes the command DEFINE. Based on the prompts you then define your fields, field lengths, and Left or Right Justification. These tasks are fast. Additional fields can be added after the fact (but it pays to read the documentation). To enter data in a new file (or add additional records) you invoke APPEND, but there are other ways to do this too, i.e. by preselected fields, repeating strings similar to those in other records and so on. At any time, a variety of items can be displayed, with or without record numbers, selected fields and their content; numerical data can be formatted (similar to PRINT USING), records can be counted to certain criteria, some basic statistics can be done on the records, and also records can be tagged according to certain criteria. That's a very interesting feature. Suppose you wanted to show those records which fitted a certain criteria, (e.g. contributors who gave more than $1000), you can do this by tagging the records with a selected code letter. By invoking the command LISTOPT you can specify which records with which tags you wish to display. The DBPRINT module allows you to print. You are able to change header labels, but to my knowledge page numbering is not possible (a minor inconvenience). Double/Triple spacing is, so is using a separation character between fields, using record numbers or suppressing them, and using either form feed or manual feed (currently, I'm using off-size 13 inch paper and it allows one to format for that). Also, printouts can be enhanced. By that I mean, the system supports various printers that are capable of different print sizes or bold print, and PARTS of the records can be selectively enhanced. The DBFORM function is similar to

Continued on page 9

BEXECK Jan 84 p 16

HOOKS INTO BASIC V1.8

BY: RICK TRETHEWEY

LISTING CONTINUED FROM LAST MONTH

```
        10 ;
        20 ; SAVE COMMAND   B9C7
        30 ;
5450    40 SAVIT JSR CLRLST     CLEAR USED TRACK LIST
        50       STA TEMP1      SHOW "SAVE"
        60 SAVI1 LDA SRCSTR+5   CHECK FILE TYPE
        70       BNE SAVI2      ASM FILE? ==> SAVI2
        80       JSR COPYER     BASIC! UPDATE FILE PARAMS
        90 SAVI2 JSR GETIN      GET FILE NAME FROM BASIC
5500   100       JSR SWAP       * DOS CONTEXT *
       110       JSR CRLF       DO CLEAN UP <CR><LF>
       120       JSR SORT       MARK USED TRACKS IN LIST
       130       LDA SRCSTR+4   GET NEEDED FILE LENGTH
       140       STA TEMP       SAVE IT
       150       JSR CHKNAM     SEE IF NAME IS IN DIRECTORY
       160       BCC FND        NAME NOT IN DIR? ==> FND
       170       JMP WRITF      YES! OLD FILE! ==> SAVE IT!
       180 FND   LDY #$00       INIZ LIST POINTER
       190       STY COUNT      INIZ FREE TRACK COUNTER
5600   200       LDA FNDNUM+1   GET FLOPPY SIZE
       210       JSR BCDH       MAKE IT HEX
       220       STA MAXVAL     SAVE MAX. TRACK #
       230 FN1   STY STTK       SAVE 1ST FREE TRACK #
       240 FN2   LDA LIST,Y     LOOK AT LIST
       250       BNE FN3        USED TRACK? => FN3
       260       INC COUNT      FREE TRACK! BUMP TRACK COUNTER
       270       LDA COUNT      FETCH COUNTER
       280       CMP TEMP       SEEN ENOUGH ROOM FOR FILE?
       290       BEQ NEWSAV     YES! ==> NEWSAV
5700   300       BNE FN4        NO! ==> FN4
       310 FN3   LDA #$00       INIZ ACC.
       320       STA COUNT      INIZ TRACK COUNTER
       330 FN4   CPY MAXVAL     AT END OF DISK?
       340       BEQ FN5        YES! SHOW ERROR ==> FN5
       350       INY            BUMP LIST POINTER
       360       TAX            CHECK ACC. CONTENTS
       370       BEQ FN1        COUNTER RESET! ==> FN1
       380       BNE FN2        MAINTAIN COUNTER & LOOP ==> FN2
5800   390 FN5   JSR STROUT     TELL USER WHAT WENT WRONG
       400       .BYTE 'NO ROOM ON THIS DISK',$0
       410       JMP ERRQ       AND EXIT THROUGH BASIC ERROR
       420 ;
5830   430NEWSAV TYA            MOVE ENDTK POINTER TO ACC.
       440       JSR HBCD       MAKE IT BCD
       450       STA ENDTK      SAVE IT FOR DIRECTORY ENTRY
       460       STA MAXVAL     GIVE IT TO 65D FOR SAVE
       470       LDA STTK       GET STARTING TRACK
       480       STA TRAKX      GIVE IT TO 65D FOR SEEKX
       490       JSR SEEKX      MOVE HEAD TO 1ST TRACK
5900   500       LDA STTK       GET STTK AGAIN
       510       JSR HBCD       ALSO MAKE IT BCD
       520       STA STTK       PUT IT BACK FOR DIR. ENTRY
       530       LDX TEMP1      CHECK IF "SAVE" OR "MAKE"
       540       BNE NEWS0      MAKE? ==> NEWS0
       550       JSR FILSAV     SAVE! SAVE FILE ON DISK
       560 NEWS0 LDA #$01       INIZ
       570       STA SECT       SET TO 1ST DIRECTORY SECTOR
       580 NEWS1 JSR DIRIN      READ IN DIRECTORY
       590       LDY #$00       INIZ DIRBUF POINTER
6000   600 NEWS2 LDA DIRBUF,Y   FETCH CHARACTER FROM DIR.
       610       CMP #'#        EMPTY SPOT ?
       620       BEQ NEWS4      YES! ==> NEWS4
       630       TYA            NO, PUT PTR. IN ACC.
       640       AND #$F8       MASK OFF LOW 3 BITS
       650       CLC            SET UP FOR ADD
       660       ADC #$08       ADD 8 TO POINT TO NEXT ENTRY
       670       TAY            PUT RESULT BACK IN Y
       680       BNE NEWS2      LOOP 'TIL PAGE END
       690 NEWS3 INC SECT       BUMP DIR. SECTOR #
6100   700       LDA SECT       AND FETCH IT
       710       CMP #$03       DONE 1 & 2 ?
       720       BNE NEWS1      NOT YET ==> NEWS1
       730       JSR STROUT     YES! TELL USER NO ROOM IN DIR
       740       .BYTE 'DIRECTORY FULL',$0
       750       JMP ERRQ       AND GO THROUGH BASIC ERROR
       760 ;
6170   770 NEWS4 LDX #$00       INIZ FILE NAME INDEX
       780 NEWS5 LDA INBUF,X    FETCH CHARACTER FROM INBUF
       790       STA DIRBUF,Y   SAVE IN DIRBUF
6200   800       INY            BUMP DIRBUF INDEX
       810       INX            BUMP FILE NAME INDEX
       820       CPX #$06       DONE FILE NAME?
       830       BNE NEWS5      NO! ==> NEWS5
       840       LDA STTK       YES! GET START TRACK
       850       STA DIRBUF,Y   PUT IN DIRBUF
       860       INY            BUMP DIRBUF INDEX AGAIN
       870       LDA ENDTK      GET END TRACK
       880       STA DIRBUF,Y   PUT IT IN DIRBUF AS WELL
       890       JSR WRITE+3    WRITE OUT DIRECTORY SECTOR
6300   900       JSR STROUT     TELL USER ALL O.K.
       910       .BYTE 'SAVED:',$0
       920       LDY #$00       INIZ NAME INDEX
       930 NEWS6 LDA INBUF,Y    FETCH CHARACTER OF NAME
       940       JSR CUTCH      PRINT IT (65D!)
       950       INY            BUMP NAME INDEX
       960       CPY #$06       END OF NAME?
       970       BNE NEWS6      NO! ==> NEWS6
       980       JSR CRLF       DO CLEAN UP <CR><LF>
       990       JMP CUT        AND GO BACK TO BASIC (QUIT)
6400  1000 ;
      1010 WRITF LDA DIRBUF,Y   FETCH STARTING TRACK
      1020       JSR BCDH       MAKE IT HEX
      1030       STA STTK       SAVE IN STTK
      1040       INY            BUMP DIRBUF POINTER
      1050       LDA DIRBUF,Y   FETCH END TRACK
      1060       STA MAXVAL     GIVE IT TO 65D
      1070       JSR BCDH       MAKE IT HEX
      1080       SEC
      1090       SBC STTK       SUBTRACT STARTING TRACK
6500  1100       TAY            PUT IN Y
      1110       INY            +1 TO GET CURRENT FILE LENGTH
      1120       CPY SRCSTR+4   COMPARE TO NEEDED SPACE
      1130       BCS WRTF1      O.K.! ==> WRTF1
      1140       JSR STROUT     TOO SMALL! TELL USER!
      1150       .BYTE 'FILE TOO SMALL',$0
      1160       JMP ERRQ       AND EXIT THROUGH BASIC ERROR
      1170 WRTF1 LDA STTK       GET START TRACK
      1180       JSR HBCD       MAKE IT BCD AGAIN
      1190       JSR FILSAV     SAVE FILE ON DISK
6600  1200       JSR STROUT     TELL USER ALL IS WELL
      1210       .BYTE 'FILE UPDATED',$D,$A,$0
      1220 CUT   JSR SWAP       * LANGUAGE CONTEXT *
      1230       PLA            CANCEL JSR TO HOOKS
      1240       PLA
      1250       RTS            AND QUIT
      1260 ;
      1270 ; CLEAR USED TRACK LIST
      1280 ;
      1290 CLRLST LDY #$4C      GET # OF TRACKS
6700  1300       LDA #$00       INIZ
      1310 CLRL1  STA LIST,Y    CLEAR LIST
      1320       DEY            DECREMENT POINTER
      1330       BPL CLRL1      LOOP 'TIL DONE
      1340       RTS            AND QUIT
      1350 ;
      1360 ; UPDATE TXTPTR PAST KEYWORD
      1370 ;
      1380 ADD4   LDA #$04      INIZ
      1390       JMP ADDON+1    ADD 4 TO TXTPTR
6800  1400 ;
      1410 ; GET FILE NAME FROM BASIC
      1420 ;
      1430 GETIN  LDA #$03      INIZ
      1440       JSR ADDON+1    MOVE PAST KEYWORD
      1450 GIN0   JSR CHRGET    FETCH NEXT CHARACTER
      1460       JSR FRMEVL     EVALUATE IT
      1470       JSR CHKSTR     MAKE SURE IT'S A STRING
      1480       JSR FREFAC     FIND IT
      1490       CMP #$07       LENGTH > 7 ?
6900  1500       BCC GIN1       NO, O.K. ==> GIN1
      1510       JMP SNERR      TOO LONG! SYNTAX ERROR!
      1520 GIN1   STX GIN2+1    SAVE STRING ADDRESS LSB
      1530       STY GIN2+2     AND MSB
      1540       STA GIN3+1     AND LENGTH
      1550       LDY #$00       INIZ POINTER
6960  1560 GIN2   LDA $FFFF,Y   FETCH CHARACTER OF STRING
```

Continued on page 6

```
6970 1570        STA INBUF,Y    SAVE IT IN INBUF
     1580        INY            BUMP POINTER
     1590 GTN3   CPY #$FF       AT END OF STRING?
7000 1600        BNE GTN2       NO ==> GTN2
     1610        LDA #'         LOAD A <SP>
     1620 GTN4   CPY #$06       FILLED 6 CHARS. IN INBUF?
     1630        BEQ GTN5       YES! ==> GTN5
     1640        STA INBUF,Y    NO, SAVE <SP> IN INBUF
     1650        INY            BUMP POINTER
     1660        BNE GTN4       AND LOOP
     1670 GTN5   RTS            AND QUIT
     1680 ;
     1690 ; SEE IF FILE NAME IN "INBUF" EXISTS IN DIR.
7090 1700 ; RETURNS WITH CARRY SET IF "YES" - CLEAR IF "NO"
     1710 ;
7120-1720 CHKNAM LDA #$01       INIZ
     1730        STA SECT       START WITH SECTOR 1
     1740 CHKN1  JSR DIRIN      READ IN DIRECTORY SECTOR
     1750        LDX #$00       INIZ
     1760        LDY #$00       INIZ
     1770 CHKN2  LDA DIRBUF,Y   FETCH CHARACTER FROM DIR
     1780        JSR CASECK     MAKE IT ALL CAPS
     1790        STA CHKN0+1    SAVE IT
7200-1800        LDA INBUF,X    FETCH CHARACTER FROM INBUF
     1810        JSR CASECK     MAKE IT ALL CAPS
     1820 CHKN0  CMP #$FF       COMPARE WITH DIR. ENTRY
     1830        BNE CHKN3      NO MATCH? ==> CHKN3
     1840        INX            YES! BUMP MATCH COUNTER
     1850        INY            BUMP ENTRY INDEX
     1860        CPX #$06       MATCHED ENTIRE NAME?
     1870        BNE CHKN2      NO! LOOP! ==> CHKN2
     1880        SEC            SHOW MATCH!
     1890        RTS            QUIT (Y POINTS TO STIK)
7300 1900 CHKN3  LDX #$00       RESET MATCH COUNTER
     1910        TYA            PUT Y IN ACC.
     1920        AND #$F8       MASK TO # OF 8'S
     1930        CLC
     1940        ADC #$08       ADD ENTRY LENGTH
     1950        TAY            PUT RESULT BACK IN Y
     1960        BNE CHKN2      LOOP 'TIL DONE
     1970        INC SECT       BUMP SECTOR #
     1980        LDA SECT       FETCH IT
     1990        CMP #$03       DONE BOTH SECTORS?
7400 2000        BNE CHKN1      NO! LOOP!
     2010        CLC            YES! SHOW NO MATCH
     2020        RTS            AND QUIT
     2030 ;
     2040 ; MAKE FILE COMMAND
     2050 ; COMMAND SYNTAX IS -  MAKE"FNAME",NT
     2060 ;
     2070 MAKER  JSR GETIN      GET NAME FROM BASIC
     2080        JSR CHKCOM     FIND COMMA TERMINATOR
     2090        JSR GETBYT     EVALUATE FILE SIZE REQUEST
7500 2100        STX TEMP       SAVE FILE LENGTH
     2110        STX TEMP1      SHOW "MAKE"
     2120        JSR SWAP       * DOS CONTEXT *
     2130        JSR CLRLST     CLEAR USED TRACK LIST
     2140        JSR SORT       MARK USED TRACKS
     2150        JSR CHKNAM     SEE IF NAME IS IN DIRECTORY
     2160        BCC MAK1       NO. O.K. ==> MAK1
     2170 NIUER  JSR STROUT     YES! SAY SO!
     2180        .BYTE 'NAME IN USE',$0
     2190 ERRQ   JSR SWAP       * LANGUAGE CONTEXT *
7600 2200        JMP TYPERR     JUMP TO BASIC'S ERROR REPORT
     2210 MAK1   JMP FND
     2220 ;
     2230 ; RENAME COMMAND
     2240 ; COMMAND SYNTAX - RENAME "OLDNAM" TO "NEWNAM"
     2250 ;
     2260 RENAME JSR CHRGET     ADJUST FOR "RENAME" LENGTH
     2270        JSR CHRGET
     2280        JSR GETIN      GET NAME FROM BASIC
     2290        JSR SWAP       * DOS CONTEXT *
7700 2300        JSR CHKNAM     FIND NAME IN DIRECTORY
     2310        BCS REN1       O.K. ==> REN1
     2320        LDA #$0C       FILE NOT FOUND
     2330        JMP ERROR      USE 65D'S ERROR REPORT
     2340 REN1   TYA            PUT DIRBUF INDEX IN ACC.
     2350        AND #$F8       MASK TO 1ST CHAR OF NAME
     2360        STA REN4+1     SAVE IT
7970 2370        LDA SECT       GET DIRECTORY SECTOR
7780 2380        STA REN3+1     SAVE IT TOO
     2390        JSR SWAP       * LANGUAGE CONTEXT * AGAIN
7800-2400        JSR CHRGOT     RE-FETCH LAST CHAR FROM TEXT
     2410        CMP #TOTK      "TO" ?
     2420        BEQ REN2       YES! O.K. ==> REN2
     2430        JMP SNERR      NO! SYNTAX ERROR
     2440 REN2   JSR GTN0       GET NEW NAME FROM BASIC
     2450        JSR SWAP       * DOS CONTEXT *
     2460        JSR CHKNAM     SEE IF NEW NAME IN USE
     2470        BCC REN3       NO. O.K. ==> REN3
     2480        JMP NIUER      YES! TELL USER & QUIT
     2490 REN3   LDA #$FF       GET RENAME SECTOR
7900 2500        STA SECT       GIVE IT TO 65D
     2510        JSR DIRIN      READ IN DIRECTORY SECTOR
     2520 REN4   LDY #$FF       FETCH NAME POINTER
     2530        LDX #$00       INIZ COUNTER
     2540 REN5   LDA INBUF,X    GET NEW FILE NAME
     2550        STA DIRBUF,Y   SAVE IN DIRECTORY BUFFER
     2560        INY            BUMP DIRBUF POINTER
     2570        INX            BUMP FILE NAME POINTER
     2580        CPX #$06       AT END OF NAME ?
     2590        BNE REN5       NO! LOOP! ==> REN5
8000 2600        JSR WRITE+3    YES! WRITE OUT DIRBUF
     2610        JMP OUT        AND QUIT!
     2620 ;
     2630 ; VIEW COMMAND
     2640 ;
     2650VIEWIT  LDY #$00       INIZ
     2660        LDA VARTAB     GET TOP OF VARIABLE TABLE
     2670        STA VARPTR     GIVE TO VARIABLE POINTER
     2680        LDA VARTAB+1   HANDLE MSB
     2690        STA VARPTR+1
8100 2700 V0     LDX VARPTR+1   GET POINTER MSB
     2710        CPX ARRTAB+1   COMPARE TO ARRAY START
     2720        BNE V1         MSB'S DON'T MATCH? ==>
     2730        LDA VARPTR     YES! CHECK LSB'S
     2740        CMP ARRTAB
     2750        BNE V1         NO! PRINT VARIABLE
     2760        JMP VOUT       YES! QUIT!
     2770 V1     LDY #$00       INIZ
     2780        LDA (VARPTR),Y FETCH VARIABLE DATA
     2790        STA VARNAM     SAVE NAME 1ST CHARACTER
8200-2800        AND #$7F       MASK BIT 7
     2810        JSR OUTCH      PRINT IT
     2820        CMP (VARPTR),Y BIT 7 SET?
     2830        BEQ V2         NO! ==> V2
     2840        INC STRFLG     YES! BUMP FLAG
     2850 V2     INY            BUMP INDEX
     2860        LDA (VARPTR),Y FETCH VARIABLE 2ND CHAR.
     2870        STA VARNAM+1   SAVE IT TOO
     2880        AND #$7F       MASK AGAIN
     2890        JSR OUTCH      PRINT IT
8300 2900        CMP (VARPTR),Y BIT 7 SET?
     2910        BEQ V3         NO! ==> V3
     2920        LDX STRFLG     YES! SEE IF BOTH SET
     2930        BNE V5         YES! ==> V5
     2940        JMP STRING     NO! IT'S A STRING ==>
     2950 V5     LDA #'%        SHOW INTERGER VARIABLE!
     2960        STY STRFLG     SAVE INDEX
     2970        JSR OUTCH      PRINT "%"
     2980        JMP V6         AND SKIP A BIT
     2990 V3     JSR $0F8A      FETCH VARIABLE CONTENTS
8400 3000        JSR GETVAR     PUT IN F.P. ACC.
     3010 V6     JSR STROUT     PRINT "= "
     3020        .BYTE '= ',0
     3030        LDA VARNAM     GET VARIABLE NAME
     3040        BPL V4         FLOATING POINT? ==> V4
     3050        LDY STRFLG     INTERGER! GET STRFLG
     3060        INY            ADD 1
     3070        INY            ADD 1 MORE
     3080        LDA (VARPTR),Y FETCH VARIABLE MSB
     3090        TAX            PUT IN X
8500 3100        DEY            BACK UP 1
     3110        LDA (VARPTR),Y FETCH LSB
     3120        JSR PNUMBR     PRINT # IN A-X PAIR
     3130        JMP NEXT       GO TO NEXT VARIABLE ==>
     3140 V4     JSR ASCII      CONVERT CONTENTS TO ASCII
     3150        JSR BASPRT     PRINT NUMBER
     3160 NEXT   JSR CRLF       DO CLEAN UP <CR><LF>
     3170        LDA VARPTR     GET VARPTR
8980 3180        CLC
```

Continued on page 7

```
3190        ADC #$07        ADD ENTRY LENGTH
3200        STA VARPTR      SAVE RESULT
3210        BCC NX1         HANDLE PAGING
3220        INC VARPTR+1
3230 NX1    LDA #$00        INIZ
3240        STA STRFLG      CLEAR STRFLG
3250        JMP V0          AND LOOP!
3260;
3270STRFLG .BYTE 0
3280;
3290 VOUT   JSR ADD4        MOVE TXTPTR PAST "VIEW"
3300 VOUT1  PLA             CANCEL JSR TO HOOKS
3310        PLA
3320        RTS             AND GO BACK TO BASIC
3330;
3340STRING  STY STRFLG       SAVE POINTER
3350        JSR STROUT      DISPLAY "$= "
3360        .BYTE '$= !',0
3370        LDY STRFLG      GET POINTER BACK
3380        INY             +1
3390        LDA (VARPTR),Y  FETCH VARIABLE DATA
3400        STA STRFLG      SAVE LENGTH OF STRING
3410        BEQ NEXT        NULL? ==> NEXT
3420        DEC STRFLG      SUBTRACT 1
3430        INY             BUMP POINTER
3440        LDA (VARPTR),Y  FETCH ADDRESS LSB
3450        STA VARPNT      SAVE IN POINTER
3460        INY             BUMP INDEX
3470        LDA (VARPTR),Y  FETCH MSB
3480        STA VARPNT+1    SAVE IT TOO
3490        LDY #$00        INIZ
3500 STR1   LDA (VARPNT),Y  FETCH CHARACTER
3510        CMP #'          IS IT ASCII?
3520        BCS STR2        YES! ==> STR2
3530        ADC #'A-1       NO! MAKE IT A-Z
3540        PHA             SAVE ON STACK
3550        LDA #'^         SHOW CONTROL CHARACTER
3560        JSR OUTCH       PRINT CARET
3570        PLA             RETRIEVE LETTER
3580 STR2   JSR OUTCH       PRINT CHARACTER
3590        CPY STRFLG      DONE?
3600        BEQ NEXT        YES! ==> NEXT
3610        INY             NO! BUMP INDEX
3620        BNE STR1        AND LOOP!
3630 ;
3640 ; KILL COMMAND
3650 ; COMMAND SYNTAX IS - KILL"FNAME" (,"FNAM2"...)
3660 ;
3670 KILL   JSR ADD4        MOVE PAST "KILL"
3680        JSR NONUMR      GIVE FILE NAME TO 65D
3690        JSR FNDNAM      FIND NAME IN DIRECTORY
3700        LDA #'#         LOAD "NULL" ENTRY BYTE
3710        INX             BUMP X
3720        LDY #$08        INIZ COUNTER
3730 KILL1  STA DIRBUF,X    ERASE ENTRY
3740        DEX             DECREMENT POINTER
3750        DEY             DECREMENT COUNTER
3760        BNE KILL1       LOOP 'TIL DONE
3770        LDA #DIRBUF     WRITE "DIRBUF" BACK OUT
3780        STA ADRL
3790        LDA #DIRBUF/256
3800        STA ADRH
3810        JSR SAVEM+$12
3820        JSR SWAP        RETURN TO LANGUAGE CONTEXT
3830        JSR CHRGOT      REVIEW LAST CHAR. SEEN
3840        BEQ KILL2
3850        CMP #',         WAS IT A COMMA?
3860        BNE KILL2       NO! ==> KILL2
3870        JSR CHRGET      YES! FETCH NEXT CHARACTER
3880        JMP KILL+3      RESTART CYCLE
3890 KILL2  PLA             RETURN TO BASIC
3900        PLA
3910        RTS
3920 ;
3930 ; USED TRACK LIST
3940 ;
3950 LIST   .BYTE $00
3960        *=*+$4C
3970 ;
3980 ; HOOKS SCRATCHPAD
3990 ;
4000 RESLO  .BYTE $00
4010 RESHI  .BYTE $00
4020 FIFTH  .BYTE $00
4030 STIK   .BYTE $00
4040 ENDTK  .BYTE $00
4050 COUNT  .BYTE $00
4060 POINT  .BYTE $00
4070 TEMP   .BYTE $00
4080 TEMP1  .BYTE $00
4090 ;
4100 ; CALL COMMAND
4110 ; COMMAND SYNTAX IS - CALL ADDR
4120 ;
4130 CALR   JSR ADD4        MOVE PAST "CALL"
4140        JSR CHRGOT      REFETCH CHAR. AT TXTPTR
4150 CALR1  JSR FRMEVL      EVALUATE EXPRESSION
4160        JSR QUINT       MAKE IT AN INTERGER
4170        LDA FACLO       GET ADDRESS LSB
4180        STA JUMPER+1    SET JUMP POINTER
4190        LDA FACMLO      GET ADDRESS MSB
4200        STA JUMPER+2    SAVE IT TOO
4210JUMPER  JSR $FFFF       EXECUTE REQUESTED CODE!
4220        PLA             RETURN TO BASIC
4230        PLA
4240        RTS
4250;
4260; WAIT COMMAND - SYNTAX:  WAIT I,J,K
4270;
4280 WAIT   JSR ADD4        UPDATE TXTPTR
4290        JSR CHRGOT      FETCH 1ST CHAR. IN EXP.
4300        JSR $1666       EVALUATE I & J
4310        STX FORPNT      X = VALUE OF J
4320        LDX #$00
4330        JSR CHRGOT      SEE IF K IS PRESENT
4340        BEQ WAIT1       NO! ==>
4350        JSR $166C       YES! CHKCOM & EVALUATE K
4360 WAIT1  STX FORPNT+1
4370        LDY #$00
4380 WAIT2  LDA (POKER),Y   CHECK ADDRESS "I"
4390        EOR FORPNT+1    EOR WITH K
4400        AND FORPNT      MASK WITH J
4410        BEQ WAIT2       WAIT FOR NON-ZERO RESULT
4420        BNE JUMPER+3    EXIT
4430;
4440 TCOD0  LDY #$00        INIZ
4450        LDA $07DB       CHECK STATUS
4460        CMP #$18        TRACE ENABLED?
4470        BEQ TCOD1       NO! ENABLE! ==>
4480        LDY #$05        YES! GET DISABLE POINTER
4490 TCOD1  LDX #$00        INIZ
4500        LDA TRIBL,Y     FETCH INSTRUCTION
4510        STA $07DB,X     MODIFY BASIC
4520        INY             BUMP FETCH INDEX
4530        INX             BUMP PUT INDEX
4540        CPX #$05        DONE?
4550        BNE TCOD1+2     NO! LOOP!
4560        JMP UPDATE      YES! QUIT
4570;
4580 TRIBL  .BYTE $20,$D8,$1C,$EA,$EA
4590        .BYTE $18,$90,$02,$E6,$C8
4600;
4610        *=$BE1B
4620 ;
4630 ; PATCH TO EVAL TO ALLOW HEXADECIMAL EXPRESSIONS.
4640 ;
4650 HEXEVL CMP #'$         IS IT A HEX NUMBER ?
4660        BEQ HEXFLT      YES! ==> HEXFLT
4670        CMP #'.         NO, REPLICATE OVERWRITTEN CODE
4680        BNE HEX6        AND CONTINUE
4690        JMP ASCFP
4700 HEX6   JMP $0DC7
4710 HEXFLT LDY #$00        INIZ
4720        STY RESLO       CLEAR RESULT LSB
4730        STY RESHI       CLEAR RESULT MSB
4740 HEX1   JSR CHRGET      FETCH CHARACTER FROM TEXT
4750        BEQ HEX3        END OF LINE ? ==> HEX3
4760        CMP #'0         CHECK FOR OTHER LEGAL CHARS.
4770        BCC HEX3
4780        JSR CASECK
4790        CMP #'Z+1
4800        BCS HEX3
```

Continued on page 8

```
10210 4810        CMP #';
      4820        BEQ HEX3
      4830        SEC
      4840        SBC #'0         SUBTRACT ASCII
      4850        CMP #$A         CORRECT FOR HEX A-F
      4860        BCC HEXF2-2
      4870        SBC #$7
      4880        LDY #$00        INIZ
      4890 HEXF2  ASL RESLO       MULTIPLY RESULT BY 16
10300 4900        ROL RESHI
      4910        INY
      4920        CPY #$04
      4930        BNE HEXF2
      4940        CLC
      4950        ADC RESLO       ADD RESULT TO NEXT #
      4960        STA RESLO       SAVE RESULT
      4970        BCC HEX1        HANDLE PAGING
      4980        INC RESHI       BUMP MSB
      4990        BNE HEX1        OVERFLOW? NO! ==> HEX1
10400 5000        JMP SNERR       YES! SHOW ERROR
      5010 HEX3   LDA RESHI       GET MSB
      5020        STA FACHI       GIVE TO F.P. ACC.
      5030        LDX RESLO       GET LSB
      5040        STX FACMHI      SAVE IT TOO
      5050        LDX #$90        SET UP
      5060        SEC
      5070        JSR FLOAT       MAKE NORMAL F.P. #
      5080        LDX RESLO       REFETCH LSB
      5090        RTS             AND QUIT
10500 5100 ;
      5110 ; PATCH TO ALLOW NAMED GOTO'S.
      5120 ;
      5130 NEWGO  BCS LINE        LETTER? ==> LINE
      5140        JMP LINGET      NUMBER! MAKE IT BINARY
      5150 LINE   JSR PTRGET      LOOK UP VARIABLE
      5160        JSR GETVAR      PUT IT IN F.P. ACC.
      5170        JSR QUINT       MAKE IT AN INTERGER
      5180        LDA FACLO       GET LSB
      5190        STA POKER       SAVE FOR "GOTO"
10600 5200        LDA FACMLO      DO MSB TOO
      5210        STA POKER+1
      5220        RTS             AND QUIT
      5230 ;
      5240 ; REPLACEMENT FOR "IF" TO ALLOW LIMITED "ELSE"
      5250 ;
      5260 NEWIF  JSR FRMEVL      EVALUATE EXPRESSION
      5270        JSR CHRGOT      CHECK CHAR. AFTER EXPRESS.
      5280        CMP #GOTOTK     "GOTO"?
      5290        BEQ NEWIF1      YES! ==> NEWIF1
10700 5300        LDA #THENTK     "THEN"?
      5310        JSR CHKCHR      CHECK TEXT
      5320NEWIF1  LDA FACEXP      CHECK TRUE/FALSE
      5330        BEQ FALSE       FALSE ==>
      5340        JMP $0941       TRUE! RETURN TO BASIC
      5350 FALSE  LDY #$00        INIZ
      5360 F1     LDA (TXTPTR),Y  CHECK REST OF LINE
      5370        BEQ NOREM       E.O.L.? ==> NOREM
      5380        CMP #REMTK      NO, "REM"?
      5390        BEQ F2          YES! ELSE! ==> F2
10800 5400        INY             NO! BUMP POINTER
      5410        BNE F1          AND LOOP!
      5420 F2     JSR ADDON       UPDATE TXTPTR
      5430        JSR CHRGET      FETCH 1ST CHARCTER
      5440        JMP GOTO        TREAT AS "GOTO"
      5450 NOREM  JMP REM
      5460 ;
      5470SETADR  LDA #$00        INIZ
      5480        STA ADRLX       SET TO PAGE START
      5490        LDA STRFLG      GET MSB FROM "PACK"
10900 5500        STA ADRHX       GIVE TO 65D
      5510        RTS             AND QUIT
      5520 ;
      5530 WRITE  JSR SETADR      SET ADDRESS
      5540        JSR LOAD        LOAD HEAD
      5550        JSR SAVEX       WRITE OUT SECTOR
      5560        JMP UNLOAD      UNLOAD HEAD & QUIT
      5570 ;
      5580 READ   JSR SETADR      SET ADDRESS
      5590        JSR LOAD        LOAD HEAD
      5600        JSR CALLX       READ SECTOR
11000 5610        JMP UNLOAD      UNLOAD HEAD & QUIT

11020 5620 ;
      5630 BCDH   PHA             SAVE ORIGINAL ON STACK
      5640        AND #$F0        MASK TO HIGH NYBBLE
      5650        LSR A           SHIFT RIGHT 4 BITS
      5660        LSR A
      5670        LSR A
      5680        LSR A
      5690        TAX             SAVE AS COUNTER
11100 5700        LDA #$00        INIZ
      5710 BC1    CLC
      5720        ADC #$A         ADD 10
      5730        DEX             DECREMENT COUNTER
      5740        BNE BC1         LOOP 'TIL DONE
      5750        STA TMP         SAVE RESULT
      5760        PLA             RETRIEVE ORIGINAL VALUE
      5770        AND #$F         MASK TO LOW NYBBLE
      5780        CLC
      5790        ADC TMP         ADD 10'S
11200 5800        RTS             AND QUIT
      5810 ;
      5820 HBCD   LDX #$FF        INIZ
      5830        SEC             SET UP FOR SUBTRACT
      5840 HB1    INX             BUMP COUNTER
      5850        SBC #$A         SUBTRACT 10
      5860        BCS HB1         LOOP 'TIL RESULT GOES NEG.
      5870        ADC #$A         RESTORE LAST SUBTRACTION
      5880        STA TMP         SAVE REMAINDER
      5890        TXA             PUT COUNTER IN ACC.
11300 5900        ASL A           SHIFT LEFT 4 BITS
      5910        ASL A
      5920        ASL A
      5930        ASL A
      5940        CLC
      5950        ADC TMP         ADD REMAINDER BACK IN
      5960        RTS             AND QUIT
      5970 ;
      5980 ; COUNT SECTORS ON CURRENT TRACK
      5990 ; RESULT IN "FIFTH"
11400 6000 ;
      6010 CNTS   LDA #$FF        INIZ
      6020        STA FIFTH       INIZ SECTOR COUNT
      6030        LDA #$01        INIZ
      6040        STA SECT        START WITH SECTOR 1
      6050        JSR LOAD        LOAD HEAD
      6060        JSR FIND        FIND SECTOR 1
      6070 CNT1   INC FIFTH       BUMP COUNTER
      6080        JSR DUMRED      ATTEMPT DUMMY READ
      6090        BCS CNT1        GOOD! LOOP! ==>
11500 6100        JMP UNLOAD      DONE WITH TRACK! QUIT!
      6110;
      6120 ; PATCH TO PRINT COMMAND TO ALLOW HEX OUTPUT.
      6130 ; COMMAND SYNTAX - PRINT $,VARNAM
      6140 ; NOTE: CURRENT VERSION NOW HAS 32 BIT ACCURACY
      6150 ;
      6160        CMP #'$         HEX PRINT?
      6170        BEQ HPRINT      YES! ==>
      6180        JMP FRMEVL      NO! EVALUATE EXPRESSION!
      6190HPRINT  LDY #$01        INIZ INDEX
11600 6200        LDA (TXTPTR),Y  LOOK PAST "$"
      6210        CMP #',         FIND A COMMA?
      6220        BEQ HPR3        YES! ==>
      6230        LDA #'$         NO! RESTORE "$" IN ACC.
      6240        BNE HPRINT-3    AND RETURN
      6250 HPR3   JSR CHRGET      THROW AWAY COMMA
      6260        JSR CHRGET      FETCH NEXT CHARACTER
      6270        JSR FRMEVL      EVALUATE EXPRESSION
      6280        LDA #$00        INIZ
      6290        STA STTK        CLEAR OUTPUT FLAG
11700 6300        BIT FACSGN      NEGATIVE VALUE?
      6310        BPL HPR1        NO! ==>
      6320        LDA #'-         YES! FETCH "-"
      6330        JSR OUTDO       PRINT IT
      6340        LSR FACSGN      MAKE VALUE POSITIVE NOW
      6350 HPR1   JSR QUINT       MAKE IT AN INTERGER
      6360        LDA #'$         SHOW IT'S HEX
      6370        JSR OUTDO       PRINT "$"
      6380        LDA FACHI       FETCH RESULT MSB
      6390        JSR HOUT        PRINT IT (MAYBE)
      6400        LDA FACMHI      FETCH RESULT NMSB
11810 6410        JSR HOUT        PRINT IT
```

Continued on page 9

```
11820 6420        LDA FACMLO      FETCH RESULT NLSB
      6430        JSR HOUT        PRINT IT
      6440        LDA FACLO       FETCH RESULT LSB
      6450        JSR HOUT        PRINT IT
      6460        LDA STIK        SEE IF ANY OUTPUT
      6470        BNE HPR2        YES! ==> HPR2
      6480        LDA #'0         NO! GET A "0"
      6490        JSR OUTDO       AND PRINT IT
11900 6500 HPR2   PLA             CANCEL A JSR
      6510        PLA
      6520        JMP $0A32       DO CHRGOT & CONTINUE
      6530;
      6540 HOUT   PHA             SAVE ON STACK
      6550        LSR A           SHIFT RIGHT 4 BITS
      6560        LSR A
      6570        LSR A
      6580        LSR A
      6590        JSR HO          DISPLAY # (MAYBE)
12000 6600        PLA             RETRIEVE ORIGINAL VALUE
      6610        AND #$0F        MASK TO LOW NYBBLE
      6620 HO     BNE HO1         NON-ZERO? ==> HO1
      6630        LDX STIK        FETCH OUTPUT FLAG
      6640        BNE HO1         DONE ANY YET? YES! ==> HO1
      6650        RTS             NO, SUPPRESS LEADING ZERO
      6660 HO1    INC STIK        SHOW OUTPUT!
      6670        CLC
      6680        CMP #$A         CORRECT FOR HEX
      6690        BCC HO2
12100 6700        ADC #$06
      6710 HO2    ADC #'0         MAKE NUMBER ASCII
      6720        JMP OUTDO       PRINT IT AND QUIT
      6730;
      6740 INBUF .BYTE '          '
      6750;
      6760 STBLK  .BYTE $00
      6770 ENBLK  .BYTE $00
      6780 GAPLEN .BYTE $00
      6790 STGAP  .BYTE $00
12200 6800;
      6810 FILGET JSR GETN        GET NAME FROM BASIC
      6820        JSR SWAP        * DOS CONTEXT *
      6830        JSR CHKNAM      FIND NAME IN DIRECTORY
      6840        BCS FILG1       YES! ==> FILG1
      6850        LDA #$FF        NO! SHOW "NOT FOUND"
      6860        STA STIK
      6870        STA ENDIK
      6880        BMI FILG2       AND SKIP A BIT
      6890 FILG1  LDA DIRBUF,Y    GET STARTING TRACK #
12300 6900        JSR BCDH        MAKE IT HEX
      6910        STA STIK        GIVE IT TO HOOKS
      6920        INY             BUMP POINTER
      6930        LDA DIRBUF,Y    GET ENDING TRACK #
      6940        JSR BCDH        MAKE IT HEX TOO
      6950        STA ENDIK       GIVE IT TO HOOKS
      6960 FILG2  JSR SWAP        * LANGUAGE CONTEXT *
      6970        JSR CHKCOM      FIND COMMA TERMINATOR
      6980        JSR PTRGET      FIND VARIABLE
      6990        STA FORPNT      GIVE POINTERS TO BASIC
12400 7000        STY FORPNT+1
      7010        JSR CHKTYP      MAKE SURE ITS NUMERIC
      7020        LDA STIK        USE STIK AS MSB
      7030        LDY ENDIK       USE ENDIK AS LSB
      7040        JSR GIVAYF      GIVE VALUE TO VARIABLE
      7050        PLA             CANCEL JSR TO HERE
      7060        PLA
      7070        JMP $1ACB       GIVE VALUE TO VAR. & QUIT
      7080;
12790 7090        .END HOOK1
```

★

DBPRINT, except that it does forms, labels, index cards, etc., concatenate fields for labels, etc. To create a format file, you need a text editor; this is an inconvenience for those who don't have one and actually a plus for those that do. Continued on page 10

The DBSORT module sorts very rapidly and sorting can be done on several fields or parts of them. DBCLEAN is a utility to get rid of deleted records. All in all, this is a very good package and amazingly easy to use. As mentioned, the documentation is good, it includes lots of on-line help and a sample case history both on paper and in demo files.

★

EXPANSION FOR 6502 COMPUTERS

Carlos M. Cortes
Belzu 3048
1636 Olivos
Argentina

Summary: Simple expansion motherboard that provides four additional slots, two 8K and two 2K wide. It uses only two TTL IC's, and is designed for an OSI Superboard, but adaptable to other 6502 systems.

Requires: Superboard or 6502-based computer & hardware components

This expansion board was designed with simplicity and low cost as objectives. It consists in a motherboard which supports up to six connectors for different cards, an input cable which plugs directly in the Superboard 40 pin expansion socket and a pair of TTL chips which do all the decoding.

I am currently using this expansion to support cards like: one with 8K of additional RAM which takes my system from its original 8K to 16K, an EPROM programmer for 2716/2732 chips and a general purpose I/O board adapted to output control signals to drive a stepper motor and to input data from a variety of sensors. Next project is an A/D (Analog to Digital) board to make the computer able to work with analog signals.

It was designed for an OSI Superboard but it can be adapted to most 6502-based computers. As described here, only four of the six slots are populated, covering a total of approximately 20K of the memory map.

The circuit:

Table 1 summarizes the memory locations used up by this expansion when used with a C1P or Superboard. The schematic for the motherboard is shown on figure 1. IC1 is a 74LS139

Table 1 : Slot decoding for the expansion

SLOT	Address(hex)	Address (dec)	Size
Superboard RAM	$0000-$1FFF	0-8191	(8K)
J5	$2000-$3FFF	8192-16383	8K
J6	$4000-$5FFF	16384-24575	8K
J7	$6000-$67FF	24576-26623	2K
J8	$6800-$6FFF	26624-28671	2K
(J9)*	$7000-$77FF	28673-30719	(2K)
(J10)*	$7800-$7FFF	30720-32767	(2K)

* optional

Figure 1

Schematic diagram for the expansion board. Device select lines S9 and S10 can be used to add two additional 2K slots.

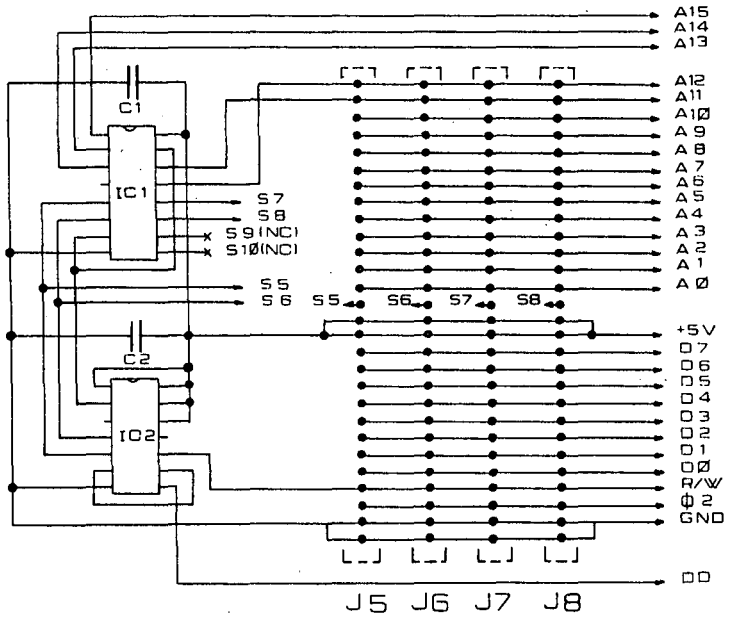
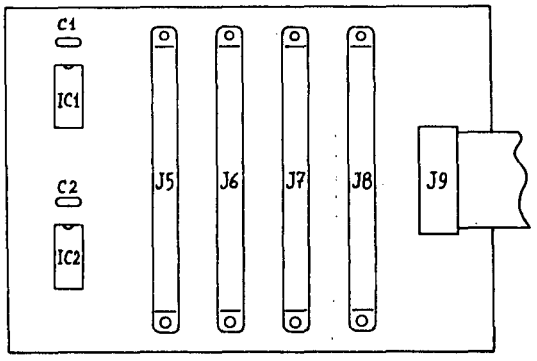


Figure 2

Component layout of the motherboard. The DIP jumper connects the output socket J9 with the expansion socket on the superboard.



D&N MICRO PRODUCTS, INC.

3702 N. Wells St.
Fort Wayne, Ind. 46808
(219) 484-6414

TERMS $3.00 shipping. Foreign orders add 15%, Indiana residents add 5% sales tax.

COMPUTER

MICRO-80 COMPUTER

Z-80A CPU with 4Mhz clock and CP/M 2.2 operating system. 64K low power static memory. Centronics parallel printer port. 3 serial ports. 4" cooling fan. Two 8" single or double sided floppy disk drives. IBM single density 3740 format for 243K or storage, double density format for 604K of storage. Double sided drives allow 1.2 meg on each drive. Satin finish extruded aluminum with vinyl woodgrain decorative finish. 8 slot backplane, 48 pin buss compatible with OSI boards.

MODEL 80-1200 **$2995**
2 8" Single sided drives

MODEL 80-2400 **$3495**
2 8" Double sided drives

MICRO-65 COMPUTER

6502 CPU with 2Mhz clock and DOS-65 operating system. 48K of low power static memory. 2 serial ports and 1 Centronics parallel port. 2 8" single or double sided drives. Satin finish extruded aluminum with vinyl woodgrain finish. 8 slot backplane, 48 pin buss compatible with OSI. Will run OSI 65D and 65U software.

MODEL 65-1 **$2995**
2 8" Single sided drives

MODEL 65-2 **$3495**
2 8" Double sided drives

BP-580 8 Slot Backplane ..... **$ 47**
OSI 48 pin Buss compatible

MEM-CM9 MEMORY/ FLOPPY CONTROLLER

24K memory/floppy controller card uses 2114 memory chips, 1 8K and 1 16K partition. Supports OSI type disk interface

24MEM-CM9 ..... **$325**
16MEM-CM9 ..... **$260**
8MEM-CM9 ..... **$180**
BARE MEM-CM9 ..... **$ 50**
Controller on assembled unit add ..... **$ 90**

BIO-1600 Bare IO card ..... **$ 50**
Supports 8K of memory, 2 16 bit parallel ports, 5 serial ports, with manual and Molex connectors.

PRINTERS

Okidata

ML82A, 120 cps, 10" **$409**
ML83A, 120 cps, 15" **$895**
ML84 Parallel, 200 caps, 15" **$1150**

C. Ioth

8510AP Prowriter, parallel **$419**
120 cps, correspondence quality
8510APD Prowriter, serial **$585**
F10-40PU Starwriter, parallel **$1319**
Letter quality daisy wheel
F10-40RU Starwriter, serial .. **$1319**
F10-55PU Printmaster **$1610**
parallel, Letter quality daisy wheel
F10-55RU Printmaster, serial **$1610**

DISK DRIVES AND CABLES

8" Shugart SA801 ..... **$385**
single sided
8" Shugart SA851 **$585**
double sided
FLC-6 6 ft cable from D&N **$69**
or OSI disk controller to 8" drive
5¼" MPI B51 disk drive with .. **$450**
cable, power supply and cabinet. Specify computer type.
FLC-5¼ cable for connection . **$75**
to 5¼ drive and D&N or OSI controller, with data separator and disk switch. Specify computer type

HARDWARE

OSI COMPATIBLE

IO-CA10X Serial Printer Port .. **$125**
Specify Device #3 or #8
IO-CA9 Parallel Printer Port .. **$150**

CMOS-MEM

64K CMOS static memory board, uses 6116 chips, 3 16K, 1 8K and 2 4K blocks, Partitionable for multi-user, OSI type disk controller, 2 IO mapped serial ports for use with D&N-80 CPU. Ideal way to upgrade from cassette to disk.

64K CMOS-MEM **$490**
48K CMOS-MEM **$390**
24K CMOS-MEM **$250**
16K CMOS-MEM **$200**
Controller add . **$ 90**
2 IO mapped serial ports add. **$125**
on assembled memory board
Z80-IO 2 IO mapped serial **$160**
ports for use with D&N-80 CPU card
FL470 Disk Controller **$155**
Specify 5¼ or 8" drive

STANDARD CP/M FOR OSI

D&N-80 CPU CARD

The D&N-80 CPU allows the owner of an OSI static memory computer to convert to Industrial Standard IBM 3740 single density disk format and CP/M operating system. Double density disk operation is also supported for 608K of storage on an 8" diskette. When used with a 5¼" disk system 200K of storage is provided. Includes parallel printer and real time clock. Also available for polled keyboard and video systems. Compatible with C2, C3, C4 and 200 series OSI computers.

D&N-80-P **$349**

CP/M 2.2 ..... **$150**

64K CMOS-MEM with D&N-80 CPU card **$450**

HARD DISK DRIVER **$140**
Allows D&N-80 CPU board to control OSI 40 or 80 meg hard disk unit. Will not destroy OSI files. Will also allow for a true 56K CP/M system. Specify 40 or 80 meg drive.

BUSS TRANSFER **$135**
Allows for D&N-80 and OSI CPU to be in the computer at the same time. Toggle switch provides for alternate CPU operation.

DISK TRANSFER **$100**
Utility program to transfer OSI CP/M format disk to IBM 3740 single density format. Will also transfer IBM to OSI format.

SYSTEM HARDWARE REQUIREMENTS

D&N-80 CPU, D&N FL470 or OSI 470 controller, 48K memory at 0000-BFFF, 4K memory at D000--DFFF, two disk drive cables.

FORMAT TRANSFER **$15**
You supply software on 8" diskette D&N will transfer OSI CP/M format to IBM 3740 CP/M format. Can also transfer IBM 3740 CP/M format to OSI CP/M format. Original diskette returned.

dual two-to-four decoder; when the three higher address lines A15, A14 and A13 are connected as shown, pins 4, 5, 6, and 7 go low when a location in the corresponding first four 8K blocks in the memory map are addressed. Pin 4 is left unconnected because the first 8K are already populated inside the Superboard.

Pin 7 decodes the fourth block ($6000-$7FFF) and I have chosen to split it further into four blocks of 2K each, to accommodate more efficiently those cards that take only a few memory locations (like the PIA and EPROM programmer cards). So, pin 7 is routed to the second decoder inside the LS139 package, together with A12 and A11 which address each one of the 2K blocks. The resulting select lines are labeled S7, S8, S9, and S10; only the first two of them are used in this version of the expansion motherboard.

The data bus on the Superboard is buffered by two 8T28 transceivers, U6 and U7 in the User's Manual schematic. They normally act as output devices and to input data from outside the DD (Data Direction) line must be pulled low. That job is performed by IC2, a 7420 dual 4 input NAND gate. When the computer needs to read data from any location within the memory segment used by the motherboard, one of three pins 5, 6, or 7 of IC1 will go low and the R/W (Read/Write) line will be high. That combination makes IC2 force the DD line low.

Assembly details:

The printed circuit for the motherboard was made using a single sided copperclad. It is not possible to route all the interconnection lines for this project on a single copper layer, so some jumper wires were used. As the expansion boards that I insert on the motherboard are also one sided, only one row of the connector's pins are used. Double sided printed circuit with plated through holes are more efficient in inter-connection density but they are much more expensive and difficult to process at home.

Figure 2 shows the component layout on the motherboard. The foil pattern used follows closely the lines as drawn on the schematic diagram.

Both IC's are mounted on sockets. The lines to interface the motherboard are connected to J9, which is a 40 pin DIP socket. Connection to the computer is made via a 16" long DIP jumper. C1 and C2 are decoupling capacitors and to be effective they should be mounted close to their respective IC's.

This expansion can be adapted to other 6502 based computers by using an appropriate connector on the computer side of the jumper cable and rearranging the wiring of J9, the interface socket on the motherboard.

Power consumption for both IC's is under 20 mA; so the +5V line can be tied to the computer supply. Note that the Superboard expansion connector does not provide a +5V output so the connection should be made with a separate wire. If the boards to be inserted into the expansion slots are going to draw current levels in excess of what the computer source can supply, the +5V input to the motherboard must be connected to an independent source.

★

EXPANDING THE C1P/SBII

by: David Tasker
111 Bass Highway
Tasmania, Australia 7303

Continued from Aug 83 issue

Part II
A systems approach.

A few preliminary design notes.

Most hardware expansions for any computer system tend to be designed within the following parameters.

a) One of a kind.
b) No provision for additional expansion.
c) Adapt if possible something existing.

What I tried to do with the motherboard concept was to allow for all future expansion needs, allow for changes in technology with plug-in modules, and very importantly, use professional techniques up to the limit of home assembly manufacture.

All of the PC Boards with the exception of the video board and the 24K CMOS card are single-sided boards.

The most complex board to assemble, the disk board, uses just 30 straps.

The system allows for modular expansion, in stages, of what I believe to be in "the need for" order, i.e.

1) Add memory to the computer (up to 8K).
2) Add a motherboard.
3) Add more memory and/or
4) Eprom
5) PIA/VIA (I/O)
6) Disk Controller with 5" disks
7) Upgrade to C4/C8 capability with videoboard
8) Add 8" disk, if required.

DATA Separators for either 5" or 8" are available to enable you to use many of the - "on special disk" drives.

Stage 1

Adding memory (up to 8K of RAM).

Assuming that you have populated the complete RAM area on the C1P/SBII, you will now be thinking of expanding to more memory with the ultimate aim of going to a disk system.

All C1P/SBII have a 40 pin expansion connector. V6 & V7 on the C1P/SBII need to be populated with either 8T28's or 8T26's. Whichever you choose to use (the circuits assume in any case 8T28's) then you must continue to use the same devices on each of the expansion boards. In truth it does not matter which you use on RAM cards (think about it), but for I/O, disks or Eprom, it is important.

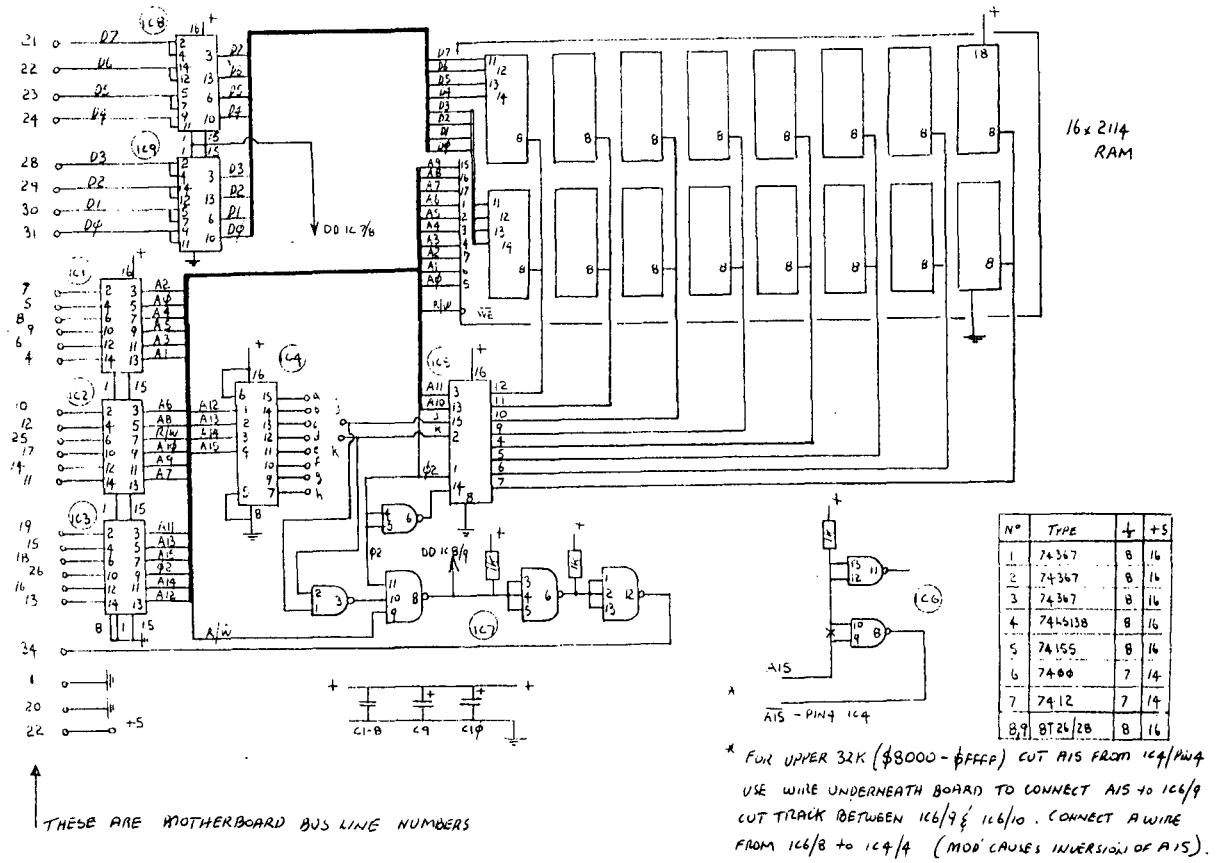
If you use 8T26s and later decide to add an Eprom board with my circuits, then you will need to modify the Eprom board to take an inverting data buffer.

NOW - back to the RAM.

The 1st used RAM card (you must specify) is 4" longer than all other cards. It is a single board, but has two sections on it, each electrically as well as physically connected. The two sections are electrically joined by a double width of copper clad edge connector pattern.

On one side of the pattern is a 40 pin socket to accommodate a 40 pin Ribbon cable from the C1P/SBII. This board also contains a buffer I.C. to buffer R/W 02, IRQ, NMI and the DD line.

No address buffers are provided on this section of the card. Address buffers for this card are on the RAM side

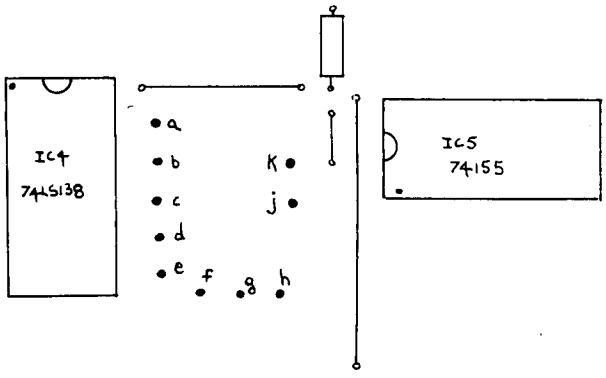


N°	TYPE	⏚	+5
1	74367	8	16
2	74367	8	16
3	74367	8	16
4	74LS138	8	16
5	74155	8	16
6	7400	7	14
7	7412	7	14
8,9	8T26/28	8	16

8K (2x4K) STATIC RAM. - 2114 - TYPE

D. TASHER 6/81

EXPANDED STRAPPING 2114-8K RAM CARD



EXAMPLES-

C1,C4, SUPERBOARD,2nd 8K (1st add on MEM CARD)
link - c-j (9-12k) $2000-$2FFF 8k on
d-k (13-16k) $3000-$3FFF CPU board

C1,C4, SUPERBOARD,3rd 8K (2nd add on MEM CARD) $0000-$1FFF
link - e-j (17-20k) $4000+$4FFF
f-k (21-24k) $5000-$5FFF

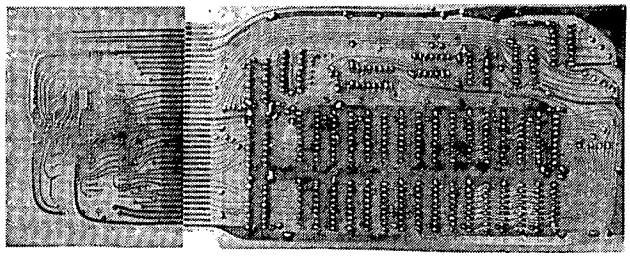
STRAP POINT-

a = 0 - 4k :-$0000~$0FFF 8K ON SUPERBOARD
b = 5k- 8k :-$1000~$1FFF C1P-C4 ROM MACHINES
c = 9k-12k :-$2000~$2FFF
d = 13k-16k:-$3000~$3FFF AIS is not inverted.
e = 17k-20k:-$4000~$4FFF i.e. standard board.
f = 21k-24k:-$5000~$5FFF
g = 25k-28k:-$6000~$6FFF
h = 29k-32k:-$7000~$7FFF

j = 1st 4k BLOCK
k = 2nd 4K BLOCK

To use RAM CARD in upper 32k - invert AIS as shown on circuit diagram - on IC4. i.e. MEM from 8000 to FFFF.

Note: $A000 - $BFFF (40k-48k) = BASIC IN ROM
$C000 - $C0FF = DISK I/O
$C700 - $C7FF = 16 PIN I/O
$D000 - $DFFF = VIDEO & KEYBOARD
$E000 - $E7FF = COLOUR RAM (4P)
$F000 - F0FF = CASS PORT C1
$FC00 - FCFF = CASS PORT C4
$FD00 - FFFF = MON' ROM C1,C4



of the edge connector pattern. Each subsequent plug-in card has its own address and data buffers. The motherboard has a set of address buffers to allow for multiple cards without loading the processor bus (CPU).

As mentioned, the other half of the 1st RAM card has an 8K block of 2114 RAM on it.

This completed board can fit inside your C1P case or above the SBII.

+5 volts for the RAM can be obtained from your existing power supply if it will provide 1 amp more. The 5 volt 5A regulator (78H05) that may be fitted to some C1Ps may run very hot. Also you may need to mount the bridge rectifier onto the case to act as a heat sink.

If your TV becomes jittery then you need to upgrade your power supply or provide a separate +5 volt feed.

Connection to the RAM card from the CPU is made via the 40 pin ribbon cable.

Next month, parts, construction notes, and more drawings.

★

OSI ROM ROUTINES

By: Leroy Erickson
Courtesy of OSMOSUS NEWS
3128 Silver Lake Rd.
Minneapolis, MN 55418

On any standard OSI Challenger II (C4P or C8P) or Challenger III, the "boot program" is contained in a 2k byte ROM (a 2316). Such a ROM contains 8 "pages", where a page is 256 decimal or 100 hexadecimal bytes. In OSI's ROMs, each page is a totally self-contained program. Out of the 8 available pages, each system uses only 2 or 3 of them addressed at $FF00, $FE00 and, maybe, $FD00. The 8 pages in one standard OSI ROM, the Synertek "SYNMON", contain the following routines:

Jumper Pin	Page #	Name	Address	Description
14	0	65V2P	$FE00	65V Monitor for 540 Video and ASCII keyboard
13	1	65VB73	$FF00	ROM BASIC Support for 540 Video & ASCII keyboard
12	2	65K	$FD00	Polled keyboard Driver
11	3	65VK	$FE00	65V Monitor for 540 Video and Polled Keyboard

Continued on page 17

High Resolution Color Graphics

Our new Color Plus board provides 256 x 192 high-resolution graphics with 15 colors. Two 8-bit resolution joystick interfaces are included. Software extensions to OS65-D BASIC provide a superset of APPLE II® graphics instructions.

Color Plus connects to the standard 48-pin bus or the 16 pin bus.

Pricing:

CP-8 for C8 or C3 computers:	$195
CP-4 for C4 computers (5V only):	$245
CP-bare Bare board with software:	$ 75

Generos Operating System

Generos is a new operating system for OSI computers. Features include descriptive error messages, optimized disk usage, device independence, and:

- ASSEM — Fast disk based assembler generates relocatable code.
- BASIC — Basic Interpreter
- DDT — Machine language debugger with single step, trace, more.
- TECO — Most powerful and widely used text editor ever.

Currently available for 48K 8-inch systems. Call or write about availability for other configurations. Cost: $85

Low Power Memory Board

Our popular MEM + board is ideal for:

- Partitions for multi-user systems
- 64K CP/M systems when combined with the D&N-80 CPU board
- Upgrading systems where backplane space, low power consumption, and/or low heat dissipation is required

Options include:

- OSI compatible floppy disk controller — protects against disk crashes caused by power failures
- Real time clock/calendar — Date and time with battery backup
- Centronics parallel printer interface — Supported by software that automatically patches OS65D and OS65U
- One year warranty

MEM + includes the following features:

- Memory chips in quality production sockets; high reliability machine screw sockets optional
- Low power consumption
- Uses 2K x 8-bit memory chips — will accept 2716-type EPROMS.
- Versatile addressing

NEW LOW PRICES!!

Bare — Solder masked and silkscreened $75
Software and documentation provided

16K	$200		
24K	$250	Disk controller	$85
32K	$300		
40K	$350	Real time clock	$65
48K	$390		
52K	$415	Centronics interface	$45
56K	$440		
64K	$490	High-rel sockets add 15%	

VISA, MasterCard, personal checks and C.O.D.s all accepted. Add $5 per board for shipping and handling.

To order, or for more information, contact:

Fial Computer
5221 S.W. Corbett
Portland, Oregon 97201

(503) 227-7083

Generic Computer Products

5740 S.E. 18th Ave. Portland, OR 97202

LISTING TO ACCOMPANY LETTER ON PAGE 22

```
10  I$=CHR$(34)
20  PRINT"ENTER;      "I$;I$"A ,B; C: "I$
30  REM THE TEST
40  POKE2888,0:POKE8722,0:POKE2972,13:
    POKE2976,13
50  POKE2968,13:POKE2797,0:POKE2794,13
    POKE202,255:REM POKE206,255 FOR V3.2
60  :
70  INPUTA$:REM NO SPACE
80  :
90  POKE2968,34:POKE2797,63:POKE2794,32:
    POKE202,32:REM DON'T FORGET TO PUT BACK
100 POKE2888,27:POKE8722,27:POKE2972,58:
    POKE2976,44
110 :
120 PRINT:PRINTA$
130 END
140 :
150 REM THE FOLLOWING CODE IS USED FOR DISK READ
160 REM WITH THE SAME MODIFICATIONS.
170 :
180 POKE2888,0:POKE8722,0:POKE2972,13:POKE2976,13
190 POKE202,255:REM ALLOWS LEADING BLANKS
200 REM 206 ON V3.2
210 POKE2968,13:REM ALLOWS THE "
220 :
230 INPUT#6,A$:REM INPUT FROM THE DISK (STRING)
240 :         :REM CAN BE BUFFER #7
250 POKE202,32:POKE2968,34
260 REM 206 ON V3.2
270 POKE2888,27:POKE8722,27:POKE2972,58:POKE2976,44
280 END
```

```
1   REM J.L.POTTIER 32 RUE PAUL DIOMEDE
2   REM CLERMONT-FERRANT 63100 FRANCE
3   REM TEL:(73) 30 88 22
4   :
5   REM SUPERBOARD II WITH OS65DV3.3 & 48 K
6   :
10  REM PROGRAM LISTER JLP 8/9/83
20  REM THIS PROGRAM LIST SEVERAL PGM FROM DISK A
30  REM USING DEVICE #5
40  REM MEM#5 = 7000 AND UP (FOR 32 K):B000 FOR 48 K
50  POKE 133,111:CLEAR:REM LIMIT MEMORY TO 6FFF
60  Q$=CHR$(34):NM=40:F$="END":S$=Q$+"*"+Q$+";"
70  PRINT"ENTER THE PROGRAM NAMES OR"
80  PRINT"ENTER "Q$"END"Q$" TO START THE LISTING
90  INPUT"DEVICE(1-2)";D$:P$="LIST#"+D$:L$="PRNT#"+D$+",
    CHR$(12)"
100 DISK!"MEM 7000,7000":REM ORIGINALLY $7000
110 PRINT#5,"NEW"
120 FORI=1TONM
130 PRINT"ENTER ";I;" NAME OF THE PROGRAM ?";:INPUTA$
140 IFA$=F$THEN170
150 GOSUB230
160 NEXTI
170 PRINT#5,"POKE133,191:NEW"
175 PRINT#5,"DISK!"+Q$+"IO 02,02"+Q$+":REM LIST FINISHED"
180 DISK!"MEM 7000,7000":REM RESET POINTER
190 REM LANCE LA PROCEDURE DE LISTING
200 DISK!"IO 10,02"
210 END
220 :
230 PRINT#5,"DISK!"+Q$+"LOAD "+A$+Q$
235 PRINT#5,"PRINT#"+D$+","+Q$+"**************************"
240 PRINT#5,"PRINT#"+D$+","+S$+Q$+STR$(I)+") LISTING OF = "+A$+Q$;
242 PRINT#5,";TAB(25)"+Q$+"*"
245 PRINT#5,"PRINT#"+D$+","+Q$+"**************************"
250 PRINT#5,"PRINT#"+D$
260 PRINT#5,P$
270 PRINT#5,L$
280 RETURN
```

J. L Pottier
Clermont-Ferrand 63100 France

* * * * *

AD$

Send for free catalog, Aurora Software, 37 South Mitchell, Arlington Heights, IL 60005. Phone (312) 259-3150.

* * * * *

48 K C8PDF with Intertec Intertube II serial monitor, two 8 inch floppy drives, CA10-X board with serial printer port. Excellent condition. CPU and disk cabinets have ultra quiet fans in them. Disk drive motors have a separate power switch so they can be shut down when not in use. This machine is very quiet! Software included: OS65D, OS65D plus assembler editor and extended monitor, OS65U, DMS (14 disks), WP-2, AMCAP, MDMS, Aardvark super disk, and copies. Approx. 50 blank disks. 9 notebooks full of documentation on software and hardware. All for $1500 plus freight. Bob Bernard, 2253 Ringling Blvd., Sarasota, FL 33577 (813) 953-5363.

* * * * *

32K C1P Series 2 Single Disk Drive System OS-65D3.3 with extended monitor / assembler. Excellent condition. Full documentation, Sams Manual, best offer. AIS, 3517 Dunedin Dr.#204, Chesapeake, VA 23321. 804-484-8856

* * * * *

FOR SALE: 8K RAM C1P Superboard II. Comes w/documentation, PEEK Issues January '83 to present, OSI Greatest Hits, Timetrek, Labyrinth games. $235. Phone 703-338-7532, ask for Ralph.

* * * * *

OSI parts. Color video 540B board $100. CPU 502 board with 8K RAM, 8K BASIC - $100. New Polled 542 keyboard - $50, no case. Bare 6100 board with CMOS 6100 processor and documentation - $50. UCSD Pascal and FORTRAN for 8 inch disks, video or serial - $300. Ron M. Battle, 1135 Princeton NE, Albuquerque, NM 87106, (505) 265-7345.

* * * * *

C2-4P cassettes for sale, WP 6502 Word Processor, worth $50, will sell for $25. Also, selling Invaders, Battleship, Wizard's City, Galaxia, Time Trek, Meteorites, and Machine Code Renumberer. Worth over $40, will sell for $15 or best offer. Contact Andrew Budson, 56 Monadnock Rd., Wellesley Hills, MA 02181, (617) 235-7899.

```
10      4    65VB76   $FF00
   ROM BASIC Support  for  540
   Video & Polled keyboard

9       5    65H      $FF00?
   CD-74 Hard Disk Boot Code

8       6    65A      $FE00
   Serial Monitor

7       7    65F3     $FF00
   "H/D/M?" Floppy Disk Boot
```

The jumper socket pin numbers are for the 502 & 505 CPU boards. Jumper socket pins 1, 2 & 3 are the select lines for addresses $FD00, $FE00 & $FF00, respectively. A standard BASIC in ROM system thus has the following 3 jumpers set - pins 1 to 12, 2 to 11, & 3 to 10. To convert to a floppy disk system, simply connect pin 3 to pin 7 rather than pin 10. To convert to a standard ASCII keyboard, connect pin 2 to pin 14, and pin 3 to pin 13, while leaving pin 1 open. To convert to a disk based serial system, connect pin 2 to pin 8 and pin 3 to pin 7. A serial BASIC in ROM system cannot be supported with this boot ROM.

This listing (see Listing 1) is the one for page 7, the floppy disk boot code. To follow what the routine is doing, start with the 6502's three interrupt vectors. On receiving an NMI interrupt (pin 6 of the 6502 pulled to ground), a jump is made to the address contained in locations $FFFA & $FFFB. For an IRQ (pin 4) or BRK instruction, the address in locations $FFFE & $FFFF is used. For a RESET (pin 40, which is connected to the Break key), addresses $FFFC & $FFFD are used. Notice that the contents of

LISTING 1
SYNMON ROM Page 7 - Floppy Disk Boot Code

```
10  FF00                *=$FF00
20                      ;
30  00FD=               LODADR=$00FD
40  00FF=               PAGCNT=$00FF
50  2200=               STRTAD=$2200
60  C000=               DSKPIA=$C000
70  C010=               DSKACI=$C010
80  D000=               SCREEN=$D000
90  D0C6=               HD0C6 =SCREEN+$C6
100 FC00=               SERPRT=$FC00
110 FD00=               HFD00=$FD00
120 FE00=               HFE00=$FE00
130 FE01=               HFE01=$FE01
140 FE0B=               HFE0B=$FE0B
150 FEED=               HFEED=$FEED
160 FEFC=               HFEFC=$FEFC
170 0130=               NMIADR=$0130
180 01C0=               IRQADR=$01C0
190                     ;
200                     ; *** DISK BOOT SUBROUTINE ***
210                     ;
220 FF00 A000   HFF00 LDY #$00      ;Select Data Direction Reg-
                                     ister A
230 FF02 8C01C0       STY DSKPIA+1 ;
240 FF05 8C00C0       STY DSKPIA   ;Assign Port A as all INPUT
250 FF08 A204         LDX #$04     ;Select I/O Port A
260 FF0A 8E01C0       STX DSKPIA+1 ;
270                                ;
280 FF0D 8C03C0       STY DSKPIA+3 ;Select Data Direction Reg-
                                     ister B
290 FF10 88           DEY          ;Get an FF
300 FF11 8C02C0       STY DSKPIA+2 ;Assign Port B as all OUTPUT
310 FF14 8E03C0       STX DSKPIA+3 ;Select I/O Port B
320 FF17 8C02C0       STY DSKPIA+2 ;Write Port B = all high
                                     (FF)
330                                ;
340 FF1A A9FB         LDA #$FB     ;Set step direction line to
                                     'IN'
350 FF1C D009         BNE HFF27    ;Skip for first pass
360                                ;
370 FF1E A902   HFF1E LDA #$02     ;Test for 'Track 0' true
380 FF20 2C00C0       BII DSKPIA   ;Read Port A & mask with
                                     TRK0 bit
390 FF23 F01C         BEQ HFF41    ;True - exit this loop
400 FF25 A9FF         LDA #$FF     ;Else, set step dir line
                                     to 'OUT'
410 FF27 8D02C0 HFF27 STA DSKPIA+2 ;Set step direction to
                                     given value
420 FF2A 2099FF       JSR HFF99    ;Wait 12 clock cycles
430 FF2D 29F7         AND #$F7     ;Select 'STEP' function
```

Continued on page 18

```
440 FF2F 8D02C0        STA DSKPIA+2 ;
450 FF32 2099FF        JSR HFF99    ;Wait 12 clock cycles
460 FF35 0908          ORA #$08     ;Turn off 'STEP' function
470 FF37 8D02C0        STA DSKPIA+2 ;
480 FF3A A218          LDX #$18     ;Wait 30,000 clock cyles
490 FF3C 2085FF        JSR WAIT     ;  (30 OR 15 ms)
500 FF3F F0DD          BEQ HFF1E    ;Loop back for more steps
510                                 ;
520 FF41 A27F   HFF41  LDX #$7F     ;Lower the head
530 FF43 8E02C0        STX DSKPIA+2 ;
540 FF46 2085FF        JSR WAIT     ;Wait about 150,000 cycles
550                                 ;
560 FF49 AD00C0 HFF49  LDA DSKPIA   ;Wait for the index hole
570 FF4C 30FB          BMI HFF49    ;
580                                 ;
590 FF4E AD00C0 HFF4E  LDA DSKPIA   ;Wait until the index hole
                                     is gone
600 FF51 10FB          BPL HFF4E    ;
610                                 ;
620 FF53 A903          LDA #$03     ;Reset the ACIA
630 FF55 8D10C0        STA DSKACI   ;
640 FF58 A958          LDA #$58     ;Select - Receive interrupt
                                     disabled Xmit interrupt
                                     disabled, 8 data bits,
                                     even parity, 1 stop bit,
                                     /1 clock
650 FF5A 8D10C0        STA DSKACI   ;
660                                 ;
670                                 ;
680                                 ;
690 FF5D 2090FF        JSR GETCHR   ;Get a byte from the disk
700 FF60 85FE          STA LODADR+1 ;Store as load address hi
                                     and save it in X
710 FF62AA             TAX          ;
720 FF63 2090FF        JSR GETCHR   ;Get another byte
730 FF66 85FD          STA LODADR   ;Store as load address low
740 FF68 2090FF        JSR GETCHR   ;Get a third byte
750 FF6B 85FF          STA PAGCNT   ;Store it as # of pages
                                      to load
760 FF6D A000          LDY #$00     ;Clear index register
770                                 ;
780 FF6F 2090FF HFF6F  JSR GETCHR   ;Get a data byte
790 FF72 91FD          STA (LODADR),Y;Save it at current
                                     location
800 FF74 C8            INY          ;Bump index
810 FF75 D0F8          BNE HFF6F    ;Loop until a page is full
820 FF77 E6FE          INC LODADR+1 ;When a page is full, incr
                                     addr hi, decr the # of
                                     pages to load
830 FF79 C6FF          DEC PAGCNT   ;
840 FF7B D0F2          BNE HFF6F    ;Loop until all pages are
                                     done
850 FF7D 86FE          STX LODADR+1 ;Then, restore addr hi
860                                 ;
870 FF7F A9FF          LDA #$FF     ;Lift the head
880 FF81 8D02C0        STA DSKPIA+2 ;
890 FF84 60            RTS          ;Go home, page zero is
                                     loaded
900             ;
910             ; *** Timed Wait Routine ***
920             ;
930             ; Wait 1250 * X + 11 machine cycles
940             ;
950 FF85 A0F8   WAIT   LDY #$F8     ;2 ; Get a 248, decimal
960                                 ;
970 FF87 88     HFF87  DEY          ;2 ; Inner loop - wait 1240
980 FF88 D0FD          BNE HFF87    ;2/3 ; machine cycles
990                                 ;
1000 FF8A 55FF         EOR PAGCNT,X ;4 ; Waste 4 cycles
1010 FF8C CA           DEX          ;2 ; Wait X * 1250 cycles
1020 FF8D D0F6         BNE WAIT     ;2/3 ; Loop until done
1030 FF8F 60           RTS          ;6 ; Go home after X*1250+
                                     11 cycles
1040      ;
1050      ; *** Get a byte from the disk ***
1060      ;
1070 FF90 AD10C0 GETCHR LDA DSKACI  ;Wait for ACIA receive flag
1080 FF93 4A           LSR A        ;
1090 FF94 90FA         BCC GETCHR   ;
1100 FF96 AD11C0       LDA DSKACI+1 ;It's there, get the byte
```

Listing continued on page 20

those last 2 locations is $FFA0, indicating that when you press the Break key, all system hardware is initialized (RESET also does that) and a jump to $FFA0 is made. From there on, you're on your own. If you have any questions, mail them to me and I'll try to answer them through PEEK. Have fun!

★

READER PROFILES

Johnson's Flowers in Midland has been using an OSI computer for the last year and a half to handle charge accounts and billing. The system has a 7 meg hard disk and is now handling 8000 accounts expandable to a maximum of 10000. The programs are written in Basic under 65U except for a couple USR routines to move large data blocks.

Previous to installing the computer, one employee spent twenty hours a week handling the paperwork and bills often did not go out until a month after the sale. The accounts receivable program prints statements at the end of each day resulting in a more prompt payment.

The program displays two different menus, one for the sales clerks and the other for the manager. The clerk menu allows charge sales, received on account, voided sales, addition of new accounts and address changes. The manager menu allows deletion of customers, editing of all data including transactions, printing of statements, listing of receivables by age, daily sales summary, and a check of file integrity.

The customer's telephone number is used as a key to access his account. In case the account is not found with a phone number, a search can be done for any string. Thus a listing of all 'SMITH's' can be displayed. This feature is useful when a telephone number has been changed or is unknown. Both manager and clerk can use this function to search for a name or name fragment.

A history of transactions is kept in each account along with comments the manager may wish to insert. While taking a telephone order, the clerk can display the account history. If the account has a large unpaid balance, the order can be refused on the spot. Many customers want the same thing sent as last year but can't remember exactly what it was. This information is kept in their account. Comments can be dated in the future and will be printed out on that date. This feature is useful in reminding forgetful husbands to get flowers for that special date.

At the end of the business day, the program searches the transaction files and prints a bill for every account with a sale today. An account with a sale exactly 30 days ago with an unpaid balance will also get a bill with service charges automatically added. The manager can add flags to the accounts to alter the normal billing cycle. Regular customers like churches are billed only once a month instead of after each transaction. A flag can also be added to delinquent accounts to prevent further charges being made to the account. With the manual billing system, flowers were usually delivered long before it was realized this was a bad account. The receivables aging program also lists accounts with long standing unpaid balances.

Bill Johnson
Earl D. Morris
Midland, MI 48640

* * * * *

ED:

I have been the "proud" owner of an OSI C3A system since 1979 when I was talked into purchasing it by an employee who is no longer with me. I am learning about computers and have received some help from your magazine although I should start back in grade school.

```
1110 FF99 60     HFF99 RTS            ;And go home
1120                   ;
1130 FF9A 48     HFF9A .BYTE 'H/D/M?';*** Request Message ***
1130 FF9B 2F
1130 FF9C 44
1130 FF9D 2F
1130 FF9E 4D
1130 FF9F 3F
1140             ;
1150             ; *** RESET Entry Point ***
1160             ;
1170 FFA0 D8     RSTADR CLD           ;Clear the decimal flag
1180                                  ;* Clear the screen *
1190 FFA1 A2D8         LDX #$D8       ;Get the high video page
                                       # + 1
1200 FFA3 A9D0         LDA #$D0       ;Get the low video page #
1210 FFA5 85FE         STA LODADR+1   ;Store it in an indirect
                                       register
1220 FFA7 A000         LDY #$00       ;Clear the low byte of
                                       the reg
1230 FFA9 84FD         STY LODADR     ;
1240 FFAB A920         LDA #$20       ;Get a blank
1250                                  ;
1260 FFAD 91FD HFFAD   STA (LODADR),Y ;Clear a char
1270 FFAF C8           INY            ;Bump the index
1280 FFB0 D0FB         BNE HFFAD      ;Loop till a page is done
1290 FFB2 E6FE         INC LODADR+1   ;Then incr the page #
1300 FFB4 E4FE         CPX LODADR+1   ;Done with the screen?
1310 FFB6 D0F5         BNE HFFAD      ;No, keep going
1320                                  ;*Reset the serial port*
1330 FFB8 A903         LDA #$03       ;Reset the ACIA
1340 FFBA 8D00FC       STA SERPRI     ;
1350 FFBD A9B1         LDA #$B1       ;Select - enable xmit &
                                       recv interrupts, 8 bit,
                                       no parity, 2 stop bits,
                                       /16 clock
1360 FFBF 8D00FC       STA SERPRT     ;
1370                                  ;
1380                                  ;*Print the request mes-
                                       sage*
1390 FFC2 B99AFF HFFC2 LDA HFF9A,Y    ;Get a char
1400 FFC5 300E         BMI HFFD5      ;Skip when 'CLD' reached
1410 FFC7 99C6D0       STA HD0C6,Y    ;Start at 4th line, 6th
                                       col
1420 FFCA AE01FE       LDX HFE01      ;Send to serial only if
                                       valid else, skip
1430 FFCD D003         BNE HFFD2      ;
1440 FFCF 200BFE       JSR HFE0B      ;Call serial out routine
1450 FFD2 C8     HFFD2 INY            ;Loop
1460 FFD3 D0ED         BNE HFFC2      ;
1470                                  ;*Get the response*
1480 FFD5 AD01FE HFFD5 LDA HFE01      ;Test for video or serial
                                       system
1490 FFD8 D005         BNE HFFDF      ;Skip if video
1500 FFDA 2000FE       JSR HFE00      ;Get char from serial
                                       device then skip else,
                                       get char from keyboard
1510 FFDD B003          BCS HFFE2     ;
1520 FFDF 20EDFE HFFDF JSR HFEED      ;
1530 FFE2 C948   HFFE2 CMP #$48       ;Is it an 'H'?
1540 FFE4 F00A         BEQ HFFF0      ;Yes, do hard disk boot
1550 FFE6 C944         CMP #$44       ;Is it a 'D'?
1560 FFE8 D00C         BNE HFFF6      ;No, go to ROM monitor
1570 FFEA 2000FF       JSR HFF00      ;Yes, load track zero
1580 FFED 4C0022       JMP STRTAD     ;Then go to $2200
1590                                  ;
1600 FFF0 4C00FD HFFF0 JMP HFD00      ;Go to hard disk boot
                                       routine
1610                                  ;
1620 FFF3 2000FF       JSR HFF00      ;***Unreachable code***
1630                                  ;
1640 FFF6 6CFCFE HFFF6 JMP (HFEFC)    ;Enter ROM monitor
1650                                  ;
1660 FFF9 EA           NOP            ;***Unreachable code***
1670                                  ;
1680 FFFA 3001   HFFFA .WORD NMIADR   ;NMI Vector (None-
                                       maskable Interrupt)
1690 FFFC A0FF   HFFFC .WORD RSTADR   ;RESET Vector
1700 FFFE C001   HFFFE .WORD IRQADR   ;IRQ Vector (Maskable
                                       Interrupt and BRK
                                       software interrupt)
                       .END
```

You have asked for letters from business users. I purchased this system to use in my business, specifically to perform the coordinate geometry computations required in land surveying. I was even able to buy a program. Needless to say, the program did not work. The author would supply an updated version for another $1000.00. I suggested where he could go and hired a programmer. I now have an excellent computation program with coordinate storage that meets all of my survey needs. The system also drives a Houston Instrument plotter and will output a quality drawing although I do not have an annotation routine.

With the hardware I had available, the next step was to upgrade some of my office procedures, so we wrote a job cost/payroll program that takes input from the daily time cards, computes payroll, writes the checks, updates totals for tax reports, and updates job costs for year to date reports.

I then purchased a General Ledger program, but to date have not been satisfied with its performance. Programmers need to learn to write operation manuals.

Somewhere in there, I added a Word Processing program and a NEC spinwriter to the system which is used for repeated technical material. Letters are still written on a standard typewriter.

Orwic A. Johnson
Columbus, IN 47201

* * * * *

ED:

Listed here is a fact sheet describing the hardware, software, and uses of the OSI-C-3 system we have at our Residency Office.

Our office is one of forty-four field offices of the Virginia Dept. of Highways and Transportation. We maintain approximately 1,475 miles of interstate, primary, and secondary roads in Frederick and Shenandoah Counties with a complement of 165.

This office is the first to utilize a micro computer in its operations. A neighboring residency is using ours to keep their road stock also via a telephone line and modem. We set up the system for them and back up their diskettes when they are finished. The

procedure is cumbersome, but is much faster than the old way we were both using.

We do not do much of our own programming, but there are a few small programs which we have implemented.

We really like our OSI and look forward to expanding it to multi-user and getting a couple of better printers.

Equipment-
OSI - C-3 (OEM) Standard 48k Dual 8" Floppy
Epson MX-80 Printer
Micro-Term, Inc. ACT-5A CRT
UDS - 300 Baud Auto Answ./ Modem

Software-
Inventory IHS Computer Services
DBMS Valley Micro Computer DBMS
Scribe (tm) Word Processor IHS Computer Services
Busi-calc Spreadsheet Micro Software Int'l

USES:

Inventory- Initial and justifying reason for purchase of micro for use in a field office of VDH & T. All road stock (pipes, stone, liquid asphalt, bridge parts, etc.) are kept track of on the C-3 using a program written by Dr. Dan Sweger of IHS Computer Services. Before, all inventory record keeping was done on a 40 year old Kardex system, and still is in the 43 other Residencies throughout the state and in our 9 District Offices. (OS-65U V.1.2)

DBMS- A data base management system which is a much more user friendly DMS compatible filing system written by Valley Micro Computers and IHS Computer Services.

At present we have several data files built which are being used to track permit applications, traffic count data, road resurfacing projects, and rights of way widths by route. (OS-65U V.1.42)

The Scribe- A word processor written by Dr. Sweger which has most of the bells and whistles indigenous to all word processors plus it interfaces to DMS files. A very slick package which is really more beneficial to a business oriented towards the private sector rather than government. It is very useful to us for writing reports and other types of in-house documents as we are lacking a letter quality printer. (OS-65U V.1.3)

Busi-calc- The only electronic spreadsheet we were able to obtain without first obtaining additional expensive software to support other operating systems.

Very nice to use, but limited in that it only has the basic four function mathematical functions available. We use it for financial forecasting, planning, and "forms" duplication. The sheet is somewhat slow to use as compared to Visicalc or Supercalc. The entire screen is rewritten after each entry. With a 48k C-3, sheet size is only 22k. But there are some nice "hidden" features which were inadvertently discovered which make it easier to use. (Files do not have to be "pre-created".) I have not given a more detailed description because I assume a general understanding of an electronic spreadsheet. Busi-calc certainly does yeoman service as a basic tool, but it is not in the same league as some of the more popular programs available of similar ilk.

W. H. Bushman
Resident Engineer
Commonwealth of Virginia
Dept. of Hwys & Transportation
Edinburg, VA 22824

LETTERS

ED:

I am writing this letter to express my high regard for my OSI computer, and my great sorrow that there are not more machines like mine in the world. The system that serves me with such felicity consists of the following: OSI C4P-MF (48K) computer, OS-65D V3.2 operating system, WP6502 with DQ-Justify word processor, DQ-Secretary utility, MX-80 printer, Aardvark BEXEC*, Aardvark Machine Code Editor and other Aardvark utilities.

I received my system in June of 1981, and I have made certain hardware modifications to it. At first the two SHIFT keys did not perform the same function, when writing l/c characters. Therefore, I cut the offending key clear, and hard wired the two SHIFT keys in parallel. Next, I purchased and installed a D&N Micro Products DSK-SW, that automatically turns off the disk drives when they are not in use. Finally, I replaced my

SYNMON monitor ROM with a SYNKEY EPROM from Micro-Interface. This last gives me true typewriter fingering. The Machine Code Editor, that I mentioned above, is a full screen editor, which I much prefer to the single line editor resident in OS-65D V3.3.

Recently, I have had an opportunity to work with an Eagle II computer, which employs a CP/M operating system, and I find it most inconvenient to program. I am spoiled because I have become accustomed to the following convenient features of my OSI operating system:

1. Upon boot-up it comes up in nine-digit BASIC in the immediate mode, which is an interpretative BASIC. That is, you start running or writing programs without delay.

2. Upon boot-up the operating system immediately runs the executive program, BEXEC*, as the first item of business. This is a BASIC program, which you are at liberty to modify, therefore, you can call any other program on your menu automatically at your option. It makes booting-up a lot simpler, if you have some favorite program that you use consistently.

I have made significant use of the trigonometric functions provided in my OSI Microsoft nine-digit BASIC. Nine digits provide ample precision for the surveying calculations that I do. The CP/M machine, that I have recently had some experience with, only has six-digit MBASIC available to it. There is CBASIC language available which provides fourteen digits of precision in the functions, but this language must be "compiled" before you can run it. It is exceedingly time consuming when writing and editing a program, because you can't see it run until you have compiled it. For a long program of 200 to 300 lines it can take a minute or more just to compile it.

I have recently attended a Computer Show, and all the computers are offering the CP/M operating system, and no one is featuring the convenient nine-digit BASIC. I don't need fourteen digits, but I do need more than six digits for precise land surveying.

Probably the reason that OSI is languishing is because there are too few customers that appreciate the conveniences inherent in its systems. On the other hand, I was initially greatly dissatisfied with the OSI keyboard, until I discovered how to correct it, no thanks to OSI.

Computer programming can be a pleasure and a challenge, but without an appreciative audience, it can be lonely.

Carl M. King
Sarasota, FL 33579

* * * * *

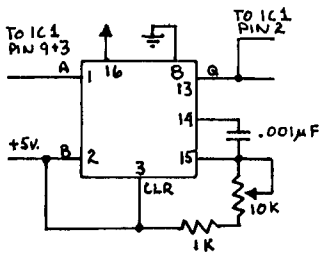
ED:

ERRATA on my May '83 (Vol 4 No. 5) article on Building a Data Separator:

The 74LS121 specified for IC2 is not made in an LS version. Use the 74121 instead, or use the circuit below.

The power connections to IC2 (74121) are wrong. +5v should go to pin 14 (there is no pin 16) and pin 7 should be connected to ground. The other pinouts are correct.

The 74121 is not too easy to find. While I have not actually built it, the following circuit based on the readily available 74LS123 should work. Sorry for any inconvenience caused by these errors.



J. F. McConkey, III
Rockville, MD 20855

P.S. Have any of you hackers tried interfacing the $50 Vicmodem or Hesmodem (originally for the Commodore 64) to an OSI yet? IF not, does anyone know what signals it requires out of the computer? It looks like it might be a simple, low cost way to add a modem, but I haven't had time to try it yet.

* * * * *

ED:

I have hopefully two simple questions to ask.

1) Does anyone know a relatively simple way to run OS-65D 3.3 on disk for the C1P on a C2-4PMF?

2) I only have a single disk drive and would like to know how to: -

a) Re-establish track zero on a disk (if I boot up another disk first, I can use my "bad" disk) and keep the rest of the information?

b) Also, how to transfer one disk to another?

Baxter B. Wilson III
Houston, TX 77017

Baxter:

The answer to your first question is NO. The reason for this is that the C1P and C2-4PMF have different memory allocations, that is the main reason why it can't be done simply. Everywhere in OS-65D that addresses an Input Output device would have to be changed. It would be easier and simpler to just buy the correct version of OS-65D for your machine.

Brian

In answer to question #2:

a) COPIER copies Track 0 through Track X. Just tell it Track 0 thru Track 0.

b) 3.3 has a single disk copy routine (COPIER). Use it. 3.2 mandates LOAD file, change disks, SAVE file.

Ed

* * * * *

ED:

Here is the answer to the question of Frank Glandorf which appeared in the November '83 issue of PEEK(65).

The following code allows you to enter leading spaces and quotation marks in a word processor. It works on my Superboard II with v3.3 and may also work with v3.2. Note that POKE202 as indicated must be changed to POKE206 for v3.2. Also, failure to reset the POKE back to a value of 32 will cause the program to stop on a space.

I have also enclosed a program that will list out a number of programs. I hope that someone will find it useful.

LISTING ON PAGE 16

PEEK (65)
The Unofficial OSI Users Journal

P.O. Box 347
Owings Mills, Md. 21117

BULK RATE
U.S. POSTAGE
PAID
Owings Mills, MD
PERMIT NO. 18

DELIVER TO:

GOODIES for OSI Users!

PEEK (65)
The Unofficial OSI Users Journal

P.O. Box 347 • Owings Mills, Md. 21117 • (301) 363-3268

( )	**C1P Sams Photo-Facts Manual.** Complete schematics, scope waveforms and board photos. All you need to be a C1P or SII Wizard, just	$7.95 $ _____
( )	**C4P Sams Photo-Facts Manual.** Includes pinouts, photos, schematics for the 502, 505, 527, 540 and 542 boards. A bargain at	$15.00 $ _____
( )	**C2/C3 Sams Photo-Facts Manual.** The facts you need to repair the larger OSI computers. Fat with useful information, but just	$30.00 $ _____
( )	**OSI's Small Systems Journals.** The complete set, July 1977 through April 1978, bound and reproduced by PEEK (65). Full set only	$15.00 $ _____
( )	**Terminal Extensions Package** - lets you program like the mini-users do, with direct cursor positioning, mnemonics and a number formatting function much more powerful than a mere "print using." Requires 65U.	$50.00 $ _____
( )	**RESEQ** - BASIC program resequencer plus much more. Global changes, tables of bad references, **GOSUBs** & GOTOs, variables by line number, resequences parts of programs or entire programs, handles line 50000 trap. Best debug tool I've seen. MACHINE LANGUAGE - VERY FAST! Requires 65U. Manual & samples only, $5.00 Everything for	$50.00 $ _____
( )	**Sanders Machine Language Sort/Merge** for 0S-65U. Complete disk sort and merge, documentation shows you how to call from any BASIC program on any disk and return it or any other BASIC program on any disk, floppy or hard. Most versatile disk sort yet. Will run under LEVEL I, II, or III. It should cost more but Sanders says, "...sell it for just..."	$89.00 $ _____
( )	**KYUTIL** - The ultimate OS-DMS keyfile utility package. This implementation of Sander's SORT/MERGE creates, loads and sorts multiple-field, conditionally loaded keyfiles. KYUTIL will load and sort a keyfile of over 15000 ZIP codes in under three hours. Never sort another Master File.	$100.00 $ _____
	BOOKS AND MANUALS (while quantities last)	
( )	**65V Primer.** Introduces machine language programming.	$4.95 $ _____
( )	**C4P Introductory Manual**	$5.95 $ _____
( )	**Basic Reference Manual** — (ROM, 65D and 65U)	$5.95 $ _____
( )	**C1P, C4P, C8P Users Manuals** — ($7.95 each, please specify)	$7.95 $ _____
( )	**How to program Microcomputers.** The C-3 Series	$7.95 $ _____
( )	**Professional Computers Set Up & Operations Manual** — C2-OEM/C2-D/C3-OEM/C3-D/C3-A/C3-B/C3-C/C3-C'	$8.95 $ _____

( ) Cash enclosed ( ) Master Charge ( ) VISA

Account No. _____ Expiration Date _____

Signature _____

Name _____

Street _____

City _____ State _____ Zip _____

TOTAL	$_____
MD Residents add 5% Tax	$_____
C.O.D. orders add $1.65	$_____
Postage & Handling	$ 3.50
TOTAL DUE	$_____

POSTAGE MAY VARY FOR OVERSEAS