INSIDE

Column One

Once again, the rumor machine has been working overtime with all sorts of bad news about OSI. We like to think that we have heard most of them and have also secured some pretty reliable information from those who ought to know.

Here are the facts, as we understand them:

In September, a sizeable note from OSI to the Bank of America came due;

OSI had pledged their assets to repay the note;

On August 29, the bank moved in to "protect their assets";

The first step was to stop everything to take an inventory;

Employees were laid off, but expect to be called back;

The Bank of America now controls OSI assets, but sales from inventory are continuing;

Negotiations for substantial additional venture financing, which were already under way when the bank stepped in, are continuing;

Long range plans are still being made;

Something concrete should happen (resumption of business, perhaps with a new direction, or...) by about mid-October.

So, what does all of this mean to PEEKers? Who knows? We have seen OSI come through too many changes to write them off... yet. They may be back stronger then ever.

On the other hand, with Osborne and others going belly up recently, they may not make it. So then what?

Suppose you own an OSI computer, and the company drops out of sight? Who will fix it? Who will support it? Who will sell you parts and software?

OK, now everybody who has depended on the factory for all those things, raise your hand. Hmm, OK, you can put your hands down now, both of you. The rest of us will keep on getting our support and parts and software from the same independent sources where we have been getting them all along.

Naturally, we believe the primary source of such support and software and parts is PEEK(65), meaning you, our readers and advertisers.

If anybody doubts that, have a look at the issue you hold in your trembling hands. You will find articles and ads and listings for everything from a new (old) text editor for small machines to entire business hard-disk systems, including a new and better data base manager which runs multi-user on a hard disk machine, an azimuth calculator for amateur astronomers and a whole new DOS.

In case you were thinking maybe your business application could use the new DBMS, but would require a hard disk, have a look at the Denver Board ad -- seems we OSIers are among the first to be able to use the new removable-media hard disk units for backup.

And last but certainly far from least, we begin in this issue our software listing. You will find everything from a snake-eating game to complete systems to run banks and clothing stores -- and much more. We plan to continue these listings as long as we have items to list, so if you still haven't sent us a description of your gem, get it in the mail!

We have received requests that the "Casette Corner" feature be reinstated. We agree that this was a valuable feature, and would love to resume it. What we need is an author. All you cassette hotshots out there, get with it. Anyone who would like to share experiences with non-disk machines, consider taking over the job. It pays magnificently (at least $10/month) and the only requirement is to send us a sample column...

Another reader wonders about a SII users' club. Who knows of one?

al

TEC65 REVIEW

By: David A. Jones
9226 N.W. 17 Street
Coral Springs, FL 33065

TEC65
6502 Program Exchange
2920 West Moana, Reno NV 89509
$35.00   ClP W/HEXDOS 4.0
$45.00   ClP W/OS65D 3.2
$45.00   ClP/C4P W/OS65D 3.3

TECO is a text editing language by Digital Equipment Corp , the minicomputer manufacturer Notice that I said a text editing language. not just an editor or a word processor.

You can actually write programs (called macros) in TECO that will open one or more files, edit, combine, format, extract excerpts of, or almost anything else you would want to do with a text or data file and then write the output to one or more new files. The most common application though is as an interactive text editor. As such, it is one of the most powerful editors available today. Because TECO is very versatile, it is necessarily complex and many of its features go unused even among experienced users.

TEC65 is a subset of the most frequently used commands of TECO. To quote from the 6502 Program Exchange catalog, it can:

Change the third occurrence of the word "apple" to "orange,"

Change the first seven occurrences of the word "men" to "women,"

Make the second paragraph of a text into the fourth paragraph,

Print every line that contains the word "cycle,"

Search for a string "stoXe" where "X" is a wildcard character. Thus "stoXe" would match with "stone", "store", or "stove,"

Print 40 copies of a form letter without interruption""

.......and more.

Several commands in TEC65 vary from TECO, all apparently to make life easier. Among these; A is replaced with EA, PW becomes EW, and EO serves both as the ER and EW commands   A single escape character does a -T and a line feed does an LT. Control G is equivalent to V or 0TT.   None of this is really objectionable, but it does slow down those of us who type mechanically. I use TECO-11 daily at work and have to reprogram myself when I come home Being so used to typing on different keyboards, I wonder why I should complain about this. I did, however, change the function of the line feed to work as an escape character when the shift lock is not down. This facilitates search and replace commands when in lower case.

TEC65 is all machine code and requires 5K of memory. Implemented on a 32K system under HEXDOS 24K of text buffer is available. With a stock system no special patches or hooks are required.

Since my system is not stock, I had to make several overlays to HEXDOS and one to TEC65. My input and output routines are customized in EPROM to support my video and parallel ports modifications. Overlaying the HEXDOS input routines at $08B3 and output routine at $0992 with patches accomplishes this task. Changing the input/output vectors at $0218 and $021A is not the proper way to do this as the disk file management code is part of HEXDOS's input/output. CEGMON support is also located here as are the HEXDOS definitions of the control keys. Also, for HEXDOS users who want to disable the control C function of the repeat key, overlay NOP's on the 14 bytes starting at $0454. This area should currently start with $A9. Many cassette games use the repeat key as a fire control.

Another item of interest is the start address of TEC65 at $0F00. It's not given in the documentation but is necessary to make work or backup copies.

Use:
SAVE#3,3838:SAVE#4,3838+2048: SAVE#5,3838+4096 to put it on tracks 3, 4 and 5. It took me quite a while to glean the necessary information to this out of the HEXDOS manual.

TEC65 is called into memory via the BASIC command LOAD $TEC65:?USR(-5)4096. I have a short BASIC program that does this and also POKES in the patches to HEXDOS and sets the default parameters for text justification. This is a clean way to do it and it allows both HEXDOS and TEC65 to remain at their original version levels. A word of warning is in order here.   HEXDOS BASIC programs start at $0B00 and TEC65 starts at $0F00 so the calling program is limited to less than 1K. Another caution: TEC65's initialization table is located at $2008. Once initialized, the table is no longer required and it is overwritten as it lies within the text buffer. When you want to make a backup or work copy, don't enter text to see if it works before you save it to disk. I lost a lot of sleep until these 2 facts dawned on me. The initial justification parameters are also stored here. See table 1.

To save text to a file, you must have created one beforehand large enough to receive the text. Too bad we can't create one dynamically from within TEC65 that would be just the right size.   Should you enter more text into the buffer than your output file will handle, all is not lost. Just put the excess into one of the Q registers and save the rest as file nr. 1. Clear the buffer and recall the excess from the Q register and save that as file nr. 2. Assuming you have file nr.1 and file nr. 2 available. Anytime you're interested in the length of the file in the buffer. you can use the commands ZL$.=$$ and the character count will be displayed.

All in all, - I really like TEC65. It is character rather than line oriented and you can move the cursor to any point in the text almost instantly by using the search command. You don't have to worry about reformatting when you insert or delete text in the middle of a line, and carriage returns are only necessary for paragraphing and lines that you don't want justified, columns of numbers or data for example. Screen editors are nice sometimes, I use mine for program editing, but I feel

the character editors are more efficient for document preparation. This program has been available since 1978 on other 6502 based machines and appears to be bug free. I would expect the OS65D version to be as good. In fact, when it becomes compatible with OS65D version 5.1 just released from OSIO maybe new files could be created from within the editor.

Although not originaly part of TECO, a justification routine has been added to TEC65. The justification works erratically sometimes unless you reboot TEC65 and load the file anew before printing. I have informed the exchange of this anomaly and have hopes of them providing a fix. Since this is a freebie included with the package, I won't downgrade TEC65 on this point alone.

A second freebie included is a set of 10 macro commands on one disk file. Not only are they useful, but they are educational in the use of TECO commands as well.

A clearly written 22 page manual supplemented with OSI specific information is included. A very nice program at a reasonable price.

TABLE 1

$200D JUSTIFICATION FLAG,00= ON
$200E PAGE FLAG 80=ON
$200F PAGE NUMBER FLAG,80=ON
$2010 PAUSE FLAG 80=ON
$2012 LINE LENGTH,VALUE+2
$2015 PAGE LENGTH,VALUE
$2016 PAGE NUMBER,STARTING NUMBER
$2017 LINE SPACING,0=SINGLE, 1=DOUBLE, 2=TRIPLE
$201E TAB SPACING,VALUE
$204B DELETE CHAR,5F=SHIFT O, 7F=RUBOUT

★

AZIMUTH READING

By: Steve McGinnis
505 Smith Street
Ridgway, PA 15853

I finally started feeling guilty about not supporting PEEK so I am enclosing an OSI program I wrote after reading the April, 1982 issue of Astronomy (published by Astro-Media Corp., Milwaukee, WI).

The program will give an azimuth reading (any good compass with degree markings is close) and an altitude reading (90 degrees is up!) of any object in the sky. All

ARCTURAS
ALTITUDE= 62.9525 DEGREES
AZIMUTH = 221.225 DEGREES
TIME: 8 : 15 DATE: 7 / 29

OBJECT: ARCTURAS
RT.ASC: 14.2404
DEC. : 19.3721
TIME: 8 : 15 DATE: 7 / 29

```
1     REM ADAPTED FROM APRIL 1982 ASTRONOMY ARTICLE WRITTEN
      BY P. BURKE
5     REM ADAPTED FOR OSI C1PII BY S. MCGINNIS
10    FOR CL=1 TO 25:PRINT:NEXT CL
20    PRINT"ALT-AZIMUTH SETTING CIRCLES"
30    PRINT:PRINT
35    INPUT"OBJECT";O$
40    REM DAY NUMBER*****************************
50    PRINT"INPUT THE CURRENT MONTH # (1-12)"
60    INPUT MO
70    PRINT"INPUT THE DAY NUMBER (1-31)"
75    INPUT DAY
80    DN=INT(275*MO/9)-(2*INT((MO+9)/12))+DAY-30
90    PRINT"DAY NUMBER = ";DN
100   REM MEAN SIDEREAL TIME*******************
110   REM1983:K=6.60649392,1984:K=6.59057904,ETC.REF ASTRONOMY
      APR 1982
120   GMST=6.60649392+(0.0657098232*DN)
130   REM SIDEREAL  TIME AT CURRENT TIME**********
140   PRINT"INPUT CURRENT TIME IN HOURS"
150   INPUT HR
155   TI=HR
160   PRINT"NOW INPUT # OF MINUTES AFTER THE HOUR"
170   INPUT MIN
180   HR=HR+MIN/60
185   SI=MIN
190   REM CONVERT TO 24 HOUR CLOCK
200   PRINT"INPUT '1' IF EVENING HOURS, '0' IF MORNING HOURS"
210   INPUT CODE
220   IF CODE=1 THEN HR=HR+12
230   REM ADD TO GET UT(GMT)*******************
240   UT=HR+4
250   GMST=GMST+(1.0027379093*UT)
260   PRINT"GMST= ";GMST
265   REM GO GET LONGITUDE*********************
270   GOSUB 1270
360   REM LOCAL SIDEREAL TIME*******************
370   LST=GMST-LGT
380   IF LST>24 THEN LST=LST-24
390   IF LST>24 GOTO 380
400   REM LST TO DEGREES***********************
410   LST=LST*15
420   PRINT"LST= ";LST
422   INPUT"1=SETTING CIRCLES,2=FIND R.A.";X
425   IF X=2 THEN 690
430   PRINT"INPUT RIGHT ASCENSION AS HOURS,MINUTES"
440   INPUTHR,MIN
460   MIN=MIN/60
470   RA=HR+MIN
472   REM RA TO DEGREES***********************
475   RA=RA*15
480   REM HOUR ANGLE**************************
490   HA=LST-RA
500   PRINT"HOUR ANGLE= ";HA
510   PRINT"INPUT THE DECLINATION AS DEGREES,MINUTES"
520   INPUTDGS,MIN
540   MIN=MIN/60
550   DGS=DGS+MIN
560   REM D=DECLINATION**********************
570   D=DGS
572   REM GO GET LATITUDE********************
575   GOSUB 1570
622   REM RADIANS***************************
623   C=.0174532925
625   D=D*C:L=L*C:HA=HA*C
630   A=(SIN(D)*SIN(L))+(COS(D)*COS(L)*COS(HA))
635   AS=ATN(A/SQR(-A*A+1))
650   AZ=(SIN(D)-(SIN(L)*SIN(AS)))/(COS(L)*COS(AS))
655   AC=-ATN(AZ/SQR(-AZ*AZ+1))+1.5708
660   AZ=AC/C
665   IF SIN(HA)>0 THEN AZ=360-AZ
```

3

```
666    POKE517,255
667    PRINTO$
668    PRINT"ALTIUTDE= ";AS/C;" DEGREES"
670    PRINT "AZIMUTH= ";AZ;" DEGREES"
672    PRINT"TIME:";TI;":";SI;" DATE:";MO;"/";DAY:PRINT:PRINT
675    POKE517,0
680    INPUT"CONT (Y/N)";X$:IF X$="Y" THEN 35
685    END
690    REM R.A. & DEC.****************************
700    REM L=LATITUDE, A=ALTITUDE, AZ=AZIMUTH
710    INPUT"INPUT ALTITUDE IN DEGREES";A
720    INPUT"INPUT AZIMUTH IN DEGREES";AZ
721    FLAG=0
722    IF AZ>180 THEN AZ=360-AZ:FLAG=1
725    REM RADIANS*******************************
730    C=.0174532925
745    REM GO GET LATITUDE**********************
750    GOSUB  1570
760    L=L*C:A=A*C:AZ=AZ*C
770    SD=(SIN(A)*SIN(L))+(COS(A)*COS(L)*COS(AZ))
775    D=ATN(SD/SQR(-SD*SD+1))
780    CHA=(SIN(A)-(SIN(L)*SD))/(COS(L)*COS(D))
785    HA=-ATN(CHA/SQR(-CHA*CHA+1))+1.5708
790    D=D/C:HA=HA/C
795    IF FLAG=1 THEN 810
800    IF SIN(HA)>=0 THEN HA=360-HA
810    HA=HA/15
820    RA=LST/15-HA
825    IF RA<0 THEN RA=RA+24:GOTO 825
830    POKE 517,255
835    PRINT"OBJECT: ";O$
840    PRINT"RT.ASC: ";RA
850    PRINT"DEC.  : ";D
860    PRINT"TIME:";TI;":";SI;"DATE:";MO;"/";DAY:PRINT:PRINT
865    POKE 517,0
870    INPUT"CONT (Y/N)";X$:IF X$="Y" THEN 35
900    END
1200   REM SUBROUTINES****************************
1270   REM LONGITUDE******************************
1290   DGS=78:MIN=44:SEC=30
1300   SEC=SEC/3600
1310   MIN=MIN/60
1320   DGS=DGS+MIN+SEC
1330   REM DECIMAL HOURS**************************
1350   LGT=DGS/15
1360   RETURN
1570   REM LATITUDE*******************************
1590   DGS=41:MIN=19:SEC=0
1600   SEC=SEC/3600
1610   MIN=MIN/60
1620   L=DGS+MIN+SEC
1625   RETURN
1630   END
```

that is needed is the right ascension (R.A.) and the declination (DEC.) of the object in question. These numbers can be found in any good astronomy handbook. Line number 422 gives the option of 1) finding the azimuth and altitude from the celestial coordinates or 2) finding the celestial coordinates from the azimuth and altitude. (Enclosed is an output for the star Arcturas, as an example.)

The only changes that have to be made to the program are as follows:

NO.   COMMENT

240   The adjustment from local time is entered here. Pennsylvania is four hrs. behind Universal Time (UT).

1290   Your longitude in Degrees (DGS), Minutes (MIN), and Seconds (SEC).

1590   Your latitude in Degrees (DGS), Minutes MIN), and Seconds (SEC).

There was a recent issue of Astronomy that gave a computer version of the April, 1982 article, but I still prefer this program. Since the mathematics is well documented in the Astronomy article, I will not repeat it here.

★   ★

by: Rick Trethewey
8 Duran Court
Pacifica, CA 94044

As promised, this month we will address and list the machine code program TRMCOD.

The next step is to create a file on an OS-65D diskette of at least 9 tracks. This time, the name doesn't matter. Serial systems will have to change the source code at lines 260 and 270 to reflect the address of their modem port, and line 390 to properly configure the ACIA. Using the OSI Assembler/Editor (or similar utility), enter the source for the machine code. Note that the origin address for this code is $B000, so make sure that you protect high memory before assembling this program. On the OSI Assembler, enter "HAFFF" to do this. Now, assemble the code to memory. If all has gone well to this point, hit <BREAK>. Now take out the OS-65D diskette and boot up on the OS-65U diskette with the TRM65U files on it. After BEXEC* has run it's course, run DISK.

The first thing DISK asks for is the disk address for the operation. This entry must be made in decimal, but you should have that information written down from when you created the files and then ran DIR. Next, DISK asks how many bytes are to be read or written. In our case, we need to write about 6 pages of RAM, so respond with "$600", the hexadecimal value. When DISK asks for the RAM address, enter "$B000". Lastly, DISK asks if you want to read or write. Enter "W" for "write". DISK will now do the rest for you.

Alright, you've captured the data on disk, now how do you get at it? That's where the program "PRINT" comes in. PRINT will dump the contents of a TRM65U file to either the console or printer. The file selection routine is the same one as contained in the TRM65U executive. Also, by making the appropriate POKEs to 11661 and 11662 and by using "4" as the output device number, it is possible to use PRINT to transfer programs captured in text form to the indirect file in memory.

Good luck.

```
 10 ; SMART TERMINAL PROGRAM FOR OS-65U V1.2
 20 ;
 30 CTRLB =$02
 40 CTRLC =$03
 50 CTRLD =$04
 60 BS    =$08
 70 CTRLI =$09
 80 LF    =$0A
 90 CR    =$0D
100 CTRLO =$0F
110 CTRLQ =$11
120 CTRLS =$13
130 POSCNT=$16
140 SP    =$20
150 SKIP2 =$2C
160 RUBOUT=$7F
170 FACHI =$AF
180 FACMHI=$B0
190 FACMLO=$B1
200 FACLO =$B2
210 STACK =$0100
220 KEYLST=8960
230 BUFFER=$A000
240 KYBD  =$DF00
250 MDCTRL=$F7D3
260 STATUS=$FC00      <-  CHANGE ON SERIAL SYSTEMS
270 MODEM =$FC01      <-  CHANGE ON SERIAL SYSTEMS
280 OUTDO =$0AEE
290 FLOAT =$1B44
300 ASCII =$1CEC
310 OUFLAG=11686
320;
330         *=$B000
340;
350 START  LDA #$34       GET UART INIZ FOR A-15
360         STA MDCTRL     SET TO MODEM PORT
370         LDA #$03       LOAD CLEAR COMMAND
380 S1      STA STATUS     INIZ ACIA
390         LDA #$16       <- CHANGE ON SERIAL SYSTEMS
400 S2      STA STATUS     CONFIGURE MODEM PORT
410         LDX OUFLAG     GET CONSOLE DEVICE NUMBER
420         LDA INTBL-1,X  FETCH LSB OF INPUT CODE
430         STA CNSLIN+1   SAVE IN CNSLIN
440         LDA INTBH-1,X  GET MSB
450         STA CNSLIN+2   SAVE IT TOO
460         JSR GETVAR     COMPUTE COMMAND FROM BASIC
470         TSX            GET CURRENT STACK POINTER
480         STX STKPTR     SAVE IT
490         LDA FACLO      GET COMMAND NUMBER
500         BEQ P0         INITIAL ENTRY? => P0
510         AND #$0F       NO, MASK TO LOW NIBBLE
520         CMP #$01       TGLON?
530         BNE S3         NO => S3
540         JMP TGLON1     YES! RE-ENTER CODE
550 S3      CMP #$02       TGLOFF?
560         BNE S4         NO => S4
570         JMP TGLOF1     YES! RE-ENTER CODE
580 S4      CMP #$03       SAVE/WRITE OUT BUFFER ?
590         BNE S5         NO => S5
600         JMP SAVE8      YES! RE-ENTER CODE
610 S5      CMP #$04       RETURN FROM FILE OPEN?
620         BNE S6         NO => S6
630         JMP SEND1      YES! ==> SEND1
640 S6      CMP #$05       RETURN FROM TRACK FETCH?
650         BNE S7         NO => S7
660         JMP SENDX      YES! => SENDX
670 S7      JMP P0         UNRECOGNIZED COMMAND => P0
680;
690 INTBL   .BYTE TTYIN,KBPOLL
700 INTBH   .BYTE TTYIN/256,KBPOLL/256
710;
720 TGLDUP  LDA P7
730         EOR #$0C
740         STA P7
750;
760 P0      JSR XIN        CHECK MODEM FOR INCOMING
770         BCC P1         NOTHING? => P1
780         AND #$7F       CHAR. REC'D! MASK PARITY BIT
790         JSR CNSLOU     SEND TO CONSOLE
```

REVIEW

SYSTEMS GENERATOR
VERSION 83.1

by:
PEEK(65)
Tom Hill

Three weeks ago I was asked by the folks here at Peek(65) to make a review of Comptrol Systems' SYSTEMS GENERATOR. Anyone who has had the opportunity to test any new software knows the feeling of apprehension when I opened the box delivered to us by Comptrol and found twelve diskettes and a manual.

As with any new software package, the first thing I did was to read the manual from cover to cover. This manual being quite thick (about an inch and a half), took about three hours to digest. It is well organized and written totally with the end user in mind. It was easy to read and appeared to be quite informative, but as we all know, if all of the software packages that we owned performed as well as the manuals claimed they would, we could probably run the entire U.S. government from our desk tops. The manual put aside, it was time to get to work.

I wondered, how long is this monster going to take to set up, and once I do manage to get it set up, will it work at all?

I must say that it was a pleasant surprise. Following the easy to understand instructions, I was able to get the system up and running in about an hour.

As the manual states, the system requires OSU V1.43 or later, so the first thing I did was to install the new OSU. This upgraded version of the operating system has plenty of enhancements over the earlier versions, and we hope to do a review of it at a later date. Using the "Install" program provided with OSU, the floppies were easily uploaded to our CD-74 hard disk. The only modification to the Systems Generator disks was to change the name of the BEXEC* program. This was done because SYSGEN was loaded into an existing subsystem on our hard disk. An easy to read, step by step procedure for this change is listed in the manual.

If you are going to use the system on a floppy disk ma- chine the only thing that you have to do, after making your initial backup of the master disks, is to run them. The reason that there are so many floppies, is that in order to prevent the hassles of con- stantly moving disks around, many of the most frequently used programs are repeated on all of the disks. For those of you who have Denver Boards in your machines, there is even a section in the manual on configuring the system for them.

The next step was to see how hard it would be to actually run the system.

SURPRISE! SURPRISE!

The main menu came up, and all of the utilities worked just fine.

The Systems Generator is a powerful data base manager de- signed for the end user, using end user terms. It incorporat- es OSI's record locking making the system fully configured for timesharing. Master files are called "MAIN STORAGE AREAS" and key files are "INDEX STORAGE AREAS". I must admit that this took some time to get use to, but for the end user these terms should be easy to understand. When de- signing a main storage area, the operator has the ability to define the rows and columns on the screen for inputting information, determine if the field is to contain alpha or numeric characters, and whe- ther it is to be checked for "DATE YYMMDD", "decimal 00.00" or no checking at all. There is also a facility to con- ditional check the data for as many as two conditions (=,<>, <,>,=<,+>) as well as checking for duplicate information in the same field.

One of the most powerful fea- tures of the system is the use of INDEX STORAGE (key files) areas. The index storage ar- eas are designed to be used in conjunction with the main storage areas, in that, if properly used, they are con- tinually updated. Whenever an addition or change is made to a field in the main storage area that has a corresponding index storage area, that index is also updated. The greatest advantage to this is that if the field in the main storage area is one that the user has designated as one that is to be checked for duplicates, in- stead if the system taking up valuable time searching the main storage area, it checks the index area. Because the index area is continually up- dated, there is no need to recreate or update the index file before printing a report. All the user has to do is sort the area or index with the fast sort supplied then print his report.

From adding and editing infor- mation, the next logical pro- cedure is to update and per- form statistical information on our data. The Systems Generator will do just about anything that the user wants. It allows the operator to glo- bally replace the information within a field, move infor- mation to another field, and perform calculations on fields of information (addition, sub- traction, multiplication, di- vision) and then store the results in yet another field. All of these functions can be done either conditionally or unconditionally. The user al- so has the ability to per- form the above functions on multiple storage areas. The results can then be stored in one of the storage areas that the user is working with or a completely separate storage area.

The basic reports are also quite versatile. The user can generate reports on MAIN STORAGE AREAS or INDEX STORAGE AREAS. These reports can be in the form of an area dump or in indexed order and have the ability to report the main storage file record number (something that was not a- vailable on OS-DMS). The re- ports can be printed on screen or printer, and because the Systems Generator uses PRTMAP the user is no longer stuck with using the device 5 printer only. He now has his choice between device 3, 5, or 8. The reports can be printed in the old fashioned DMS way (ie. horizontal or vertical) or they can be printed on paper exactly the way the input screen was designed.

Think you've heard it all? The best is yet to come!!

There are three more features available with the Systems Generator that really make it shine. They are: design sin- gle or multiple storage area updates, design custom re- ports, and design user menus. In designing updates and re- ports, the user now has the opportunity not only to per- form very sophisticated up- dates and print reports that once took laborious time and effort in trial and error to get the proper layout, he now can store all of these para-

meters. There is even a routine that allows the user to define and store the length of a form and place information virtually any place on the paper he wishes. This is extremely important and takes all of the hassle out of hardcoding a program to fill in a preprinted form. The next time the user needs this information, all he has to do is to call the reports or the updates from their respective menus and the system will act automatically, even to the point of accessing the correct main storage areas and index storage areas.

The "Design A User Menu" portion of the Systems Generator is the most impressive portion of the system. It allows the user to define his own sub menu inserting only those options that are needed for the particular data base he is working with. When this is done, he does not have to know the names of any of the storage areas that he is going to work with, all he has to do is select his menu and password and the system opens all the storage areas pertaining to that menu. It also opens all the index files and performs automatic updates, thus allowing even the most computer illiterate person to

become quickly conversant with his data base.

Now that you have read the meat of this review, you are probably saying to yourself, "sounds good for someone who is going to start a new data base, but I don't want to spend the time to convert or rekey all the information from my current OS-DMS files to the Systems Generator." Never fear, Comptrol Systems comes through again! They have provided a nice little routine for converting OS-DMS files to the System Generator files. All the user has to do is create a MAIN STORAGE area with the Systems Generator with fields the same size as the fields in the DMS files and run the "Convert an OSI-DMS file to a SYSGEN 83.1 Main Storage Area" routine and all the work is done by the machine. This may take a while, but I would much rather have the computer do the work while I relax with my coffee.

For those of you who like to do their own custom programming, the manual supplied with the SYSTEMS GENERATOR provides plenty of help. It lists all the error codes that you may encounter, complete storage area layouts, and listings of the storage area variables,

and special variables inherent in the system.

What about bugs and fixes? Comptrol Systems has sent us a couple fixes for bugs that they have discovered. One was minor and only required changing a few lines of code. The other was quite lengthy, but to show that they are more than willing to support their system, the letter that came with the changes states, "if you want, you may send us your original diskettes for the SYSTEMS GENERATOR - VERSION 83.1 and we will refresh them AT NO COST TO YOU other than postage to us." Who could ask for more.

If you are still not sure if the SYSTEMS GERERATOR is the right data base for you, then read the Comptrol Systems ad. Call them with your questions. We have found them easy to get hold of and quite helpful with all of our questions. After you have talked to them, if you are still not sure about their system, then take them up on their FREE 30 days trial. We are sure that once you have had hands on experience with the SYSTEMS GENERATOR you won't want to give it up.

★

```
800  P1       JSR CNSLIN       CHECK CONSOLE FOR KEYPRESS
810           BEQ P0           NOTHING? => P0
820           CMP #CTRLD       KEYPRESS! <CTRL>'D'?
830           BEQ TGLDUP       YES! TOGGLE DUPLEX
840           CMP #CTRLB       <CTRL>'B'?
850           BNE P2           NO => P2
860           JMP B1           YES! EXIT =>B1
870  P2       CMP #CTRLI       <CTRL>'I'
880           BNE P3           NO => P3
890           JMP TGLSAV       YES! TOGGLE SAVE ON/OFF
900  P3       CMP #RUBOUT      LC <RUBOUT> ?
910           BNE P4           NO => P4
920           SBC #$20         YES! CHANGE TO UPPER (C=1)
930  P4       CMP #RUBOUT-$20  <RUBOUT>?
940           BNE P8           NO => P8
950           LDA #BS          YES! FIRST LOAD BACKSPACE
960           JSR OUTDO        PRINT IT
970           LDA #SP          THEN LOAD A SPACE
980           JSR OUTDO        PRINT IT TOO
990  P5       LDA #BS          GET BACKSPACE AGAIN
1000 P6       JSR XMIT         SEND CHARACTER TO HOST
1010 P7       BIT CNSLOU       ECHO TO CONSOLE IF FULL DUPLEX
1020          JMP P0           AND LOOP
1030 P8       CMP #CTRLO       <CTRL>'O'?
1040          BNE P6           NO => P6
1050          JMP SEND
1060 ;
1070 B1       LDA SAVFLG       CHECK IF SAVE IS ON
1080          BEQ BACK         NO ==> BACK
1090          LDA #$04         YES! SHOW BUFFER DIRTY
1100          STA CMDFLG       SAVE IN COMMAND FLAG
1110 ;
1120 BACK     LDA #$00         INIZ MSB TO 0
1130          LDY CMDFLG       LOAD CMDFLG AS LSB
1140          STA CMDFLG       CLEAR CMDFLG
1150          LDX STKPTR       GET PROPER STACK POINTER
1160          TXS              MAKE IT CURRENT
1170          JMP ($0008)      GIVE CMD. TO BASIC & GO BACK
1180 ;
1190 TGLSAV LDA SAVFLG         IS SAVE ON ?
1200          BNE TGLOFF       YES! TURN IT OFF
1210          LDA #$01         NO, SET UP FOR TGLON
1220          STA CMDFLG       GIVE TO COMMAND FLAG
1230          JMP BACK         GO BACK TO BASIC FOR NOW
1240 TGLON    LDA FACLO        GET RESULT OF ATTEMPT
1250          AND #$F0         CHECK ERROR STATUS
1260          BEQ TGLON1       O.K. TURN SAVE ON
1270          JMP P0           PROBLEM! RE-ENTER MAIN LOOP
1280 TGLON1 LDA #$01           INIZ
1290          STA SAVFLG       SHOW SAVE IS ON
1300          LDA #BUFFER      RESET BUFFER POINTER
1310          STA SAVD+1       GIVE IT TO SAVE
1320          LDA #BUFFER/256
1330          STA SAVD+2       DON'T FORGET MSB
1340          STA SAVC+2       ALSO SET EOF MARKING PTR
1350          LDA #$00         INIZ
1360          STA INDEX        SET SAVE BUFFER POINTER
1370          JMP P0           RE-ENTER MAIN LOOP
1380 TGLOFF LDA #$00           INIZ
1390          STA SAVFLG       TURN SAVE OFF
1400          LDA #$02         SHOW RETURN TO TGLOF1
1410          STA CMDFLG       GIVE TO COMMAND FLAG
1420          JMP BACK         GO BACK SHOWING TURN SAVE OFF
1430 TGLOF1 JMP P0             RE-ENTER MAIN LOOP ON RETURN
1440 ;
1450 SAVE     LDY SAVFLG       SEE IF SAVE IS ON
1460          BNE SAVE1        YES! HANDLE IT => SAVE1
1470          RTS              NO! QUIT
1480 SAVE1    LDY INDEX        FETCH SAVE BUFFER POINTER
1490 SAVD     STA BUFFER,Y     SAVE CHARACTER IN BUFFER
1500          INY              BUMP INDEX
1510          BNE SAVE3        WATCH OFR PAGING
1520          INC SAVD+2       BUMP MSB ON PAGE CROSS
1530          INC SAVC+2       BUMP EOF MARKER MSB TOO
1540          LDA SAVD+2       CHECK MSB
1550          CMP #BUFFER/256+14
1560          BEQ SAVE4        BUFFER FULL! SAVE TO DISK!
1570 SAVE3    STY INDEX        SAVE BUFFER POINTER
1580          LDA #$FF         GET EOF MARKER
1590 SAVC     STA BUFFER,Y     SAVE IT IN BUFFER
1600          RTS              AND QUIT
```

Listing continued

```
1610;
1620 SAVE4    LDY #$00       INIZ OVERFLOW POINTER
1630          LDA #CTRLS     FETCH <CTRL>'S'
1640          JSR XMIT       TELL HOST TO STOP
1650 SAVE5    LDX #$40       INIZ DELAY COUNTER LSB
1660          STX COUNT      INIZ DELAY COUNTER MSB
1670 SAVE6    JSR XIN        CHECK HOST FOR INCOMING
1680          BCC SAVE7      NOTHING=> DECREMENT TIMER
1690          AND #$7F       MASK OFF PARITY BIT
1700          STA INBUF,Y    CHAR. REC'D, SAVE IT
1710          INY            BUMP POINTER
1720          BNE SAVE5      RESET TIMER AND LOOP
1730 SAVE7    DEX            DECREMENT TIMER LSB
1740          BNE SAVE6      LOOP 'TIL 0
1750          DEC COUNT      DECREMENT TIMER MSB
1760          BNE SAVE6      LOOP 'TIL TIME OUT
1770          STY FWDPT      SAVE OVERFLOW POINTER
1780          LDA #$03       GET COMMAND BYTE
1790          STA CMDFLG     SHOW RETURN TO SAVE8
1800          JMP BACK       RETURN TO BASIC
1810;
1820 SAVE8    LDY FWDPT      ANY OVERFLOW ?
1830          BEQ SAVE10     NOTHING? => SAVE10
1840          LDY #$00       INIZ
1850 SAVE9    LDA INBUF,Y    FETCH CHAR. FROM OVERFLOW
1860          STA BUFFER,Y   SAVE IN BUFFER
1870          STY COUNT      SAVE OVERFLOW POINTER
1880          JSR OUTDO      PRINT CHARACTER
1890          LDY COUNT      RETRIEVE POINTER
1900          INY            BUMP POINTER
1910          CPY FWDPT      ARE WE DONE ?
1920          BNE SAVE9      NO! LOOP!
1930          LDY FWDPT      FETCH OVERFLOW POINTER
1940 SAVE10   STY INDEX      SAVE AS BUFFER POINTER
1950          LDA #$FF       LOAD EOF CHARACTER
1960          STA BUFFER,Y   SAVE IT AT END OF OVERFLOW
1970          LDA #BUFFER/256
1980          STA SAVD+2     RESET BUFFER ADDRESS
1990          STA SAVC+2     AND EOF MARKING PTR
2000          LDA FACLO      REFETCH RESULT BYTE
2010          AND #$F0       MASK TO ERROR FLAG
2020          BNE SAVE12     ERROR! ==> SAVE12
2030 SAVE11   LDA #CTRLQ     LOAD <CTRL>'Q'
2040          JSR XMIT       TELL HOST TO CONTINUE
2050          LDX STKPTR     GET STACK POINTER
2060          TXS            MAKE IT CURRENT
2070          JMP P0         RE-ENTER MAIN LOOP
2080;
2090 SAVE12   LDA #$00       INIZ
2100          STA SAVFLG     TURN SAVE OFF
2110          BEQ SAVE11     AND CONTINUE
2120;
2130; ROUTINE TO SEND CHARACTER OUT MODEM PORT
2140;
2150 XMIT     PHA            SAVE CHAR. ON STACK
2160 X1       LDA STATUS     CHECK ACIA
2170          LSR A
2180          LSR A
2190          BCC X1         WAIT FOR CLEAR TO SEND
2200          PLA            RETRIEVE CHARACTER
2210 X2       STA MODEM      SEND IT
2220          RTS            QUIT
2230;
2240 CNSLIN   JMP $FFFF      CONSOLE INPUT ROUTINE
2250;
2260 CNSLOU   STA CNSLO1+1   SAVE DATA
2270          JSR OUTDO      PRINT CHARACTER
2280          LDX #$00       INIZ
2290          STX POSCNT     CLEAR BASIC'S LINE POINTER
2300 CNSLO1   LDA #$FF       RETRIEVE CHARACTER
2310          JMP SAVE       DO ANY SAVES AND QUIT
2320 ;
2330; SERIAL CONSOLE INPUT ROUTINE
2340;
2350 TTYIN    LDA $FC00      CHECK ACIA
2360          LSR A
2370          BCC TTYIN2     NOTHING THERE? => TTYIN2
2380 TTYIN1   LDA $FC01      YES! FETCH CHAR. FROM CONSO
2390          AND #$7F       MASK OFF PARITY
2400          RTS            NO, GO BACK WITH CHARACTER
2410 TTYIN2   LDA #$00       SHOW NOTHING REC'D
```
navigation
                                                          cont. page 16

boilerplate
COMPUTER REPAIRS
C1P - C2P - C4P

Have your personal computer serviced by a qualified technician familiar with OSI hardware. We will evaluate your computer and notify you of what should be done and how much it will cost. Any repairs will be made only by your approval. Please include a description of the problem if it is intermittent. Minimum charge is $20 whether repairs are made or not.

C1P SERIES I OWNERS: A 24/32 column video mod. is available! Go back to a 24 column display at any time with just a flick of a switch. Uses the prototype area without piggybacking any IC's. Only $89.95 installed.

Please include $4.50 return postage.

The Computer Shelter
8533 Pacific Hwy SE
Olympia, WA 98503

boilerplate
"Computer Business Software"

"CBS"

• **INTEGRATED BUSINESS SYSTEM**

— FEATURING —
• Accounts Receivable
• Inventory Control
• Order Entry/Invoicing
• Accounts Payable
• General Ledger
• Payroll

• **BUSI-CALC**
"An electronic worksheet"

— FEATURING —
• Local and General Formatting
• Replication
• Variable Column Widths
• Editing
• Insertion/Deletion of Rows and Columns
• Protected Entries
• Help Screen
• Flexible Printing
• Complete User Manual

MICROSOFTWARE INTERNATIONAL
3300 South Madelyn□Sioux Falls, SD 57106
1-800-843-9538

SOFTWARE FOR OSI

EXPLANATION OF LISTING CODES

BASIC Version No./
Minimum computer/
  1=SB,SBII,ClP,C2/4P 4=C4P
  8=C8P
  O=C2/3OEM
  D=C2/3-D
  2=C200,C3A/B
  3=C300

Minimum Storage required/
  C=Cassette
  5=5 1/4" MF
  8=8"FD
  7=CD-7
  2=CD-23/28/36/74
  digit following indicates
  number of devices required

Systems Supported/
  S=Single User
  M=Multi-User
  H=Hard Disk
  R=Record Locking
  record lock assumes multi-
  user. Two may be specified.

Software Support by/
  D=Dealer
  P=Phone
  M=Modem
  N=None
  O=Other

Sold by/
  A=Author
  D=Dealer
  M=Mail order
  O=Other

Copies in Circulation/
  No. divided by 10, ie.
  1=Less than 11
  11=100-110

Price/
  Dollars only, no cents,
  tax, shipping, etc.

OS65-D*BUSINESS*VIDEO

A/R-65
/4/51/S/O/A/1/ $89
Author:
STEVE DONACHIE
MIAMI, FL 33143
Seller: SAME

System of five programs allows
entry of data from invoices,
posts to customer accounts,
shows sales by categories,
prints monthly statements.
Single 5 in. drive version
holds 176 accounts. Dual
5 in. gives 416. dual 8 in.
816 accounts. Tested in use.
For manual only, send $5.00.

OS65-D*GAME*SERIAL & VIDEO

WARTS
1.0/1/C/S/N/A/O/ $10
Author:
RICHARD LIST
PITTSBURGH, PA 15221

Seller: SAME

Move toad across road without
getting hit by car. to island,
without getting eaten by
snake, then hop on logs and
turtles across river without
falling in water and drowning.
All machine language.

OS65-D*GAME*VIDEO

LUNAR LANDER
2.0/1/C/S/P/A/1/ $20
Author:
ARTHUR ROSS
SEATTLE, WA 98188
Seller: SAME

A new version of the classic
'Lunar Lander'. Lots of extra
features, and is well for-
matted for the ClP display.

SNAKE FIGHT
/1/C/S/N/M/1/ $6
Author:
HENRY KUSKA
AKRON, OH 44313
Seller:
STEVE KUSKA
SAME

Two player game using key-
board. Each snake tries to
bite off the other snake's
tail, one unit at a time.
until one is completely
eaten.

OS65-D*OTHER*SERIAL & VIDEO

SIGAVG
3.2/8/8/S/P/M/1/ $30
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

SIGAVG extracts time varying
signals which are buried in
noise. Samples signal at
rates of 10-5000 Hz. Can be
used as digital storage scope.
Requires A/D, D/A on 430
board, and 6522 on 500, 510
board. BASIC exec. with
calls to machine code.

OS65-D*OTHER*SERIAL

ANOVA
3.2/O/8/S/P/M/1/ $25
Author:
ROBERT T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

ANOVA is a general purpose
analysis of variance program.
It can analyse any design with
combination of within and be-
tween variables. as long as
CELLS <32,#SCORES/S<33 and
total factors is <10. Also,
contains a LOAD program to
enter data.

DECISION MAKING
3.2/O/8/S/P/M/1/ $25
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

Decision making is a set of
two programs to provide com-
puter assisted decision analy-
sis. DECMAK is based on a
linear model of weighted mean
ranks and provides sensitivity
analysis options. FDM is a
similar program based on fuzzy
set theory. Many options in
each.

MULREG
3.2/O/8/S/P/M/1/ $25
Author:
ROBERT T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

MULREG performs multiple
linear regression for data
files stored on disk. Com-
putes Regression Equation.
Var-Covar Matrix, Coef.
Estimates. Residual Plots.
Residual Listing. Program
LOAD permits loading data
on disk.

OPTICAL
3.2/O/8/S/P/M/1/ $40
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

OPTCAL is an optimization
program which solves for Y as
a function of up to 4 input
variables which can be ex-
pressed as linear, squared or
interaction terms. Ranges
for the 4 X-vars. can be
specified. Designed to be
used with good multiple
regression program.

PRCOMP
3.2/O/8/S/P/M/1/ $150
Author:
ROBERT T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

PRCOMP is a series of 18
compatible programs for ac-
quiring and analyzing paired-
comparison data. Includes
interactive data gathering
using a sort algorithm. If
you know what this means.
you know this is a real
buy!

PRINCP
3.2/O/8/SH/P/M/1/ $20
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

PRINCP performs principal
components analysis of data
previously stored on disk.
Data Matrix size limited

only by memory. Program extracts up to 5 components or roots.

OS65-D*OTHER*VIDEO

CASH FLOW
HEXD/1/C1/S/P/A/1/ $20
Author:
JOHN T. ROECKER
MINNEAPOLIS. MN 55410
Seller: SAME

A household cash flow/budget program which categorizes checks into budget categories. Monthly and yearly totals are reported. Checks may be sorted by number. budget category, or amount. Disk or tape files are maintained for each month. Also available in 5 1/4" MF format.

CHECKWRITER
3.3/4/51/S/P/A/1/ $19
Author:
RUDY POLACEK
WOODSIDE. CA 94062
Seller: SAME

User-defined one letter code, and amount to print a complete check. Allows for input of checks that are written away from the home. Uses NEBS check forms and has your code letters and balance always visible on the screen. Accumulates year to date total for taxes and budget.

LONG DIVISION WITH REMAINDER
1.43/1/C/S/N/M/1/ $6.
Author:
HENRY KUSKA
AKRON, OH 44313
Seller:
STEVE KUSKA
SAME

Division drill using split screen. Student does problem on one side, computer then does the problem on the other side.

QUICKTEXT
1.0/1/C/S/P/A// $30
Author:
ARTHUR F. ROSS
SEATTLE. WA 98188
Seller: SAME

A bargain basement text editor. Line oriented, uses simple commands for altering and formatting text, and includes a variety of print options and storage capabilities using cassettes.

SCHOOL PROGRAMS
HEXD/1/C1/S/P/A/1/ $15
Author:
JOHN T. ROECKER
MINNEAPOLIS. MN 55410
Seller: SAME

Two programs, spelling list

and math tutor. which are useful for school children. Graphics are used to make the drill entertaining. Only positive feedback is provided to the student. Also available in 5 1/4" MF format.

OS65-D*UTILITY*SERIAL & VIDEO

CORRECTED GARBAGE COLLECTOR
ROM/1///O/M/3/ $15
Author:
SOFTWARE SOLUTIONS
P. O. BOX 3753
SEATTLE, WA 98124
Seller: SAME

Fully corrected machine code to solve infamous OSI/MICRO-SOFT ROM BASIC garbage collector problem(subscripted string bug).Implemented in a replacement BASIC3 EPROM. Installation requires the cutting of 3-5 board traces (depending on model) and the addition of 3-5 jumpers. 100% guarantee.

ENHANCED MONITOR EPROM
ROM/1///O/M/2/ $15
Author:
SOFTWARE SOLUTIONS
P. O. BOX 3753
SEATTLE, WA 98124
Seller: SAME

Monitor EPROM for the Series 2 C1P/SB. HEXDOS compatible. Corrected keyboard routine.one key output control, screen editor, BASIC shorthand, terminal emulator with up/download & protocol control, 65V compatible machine code dump, screen print. & one key jump to machine code. 100% guarantee.

ENHANCED OS65D 3.3
3.3/1/51/S/O/A/2/ $30
Author:
RAY LYDON
GRAFTON, OH 44044
Seller: SAME

Many new functions such as sys command for catalog, 10 active files. end of file/end of volume processing, background printing, file append, dynamic file buffers, command file processing, reads long strings from disk and keyboard, plus much more!

ENHANCED ROM BASIC
ROM/1///O/M/2/ $30
Author:
SOFTWARE SOLUTIONS
P. O. BOX 3753
SEATTLE. WA 98124
Seller: SAME

Two EPROMs to replace BASIC1/4 in the Series 2 C1P/SB. Enhanced video driver with one key control of 24/48 format. one key screen clear. & true backspace. Corrected error codes

& modified input routine to allow all 96 printable ASCII characters. Requires board modifications. 100% guarantee.

EXTENDED ARITHMATIC PROCESSOR
ALL/O/81/M/D/A/11/ $95
Author:
NORTHEAST FINANCIAL SYSTEMS
WEST NYACK, NY 10994
Seller: SAME

Transient utility. Permits 13 digit input and 23 digit output precision. Has complete rounding function. Replaces NULL command. 39 page manual with examples. 8" disk only.

PACKER
1.0/1/C/S/N/A/*/ $13
Author:
RICHARD LIST
PITTSBURGH, PA 15221
Seller: SAME

Removes unnecessary spaces from BASIC program. combines lines if possible, optionally removes REMs. Fast. all machine language. Shortens program and makes it run more efficiently.
* 200 copies of original BASIC version sold by Aardvark.

PIA-COMM
/4/51/S/O/A/1/ $35
Author:
STEVE DONACHIE
MIAMI, FL 33143
Seller: SAME

Transfer text, programs and data files between 5 and 8 in. disks by connecting the PIA ports of two C-2/4/8 computers (A-15 board ports A or B) leave serial port free for terminal or printer. Includes software on both sizes of disks and instructions for cable and operation.

OS65-D*UTILITY*VIDEO

AUTOLOAD GENERATOR
/1/C1/S/P/A/1/ $18
Author:
JOHN S. FINE
MEDFORD, MA 02155
Seller: SAME

If you load any large machine language programs (your own or purchased) from cassette. use the AG to make loading much easier. The AG translates tapes from the assembler output format to an autoload format which is about 2.5 times faster and much more reliable.

FULL-VIEW BASIC EDIT
/4/51/S/P/A/1/ $20
Author:
MIKE F. PUTNAM
ROSEVILLE. MN 55113

Seller: SAME

This editor features a full screen window into a BASIC program. It can be moved forward or backward a line or a screen at a time. Has 27 functions for edit, text search, and screen dump. Uses top 4K of memory-all machine code. C2/4P-MF or C4P-MF. State memory size.

FULL-VIEW DISK EDIT
/4/51/S/P/A/1/ $15
Author:
MIKE F. PUTNAM
ROSEVILLE. MN 55113
Seller: SAME

Provides full screen viewing and editing of disk data. Simple keystrokes control disk read, write. and track stepping. Data displayed in Hex and ASCII. Has address column which allows matching data with its actual memory address. Ideal for making OS patches. C2/4P and C4P only.

OSI BASIC UTILITIES
1.0/1/C/S/P/A/1/ $100
Author:
ARTHUR F. ROSS
SEATTLE. WA 98188
Seller: SAME

A group of very useful utilities for the C1P, including program renumbering, variable crossreferencing, memory mapping, and several others. Easy to use (menu-driven) and invaluable for program development and debugging.

SUPER MONITOR AND COMPRESS
HEXD/1/C1/S/O/M/1/ $12
Author:
JERRY E. TRAVIS
OLYMPIA, WA 98503
Seller:
THE COMPUTER SHELTER
SAME

A machine code monitor for use with BASIC programs. Useful for writing USR subroutines. patching chains.making machine code tapes. COMPRESS removes non-executable blanks and/or REMs from BASIC programs freeing up space. Ten page manual included. Also available in 5 1/4 MF format.

OS65-D*UTILITY*SERIAL

THE BROWSE SYSTEM
1.2+/O/8/SH/P/M/3/ $50
Author:
LEE CONYERS
RESTON, VA 22091
Seller: SAME

65U programmer support aids. Nine programs. Menu driven and FULL-SCREEN ORIENTED. Disk sector editor. RAM editor.

Two 6502 DISASSEMBLERs for object code resident on disk or in system memory. Utility for CRT code testing. On-line documentation facility. Annouced in Aug '83 ad in Peek.

OS65-U*BUSINESS*SERIAL

AUTO BANK - SIGMA
1.43/2/21/HR/M/A/1/ $
Author:
SIGMA, INC.
CP 5
LEVIS, QUEBEC G6V 6N6
Seller: SAME

Complete banking system with interests calculation (daily if wanted) and automatic transactions. 40 digits precision. Prints customer's monthly statement. Bundle of reports. First version has been working for 2 years. Also works fine under 65-U network and DB-1's.

BANKING SYSTEM
1.2/2/21/HR/M/A/1/ $
Author:
SIGMA, INC.
CP 5
LEVIS. QUEBEC G6V 6N6
Seller: SAME

Complete banking system with interests calculation (daily if wanted), and automatic transactions. 40 digits precision (uses X-tra Precision Basic). First version running for 2 years. 15,000 accounts. 15 types of account. Biggest 65-U system we have heard of.

CANADIAN PAYROLL
1.43/O/82/MH/M/A/1/ $995
Author:
SIGMA, INC.
CP 5
LEVIS, QUEBEC G6V 6N6
Seller: SAME

Complete payroll with all deductions possible. Currently used in Quebec and New Brunswick by different companies (construction. retail. etc...) Very versatile and easy to use.

CHECKPOINT-OF-SALE
1.42/D/71/MH/M/A/4/ $6950
Author:
SILEO COMPUTER SYSTEMS
DENVER. CO 80209
Seller: SAME

Completely interactive P.O.S. Retail management system featuring price look up cash ticketing or invoicing, quotes. purchasing, price tags, sales analysis, complete accounting handles 5 stores-150,000 items perfect for auto parts, hardware,sporting goods.hobby/toy clothing,shoe&furniture stores

COMPREHENSIVE INSURANCE SYSTEM
1.44/D/71/M/O/D/4/ $
Author:
LEON HAVERLY
MARIETTA, GA 30064
Seller:
COMPUWORK
ATLANTA, GA 30341

Accounting, management, Marketing, Client Services, Word Processing, Utilities. Rating.
A total system for independent agents.

IHS ACCOUNTS RECEIVABLE
1.42/O/82/SH/M/A/2/ $395
Author:
IHS COMPUTER SERVICE
ROUTE 1, BOX 201B
PORT REPUBLIC, VA 24471
Seller: SAME

Used in IHS Bookkeeper. System menu driven/screen formatted. Employs Master & Transaction Files plus Customer Information File. Functions include: invoicing, sales journal, balance frwd statements w/aging; payment by invoice;auto and/or cyclic billing; more. Manual - $20. Dealer inquiries invited.

IHS BOOKKEEPER
1.42/D/71/R/M/A/1/ $1500
Author:
IHS COMPUTER SERVICES
ROUTE 1, BOX 201B
PORT REPUBLIC, VA 24471
Seller: SAME

IHS Bookkeeper is a completely integrated accounting system including Point of Sale invoicing & inventory control, A/R. G/L, A/P. Other functions available. May be customized. Uses DMS files. Runs on DB's. Brochure - $2. Manual - $40. Dealer inquiries invited.

IHS GENERAL LEDGER
1.42/O/82/SH/P/A/2/ $395
Author:
IHS COMPUTER SERVICES
ROUTE 1, BOX 201B
PORT REPUBLIC, VA 24471
Seller: SAME

Used in IHS Bookkeeper Features include: Menu driven; screen formatting, entries labeled by subsidiary ledger; optional profit centers; optional budgeting; any numbering system may be used; cash receipts/disbursements journal included. Manual $20. Dealer inquiries invited.

PAYROLL
1.2/O/81/S/P/A/1/ $75
Author:
PATRICK CLUSMAN
FOND DU LAC, WI 54935
Seller: SAME

Payroll system with federal and state of Wisconsin tax

table. Modifiable. Supports
300 employees. reports with
check printing possible.

SOFTWEAR 65U
1.43/2/21/HR/M/A/1/ $4500
Author:
SIGMA, INC.
CP5
LEVIS, QUEBEC G6V 6N6
Seller: SAME

A total clothing merchandising
management system for company
having from 2 to 50 stores.
Takes care of purchase orders,
distribution. transfers,
sales. receipts and returns.
markdowns, supplier's record,
style inquiry and reports.
employee productivity, etc...

OS65-U*GAME*SERIAL

GAME DISK
1.2/O/81/S/N/O/1/ $25
Author:
PATRICK CLUSMAN
FOND DU LAC, WI 54935
Seller: SAME

Traditional computer games
keyed in to run under OS-65U
more than 10 games along with
some miscellaneous programs
and utilities.

OS65-U*OTHER*SERIAL & VIDEO

SIGAVG
1.2/8/8/S/P/M/1/ $30
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

SIGAVG extracts time varying
signals which are buried in
noise. Samples signal at
rates of 10-5000 Hz. Can be
used as digital storage scope.
Requires A/D, D/A on 4300
board, and 6522 on 500, 510
board. BASIC exec. with
calls to machine code.

OS65-U*OTHER*SERIAL

BOWL
1.2/O/82/S/P/A/1/ $25
Author:
PATRICK CLUSMAN
FOND DU LAC, WI 54935
Seller: SAME

Supports up to 12 teams, 9
bowlers per team. set for 33
week season. Modifiable.
SASE for more information.

DATA MANAGEMENT SYSTEM
1.4/O/81/SH/P/A/5/ $250
Author:
STAN SADLER
RITTMAN, OH 44270
Seller: SAME

User Friendly - Screen

oriented data management
system. Designed by a pro.
Use to build data files for
programs or for simply
entering and maintaining data.
Add, change, delete. compress.
sort, find, and list records.
Try it - You'll never use
another. Learn in minutes.

DECISION MAKING
1.2/O/8/S/P/M/1/ $25
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

Decision making is a set of
two programs to provide com-
puter assisted decision anal-
ysis. DECMAK is based on a
linear model of weighted mean
ranks and provides sensitivity
analysis options. FDM is a
similar program based on
fuzzy set theory. Many
options in each.

ENERGY
1.2/O/81/S/N/O/1/ $SASE
Author:
PATRICK CLUSMAN
FOND DU LAC, WI 54935
Seller: SAME

Home energy analysis program.
Worksheet and documentation
from various sources.

GASMIL
1.2/O/81/S/N/A/1/ $10
Author:
PATRICK CLUSMAN
FOND DU LAC, WI 54935
Seller: SAME

Simple gas mileage tracking
aid. Keeps total miles
driven, dollars spent and
gallons used. Simple
projections from date to year
end.

HORSE RACING ANALYSIS
1.4/O/81/SH/P/A/1/ $550
Author:
STAN SADLER
RITTMAN, OH 44270
Seller: SAME

Use for handicapping or
claiming horses. Technical
program to calculate true
speed of horses by calculating
track variant. Distance and
condition of race used in var-
iant calculation with itera-
tive procedures. Must be know-
ledgeable in racing to use.
Guaranteed results.

PRINCP
1.2/O/8/SH/P/M/1/ $20
Author:
R. T. KINTZ
ROCHESTER, NY 14610
Seller: SAME

PRINCP performs principal
components analysis of data
previously stored on disk.

Data Matrix size limited only
by memory. Program extracts
up to 5 components or roots.

RECORDKEEPING SYSTEM
1.4/O/81/SH/P/A/5/ $350
Author:
STAN SADLER
RITTMAN, OH 44270
Seller: SAME

Designed for the person who
has to fill out tax forms -
Sch. C or Sch. F. Uses data
management system for entering
and maintaining all business
transactions. including income
and expenses. Generates
reports for monthly. detail.
and summary transactions.
Simple, yet efficient.

THE SCRIBE WORD PROCESSOR
1.42/8/81/SM/P/M/2/ $195
Author:
IHS COMPUTER SERVICES
ROUTE 1, BOX 201B
PORT REPUBLIC, VA 24471
Seller: SAME

A screen oriented Word Proces-
sor fully interactive with DMS
files. Text entry & edit fea-
ture: Automatic wraparound,
full cursor control, search &
replace, block xsfer. insert/
delete text, etc. Imbedded
commands for tabs, indent.
underline. center. etc. Also
written for Denver Boards.

OS65-U*UTILITY*SERIAL & VIDEO

EXTENDED ARITHMATIC PROCESSOR
ALL/O/8/M/D/A/11/ $95
Author:
NORTHEAST FINANCIAL SYSTEMS
WEST NYACK, NY 10994
Seller: SAME

Transient utility. Permits
13 digit input and 23 digit
output precision. Has com-
plete rounding function.
Replaces NULL command.
39 page manual with examples.
8" disk only.

OS65-U*UTILITY*SERIAL

CSS MENU MANAGER
1.2/O/8/SH/P/M/1/ $50
Author:
LEE CONYERS
RESTON, VA 22091
Seller: SAME

Professional menu management
system to provide that desired
USER-FRIENDLY front-end to 65U
applications. Uses advanced
features of many CRTs. Main
menu, up to 8 sub-menus, up to
9 selections each. Turn menus
off to command mode. Multiple
response types for traversing
menus. Keywords. HELP feature.

★

15

Listing continued from page 9

```
2420        RTS              AND QUIT
2430;
2440; MODEM INPUT ROUTINE
2450;
2460 XIN   LDA STATUS       CHECK ACIA STATUS
2470       LSR A
2480       BCC X4           NOTHING THERE? => X4
2490 X3    LDA MODEM        YES! FETCH CHARACTER
2500 X4    RTS              QUIT (CARRY SHOWS CHAR. REC'
2510;
2520 GETVAR JMP ($0006)     GET VALUE OF (CM)
2530;
2540 KBPOLL LDX    #$01     INIZ
2550       JSR    POLL      POLL CONTROL ROW
2560       PHA              SAVE RESULT ON STACK
2570       INX              (X=2)
2580       LDY    #$06      INIZ ROW COUNTER
2590 KBP1  JSR    POLL      POLL ROW OF KEYBOARD
2600       BNE    KBP2      KEYPRESS? ==> KBP2
2610       DEY              NO, DECREMENT ROW COUNTER
2620       TXA              PUT X IN ACC.
2630       ASL    A         *2
2640       TAX              PUT BACK IN X
2650       BCC    KBP1      LOOP TIL ALL ROWS DONE
2660       TAY              CLEAR Y   (ACC.=0)
2670       BCS    KBP4      AND JUMP ==> KBP4 (BRA)
2680 KBP2  PHA              SAVE RESULT ON STACK
2690       TYA              PUT ROW NUMBER IN ACC.
2700       ASL    A         *2
2710       ASL    A         *4
2720       ASL    A         *8
2730       STA    KBDAT     SAVE RESULT
2740       PLA              RETRIEVE POLL RETURN DATA
2750       LDX    #$FF      INIZ X TO -1
2760 KBP3  INX              BUMP X
2770       ASL    A         DATA * 2
2780       BCC    KBP3      LOOP 'TIL CARRY=1
2790       TXA              PUT RESULT IN ACC.
2800       ADC    KBDAT     ADD ROW INDEX
2810       TAX              PUT RESULT IN X
2820       LDY TABLE-1,X    GET CHARACTER FROM TABLE
2830 KBP4  TYA              PUT CHARACTER IN ACC.
2840       BEQ    KBP6      NO KEYVALUE? ==> KBP6
2850       PLA              RETRIEVE CONTROL ROW POLL
2860       BMI    KBP5      HANDLE SHIFTS AND <CTRL> KEY
2870       CPY    LSTKEY
2880       BNE    KBP5
2890       LDY    #$00
2900       BEQ    KBP11
2910 KBP5  PHA
2920       AND    #$20
2930       BEQ    KBP6
2940       CPX    #$0C
2950       BCS    KBP6
2960       STY    LSTKEY
2970       LDY    #$1B
2980       STY    KBDAT
2990       PLA
3000       JMP    KBP11
3010 KBP6  PLA
3020       STY    LSTKEY
3030       CPY    #CR
3040       BEQ    KBP11
3050       CPY    #SP
3060       BEQ    KBP11
3070       CPY    #$00
3080       BEQ    KBP11
3090       PHA
3100       AND    #$07
3110       LDX    #$20
3120       CPY    #$00
3130       BPL    KBP7
3140       AND    #$06
3150       LDX    #$10
3160 KBP7  LSR    A
3170       BCC    KBP8
3180       BEQ    KBP9
3190       LDX    #$30
3200       .BYTE SKIP2
3210 KBP8  BEQ    KBP10
3220 KBP9  TXA
3230       EOR    LSTKEY
3240       TAY
3250 KBP10 PLA
3260       AND    #$40
3270       BEQ    KBP11
3280       TYA
3290       AND    #$1F
3300       TAY
3310 KBP11 LDX    #$08   DELAY LOOP
3320 KBP12 DEC    KBDAT
3330       BNE    KBP12
3340       DEX
3350       BNE    KBP12
3360       TYA          PUT CHARAC. IN ACC.
3370       AND    #$7F   MASK OFF BIT 7
3380       STA    KBDAT SAVE IT
3390       RTS          AND RETURN
3400 ;
3410 POLL  TXA
3420       EOR    #$00
3430       STA    KYBD
3440       STA    KYBD
3450       LDA    KYBD
3460       EOR    #$00
3470       RTS
3480 ;                Listing cont. page 17
```

```
3490 TABLE   .BYTE $B1,$B2,$B3,$B4,$B5,$B6,$B7,$00
3500         .BYTE $B8,$B9,$B0,$BA,$AD,$7F,$00,$00
3510         .BYTE $AE,$6C,$6F,$8A,$0D,$00,$00,$00
3520         .BYTE $77,$65,$72,$74,$79,$75,$69,$00
3530         .BYTE $73,$64,$66,$67,$68,$6A,$6B,$00
3540         .BYTE $78,$63,$76,$62,$6E,$6D,$AC,$00
3550         .BYTE $71,$61,$7A,$20,$AF,$BB,$70,$00
3560 KBDAT   .BYTE $00
3570 LSTKEY  .BYTE $00
3580 CMDFLG  .BYTE $00
3590 SAVFLG  .BYTE $00
3600 INDEX   .BYTE $00
3610 FWDPT   .BYTE $00
3620 COUNT   .BYTE $00
3630 INBUF   .BYTE $00,$00,$00,$00,$00,$00,$00,$00
3640 STKPTR  .BYTE $00
3650 TMP     .BYTE $00                TEMPORARY STORAGE
3660 KEYNUM  .BYTE $00                BASIC TOKEN STORAGE
3670 REG1    .BYTE $00,$00,$00        MATH REGISTER #1
3680 REG2    .BYTE $00,$00,$00        MATH REGISTER #2
3690 RESLO   .BYTE $00                RESULT LSB
3700 RESHI   .BYTE $00                RESULT MLSB
3710 FIFTH   .BYTE $00                RESULT MSB
3720 POWER   .BYTE $00                NUMBER CONVERSION EXP.
3730 NUMBER  .BYTE $00,$00,$00,$00,$00,$00,$00
3740 TOTAL   .BYTE $00,$00,$00,$00
3750 TBL1    .BYTE $01,$16,$56,$96,$36,$76
3760 TBL2    .BYTE $00,$00,$02,$40,$55,$85
3770 TBL3    .BYTE $00,$00,$00,$00,$06,$04
3780 TBL4    .BYTE $00,$00,$00,$00,$00,$01
3790;
3800 HEXADD  PHA          SAVE ORIGINAL BYTE
3810         JSR SHIFT    SHIFT 4 BITS
3820         JSR HADD     ADD FOR THIS #
3830         PLA          RETRIEVE ORIGINAL BYTE
3840         AND #$0F     MASK TO LOW NYBBLE
3850         DEC POWER    ADJUST INDEX
3860         JSR HADD     ADD FOR THIS #
3870         DEC POWER    ADJUST INDEX
3880         RTS          QUIT
3890 HADD    LDY POWER    GET INDEX TO TABLE
3900         TAX          SET COUNTER
3910         BEQ HADD2    IF 0, NO ACTION NEEDED
3920 HADD1   SED
3930         CLC
3940         LDA TBL1,Y
3950         ADC TOTAL
3960         STA TOTAL
3970         LDA TBL2,Y
3980         ADC TOTAL+1
3990         STA TOTAL+1
4000         LDA TBL3,Y
4010         ADC TOTAL+2
4020         STA TOTAL+2
4030         LDA TBL4,Y
4040         ADC TOTAL+3
4050         STA TOTAL+3
4060         CLD
4070         DEX
4080         BNE HADD1    LOOP 'TIL COUNTER ZEROES
4090 HADD2   RTS          QUIT
4100;
4110 HEXDEC  LDA #$00     ROUTINE TO CONVERT HEX NUMBER
4120         TAY          IN RESULT REGISTER TO
4130 HEXD1   STA POWER,Y  DECIMAL ASCII
4140         INY
4150         CPY #$C
4160         BNE HEXD1
4170         LDA #$05
4180         STA POWER
4190         LDA FIFTH
4200         JSR HEXADD
4210         LDA RESHI
4220         JSR HEXADD
4230         LDA RESLO
4240         JSR HEXADD
4250         LDX #$03
4260         LDY #$00
4270 HEXD2   LDA TOTAL,X
4280         PHA
4290         JSR SHIFT
```

OPTIMIZING OSI BASIC PROGRAMS - PART II

By: Dave Rich
Courtesy of Osmosus News

The last installment included several suggestions for speeding up your programs. Some of you may have recognized the loop example as a REPEAT.. UNTIL loop, not a WHILE loop. The WHILE loop would be coded:

```
IF T>25 THEN jump around loop
FOR X=0TO-1STEP-1
   .
   .
   .
X=(T>25):REM WHILE T<=25
NEXT
```

FOR-loops

One very basic means of speeding up programs is to carefully examine the code within FOR loops for any statements which can be placed outside of the loops. This is a technique used by all optimizing compilers and one you can use.

Check each assignment statement. If none of the variables on the right side of the statement are redefined within the loop, or in subroutines called from within the loop, place the statement before the FOR statement. If all items on the right side of the statement are numeric or string constants, the statement can probably be placed ahead of the FOR statement.

Be sure to check program logic to make sure that the candidate statement can really be moved. Statements which reinitialize variables must remain in the loop.

STRING CONCATENATION (or, how to avoid the collector)

The garbage collector is very important to speed but is hard to explain in a few words. You are referred to articles on how Microsoft BASIC handles strings and garbage collection.

'Garbage collection' is a process used by BASIC to clean up, or compact, variable storage when it becomes cluttered with no-longer-needed strings. Every string concatenation (A$=B$+C$+D$) leaves garbage. Constant concatenation causes frequent garbage collection, the most time-consuming process in BASIC.

Programs which build destination strings from source strings, such as command par-

```
4300         STA NUMBER,Y
4310         INY .
4320         PLA
4330         AND #$0F
4340         STA NUMBER,Y
4350         INY
4360         DEX
4370         BPL HEXD2
4380         RTS              STOP! NO PRINT!
4390 ;
4400 SHIFT LSR A
4410       LSR A
4420       LSR A
4430       LSR A
4440       RTS
4450;
4460; BASIC FILE TRANSLATE & SEND ROUTINE
4470;
4480 SEND  LDA SAVFLG        CHECK IF SAVE IS ON
4490       BEQ SEND0         NO, O.K. ==> SEND0
4500       JMP P0            YES! IGNORE COMMAND
4510 SEND0 LDA #$05          INIZ
4520       STA CMDFLG        SAVE FILE OPEN COMMAND
4530       JMP BACK          AND GO BACK TO BASIC
4540 ;
4550 SEND1 LDA FACLO         GET REQUEST RESULT
4560       AND #$10          CHECK FOR ERROR
4570       BEQ SEND2         O.K. ==> SEND2
4580       JMP P0            ERROR! QUIT
4590 SEND2 LDA #BUFFER+21    SKIP EMBEDDED HEADER
4600       STA TXTPTR
4610       LDA #BUFFER/256
4620       STA TXTPTR+1
4630       JSR CHRGET        SKIP ADDRESS BYTES
4640 BAS1  JSR ASMLIN        FETCH & PRINT LINE #'S
4650 BAS2  JSR CHRGET        FETCH CHARACTER
4660       BMI BAS4          KEYWORD TOKEN? => BAS4
4670       BEQ BAS3          E.O.L.? => BAS3
4680       JSR BAS11         IT'S JUST TEXT, PRINT IT
4690       JMP BAS2          AND LOOP TO GET NEXT CHARAC
4700 BAS3  JSR SCRLF
4710       JSR CHRGET        GET LSB NEXT LINE
4720       STA TMP           SAVE IT
4730       JSR CHRGET        GET MSB NEXT LINE
4740       BNE BAS5          NOT 0 => BAS5
4750       LDA TMP           GET LSB AGAIN
4760       BEQ BAS13         IT WAS 0 TOO! QUIT
4770 BAS5  JMP BAS1          START NEW LINE
4780;
4790 BAS4  JSR TOKOUT        PRINT KEYWORD
4800       JMP BAS2          AND LOOP
4810 BAS11 JSR CNSLOU        PRINT TO CONSOLE
4820       JSR XMIT          SEND TO MODEM
4830       JSR CKRDY         SEE IF HOST SENT <CTRL>'S'
4840       JSR CNSLIN        CHECK KEYBOARD
4850       BEQ BAS12         NO KEYPRESS? => BAS12
4860       CMP #CTRLC        YES, BUT WAS IT <CTRL>'C' ?
4870       BEQ BAS13         YES! QUIT RIGHT NOW!
4880 BAS12 RTS              NO, GO BACK
4890 BAS13 LDX STKPTR
4900       TXS
4910       JSR SCRLF
4920       LDA #$00
4930       STA CMDFLG        CLEAR COMMAND FLAG
4940       JMP P0
4950 ;
4960; TRANSLATE TOKEN TO ASCII
4970;
4980 TOKOUT AND #$7F         STRIP TOKEN OFFSET
4990        STA KEYNUM       SAVE AS KEYWORD NUMBER
5000        LDY #$00         INIZ Y
5010        LDX #$00         INIZ X
5020 TOK1   CPX KEYNUM       AT KEYWORD YET ?
5030        BEQ TOK5         YES! => TOK5
5040 TOK2   LDA KEYLST,Y     NO, GET NEXT KEYWORD CHAR.
5050        BMI TOK3         END OF KEYWORD => TOK3
5060        INY              NO, BUMP INDEX TO KEYWORD LI
5070        BNE TOK2         AND LOOP
5080 TOK3   INX              BUMP KEYWORD COUNTER
5090        INY              BUMP INDEX
5100        BNE TOK1         AND LOOP!
```

Listing continued

sers, often accomplish this by concatenating characters to the destination string one at a time. Use of pointers cuts down on the number of string concatenations and, therefore, garbage collections.

If strings to be concatenated can be placed in several variables and then concatenated in one assignment statement, this can cut down the number of garbage strings.

If you only need to test the first letter of a string, use the ASC() function and test the ASCII value, rather than something like the following:

```
INPUT "COMMAND";A$
A$=LEFT$(A$,1)
IF A$="S"THEN10235
```

Instead, use the following:

```
INPUT"COMMAND";A$:A=ASC(A$)
IF A=83 THEN 10235
    VARIABLES
    (or, first things first)
```

Whenever a variable is referenced, BASIC has to search from the beginning of its variable table for the entry of that variable. To speed execution, variables which are frequently referenced and FOR-loop variables should be placed at the beginning of the table.

To do this, reference these variables at the beginning of execution. For example, if X and Y are used frequently as temporaries and loop variables, the first executed line of your program could contain

    X=0: Y=0

to force them into the beginning of the variable table.

I hope these suggestions are of use to you.

★

CLEANING UP THE CLOCK

By: Steven P. Hendrix
Route 8, Box 81E
New Braunfels, TX 78130

Having had some intermittent memory bugs on my trusty C1P, I was relieved to discover the problem during some unrelated troubleshooting on a recent project. This problem exists, as far as I know, on every C1P and probably on other models in the Challenger line. It is negligible when operating with a 1 MHz (megahertz) clock, but can be the source of persistent and elusive bugs on a

Continued on page 20

18

```
5110 TOK5   LDA KEYLST,Y  GET CHARACTER FROM LIST      5530          CMP #BUFFER/256+$D
5120        AND #$7F      STRIP OFF BIT 7              5540          BEQ FETCH
5130        PHA           SAVE ON STACK               5550          INC TXTPTR+1
5140        STY TMP       SAVE Y                      5560          TXTPTR=*+1
5150        JSR BAS11     PRINT AND TRANSMIT          5570 CHRGOT LDA $FFFF
5160        PLA           RETRIEVE ORIGINAL CHARACTER 5580          RTS
5170        LDY TMP       RETRIEVE Y                  5590 ;
5180        CMP KEYLST,Y  SEE IF AT END OF WORD       5600 FETCH  PLA
5190        BNE TOK6      YES! QUIT!                  5610          STA SENDX2+1
5200        INY           BUMP INDEX                  5620          PLA
5210        BNE TOK5      AND RE-ENTER OUTPUT LOOP    5630          STA SENDX1+1
5220 TOK6   RTS                                       5640          LDA #$06
5230;                                                 5650          STA CMDFLG
5240ASMLIN JSR CHRGET                                 5660          JMP BACK
5250        STA RESLO                                 5670 ;
5260        JSR CHRGET                                5680 SENDX  LDA FACLO
5270        STA RESHI                                 5690          AND #$10
5280;                                                 5700          BEQ SENDX1
5290; OUTPUT CURRENT LINE NUMBER                      5710          JMP P0
5300;                                                 5720 SENDX1 LDA #$FF
5310BLNOUT JSR HEXDEC                                 5730          PHA
5320        LDX #$FF                                  5740 SENDX2 LDA #$FF
5330 BLN1   INX                                       5750          PHA
5340        LDA NUMBER,X                              5760          LDA #BUFFER
5350        BNE BLN2      SUPRESS LEADING ZEROES      5770          STA TXTPTR
5360        CPX #$07                                  5780          LDA #BUFFER/256
5370        BNE BLN1                                  5790          STA TXTPTR+1
5380 BLN2   LDA NUMBER,X                              5800          JMP CHRGOT
5390        ORA #'0                                   5810 ;
5400        STX RESLO                                 5820 SCRLF  LDA #CR
5410        JSR BAS11                                 5830          JSR XMIT
5420        LDX RESLO                                 5840          JSR CNSLOU
5430        CPX #$07                                  5850          LDA #LF
5440        BEQ BLN3                                  5860          JMP CNSLOU
5450        INX                                       5870 ;
5460        BNE BLN2                                  5880 CKRDY  JSR XIN
5470 BLN3   LDA #'                                    5890          BCC CKRD2
5480        JMP BAS11                                 5900          CMP #CTRLS
5490 ;                                                5910          BNE CKRD2
5500 CHRGET INC TXTPTR
5510        BNE CHRGOT                                        Listing continued
5520        LDA TXTPTR+1
```

```
5920 CKRD1   JSR XIN
5930          BCC CKRD1
5940          CMP #CTRLQ
5950          BNE CKRD1
5960 CKRD2   RTS
5970 ;
5980          .END TRM65U
```

★

Continued from page 18

system which has been upgraded to 2 MHz. The fix is very simple, if you're willing to cut a few traces on the processor board.

The 6502 microprocessor specifications permit each output pin to drive one standard TTL (transistor-transistor logic) load plus 130 pF (picofarads). The design on the 600 board has the phase 2 clock output driving upwards of 10 other devices with no buffering. I had not noticed . this in looking at the schematics, so I was very surprised when I put an oscilloscope on the phase 2 clock during troubleshooting of my project. The supposed square wave looks more like a 50% duty cycle sawtooth:

None of the various circuits driven by this signal is particularly fussy about the rise time of the clock, and it does not drive the RAMs directly. However. since the threshold level where the circuits treat a voltage as a logic "1" is somewhat variable, this can lead to inconsistent results. Of more concern in my case, however, is the fact that the clock signal as seen by the rest of the board is delayed by about 80 nS (nanoseconds). That may seem like a very small delay, but remember that we're talking about a signal that is already only high for 250 nS (2 MHz clock). If a particular memory chip is already a bit marginal at 2 MHz, this slight delay makes the difference between working and not working.

The trick, then, is to find an unused gate on the 600 board to be used as a buffer for the clock. Alas, every unused gate that I could find inverts the output signal with respect to the input signal. Aha!! U21 already inverts the signal for some uses, so by a little rearranging, we can use an inverting buffer. While there

are several unused gates on the board, I chose to use the section of U18 on pins 8 & 9. This produces an inverted clock signal for use in the applications presently driven by U21 pin 4, and the now unneeded section of U21 can re-invert the signal for use by the circuits which were driven directly by the processor. This does introduce a slight lag of its own into the clock signal.   Each 74LS04 has a typical delay of 9 nS, so the new clock signal lags the true clock by 18 nS.

This still beats the irregular 80 nS lag, however.

All right, time to get serious. Get out your soldering iron, wire cutters, etc. Remove the case, if any, and the 610 board if you have one. You'll need access to both sides of the board. Fortunately, none of the traces involved go underneath the integrated circuits, however. On the top side of the 600 board (component side), cut the trace going to pin 39 of U8 (the 6502) and the traces going to pins 3 & 4 of U21. Turn the board over and you will find aother trace tied to each of pins 3 & 4 of U21. Cut these also. In each case, cut the traces close to the

pins so that you don't miss a branch which would stay connected.

To add the new connections, I prefer #30 insulated wire-wrap wire soldered to the appropriate points. You may find a slightly larger size such as #22 easier to work with. Tie pin 39 of U8 to pin 9 of U18, and pin 8 of U18 to pin 3 of U21. You now have the phase 2 clock available at pins 3 (inverted) and 4 (non-inverted) of U21. Tie pin 3 of U21 to the two traces which you cut loose from pin 4, and pin 4 of U21 to the two traces which you cut loose from pin 3. This completes the patch.

If you previously found that your system would not work or was unreliable at 2 MHz, test it again after making this patch. You may also find that some memory chips which were marginal at the faster speed are now quite solid. In my case, several troublesome chips suddenly decided to behave. If you have an oscilloscope available, it will be instructive to look at the clock signal on the expansion connector before and after the modification. Maybe it will even help you resist the temptation to cut corners on buffering in your next project!

★

READER PROFILE

ED:

In the June 1983 issue of PEEK(65) you asked readers to let you know what kind of hardware and software they had and what they were using their systems for. Here it is the middle of August, and I am just getting around to responding to your request. I trust this information is not too late for you to use.

Hardware
C2-OEM (48K)
ADDS-25 terminal
Okidata Microline 92 printer

Software
OS65U versions 1.2 and 1.43
OS-DMS (1979 level)
WP-3.2 Word Processor

The system is used almost exclusively in support of organizations and activities in which my wife, our four children and I are involved. All members of the family use the word processor. It has been a life-saver in meeting tight schedules for high school term papers, book reports, etc. It has also been used by my wife to prepare the minutes of

meetings for organizations of which she has been secretary. I could go on and on about the uses which we have put the word processor.

The primary applications for which our home computer system is used are competitive swimming and fund raising for our high school band parents' organization. Preparing for a swim meet entails a large amount of clerical and manual work. The larger the meet, the greater the amount of work and the challenges for the meet director. We have used our computer for everything from keeping track of entries, to producing lane cards, to generating heat sheets, to assisting with the preparation of awards. Again, I could go on and on about the automated assists we have added to swim meet management. My wife says that the use of the computer has made her job as meet director simpler by freeing her from most of the clerical and manual work (actually saving hundreds of hours each year), and permitting her to devote more attention to people-related matters.

The largest application run on the system is used to support the semi-annual citrus fruit sales which our high school band parents' organization conducts to raise money. Two years ago the position of "Data Processing Chairman" was established after I suggested to the Board of Directors that it might be feasible to use a home computer system in conjunction with the fruit sale.

In five years my last child will graduate from high school, which means I still have five years to serve as Data Processing Chairman.

To give you some idea of the size of the application, the software is contained on three eight-inch diskettes, the master files require two diskettes. The system is used for such things as order entry, sales reporting, sales contest reporting, customer follow up and printing receipts for customers to use for income tax reporting. It is a system which seems to keep on growing. There is always a new feature or function being added. This is a system which has saved people hundreds of hours in clerical time during the past two years. The organization it has brought to conducting our fund raising events has resulted in our receiving many compliments from customers as to the effi-

ciency with which our sales are done. We honestly believe that the manner in which the computer has been employed has actually accounted for higher sales and profits than would have been possible without it. If everything goes according to plan, I will be making this application system available to OSI users in the second quarter of 1984.

We have had our home computer system for three years now. We believe it to be one of the best investments we have ever made.

David A. Weigle
Morton, IL 61550

* * * * *

ED:

So you want us to embarrass ourselves by describing the various ways we write bad BASIC in our efforts to come up with good programs? Okay.

Machines:
  Superboard II/C1P (16K)
  Sylvania 12" portable b/w TV
    for monitor
  IBM Selectric, a.k.a. COPE
    1030 Terminal
  Portable Cassette Recorder

Operating System:
  OS65V version 3.2 as
    modified by
  Progressive Computing (64
    ch. video & edit
    functions)
  BASIC-IN-ROM (microsoft
    original)

Use:
  Business (invoice &
    letters), games

I daresay many of us have tried the Polish or Aggie

(Texas A & M) method:

1) turn on computer;
2) enter first line number followed by a space;
3) try to figure out what comes next.

Now let me share with you a more enlightened approach. I learned this method from no less prestigious an institution than IBM. At the time (1974) I was employed as an accountant by a firm that was about to acquire its first computer - and IBM system 3 (a honey!). So the company sent me to a one-week introductory course at one of IBM's schools. Here's what I remember of that course:

Step 1. Define the output. What data do you want, where, and how should it appear?

Step 2. Examine the input. How and when will it appear? Is it sufficient for the task?

Step 3. Develop the operations necessary to get from input to output.

This is not to say I always follow this method. Often times I start with the operations phase. I'll work through the steps which (I hope) will lead me to the desired output. Along the way, I see what kind of input is required. This technique is most useful when you're not sure of the exact form of the output. Lastly, I put in the "bells and whistles" - menu or user instructions. The only documentation I use is a hard copy of the finished program.

My brother is a programmer for a well known wine company in California. The programming technique he described to me sounds like one of the best. After his boss tells him the objectives of the program, my brother goes into his office and shuts the door. For the next several hours, he sits with his feet upon his desk and thinks of all the wrong ways to write the program. Then he sits up and writes the correct program.

Bruce Showalter
Abilene, TX 79601

LETTERS

ED:

In the March 1983 issue of Byte, on pages 389 thru 415 is a good solid public domain CBASIC program, which I chose to pump in and use. The article "Keywords in a Fuzzy Context" by Thomas A. Smith of Camarillo, CA., accompanies 8 related programs which make up a data base for bibliographic search along the lines of so called "Fuzzy Theory", developed by Ronald Yager (see: "A Logical On-Line Bibligraphic Searcher: An Application of Fuzzy Sets" IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-10, no. 1, Jan. 1980, p. 51).

As the article explains, an item either belongs or doesn't belong. What the "Fuzzy theory" does is define the degree ot belonging. For example, suppose I were posting to this data base a paper or article about OSI DMS software. Via assignment of keywords, I could say for instance that on a scale of 1 to 10 (or 0.0 to 1.0 as is programmed), I would rate the degree ot belonging as 1.0 for OSI BUSS; 0.9 for DATA BASE; 0.7 FOR OS65U; 0.2 for TREE STRUCTURE. Now let's suppose that some time later, (after a couple of hundred articles were listed in the data base), I needed all the information I could find on Tree-Structured Data bases - then I would merely insert the various pertinent keywords with their selection rating in the 'sense of belonging' to the research project I am doing, and a listing of only those articles would appear on the list device. That's the principle of 'fuzzy theory'.

After reading the aforementioned article in BYTE, that idea looked mighty good to me, so the next thing was to punch in the program. I have CBASIC 2.06 compiler but I had never used it. I found it to be simplicity itself. Using my OSI C3, I loaded CP/M v2.24a onto a disk with the following two CBASIC 2.06 system utilities on it, viz. CBAS2 and CRUN2 (total 17K). Then I loaded also the ED program. After that, I invoked ED (actually ED FILENAM.BAS) and after receiving the ED prompt (*), I typed 'I' for insert. That's akin to typing NEW under OSU. Then I got the words 'NEW FILE' which means that the name of the file, FILENAM.BAS is placed in the directory. The next prompt is 1: (meaning program line 1) and you type in your program, being very careful to indent in the proper places for LOOPS, FN-FEND and WHILE WENDs. When you make an error, you can go back by typing control Z (gets you back into ED without clearing RAM) and doing whatever is necessary according to standard ED conventions. 'I' gets you back into the programming mode. When you are all done, you can look the program over by typing B#T which lists the whole program out. If that's OK so far, you type 'E' - that does two things, it EXITS you out of ED and it SAVES FILENAM.BAS. Finally, you must compile the program. You do that by typing CBAS2 FILENAM (no extension) and if you did a Control P first, it will do so on the printer. It will print errors under the line where they occur with the position (column) number where the error occurred so debugging is not all too complicated. To debug, you go back to ED FILENAM.BAS (the previously saved program then becomes FILENAM.BAK) which you should erase (ERA *.BAK) after all is well and done to conserve disk space. After you are reasonably sure that you've debugged successfully, you again compile (do that as often as is necessary until the words 'NO ERRORS DETECTED' appear under the program (the following information also prints out: Constant Area, Code Size, Data Stmt Area, and, Variable Area). You are now ready to run the program using the command CRUN2 FILENAM (no extension). From the above, you realize that the only way to change the compiled programs is via the ED program and the programs with .BAS extension, so you keep a disk (plus backup) of that disk. On the disk where you actually use the programs, it is not necessary to keep those files. Just keep the compiled versions with the .INT extensions.

The purpose of this short review is to acquaint you with entering programs using C-BASIC-2 under OSI CP/M. C-BASIC is at first complicated but if you are proficient with OSI's (Microsoft) Basic, C-BASIC will soon fall into place. I think it's a great and very powerful language to use! By all means buy the following two Osborne-McGraw-Hill books: "Osborne CP/M User Guide" by Thom Hogan, and "CBASIC User Guide" by Adam Osborne, Gordon Eubanks, Jr., and Martin McNiff. These two books have become my 'bible' in using CP/M and CBASIC respectively and were the best investments I have made.

Fred S. Schaeffer
Kew Gardens, NY 11415

* * * * *

ED:

I have just purchased a new DOS called DOS/65 which is a CP/M type system for the 6502 cpu. DOS/65 provides the same kind of of file compatibility between 6502 based computers as CP/M does for Z80/8080/8086/8088 based systems.

It follows very closely the CP/M structure, in fact, anyone using a CP/M manual or guide can apply most every command to DOS/65. DOS/65 will allow reading/writing of standard CP/M files also. It is made available for most 6502 based machines and could become a great DOS if we can get enough people interested in it. Its basic is Basic/E which is very much like Micro-Soft's basic.

I own an OSI C4PMF system running now with DOS/65 and it's great. I would like to ask your 6502 system members if anyone has DOS/65 and also if anyone has available any Basic/E software? There are four of us in the Detroit OSI SIG running DOS/65 and would like very much to see this DOS promoted in order to open the door to compatible software.

Don Emmons
Novi, MI 48050

* * * * *

Don:

DOS/65 is currently under negotiations for a review. With a little luck it will be ready for the next issue. In the meantime, let's hear from your friends.

Peek Staff

* * * * *

ED:

I recently purchased a used OSI C24P MF. To my amazement I then discovered that OSI doesn't make a personal computer anymore and I have been unable to find compatible software for it.

I am particularly interested in a "Wall Street Analyser" or software that would help me analyse stock options. Do you have any software that might suffice, or could you direct me to someone who might?

I have had no computer background and find the OS65D V3.0 manual very "unfriendly". I do have OS65D V3.2 software and a 1980 manual for OS65D V3.3, but no software! I am

up a stump so to speak.

Charles Huff
Staples, MN 56479

Charles:

We don't know of such a package ready to go, but something like Technical Products' Terminal Control Program (see ad this issue) to give you access to Dow Jones through The Source and others and OSI's Planner Plus will allow you to build whatever algorithms you wish or need.

Al.

* * * * *

ED:

I would like to hear from someone who has, or can copy the schematic from Electronic Systems Modem Kit #109A. It originally sold for $29.95. The company was based in San Jose, CA and is now out of business.

Bruce Showalter
Abilene, TX 79601

CALENDAR ITEM

**FIFTH ANNUAL NORTHEAST
COMPUTER SHOW AND
SOFTWARE EXPOSITION**

The Fifth Annual Northeast Computer Show and Software Exposition will be held Thursday – Saturday, November 17 – 19, 1983, at Boston's Hynes Auditorium. The Show Hours are 10:30 AM to 5:30 PM daily. This is the largest annual end-user computer event in the East and features nearly 500 displays and exhibits of microcomputers, accessories, peripherals, and software. Admission is $7.00 for adults. For more information call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167. Telephone: 800-841-7000 or 617-739-2000 (within Massachusetts).

AD$

PEEK (65)

The Unofficial OSI Users Journal

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

DELIVER TO:

GOODIES for OSI Users!

PEEK (65)
The Unofficial OSI Users Journal

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