PEEK (65)

The Unofficial OSI Users Journal

$1.75
FEBRUARY 1983
VOL. 4, NO. 2

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

INSIDE:

BLOCK DELETE FUNCTION	2
PARALLEL PRINTER INTERFACE	3
DIRECTORY RESTORER	9
ADD 8K RAM TO SBII, C1P	12
FUN WITH OSI'S UTI	14
OS-DMS KEY FILE DUMP	16
KEYWORD REVIEW	18

Column One

For some months, all of us have felt like a caller on "hold." Throughout the M/A-COM buyout/sellout, and all the changes in personnel and equipment, we have wondered if OSI would survive, and in what form.

Now it looks like our patience has been rewarded. The new Ohio Scientific, Inc. of Bedford, MA has started operations with a bang, with new products and personnel announced. First the personnel:

Kenneth E. Wortz -- President

Maurice Berez -- V.P. Finance

Michael Sorrentino -- V.P. Operations
Jim Halverson -- Product Development
Bill Bordy -- Hardware Engineering
Rick Whitesel -- Operating Systems Mgr.
Frank Valcarcel -- Marketing

The new company plans to work hard on hard disk, multiuser business systems. This fits in with the Comdex announcement of the new Masterkey 300 Multiprocessor system series.

The 300 series is a new departure for OSI, using multiple single board computers to give each user a separate CPU and 64K of RAM, instead of forcing several users to time-share a single CPU.

Also a new direction, the processor used on each "application processor" board in the 300 series is a Z80, the operating system providing a CP/M compatible OS for each AP board.

PEEK(65) readers will recall that I asked for just such a system, with record and file locking for each user, a few months ago. No, I did not realize that was exactly what OSI would announce at Comdex! It simply stood to reason that such a system was what was needed for business applications. It is nice that OSI was listening to the same market needs.

The announcement says that many OS-65U programs will also run on the new machines. That is very interesting, since 65U's disk access method is completely different from CP/M's. But the new KEYBASIC is a language and KeyOperator 1 is an operating system (whereas OS65U is a mix of language and operating system). KeyOperator 1 uses dynamic file allocation and doesn't care which supported language the file is for. Thus, OSU and CPM files may be physically adjacent on the hard disk. Incidentally, KEYBASIC is a superset of OSU Version 1.43 and thus 1.43 programs will run if Peeks & Pokes are changed to "Commands." Double Density disks holding 620K will also be supported. If past experience is any guide, the hardware will be well designed and what it does do, it will do quickly and well.

One more thing: the disk format for floppy disk will be IBM 3740, the "standard" single density format used by many other CP/M computers, so we will be able to buy software from any place, including mail order discount houses.

All in all, it sounds like OSI has done it right, with a powerful, true multi-tasking computer using the industry standard disk format and operating system, while still maintaining compatibility with the 65U programs we have written for our systems. It is debatable how much 65U software will continue to be used, but I am very glad we have that option.

Now we must wait a few weeks to see what plans there are for the polled-keyboard hobbyist systems. Of course, they are still supported and parts and repair are available from CCCC; but will they be revised, upgraded, re-designed, or manufactured in their old form? Only time will tell...

al

THE WORKINGS OF BASIC
A BLOCK DELETE FUNCTION

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

As you may recall from my column a couple of months ago, the line editing portion of the Basic interpreter on the C1P contains the logic to delete a single line from a program. However, it does not include the logic to delete a block of lines, as do some Basic interpreters on more expensive systems. For instance, if I choose to delete all lines from 100 thru 200, I must delete each line individually, by typing in its line number and a carriage return. This can be extremely tedious. This situation may not occur very often, but a simple method of doing the job would be useful. I will present here a machine language routine which will allow you to add a block deletion function to Basic.

As we saw in the earlier column, deleting a line involves searching for the line, moving the subsequent part of the program downward in memory to overwrite the line, and correcting the pointers which link each line to the next. Fortunately, all of these moderately complex functions are done by routines in ROM. In fact, it is fairly easy to fool the single-line deletion routine into deleting a whole chunk of a program (some would say dangerously easy!).

The routine in Basic's immediate mode handler appears at $A295-$A2E5. As discussed earlier, it proceeds by using the "next line" pointer in the line to be deleted, assuming that pointer designates the beginning of the program after the deleted line. We can easily fool this routine by simply setting the link pointer in the first of the lines being deleted, so that it points to the first line after the deleted section. For instance, suppose we have a program consisting of lines numbered in increments of 10, and we wish to delete lines 100 thru 200. We could set line 100's link pointer to point to line 210 and then call the normal deletion routine. The routine would then carry out the desired function, keeping the rest of the program intact.

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

LISTING 1: BLOCK DELETE

```
JSR $00C2 ; RETRIEVE THE LAST CHARACTER
JSR $A77F ; CONVERT ASCII NUMBER TO BINARY NUMBER
JSR $A432 ; SEARCH FOR LINE
LDA $AB   ; SAVE ADDRESS OF THE LINE
STA $FF
LDA $AA
STA $FE
LDA #$A4  ; CHECK FOR HYPHEN
JSR $AC03
JSR $A77F ; CONVERT SECOND LINE NUMBER
INC $11 ; INCREMENT LINE NUMBER
BNE *+4
INC $12
JSR $A432 ; SEARCH FOR LINE FOLLOWING LAST LINE TO BE DELETED
LDY #0
STA $13   ; TO PREVENT SPURIOUS INSERTIONS
LDA $AA   ; CHANGE POINTER FIELD OF FIRST LINE
STA ($FE),Y
LDA $AB
INY
STA ($FE),Y
LDA $FE   ; RESTORE POINTER TO FIRST LINE
STA $AA
LDA $FF
STA $AB
SEC       ; CALL NORMAL DELETE ROUTINE
JMP $A2A2
```

The program presented in listing 1 does the job as described. It is meant to be called with the USR function, but could also be called by SAVE or LOAD if you wish and if you do not use those verbs for their normal function. I will describe the use of this routine from USR, whose vector is at $000B, but you can use LOAD by changing its vector at $021E (decimal 542) or SAVE, with its vector at $0220 (decimal 544).

To use the routine as described, enter it at any convenient location (it is position - independent) and point USR's vector to it by POKEing the high byte of its starting address to (decimal) 12 and the low byte to (decimal) 11. For instance, if you choose to place the routine at $1000, use POKE 12,16 : POKE 11,0. Then, to delete lines 100 thru 200, for instance, type ? USR (0) 100-200 <cr> . If you wish to use LOAD, POKE the address to 543 and 542 and type LOAD 100-200 <cr> .

At the entry to the routine, Basic's next-byte pointer in GETBYTE points to the byte after the right parenthesis. The JSR $00C2 gets that next byte as Basic normally does, in preparation for the numeric conversion done by the JSR $A77F. This conversion leaves the value of the first line number (the 100 in the example above) at $0011-$0012. The JSR $A432 then seeks that line number, leaving the address of the beginning of the line at $00AA-$00AB. That address is then saved at $00FE-$00FF so that it will not be destroyed in searching for the other line.

The LDA #$A4 and the JSR check for the hyphen, signalling an SN ERROR if it is not the next character. Then the JSR $A77F converts the high line number (the 200) and stores it at $0011-$0012 as before. Since we want the deletion to be inclusive, we must search for one line number higher to find the first line which will be left as part of the program. The INC-BNE-INC sequence handles this, and then the JSR $A432 searches for the line as above.

Now to fool the delete routine. We have the addresses of the first line to be deleted and the first line after it to be retained. We must make it appear that there is only one line between the two addresses. To do this, we change the link pointer of the first line to point to the second. To do the required indirect addressing, we must first load Y with a 0. This

is as convenient a time as any to zero out location $0013, to prevent anything from being inserted in place of the lines we are deleting. Then we load the two bytes of the address of the second line (the line after 200) from $00AA-$00AB and store them to the two bytes pointed to by $00FE-$00FF. The two LDA-STA pairs then reset the line pointer ($00AA-$00AB) to point to the first line (100), and then the normal delete routine is called. It takes care of all the necessary internal housekeeping and then returns to Basic's immediate mode. Notice that if you do this in a program, the program stops.

That's all there is to it. Those of you who are thinking about tinkering with your ROMs and making some of the changes I suggested in an earlier article may want to consider including this routine. Who knows, maybe someday one of you will come up with the ultimate combination of features to finally overcome the little annoyances OSI built into these machines for us!

★

EXPANDING THE C1P
A PARALLEL PRINTER INTERFACE

By: D. B. Baker
3128 Silver Lake Road
Minneapolis, MN 55418

When I got my Ohio Scientific C1P three years ago, I had only one or two uses in mind for it. Since then, I have been continually expanding both my use of, and the hardware in, the system. Unfortunately, with the limited expandability of the C1P, I reached the limit of the hardware before I ran out of ideas for new uses.

One problem I ran into was the multiple usage of the serial port at $F000. It's the cassette interface, and a RS-232 serial interface for either a MODEM or a printer, both in one! While this can be an advantage sometimes, it becomes very difficult to use both a printer and a MODEM at the same time. If you have a disk system, it becomes a little easier: you can "spool" data to the disk when using the MODEM, and print it later, when the MODEM is no longer being used. This takes more time, and considerable software support. Clearly, I needed some expansion to allow an easier, more direct access to a printer, as well as

FIGURE 1. Printer Interface Schematic

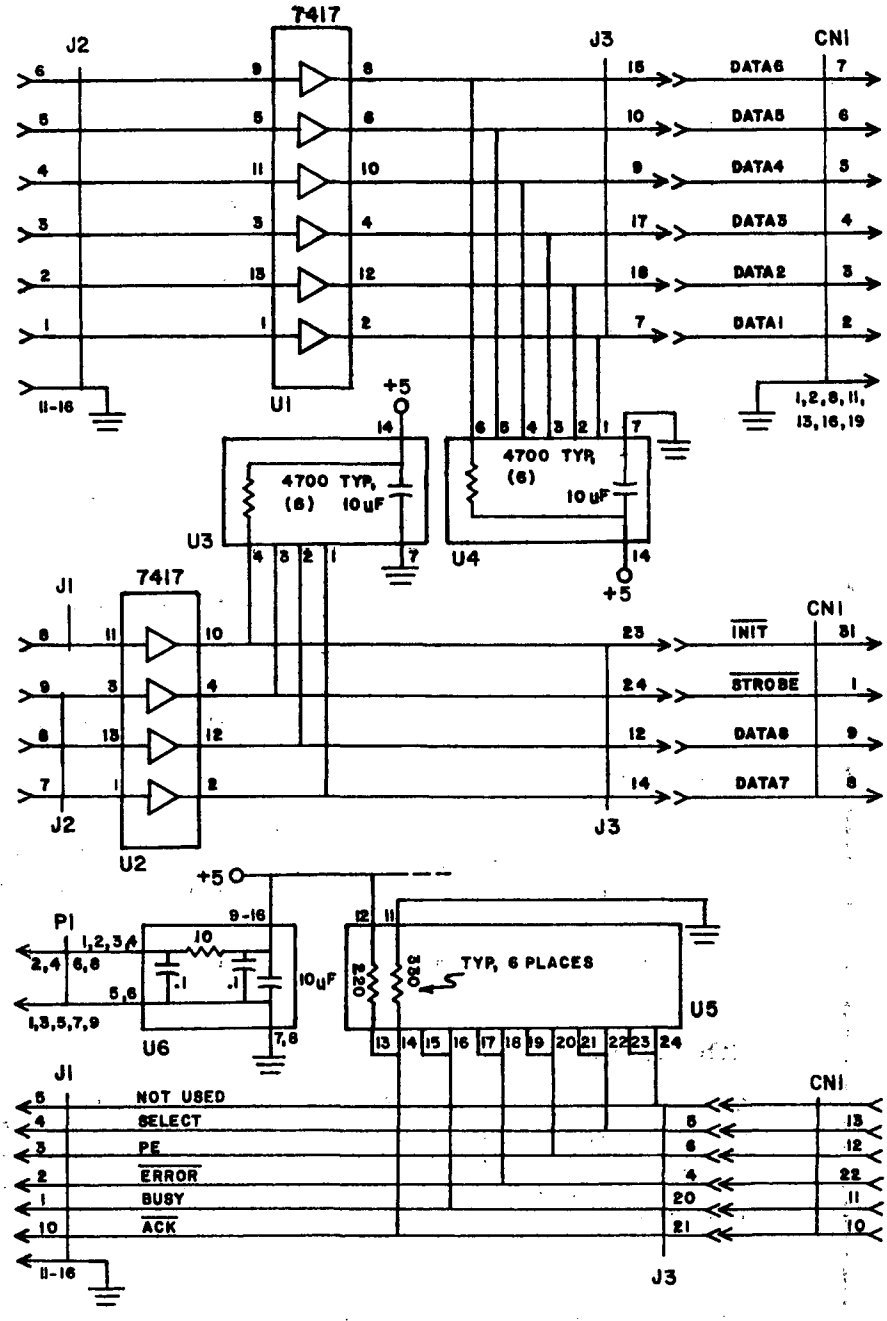
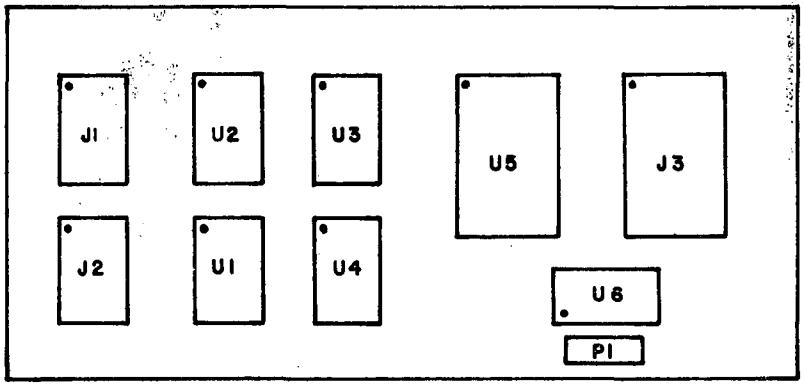


FIGURE 2. Driver/Receiver Board Layout



MULTI-PROCESSING
with the Denver Board

The Denver Board (Model DB-1) is an assembled and tested terminal expansion circuit board for expanding terminal usage on any Ohio Scientific, Inc. (OSI*) Series C2 and C3 computer system. The DB-1 is designed to reduce terminal speed loss from 80 to 90 percent when two or more terminals are added to the computer. Each terminal is also provided with an additional 16K bytes of memory.

Each DB-1 comes with a full 90-day parts and labor warranty, and a factory repair/exchange program is also available should a DB-1 that is out of warranty ever need servicing.

FEATURES

- 64K Bytes Random Access Memory (RAM)
- One Programmable Read Only Memory (PROM) for BUS arbitration and interprocessor communications.
- Six light emitting diodes (LED's) for power, master BUS indicator, transmit and receive.
- Automatic system boot switch.
- Auxiliary BUS for expansion printer I/O circuit board.
- Four reset modes:
  Power-on reset.
  Master reset (front panel).
  Individual reset from terminal with BREAK key.
  Individual reset from DB-1 with pushbutton switch.
- Memory expansion capability of 4K bytes common memory using standard OSI memory expansion circuit board.

SOFTWARE

95 percent of existing OS-65u* software is compatible with the DB-1. An OSI operating system patch program is supplied on 8-inch floppy disk as required. The patch program is copied to the user disk that contains the OSI operating system; and when the computer is turned on, the patch program will automatically tie-in.

for more information call or write:

DBi, inc.

p.o. box 7276
denver, co 80207
(303) 364-6987

Dealer Inquires Invited

* OSI and OS-65u are trademarks of M/A-COM Office Systems, INC.

provide general expansion capabilities.

EXPANSION POSSIBILITIES

I found several expansion options availble for the C1P. A disk interface and up to 24K of memory may be added with an OSI 610 board. D & N Micro Products makes an expansion connector that will "convert" the C1P to a standard OSI 48-pin buss, after which almost any "48-pin" board may be used. Aardvark Technical Services offers a couple of versions of an 8K memory expansion board. And Mittendorf Engineering has a 256 x 256 dot graphics video board.

All of these expansions seem to be excellent, yet each has its draw-backs. None provide an interface for a printer. Some are expensive, or have hidden costs. Others are "dead-ends," and don't allow further expansion. Some work with the 600 board, but not the 610, or vice-versa. It's just difficult to expand a C1P past the individual board!

For these reasons, I was delighted to run across the DC650 Motherboard and DC660 Dual PIA board made by Device Control - "a microprocessor applications company," of St. Paul, Minnesota.

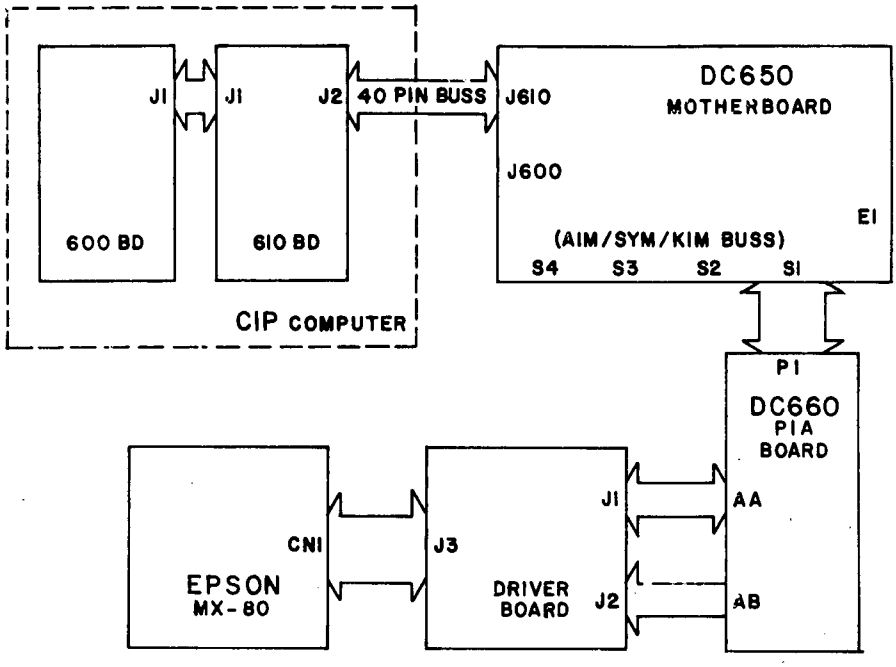
EXPANSION MOTHERBOARD AND PIA BOARD

The DC650 Motherboard converts EITHER a 600 or 610 board to the AIM/SYM/KIM buss simply by plugging the interconnect cable into the appropriate socket. When necessary, as for a 600 board, buss drivers are used to buffer the expanded buss. No rewiring is needed, and standard ribbon cable (supplied!) is used.

The DC650 Motherboard provides four 44-pin sockets, plus one "edge-pin" connector. All signals common to the AIM/SYM/KIM busses are supported, although some unique ones are not. In addition, the TEST signal, used by AIM, has been replaced by the OSI - unique signal DATA DIRECTION (DD). Instructions for generating this signal are provided.

For those signal pins that are different on different busses, appropriate jumpers on individual plug-in boards make them available. On the DC650 Motherboard, multiple use pins are uncommitted.

FIGURE 3. Printer Interface - Cabling Diagram



The DC660 Dual PIA board provides two PIAs (6820 or 6821). However, solder pads are also available to alter the chip's pin assignments to allow the use of Versatile Interface Adapters (VIAs - 6522s) in place of the PIAs. This chip is more powerful in what it does, but also costs more and is harder to program. (In either case, changing is easy, as the chips are socketed.)

Both 8-bit parallel ports of each PIA (or VIA), along with both control signals, are brought out to 16-pin DIP connectors. Six grounds are also provided for each port. This simplifies connection to peripherals.

PRINTER INTERFACE

To solve my MODEM/printer problem, I decided to use a combination of the DC650 Motherboard and DC660 Dual PIA board to build a "Centronics-type" parallel printer interface for the Epson MX-80 printer. This would allow the use of the printer and MODEM at the same time, require a minimum of external hardware, be easy to program, AND allow further expansion of the C1P with any of the many boards available for the AIM/SYM/KIM busses.

The parallel interface for an Epson (and other "Centronics-type" printers) actually requires eleven signals, plus grounds. Since more than enough pins were available in a single PIA, I decided to provide a full implementation of the interface. To make use of the full interface, more software is required, but it may be worth it for some applications. It would be nice to have the printer TELL you it is out of paper, rather than wondering why it quit!

In the interface, the only signal that is a little strange is the STROBE-(NOT) command. This signal tells the printer that valid data is available on the interface, and is a minimum 0.5 microsecond negative pulse. Usually, STROBE-(NOT) is generated with a one-shot multivibrator. However, a PIA has as one of its control lines (CA2 or CB2 depending on which port) a programmable signal that may be used for just this purpose. I decided to use this instead of the one-shot, allowing a simpler interface, and a little less software. The schematic for the interface is shown in Figure 1.

Aside from the DC660 Dual PIA board, a cable interface and driver/receiver board is required. Cable drivers are two quad-driver (7417) ICs. A simple resistive network consisting of a 220 ohm and 390 ohm resistor terminates each received line. The resistors are mounted on a 24-pin DIP

header for convenience. All sockets are wire-wrapped, using a Radio-Shack #276-1395 "prepunched perfboard" to hold the sockets. A drop of "5-minute" epoxy under each keeps them from wobbling. The output connector is a standard 24-pin wire-wrap socket. The parts layout I used is shown in Figure 2, but there is nothing critical about the layout.

The cable to the printer consists of twelve twisted pairs, terminated at one end by a 24-pin DIP plug, and an AMP 57-30360 plug, to match the Epson MX-80 connector at the other end. The complete cabling diagram for the printer interface is shown in Figure 3.

(The 40-pin cable supplied with the DC650 Motherboard was not long enough to reach from inside a C1P case to my expansion cabinet. I made up a new cable that is 24 inches long. Although not recommended by DEVICE CONTROL, it works fine. Put the 40-pin plugs on opposite sides of the cable for best cable dress. Be sure that pin 1 IS pin 1 on both ends of the cable!)

With the use of the connectors and the driver/receiver board, modularity is maintained; the printer may be easily disconnected from the interface for movement, and the interface and PIA boards may be removed from the expansion cabinet individually.

Fortunately, the hardware design of the C1P allows adding a printer port at $F400. The ACIA (serial port) is decoded to $F0XX. Likewise, the monitor ROM is decoded to addresses greater than $F800. This permits additional decoding of any address in the range of $F100 to $F7FF. For convenience, I used the "standard" address of $F400 for my parallel printer port. This assignment is reflected in the software illustrated in the listings below.

LISTING 1. Initialization of PIA and printer and Output Routine - Basic Method

```
1000 REM  C1P PARALLEL PRINTER PORT INITIALIZATION
1010 REM    PIA PORTS ADDRESSED AT 62464 ($F400)
1020 REM    TO MATCH OS65D FOR C2/C4/C8
1030 REM
1040 REM  $F400 = READ/WRITE FOR 'A' PORT
1050 REM  $F401 = CONTROL REGISTER, 'A' PORT
1060 REM  $F402 = READ/WRITE FOR 'B' PORT
1070 REM  $F403 = CONTROL REGISTER, 'B' PORT
1080 REM
1090 REM  INITIALIZE PIA PORTS AS FOLLOWS:
1100 REM    PORT A - BITS 0-6 INPUT
1110 REM    PORT A - BIT 7 OUTPUT
1120 REM    PORT B - ALL BITS OUTPUT
1130 REM    CB2 - PULSE OUTPUT ON WRITE
1140 REM    CB1 - NEGAVTIVE TRANSITION ACTIVE,
1150 REM      NO IRBQ
1160 REM
1200 PIA = 62464          :REM  DIRECTION REGISTERS
1210 POKE PIA+1,0         :REM ACCESS DATA
1220 POKE PIA+3,0         :REM  DIRECTION REGISTERS
1230 POKE PIA,128         :REM A7 OUTPUT, A0-A6 INPUT
1240 POKE PIA+2,255       :REM B0-B7 OUTPUT
1250 POKE PIA+1,4         :REM SELECT A R/W REGISTER
1260 POKE PIA+3,44        :REM SELECT B R/W REGISTER
1270 REM                        CB1 ACTIVE NEGATIVE,
1280 REM                        NO IRQB, CB2 PULSE
1290 REM                        ON WRITE TO B PORT
1300 POKE PIA,0           :REM INITIALIZE PRINTER
1310 FOR I = 1 TO 10      :REM SET INIT DELAY
1320 NEXT                 :REM 50 MICROSECONDS NEEDED
1330 POKE PIA,128         :REM SET (NOT)-INIT LINE HIGH
1340 REM DONE WITH INITIALIZATION !
1400 REM   OUTPUT CHARACTER ROUTINE
1410 REM    BASIC VERSION
1420 REM
1430 REM   COMPATIBLE WITH OS65D
1440 REM   CORRESPONDS TO ASSEMBLER
1450 REM    VERSION
1460 VEC=8983:REM $2317 - $2318
1470 ADD=9409:REM $24C1 - $24CC
1480 FOR I = 0 TO 11
1490 READ N
1500 POKE ADD+I,N
1510 NEXT I
1520 DATA 72
1530 DATA 173,0,244
1540 DATA 106
1550 DATA 176,250
1560 DATA 104
1570 DATA 141,2,244
1580 DATA 96
1590 FOR I = 0 TO 1
1600 READ N
1610 POKE VEC+I,N
1620 NEXT I
1630 DATA 192,36
```

PRINTER SOFTWARE

Although OS65D for C2/C4/C8 systems already supports a parallel printer port at $F400, this port doesn't exist on a C1P. Therefore, the I/O vectors normally associated with it point to a "null" routine within the operating system. Additionally, the driver code has been replaced with software to "correct" the C1P keyboard to look like that found on C2/C4/C8 systems.

Two items of software are all that are required to utilize the parallel interface. These are shown in Listings 1 and 2.

The first initializes both the PIA itself, and the printer. The important parts of this code are the statements that set the control bits for CB2 and CB1. Control line CB2 is programmed for a negative-going pulse output when the "B" port is written. Since this pulse lasts for one clock cycle (or 1 microsecond on a C1P), this meets the requirement of a 0.5 microsecond STROBE-(NOT) pulse.

The CB1 control line is programmed to receive the ACKNLG-(NOT) pulse from the printer. Although the Interrupt capability of the PIA is not being utilized in the examples, a negative transition of CB1 will set bit 7 of the PIA port-B status register. This isn't strictly necessary, but it does indicate that the data was accepted, and may be used in more sophisticated, interrupt driven applicatons.

OHIO SCIENTIFIC, Inc.

With our new management team, OSI is proud to announce the addition of the **KeyFamily 300** series —

MULTI-PROCESSING BUSINESS SYSTEMS

to our complete line of 200 series timesharing business computers. Utilizing state-of-the-art microprocessor technology OSI now offers the highest performance microprocessor based business system available. Each user has his own Z80A 4MHZ CPU, 64K memory, 4 channel DMA and two serial ports. A system master processor with a separate CPU, 56K of memory, 4 channel DMA and 2 serial ports handles all disk and system I/O tasks. Our separate, proprietary, 8 Megabit inter-processor communications bus provides nearly instantaneous inter-processor data transfers. Running OSI's proprietary version of the KeyOperator-1 Multiprocessing operating system allows most of the over 3000 CP/M based packages to run together with OSI's ...

KEYBASIC Version 2.0

KeyBasic 2.0 is the 65U BASIC version 1.43 compatible SUPER-BASIC language, the culmination of **your** input on 65U extensions and has many, many features unavailable in any other language. These include;

• Enhanced Extended Input • Character oriented Disk I/O • FIND command with limit • CRT Command • SWAP • WHILE WEND • KILL MultiByte to MultiByte input translation • Semaphore WAIT FOR with time limit • Enhanced Extended Output • Key Map • RANDOMIZE • TIMER • Selectable Dynamic File Allocation • RESUME • Invisible SPOOLING on 1 to 16 Queues onto 1 to 16 printers • **Record Locking** • Extended EDITOR • 4 types of Program Chaining with COMMON Verb • Up to 15 Disk Channels with individual buffers • Subroutine CALL • SuperTrace • TIME • DATE • RENAME • INSTR$ • Delete, Resequence and Renumber In Basic • PRINT USING • ON TIMER GOTO • ! and !! editor commands • ON ERROR GOTO • ERASE (delete file) • OPEN (creates file) • FIX • 16 Digit Precision • DEV$

The KeyFamily 300 series will initially be available in 4 models, the 10MB 330E and 40MB 330I (up to 4 users) and the 350J/JJ (up to 8 users). These systems will include **KeyOperator-1**, **KeyWord** Word Processing System and **KeyBasic**. Orders are now being taken for deliveries scheduled in February. KeyBasic 2.0 Language manual is available now for $25.00.

ORDER YOUR SYSTEMS NOW!!!

from your dealer or

OHIO SCIENTIFIC, Inc.
7 Oak Park
Bedford, Ma. 01730
Tel. (617) 275-4440

This initialization code is only needed once, at system "boot" time. To make it easy, I wrote the initialization code in Basic, and it may be added to BEXEC*. Suitable changes in line numbers are required, of course, to fit your BEXEC*.

(As an alternative, the BASIC program illustrated may be put into a file and called by a 'RUN"filename' command. This is the way I do it on my ClP.)

The output character subroutine simply takes an 8-bit character from the accumulator, and places it on the parallel interface if the printer is not 'BUSY'. This routine is shown in Listing 1, lines 1400 - 1630; or Listing 2, for the assembly-code version.

Note that the routine does not "strip" the high-order bit (8th): this allows access to any special features that may be installed in the printer, such as graphics capability. When a character is transmitted, the routine reads the printer status, and returns it to the calling program in the accumulator. The addresses of $A000 - $A00B are simply for convenience: the code is completely relocatable. The only "fixed" addresses are $F400-$F403 for the PIA addresses, and the I/O vector contents of $C0, $24.

Adding the driver to the operating system can be done in a couple of ways. A Basic routine may be written to POKE the code into memory, and then modify the I/O vector to point to the routine OUTCHR. This would have to be done every

LISTING 2. Output Routine - Assembly Code Method

```
100;OUTPUT CHARACTER ROUTINE
110; ASSEMBLER VERSION
120;
130; COMPATIBLE WITH OS65D.
140;
142         PIA = $F400
144         * = $A000  ;TEMP ORIGIN
150OUTCHR PHA         ;
160AGAIN  LDA PIA     ;TEST IF PRINTER BUSY
165       ROR A
170       BCS AGAIN   ;YES, WAIT UNTIL NOT BUSY
180       PLA         ;NO, SEND CHARACTER
190       STA PIA+2   ; WRITE TO PIA & STROBE
200       RTS         ;DONE, RETURN
210       * = $2317   ;MODIFY I/O VECTOR
220       .BYTE $C0,$24
```

```
.A

100               ;OUTPUT CHARACTER ROUTINE
110               ; ASSEMBLER VERSION
120               ;
130               ; COMPATIBLE WITH OS65D.
140               ;
142 F400=                  PIA = $F400
144 A000                   * = $A000  ;TEMP ORIGIN
150 A000 48       OUTCHR PHA         ;
160 A001 AD00F4   AGAIN  LDA PIA     ;TEST IF PRINTER BUSY
165 A004 6A              ROR A
170 A005 B0FA            BCS AGAIN   ;YES, WAIT UNTIL NOT BUSY
180 A007 68              PLA         ;NO, SEND CHARACTER
190 A008 8D02F4          STA PIA+2   ; WRITE TO PIA & STROBE
200 A00B 60              RTS         ;DONE, RETURN
210 2317                 * = $2317   ;MODIFY I/O VECTOR
220 2317 C0              .BYTE $C0,$24
220 2318 24
```

time that the system was booted, although it could also be made a part of BEXEC*, as is the initialization of the PIA and printer. This is the method used in Listing 1. If the majority of your programming is done in BASIC, this is the easier of the two methods.

The other way is to modify the operating system itself as it resides on the disk. This change is required only once. From then on, as the system is booted, the printer driver code is loaded into the machine. The assembly code for the driver is shown in Listing 2. Note that this code was assembled at $A000 for illustration only. I actually placed the code at $24C1 - $24CC. This overlays unused (on a C1P) code for a 550 board output routine.

In either case, the procedure is similar: assemble the driver code, find a place in the operating system to put it, place it there, and finally, modify the I/O vector for device #4 (located at $2317 - $2318) to point to the routine instead of the "null" routine. IMPORTANT: the I/O vector MUST point to one byte before the actual output routine. This is required for correct OS65D operation.

CONCLUSION

The DEVICE CONTROL DC650 Motherboard and DC660 Dual PIA board provided me with an excellent expansion system for my C1P. The first expansion project, this parallel printer port, had no problems at all; and still leaves me with one whole PIA for future projects before I need to buy another PIA board. This system seems to solve a basic problem with the C1P, that being its limited expansion possibilities.

ACKNOWLEDGMENTS

I would like to thank Leroy Erickson for his assistance and encouragement in this project. Without his help in understanding OS65D, I would not have been able to make the modifications as easily as I did. Also, I should like to thank Device Control for the assistance they gave me in providing information on using 6522 VIAs in place of the standard PIAs on the DC660 board.

★

DIRECTORY RESTORER FOR OS-65D

by: Willis H. Cook
1298 Renee Drive
Lilburn, GA 30247

Here is a little utility program, one that you may never have to use, but one that is very helpful if you ever do: a directory restorer.

I don't kow why directories go away, and it doesn't happen very often, but when it does, it is a terrible nuisance to resurrect them. If the directory track header is gone, the OSI COPIER utility can't copy the disk, so you have to copy your files onto a new disk one at a time. Even if you can make a copy, you still have to use the CREATE utility and re-create every file in order to get the names put back into the directory. In order to use this program, you must have a current copy of the directory of the disk you want to restore. If you don't have one, you must hunt through the disk and find all your files by using the command DISK!"LOAD xx", where xx is every track number where a file might start. You will get an error message if you try to load a track that is not the start of a file. If the file loads, you must LIST it to identify it. It is easier to have a current directory available.

DIRECTORY FORMAT

Disk directories are stored in two parts: disk track 8, sector 1 holds up to 32 file names, and track 8, sector 2 holds an additional 32 names. (These numbers refer to 8" disks. The track number is 12 for 5 1/4" floppies.) One disk sector corresponds to one page of memory, or 256 bytes.

The DIR utility could have used a normal 12-page buffer (8 for mini-floppies) but this would have used up an additional track on the disk. Instead, OSI provided a scratchpad buffer at location 11897 ($2E79). The buffer only holds one page of data, so the first page is read into the buffer, operated on, then is overwritten by the second page.

Each file name gets eight bytes of buffer space: six bytes corresponding to the maximum of six characters in the file name, and two bytes to define the starting and ending tracks. Eight bytes /

```
10 PRINT TAB(11)"****************************************"
20 PRINT TAB(11)"*                                      *"
30 PRINT TAB(11)"*          DIRECTORY RESTORER          *"
40 PRINT TAB(11)"*                                      *"
50 PRINT TAB(11)"****************************************"
60 PRINT : PRINT : N=1
70 PRINT"BE SURE YOUR BAD-DIRECTORY DISK IS IN THE DRIVE."
80 PRINT : INPUT"READY";A$ : IF LEFT$(A$,1)<>"Y" THEN END
90 :
100 REM             CLEAR DIRECTORY BUFFER
110 :
120 PRINT : PRINT TAB(23)"DIRECTORY PAGE";N : PRINT
130 A=11895 : REM Buffer address the same for both 8" & 5 1/4"
140 FOR I=1 TO 256 STEP 8                                  disks.
150 : FOR J=1 TO 6
160 :   POKE A+I+J,ASC("#")
170 : NEXT J
180 : POKE A+I+7,0 : POKE A+I+8,0
190 NEXT I
200 :
210 REM                 ENTER DIRECTORY
220 :
230 J=0
240 PRINT TAB(10)"ENTER 0,0,0 TO END." : PRINT : PRINT
250 INPUT "NAME, FROM, TO";A$,TS,TE
260 IF A$="0" THEN 360
270 IF LEN(A$)<6 THEN A$=A$+" " : GOTO 270
280 FOR I=1 TO 6
290 : C=ASC(MID$(A$,I,1))
300 : POKE A+I+1,C
310 NEXT I
320 POKE A+8,16*INT(TS/10)+TS-10*INT(TS/10)
330 POKE A+9,16*INT(TE/10)+TE-10*INT(TE/10)
340 A=A+8 : J=J+1
350 IF J<32 THEN 250
360 IF N=2 THEN 390
370 DISK!"SAVE 08,1=2E79/1" : REM "SAVE 12,1=2E79/1" for 5 1/4"
380 N=2 : GOTO 120                                         disks.
390 DISK!"SAVE 08,2=2E79/1" : REM "SAVE 12,2=2E79/1" for 5 1/4"
400 END                                                    disks.
```

name times 32 names gives 256 bytes, a perfect fit in the buffer. There is a slight complication, however. If you have ever read a directory sector into the video memory to see it on the screen, you noticed that the track numbers were represented by graphics symbols. If you looked up those symbols in the Character Graphics Reference Manual, you also noticed that the graphics symbols correspond to the track number converted to hex. For example, if you have a file residing on tracks 21-22, it will be represented in the directory file as FILNAM!", where FILNAM is the name of your file, ! is the character represented by $21 and " is the character represented by $22.

The advantage of this scheme is that any track number can be represented by one character, since there are 256 unique alphanumeric/graphics characters available, more than enough to represent the maximum 77 tracks on a floppy. The disadvantage is that the track numbers have to be translated from the user's input when stored in the directory, and re-translated from the directory back into decimal numbers when displayed in a directory printout. Lines 320 and 330 in the listing do the conversion from decimal to hex. Do they look familiar? The same routine is used in OSI's CREATE utility.

HOW TO USE THE PROGRAM

When you run the program, presumably it will come from a good disk. The prompt at line 80 allows you to replace that disk with the one having the bad directory. Type Y when you are ready. After a pause the program prompts you for the names and track ranges of each entry in the directory, starting with page one. For each entry, enter the file name, the starting track number and the ending track number, just like they appear in the directory display. Be sure to include OS-65D3 and BEXEC* and any other system files along with your own. If you have less than 32 files, enter 0,0,0 after the last one and they will be stored in directory page one. After another pause the program will ask you to enter any names for directory page two. Since you are finished, enter 0,0,0 again and the program terminates.

If you have more than 32 files on your disk, the program will stop you after the 32nd one and store the ones entered so far in page one. Then it will come back for the remaining ones. Enter 0,0,0 after the last one and the program will terminate.

HOW THE PROGRAM WORKS

When you answer "Y" to the prompt in line 80 the FOR...NEXT loops at lines 140 to 190 fill the buffer at 11897 with pound signs for file names and ASCII zeros for track numbers. Notice in line 130 that A, the starting address for the POKEs is 11895, not 11897 as you would expect. That is because the first POKE in line 160 is at location A+I+J. I and J are both equal to 1, so the initial address is 11897. Each time you enter a name and track numbers the program POKEs the ASCII value of each character of the name into the next available space in the buffer. If the name is less than six characters long, it is padded on the right with blanks. If the name is longer than six characters, the loop in lines 280-310 only reads the six left most characters, so there is no need to ask for the input again as OSI does in the CREATE program. Next the starting and ending track numbers are converted to hex and POKEd to the buffer. The buffer position counter A is incremented by eight and the input counter J is incremented by one. If less than 32 file names have been entered, control passes back to line 250 for another input. If 32 names have been entered or if the last entry was the termination signal 0,0,0 the buffer is stored into the appropriate directory sector on the disk. The program executes twice, each directory page being handled separately. The flag N corresponds to the page number. Notice in lines 370 and 390 the SAVE commands will have to be changed for mini-floppies.

WHAT THE PROGRAM CAN AND CANNOT DO

Obviously, the main purpose of the program is to restore directories that have become unreadable, but it can be used for other things as well. For

example, you can manually sort a directory by track or name and save it back to disk in that order. This allows you to get a sorted directory using the faster option 1 of the DIR utility rather than the directory sort routines.

You may also use it to assign a file name to a particular place in the directory. I usually assign my disk scratch file, TEMP, to the last tracks on the disk, and I like the name to appear at the end of the directory. After entering all the disk file names, enter ######,0,0 to fill the directory with nulls, then enter TEMP as the 64th entry. Then no matter how many files are added or deleted, TEMP always appears at the end of the directory.

Just remember that you are not creating files when you build a directory. In fact, the value of this utility rests on the converse of this statement -- the files are not destroyed just because the directory is damaged.

My system is a C2-4P with one 8" disk drive. I have tested the program under both versions 3.2 and 3.3 of OS-65D. I have NOT run it on a system using 5 1/4" disks and the information on the required changes comes from OSI's documentation. If you run it on a 5 1/4" system, please first try it on a spare disk and let the other PEEK(65) readers know of any difficulties you encounter.

One final note for those who want to use the program to change only one page of the directory (no one ever wants to run a program as-is): to change page 1 only, simply enter a C/R after the page 1 portion has finished and you see the message DIRECTORY PAGE 2. That will terminate the program without altering page 2. If you want to change the second page only, add the following line:

```
65N=2
```

This will take you directly to the second page modification routine without affecting page 1.

★

ADD 8K RAM TO YOUR SBII OR C1P

by: Guy Vanderwaeren
06100 Mexico 11 D.F.
Mexico

This description of a do-it-yourself board of 8K RAM to add to a SBII or C1P is something for the routined hardware freaks. Because of its relative complexity it is not recommended as a first project.

I have owned a SBII for about two years now and I am sometimes limited in program writing by the 8K RAM on my 600-board. On the other hand my economic situation does not permit me to buy the 610 extension board. Therefore, I got the idea to design an extension board with 8K RAM. As the possibility exists to be able to buy the 610 board later, it would be nice to have the extension board with switches so that it is possible to still use it later, together with the 610 board. This one comes normally with 4K of RAM, so I thought it would be very nice to be able to switch the extension board to different starting addresses, so it would be possible to change only a few switches when I populate the 610 board 1K at a time. If the 610 board is fully populated with RAM, then the extension board still can be used, because the memory map of the SBII shows an unused portion of 8K in the locations just following the 610 board RAM. The board described here has all these possibilities! It has 8K of RAM, with switchable starting addresses and ending addresses (more on this later). The addresses can be set in steps of 1K at a time.

Another thing that's good to know, is that this board, for the moment, only exists on paper, so there is a possibility that there are some errors in the circuit. Nevertheless, I was asked by the Editor to send the diagram and description anyway. I would greatly appreciate it if anyone who finds a circuit error, would write to me, via PEEK(65), so that everybody can correct it. Also, further ideas or suggestions will be welcomed!

Let's have a look now at the circuit itself. On the first glance, it seems quite complicated and the printed circuit will certainly be. The working is, on the contrary, a lot easier. There are two great blocs: the RAMs, which are U1 to U16, and the address decodification, which is the rest of the diagram.

The RAMs are connected in parallel with their address lines (A0 to A9), their data lines (D0 to D7), their power lines (GND and +5V, the whole circuit works on +5V and will consume about 1.3A max.) and the read/write line (R/W). Because the 2114 type RAM is 4 bits by 1K memory, the data lines are separated into two groups of 4 and each chip-select line (CS1 to CS8) goes to two RAMs in parallel. All the address lines, data lines, the R/W line, the clock line (02) and the data-direction line (DD) are connected to the 40 pins extension connector on the 600 board. It would be a good idea to put buffers on the R/W line, the 02 line and all the address lines, to evitate the trouble of a burning CPU. These lines are NOT buffered on the 600 board. On the other hand, the data lines are buffered, but you have to install 2 8T28 chips in the empty sockets U6 and U7 on the 600 board.

How does the decodification work? First, let's look at the circuit around U24, U25, U26. The possible locations for the RAM are from $2000 to $9FFF. The decoder U24 covers this whole range with the input of A15, A14, A13 and an enable line coming from N10, which gives a more specific address selection (see later). The 4 outputs are ored together with N12 and DD1, DD2. This, together with the clock (02) and the R/W, gives us the data direction signal (DD), which is buffered with N13 and N14 in parallel to have an open collector output with enough power.

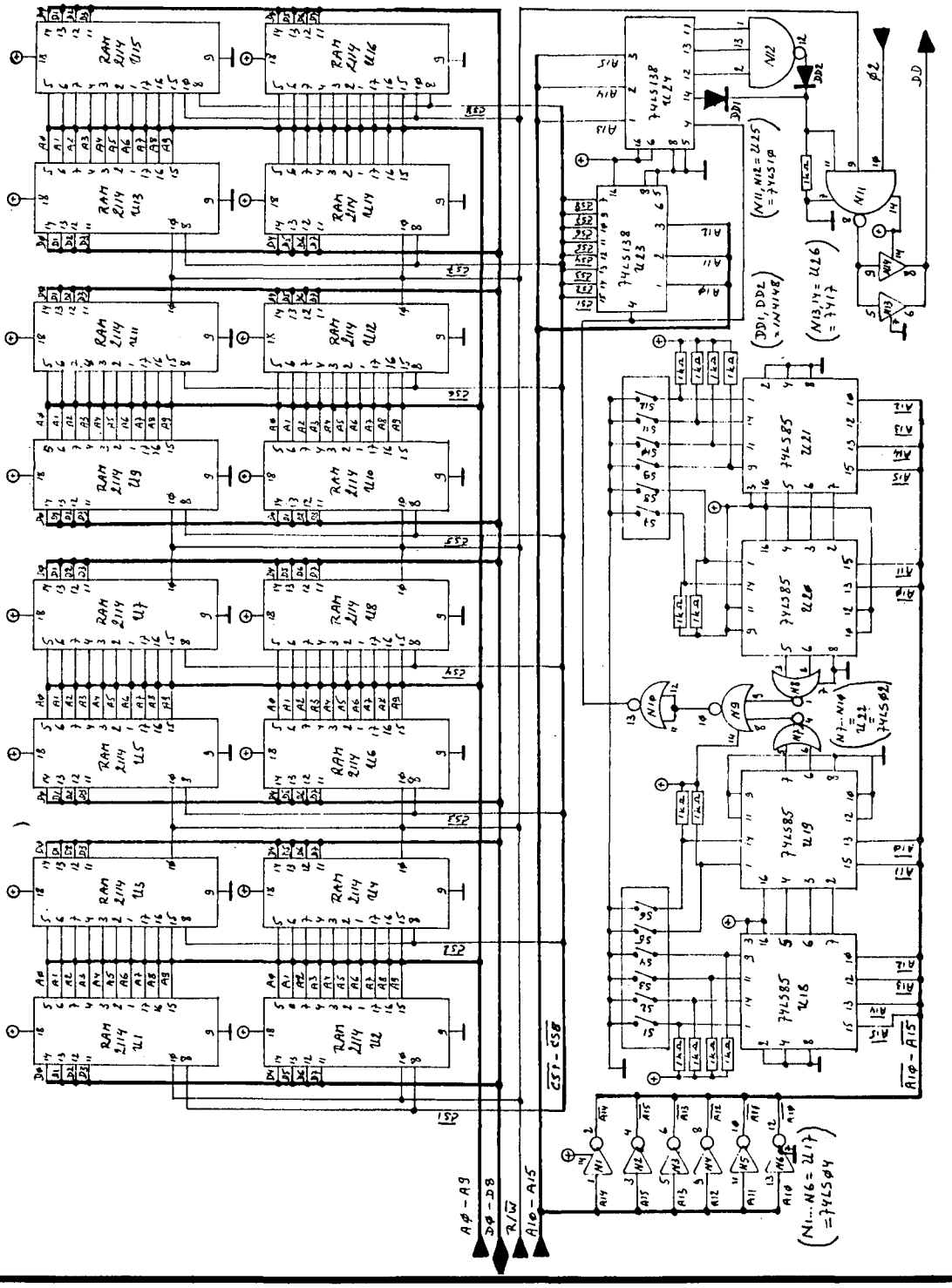
Next, we have the real address decoding with U18 to U22. Four 4-bit comparators (U18, U19, U20, U21) are used to select the 8K range. This range is set with 12 DIP-switches (S1 to S12). The first 6 of these switches set the starting address, the last 6 set the ending address. Why set the ending address? Two reasons: first to simplify the following chip-select generation, and second to allow for other uses of the memory locations in the range of $2000 to $9FFF. This of course, not on the same ones, already used for RAM.

Table 1 gives the settings of the switches for each starting address.

* * * * *TABLE 1:

Starting Address	Closed Switches
2000	3 4 5 6 10
2400	2 7 10

FIG. 1 - 8K RAM CARD



2800	2 6 8 10		5800	234 6 89 11	The comparators U18 and U19 check if the address on the bus is greater than or equal to the starting address, these two possibilities ored in N7. The other two comparators, U20 and U21, check if the address is less than or equal to the ending address, these two signals ored in N8. The range signal is further ored in N9 and inverted in N10 to have a
2C00	2 5 78 10		5C00	2345 789 11	
3000	2 56 910		6000	23456 1011	
3400	2 4 7 910		6400	1 7 1011	
3800	2 4 6 8910		6800	1 6 8 1011	
3C00	2 45 78910		6C00	1 5 78 1011	
4000	2 456 11		7000	1 56 91011	
4400	23 7 11		7400	1 4 7 91011	
4800	23 6 8 11		7800	1 4 6 891011	
4C00	23 5 78 11		7C00	1 45 7891011	
5000	23 56 9 11		8000	1 456 12	
5400	234 7 9 11				

single enable line for the chip-select generator U23. This generator has 8 output lines, used to select the 1K bloc of RAM.

This construction makes it impossible to tell which chips form the first 1K of RAM, which the second, etc., because this depends on the chosen address range of the whole 8K bloc. Don't feel very frustrated about it, the computer will know which K to use first. Of course, a close study of the decoding in each case can allow you to define this also. This concept also makes it a necessity to put all the 16 RAM chips on the board, because otherwise you may have holes in your memory.

A last word on the diagram: for simplification the address lines, data lines and chip-select lines are shown as big lines (bus). The connections to make with the 600 board (see fig.) are shown as arrows, which at the same time show the direction of the signal (from or to CPU).

I have further ideas, which I am working out. With the Editor's fiat, you may see in the (near) future a diagram of a 6K EPROM board, an EPROM programmer for 2708 and 2716, 16 parallel I/O board and a parallel Centronics - type printer interface. Who knows what's next??? Good luck.

★

FUN WITH OSI'S UTI

by: Robert S. Baldassano
4045 Ashbrook Circle
San Jose, CA 95124

Ever since I saw OSI's CA-15V, Universal Telephone Interface (UTI for short), in their slick color catalogue, I knew I had to have one someday. Here was a 300 baud, direct connect, auto-dial and auto-answer modem, with VOTRAX, that plugged right into the OSI backplane and was capable of all kinds of amazing things.

Unfortunately, these amazing things did not come cheap, the unit originally costing around $799 plus $199 for a telephone isolator. That price has escalated over time due to inflation and the addition of inflection in the VOTRAX module.

Well, thanks to a lot of unexpected overtime this past summer, I was finally able to assemble enough money to buy one without breaking the family budget.

I called my friendly and helpful OSI dealer, Rick Guido, of Computer Business Service in San Jose and asked him to order one for me. Rick had never sold a UTI to any of his customers, which is surprising since most of his clients are businessmen. So when I told him I needed the CA-CBT telephone line isolator as well, he checked with the West Coast OSI distributor, to keep me from spending unnecessary money. He was told it was only needed for multi-line systems (more on that later). With a discount from Rick, the cost of the CA-15V was now $825, more than my new CM-20 memory board had cost me. But I had the cash, and knew it was now or never, so I went ahead and ordered it.

A short time later, my UTI arrived and in checking over the documentation, it was quickly evident that the line isolator was a necessary part of the system. The CBT is required by the phone company, and in fact, could be rented. It isolates the phone line from the computer and the UTI's Ring Detection and

Connect Control circuitry, as well as the UTI Sound Duplexer, connect to it.

While Rick tried to get a CBT from the distributor, I installed and wired the supplied six RCA jacks and DA15S connector in the predrilled holes on the back of my computer and plugged the UTI into my 8PDF backplane. Each of the wires was clearly marked, and I just needed a wrench and soldering iron to install it in about 30 minutes. A multimeter is also helpful to insure that +5 VDC at 1.0 amp, +12 VDC at 130 ma and -9 VDC at 100 ma are available at the slot you want to plug it into.

The VOTRAX is a separate potted module that connects to a 16-pin DIP socket on the UTI via a ribbon cable. It can be mounted to the board with No. 6 screws and stand-offs (not supplied).

The six RCA jacks allow the UTI to control record and playback functions of two tape recorders, provide an amplified VOTRAX output for direct connection to an 8 ohm speaker, and also provide an auxiliary input to the phone line. The DA15S allows connection to the CA-CBT.

The system comes with two disks, plus backup copies. One contains demonstration programs for the VOTRAX, and the other modem demonstration programs. The operating system is OS-65D V3.2 HC with modifications. An OS-65U Level 3 version is available for registered users. After bootup a menu is displayed for the VOTRAX selections; the modem disk also uses a menu after first asking for current time inputs.

I booted the modem disk, and after setting the time was given a selection of seven functions:

```
1  CALL ANY NUMBER
2  MONITOR PHONE CALLS
3  CALL ANY MODEM
4  MONITOR FOR MODEM CALLS
5  INTERROGATE WEATHER STATION
6  WEATHER STATION
7  MODEM SELF TEST
```

The UTI operation manual explained that the first program dialed any number typed in. The program uses touch tone, or by changing a GOSUB call, pulse dialing.

The second program was more interesting, but required a tape recorder to be hooked into the system. This program would answer any call with VOTRAX voice, asking you to leave a message at the tone, then it would run the tape recorder for 70 seconds. Or, if you called and entered a password (using a touch tone phone or generator) that you preselected, you could get into your AC home control or security system to control lights or appliances or check your security status, as well as get the current time, all prompted by your friendly VOTRAX.

"CALL ANY MODEM" is a "DUMB" terminal program that also outputs to a serial printer on a polled keyboard system or a parallel printer in a Serial system.

"MONITOR FOR MODEM CALLS" answers "DUMB" terminal calls.

The next two selections are part of an auto interrogate demo that can poll a weather station at desired intervals, selection six being a simulated weather station that will send simulated statistics if called.

"MODEM SELF TEST" checks the modem with 500 input and output characters. Since I knew I couldn't use the modem yet without the CBT, I tried the self test and it passed with flying colors - so everything was O.K. - or was it?

I then decided to try the VOTRAX disc. The manual (fairly well written by the way) said the UTI would support the VOTRAX just like a CA-14A VOTRAX I/O board. It turned out that the VOTRAX software is in fact CA-14A software.

Well, I fired up the disk, and got a menu of seven programs:

```
1 TALKING CALCULATOR
2 DEMO #1 (W / INFLECTION)
3 DEMO #2
4 BLACKJACK (GRAPHICS VERSION)
5 BLACKJACK (SERIAL VERSION)
6 23 MATCHES
7 BUZZWORD
```

The first program on the menu solved problems verbally while printing them on the screen or spoke random numbers generated by the program.

The next two programs demonstrated VOTRAX with and without inflection. I was to learn later that without inflection VOTRAX sounds like a science fiction robot. With inflection, it was much more natural, but not as natural as

the method used by Texas instruments.

The other programs were standard games with voice, BUZZWORD being the most fun at first, as it generates three words that sound good together but have no real meaning.

Though not shown in the menu, the disc also contained VOTRAX drivers with from zero to two buffers so you could write your own programs.

With anticipation, I selected TALKING CALCULATOR and the program prompted me through the selection of calculator or random number generator mode. I chose calculator and entered 4+5 when prompted for an equation and - HUNG THE SYSTEM!

I re-booted and tried the other demos, all with the same result. I next checked the voltages on the 16-pin DIP and they were O.K.

The VOTRAX can be activated by two methods. The first uses a PRINT #5 statement followed by a mnemonic representation of phonetics (phonemes). For example, to say "HELLO", you would type: PRINT#5,"PAl PAl H H H 4EHl 3UH3 L 3UH3 302 4Ul 4Ul". There are sixty-three phoneme commands used with the VS-6 VOTRAX module. The second method uses DATA statements which contain binary numbers that correspond to the phonetics. These are poked into location 52480 in conjunction with a WAIT statement. This method is independent of the software in the system and can, therefore, be used in any form of BASIC.

I called Rick and told him the problem I was having and he suggested trying a direct poke to 52480. I tried this and got a sound. It appeared that the output circuitry was working.

On subsequent visits to my home, Rick and I tried to tackle the problem, thinking it was software related, and did find out it hung on the WAIT statement.

To add to my problems, Rick could not get a CA-CBT from the distributor, the phone company seemed to be out and so did OSI. He finally located one at an East coast OSI dealer for $132.06.

When the line isolator arrived, it only took about ten minutes to install and plug into a power outlet and into the modular jack of the phone outlet. I had installed a dual adapter in the outlet so I could have my phone hooked to the same line.

I loaded the modem disk and used the CALL ANY MODEM program to call CompuServe, all without touching the phone. The modem worked fine, but when I tried the VOTRAX disk, I still could not get VOTRAX voice to work.

Rick finally called in his technician and he quickly isolated the problem to a cracked trace between an eight input AND gate and the input to a tri-state buffer. A little solder across the crack and my computer was talking my ears off.

It looked like my problems were over. I tried controlling my lights by calling the computer while away from home, and it worked perfectly. I had the computer call numbers I typed in and it dialed them quickly and accurately, telling me when a voice was on the line or hanging up when the line was busy.

The UTI is capable of operating in originate or answer mode under software control. Parity and stop bits are also software selectable. I subscribe to both CompuServe and Dow Jones. CompuServe seemed to work fine, but I got a lot of garbled text over the Dow Jones circuit. In tracing the problem, I found out that the OSI software was set for eight data bits, no parity and two stop bits. According to my CompuServe documentation, two stop bits are used at 110 baud only. I changed the stop bits from two to one, and both services have worked perfectly ever since.

So, in all I have spent over $957 for this board and isolator. I could have bought a Hayes Stack Smartmodem for $279. This is also an auto-dial, auto-answer system, but it would not have the VOTRAX, nor would it have the capability to control tape recorders or other devices. There are VOTRAX units out now for $375, but again I still wouldn't have all the capability in one compact board that plugs right into the backplane.

In summary, I am happy with my purchase and plan to purchase OSI's Vocalizer I software to make speech easier to program. I also want to write or purchase a smart modem program that will work with the UTI. Other possibilities exist. How about a program that would use the UTI and Vocalizer software to give a voice output of the stock quotes? Now, if I can only get more overtime --Hmmmmm.

★

OS-DMS (V. 9/79)
CUSTOMIZED KEY FILE DUMP

by: Frederick S. Schaeffer
84-55 Daniels Street #4f
Jamaica, NY 11435

Using OS-DMS's data base utilities can take a frustrating amount of time. A few weeks ago, I was asked to provide an organization with a geographical breakdown of members vs. states of residence. I looked for a "number of occurrence" in the stat section of "STAT03/STAT3A" in vain, because this version doesn't have it. To solve this problem sensibly, rather than count totals by hand off a printout in zip key order (which takes hours for a file with 800 records) I did some heavy thinking and found a better way. I customized the "kdump" utility to do the job for me.

The prerequisite to running the amended kdump is to load a KEY FILE. In my case, I use PEEK (65) Inc.'s beautiful "KYUTIL" program which takes about 12 minutes to load/sort my keyfile. Since "KYUTIL" is capable of loading more than the contents of one field in the resulting keyfile, it is highly suitable for this "number of occurrences" counting. If you load/sort by state and zip, you get a listing that can be broken down to greater detail in regard to foreign addresses, or, if you're dealing with an organization that has several types of membership, you can get a breakdown of membership classes (i.e. revenue) by state.

Basically, what happens in the amended "kdump" is that the conventional kdump prints out (at 9600 baud) on the screen and where a breakpoint occurs the resulting count vectors to the printer, resulting (800 records) in a total run time of 2:50 minutes from menu selection to the end, and that's more like it!

You don't want to permanently change kdump so you begin with (additions underlined); and "A" ahead of line # means newly added:

Listing on page 18

OSI COMPATIBLE PRODUCTS

56K 2-MHz Ultra Low Power CMOS Static Memory Board **MEM-56K $850**

Partially Populated Boards (Specify address locations required) ... **MEM-48K $750**
MEM Board uses the new 2K-Byte Wide Static RAM chips which are **MEM-32K $550**
2716 EPROM compatible. Any 2K byte memory segment can be **MEM-24K $450**
populated with RAM or EPROM (or left empty for use of Address Space **MEM-16K $350**
by another board). Fully expandable to any memory size you will ever **MEM- 8K $250**
need. No special addressing requirements, just solder in extra sockets **MEM- 4K $200**
Extra 2K RAM Memory Chip **$24**
Optional Parallel Printer Port **-P $120**
Optional Calendar/Clock Software available in EPROM) **-T $ 25**
Both options (Disk software mods provided for use of 6522 VIA on printer). **-PT $125**

EXAMPLE USES:

C4P & C8P: Expansion to **4K RAM** of Basic workspace.
Parallel Printer Port — Reserve Serial Port for MODEM
Calendar/Clock Displaying on unused portion of screen.
Space for 5.75K of **Enhanced System Monitor EPROMS.**

All of this on 1 Board, using only one of your precious slots. Software for Enhanced System Monitor capabilities is continuously being developed and improved. As new EPROM Monitors are available, you may upgrade to them for any price differential plus a nominal $10 exchange fee. Another possibility is to fill any portion of the memory with Basic Programs in EPROM for **Power-on Instant Action.** This custom EPROM programming service is available at $25 per 2716 (Includes EPROM). Extra copies at $15 for each EPROM.

C4P-MF & C8P-DF:

Memory expansion to **48K.**
Add 6K **Memory** above BASIC for special software requirements.
Parallel Printer Interface and/or **Displaying Calendar/Clock.**
Add 1.75 K **Enhanced System Monitor ROM.**

C3: Up to **56K of Memory** Expansion — can be addressed for **Multiuser.** (Optionally, each user can have his own **Dedicated Printer Port).**

C1P,C4P & C8P FLOPPY DISC CONVERSIONS:
Memory/Floppy Board (Includes MI48P1 ROM) ..... **MEM F-16K $450**
C1P-600 Board Adapter & Cable ..... **A600/48 $ 50**
Additional Memory/Printer/Times ..... **(See MEM Board Prices)**
5¼" Drive/Case/Power Supply & Cable to MEMF Board **FD5 $399**

IEE-488 INTERFACES AND SOFTWARE:

The General Purpose Instrumentation Bus (GPIB Controller interface is available for all OSI Computers. Machine code GPIB Drivers are linked to Basic to provide easy control of IEEE-488 instruments which is equal to the best of Hewlett-Packard Controllers and far superior to most others. Basic Commands for Serial Poll, Parallel Poll, IFC Clear, full Local/Remote Control, Respond to SRQ Interrupts, Send Trigger, do Formatted Input/Output, Direct Memory Input/Output and MORE. Interface includes IEEE-488 Ribbon Cable/Connector.

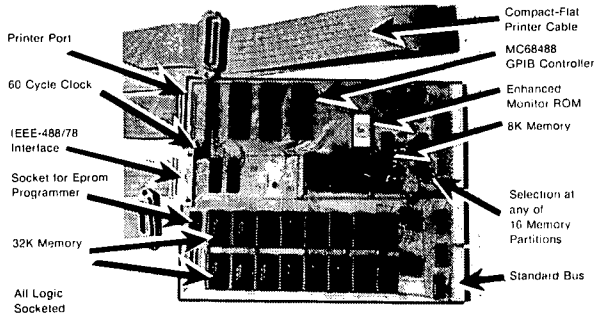
GPIB Controller Interface for C2, C3, C4 and C8 Systems ..... **GPIB 4-488 $395**
GPIB Software for **OS-65D** (Add -8 for 8" or -5 for 5") ..... **GPIB 488-D $ 70**
GPIB Software for **OS-65U** ..... **GPIB 488-U $100**
GPIB Software on two 2716 EPROMS for **ROM** based systems ... **GPIB 488-R $100**
Add Optional **Parallel Printer Interface** to GPIB 4-488 ..... **-P $120**
Add Optional **Calendar/Clock** to GPIB 4-488 ..... **-T $ 25**
Add **2K RAM** to GPIB 4-488 (Specify location, $4000-$BFFF & $D000-$EFFF available) **-M $25**
GPIB Controller for C1P, Includes Software, Clock, All Features of ROMTERMS, & space for 6K EPROM ..... **GPIB 6-488R $395**
Add Optional Parallel Printer Interface to GPIB 6-488R ..... **-P $120**

EPROMS:

C1P ROM with 24/48 Col Display for Series II, Smart Terminal, Line Editing, Corrected Keyboard Screen Clear and More ..... **ROM-TERM II $59.95**
C1P ROM with 24 Col Display. Other ROM-TERM II Features, Disk Boot, and ROM/Disc Basic Interchange ..... **ROM-TERM $59.95**
C4P-MF/C8P-DF Disk warm start, changed IRQ Vector and just flip switch for Serial or Video System with Corrected Keyboard ..... **SYNKEY $39.95**

ENHANCED MONITOR ROMS FOR USE ON GPIB 4-488 & MEM BOARDS:
Expanded Support for **C4P & C8P** Featuring Calendar/Clock, Line Edit, Smart Terminal, Memory Files, Parallel Printer Control, Corrected Keyboard, All Features of ROMTERMS, **Disk Support** with Warm Start and More **MI48P1 $59.95**
Expanded C2 Monitor with Calendar/Clock Software, Hard Disk Boot, Warm Start and Multi-User Control for C2 Systems **MIC2-1 $59.95**

IEEE-488 CONTROLLER INTERFACE



THE GPIB 4-448 INTERFACE BOARD CONVERTS ANY OSI COMPUTER INTO AN IEEE-488 INSTRUMENT BUS CONTROLLER!

BENEFITS — Provides a Sophisticated Instrumentation Controller at very low cost (often saving thousands of Dollars). The combination of IEEE-488 Instrumentation Controller and High Capacity Hard Disk file storage available on OSI Computer systems is available at a fraction of the cost required by the nearest competitor. The IEEE-488 Bus, also known as the GPIB,HP-IB or IEC-625 is the most popular International Standard for connecting instrumentation systems. This 16-line bus is designed to interconnect and control up to 15 instruments at a time. Currently, over 2000 different instruments are available to work on this bus. They include: Plotters, Digitizers, Printers, Graphic Displays, Recorders and a multitude of specialized Test/MeasurementControl Equipment.

EPROM-ABLE — Can be used with a C4-P to create a dedicated IEEE-488 controller.

C2-D MULTIPLE USER SYSTEMS

SAVE — 2 and 3 user Time Sharing Systems are available on the C2-D Winchester Disk Computer at a considerable cost savings from C3 Multiple User Systems. The 3 user C2-D System can be expanded to include a word processing printer, 4 other parallel printers and 3 serial printer interfaces.

COMPATABLE — The special C2-D Multi-User Executive Program is 100% compatable with OS-65U V1.4. The Multi-User Real Time Clock, Memory Partition Control and IRQ Interrupt Management are done on the Micro Interface Memory Board. Thus, the CPU board is not modified and remains in factory condition.

CONVERSIONS — The Up-Grade of your existing C2-D Computer to Multiple User Configuration is also available. Call for details.

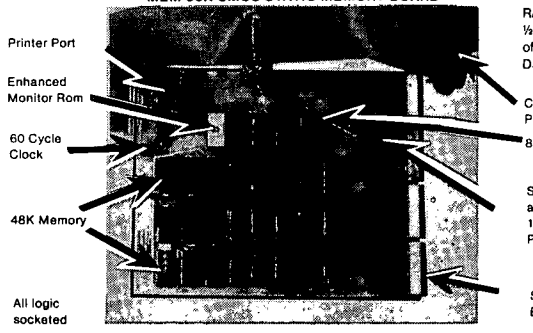
FLOPPY DISK UPGRADES FOR C1P, C4P & C8P

Our Memory/Floppy Board provides easy conversion of 502 and 600 CPU Computers to Floppy Disk Operation. The **MEMF** Board has a floppy disk interface which includes a data separator and the ability to automatically lift the disk drive heads — your floppy disk lifetime will be extended many times. You will retain the cassette interface for your existing software; which can easily be converted to Disk.

This **MEMF-16K** Board is populated with **16K RAM** (50K possible) and has features of the MEM CMOS Static Memory Board with an added floppy interface. The low power memory means extra power supply not required. **ROM Basic** is retained even when Board is populated for 48K Disc Basic. An optional Parallel Printer Port and Real Time Calendar/Clock is on board.

Complete Ready to Run conversion kits with 5¼" or 8" Disk Drives are available.

MEM-56K CMOS STATIC MEMORY BOARD



ULTRA-LOW POWER — By using CMOS Static RAM Memory, the total power consumption is about ½ Amp at 5 Volts when populated for 48K. In fact, most of power is used by the Address Line Buffers and the Data Transceivers.

MULTI-USER — Can be addressed for any of the 16 multi-user memory partitions. The low power and single memory board/partition simplify installation and provide a typical $1400 saving for a 3-user system.

MICRO-INTERFACE
3111 SO. VALLEY VIEW BLVD., SUITE I-101
LAS VEGAS, NEVADA 89102
Telephone: (702) 871-3263

Check with your local Dealer or Order Direct.
Phone orders accepted.
TERMS: Check/Money Order/Master Charge/VISA
Sent POSTPAID ON PREPAID ORDERS.
Foreign Orders: Prepaid only.
Add 5% for handling/shipping.

```
   9  S$="-":A1=390:A2=645
A 40  PRINT:INPUT"WANT TOTAL COUNTS (Y/N)";FILE$
A 41 IF FILE $<>"N" THEN PRINT "DEV TO PRINT COUNTS =5"
A 42 IF FILE $="N" THEN GOTO 49
A 43 PRINT:INPUT"TITLE(NEXT LINE)DATE";H$,DATE$
A 47 PRINT:INPUT"WHICH PRINTER - - etc.
```

(Note here follow spooler control codes to select printer which are not of general interest)

```
A 80 PRINT #5, H$
A 81 PRINT #5
A 82 PRINT #5, DATE $
```

(If using date patch, put DT$ in Line 82 and omit date input)

```
A 83 PRINT #5
```

(Now you want to bypass the regular kdump "dump" section)

```
A 200 IF FILE$="Y" THEN GOTO 600
A 600 PRINT#1:PRINT#1,"KEY FILE WITH COUNTS OF ";NA$;
A 610 PRINT#1:PRINT#1,"KEY FILE LABEL IS ";LB$;
A 620 FOR Z=K1 TO LEN (LB$):PRINT#1,TAB(37);S$;:NEXT
A 630 PRINT#1:PRINT#1,"KEY"; TAB(59);"INDEX#"
A 640 PRINT#1, "***" ; TAB(59);"********" :PRINT#PD
A 645 TMP$=Z$:CC=CC1
A 650 INPUT%K1,L$:INPUT%K1,I:Z$=MID$(L$,K2)
  :PRINT#1,Z$;TAB(60);I
A 660 IF Z$<>TMP$ THEN GOSUB 699
A 670 IF INDEX(K1)<EDF GOTO A2
A 671 PRINT#5,Z$;"TOTAL";CC:TC=TC+CC:REM PRINT LAST TOTAL
A 680 PRINT:PRINT"DONE":GOSUB 6Z000:GOSUB 60000
A 690 PRINT#5:PRINT#5,"GRAND TOTAL";TC-1;:  GOTO 510:REM 510 IS
  COMMON EXIT TO "DBMSYS"
A 699 TC=TC+CC:IF TC<2 THEN GOTO 701
A 700 PRINT#5,TMP$;"TOTAL ";CC
A 701 CC=0: GOSUB 60000
A 702 RETURN
```

(SELECT (P) PRINTER FOR BREAK POINT LIST.)

The resulting printout looks like this-

```
XYZ COMPANY DISTRIBUTION
12/13/82
NY TOTAL=32
NJ TOTAL=62
NH TOTAL=50
GRAND TOTAL 144
```

★ ★ ★

KEYWORD REVIEW

by: Dickinson H. McGuire
1125 Mainsail Drive
Annapolis, MD 21401

Keyword, Ohio Scientific's new word processor based on Palantir and using the new OSI CP/M version 2.25 is a far cry from any other word processor offered by OSI. It is also a far cry below Wordstar. Not just because it is of itself inferior--it is not; but because there are so many other things to go with it when you buy Wordstar. There is no spelling program, merged letter program, or spread sheet program.

This word processor is very easy to use. You get a set of labels to stick on your keyboard. Don't put them on the front of the keys; they don't stick too well. Put them on the top of the keys and live with the inconvenience of not being able to see the letter there.

Keyword has a couple of very nice features I haven't seen before. The first is the Lexicon. You may create a table of up to 39 phrases which can be called into your text with only two keystrokes. The second feature is the concept of Direction.

You may set the Directions as either from the cursor position to the front or to the end of your document and then go into "Rangefinding." This means that you can control the cursor movement by line, page or screen. You also may use this for reformatting, deleting, moving, copying and video attribute changing.

Speaking of video attributes, you may configure your terminal driver so that certain video attributes relate to printer attributes. In my case, Boldface on the printer is Bright on the CRT and Underscore on the printer is Underscore on the screen. This is a whole lot better than having some control code on the screen which takes up room and which I must interpret.

Other features are Block insertion, virtual files, hard and soft hyphens, pagination, and carriage returns. There is a define menu which allows you to quickly change the printer specs, a file menu which will allow you to look in the directory, rename or erase files and change disks. There is also a very useful help menu.

For $400 you get a manual and a distribution disk. This disk has no system on it and is in the new OSI format so you must also buy the new CP/M version 2.25 for $190. The manual contains a meager section on installation which works very well if you happen

to own one of the supported terminals and printers.

The supported terminals are:

ADDS Viewpoint
Hazeltine 1420
Televideo TV950
ADM3A
Dec VT 52

The supported printers are:

STARWRITER
NEC 7715
Qume Sprint 5
Epson MX-80
Basic Printer

Should you not own one of these, you will have to write your own driver. Supplied are source code for the TV950 and the MX-80. You had better be prepared to borrow a supported terminal while modifying the code otherwise you will have to use ED.

Instructions for modifying the supplied code are nonexistent. The code is, however, annotated and once you have tried it several times, you will get fairly proficient at it. The author only took three days to get it right! Of course, he is neither a CP/M expert nor an assembly programmer.

I have found only a couple of problems. My terminal is a Micro-Term ACT 5A which doesn't like NULL characters. It wants RUBOUTs, and there is no provision for changing it. The alternative is to use X-ON/X-OFF which may be part of my other problem. When there is a full page of text and you are typing on the bottom of the CRT, Keyword will decide to move the word you are typing to the next line. In order to do this it must delete the top line of the text. If you type with more than four fingers, the chances are about 80% that someone will drop a character. Wordstar on OSI used to do this until Al Black of St. Louis, MO. found a fix for it. That one was a problem with CP/M and perhaps this one is too.

★

WE STILL NEED SPECIALISTS

by: Al Peabody

Ever since computers were invented, they have been attended by special personnel: clad in white smocks, these privileged initiates alone have been allowed to enter the special temples with their raised flooring and special air conditioning to minister to the magic machines.

Small business and personal computers have changed all that. Or have they? My observation of the most successful installations of business microcomputers tells me that things are not so totally different as they might seem.

Let me give an example as an illustration. Only the names have been changed to avoid any possible embarrassment.

XYZ Engineering, Inc. manufactures complex assemblies which are then made a part of large material handling systems in factories. Every installation they produce requires original design engineering, plus the assembly of a large number of standard components from a variety of steel mills, motor manufacturers, bearing plants, etc. There is no such thing as an "off-the-shelf" installation.

Pat and Mike, the principals of XYZ, wanted to computerize the process of generating cost estimates. This would require a data base manager, with a sizeable and easily updated file of component parts and assemblies, plus a mechanism for trying out various combinations to see which one would best suit the particular requirements of each job while

yielding the lowest bid. A formidable task.

On my advice, they decided to buy an Ohio Scientific computer, a C-2 two-box with two 8" floppy disk drives, terminal and printer. They also bought the OS-DMS Nucleus, to provide them with the programs and file structures they would need to produce their data files. However, they decided not to attempt to perform their particular application with a standard DMS module; rather they elected to have custom programs written, using the standard DMS file structure and Nucleus utilities such as file create, master file edit and file merge programs to make the job easier. At this point, Pat and Mike needed a programmer. Again on my advice, they decided not to hire a "professional" programmer, but rather a very intelligent and enthusiastic young woman who was to learn programming in OSI Basic, then produce the needed programs with some help from me. This was perhaps the most questionable advice I ever gave them (or anyone else).

Please understand. The candidate programmer was plenty smart, certainly capable of learning the job. Furthermore, her husband was an "R.P." (Real Programmer), and ready to help. What is questionable is my assumption that an intelligent person can learn enough Basic programming in a relatively limited period of time to produce useful business programs without costing more in learning-time salary than a professional programmer would charge in fees for the same job. Sue, the candidate programmer, spent some frustrating months going through the process I had gone through before her, learning Basic and OSI's DMS file structure. Perhaps if I had remembered how much of my learning time occurred between midnight and 5 AM, I would have been less optimistic about the time required.

Sue was restricted to regular office hours, 9 to 5 five days per week, and had to depend on my availability plus the inadequate documentation we could pull together to help her.

Fortunately, Sue is very intelligent, and a logical thinker. Fortunately, Pat and Mike are patient and kind. Fortunately, all of them shared an enthusiasm for the potential of the computer and a determination to make it work. Long after many other businesses would have written the project off, Sue was reading and re-reading what passed at the time for OSI manuals, trying various approaches to organizing the data and the calculations, and calling on me and her husband Dick the R.P. for help when she got in a real bind.

The result has been, ultimately, a success story. Sue hardly ever calls me any more. The programs mostly work, and do most of what they were intended to do (along with a lot of things which came to mind on the way). Pat and Mike still speak to me. What is the lesson of all this? It seems to me that computers, even clever little microcomputers, can do just so much with the programs supplied with them. If you want to do just Accounts Receivable or Payroll or a Mailing List Application, and most importantly if you are willing to do the application exactly as the program was designed to work, you can buy a computer and some programs and turn the job over to your secretary or payroll clerk with very little training.

But if you want to do something different, if you want your payroll run your special way which only your family business uses, if you want to do something out of the ordinary, you must be prepared to spend much more, in money or in time, either learning to modify and write programs yourself, or to have someone on your staff learn, or to hire a "R.P." of your own. If we who deal with and understand microcomputers could just make this one fact much more clear to our prospective clients, I believe we would still sell computers... they are still certainly worth their price... but we would also have many more satisfied customers.

LETTERS

ED:

Thank you Mr. Law for your response regarding the cube. I've changed back to my original monitor ROM so work on that has been halted for a while.

My machine is a C4P with an additional power supply, D&N 24K memory board and disk

WEST COAST DISTRIBUTOR

Ohio Scientific LIQUIDATION UP to 60% off

Model	Description	Retail	Cash
	Showroom Demonstrators		
C4PMF	24k w/5" floppy & color	1995	1299
C4PDMF	48k w/dual 5" floppy	2495	1499
CD-3AP	5" disk drv w/pwr sup/cabl	699	350
C8PDF	48k dual 8" floppy/color	3495	1750
C2-OEM	48k 1mhz dual 8" 65U sys	3250	1600
C2-OEM2	48k 2mhz stat mem 65U	3750	1800
C3-OEM	48k 2mhz dual 8" 3 proc	4200	2100
C3-OEM	56k 2/4mhz CP/M compatbl	4400	2200
C3-DTS	56k 10 Mbyte 5 ser ports	9300	4650
C3-D/5	52k 5 Mbyte HD 2mhz	8000	4000
C3-C12	104k 36 MB 2 user 2/1mhz	14900	8950
C3-B33	152k 74 MB 3 user 2mhz	19500	9990
	Brand New Computers		
C3-OEM	56k 2mhz dual 8" 65U 3 proc	4400	2950
C3-DTS	56k 10 M/B HD up to 4 user	9300	5750
C3-C'	52k 36 MB HD 16 slot	13500	8500
C3-D/5	52k 5 Mbyte HD 2mhz - slick	7000	5000
	Accessories and Spare Parts		
CM-2	4k stat ram at D000 L3/CPM	125	49
CM-10	8k stat mem at D000 L3/CPM	200	79
CM-6	48k 1mhz dynam mem board	550	249
CM-9	24k 2mhz stat mem board	450	179
CM-3	16k 2mhz lo pwr stat mem	399	149
CM-11	48k 2mhz stat lo pwr ram	995	499
CA-9	Centr prl ptr intfc w/cabl	235	99
CA-9D	Diablo prl intfc w/ribcabl	200	99
CA-10-1	RS-232 ser intfc w/1st port	200	99
CA-15	Modem/Telephone intfc board	500	299
FD100-8	Siemens 8" disk driv A or B	500	299
PS5-3	5 volt 3 amp power supply	69	29
PS24	24 volt 2.5 amp disk pwr sup	110	49
PS-1	5/12/-9 volt triple pwr supp	270	129
590/5	Hard disk controller pair	600	399
510c	CPU w/3 proc 2/4 mhz	600	299
470b	Floppy disk controller board	175	79

Discounts for quantity purchase/dealers/clubs/schools
Offer limited to qty on hand. Payment by cashiers ck
Demos tested and sold as is. Inspection available.

Toll Free 1-800-854-7165 Call today

SPACE-COM International

22991 La Cadena Drive, Laguna Hills, CA 92653 (714) 951-4648

controller and cable, a Shugart, SA400 5 1/4 drive, Aardvarks basic 3 ROM and the 502 CPU board. Let me tell you about some of the fun I've had lately. Mr. Stanley Windes who wrote an article about using a Shugart SA400 in a C1P was kind enough to put me in touch with Gerald Werner who also was kind enough to make me a disc with the changes necessary on Track 0 to enable 65D to work with my system. What a thrill it was to hear the drive whirring and stepping through the necessary tracks, and seeing the BEXEC program waiting for my response on the CRT! My machine would not let me get at the program though!! After many fruitless attempts to get at the program I gave up but then remembering an article in Aardvark Journal (does it still exist?)- that said the CEGMON boot doesn't work, I wrote to Rodger Olson. His response, a couple of months later was of no help other than to tell me to replace the CEGMON with the original ROM. I could have figured that out! Anyway, I did (sob) losing my backspace and other neat stuff, but now the boot worked and I could run the programs. Prior to this I had purchased 65D 3.3 from my store, because my dealer called OSI, who told him that 3.3 would work on any drive regardless of the stepping rate. This is not the case! They next said to change a location in what did boot. This changed the DOS from 2 to 1 mhz. Needless to say, it didn't work either. What did I care though. I could boot the diskette Mr. Werner had fixed, take it out and get at the programs to track 35 on 3.3. After a couple of hours I figured I had better initialize and copy a new disk. Using 3.3 copy program I could not make any new masters. After several hours of this, in desperation I used Aardvarks instructions to copy track 0. The instructions were so good that I managed to write on track 0 of the only bootable disk I had.

My computer store and OSI are no help. I have finally decided to buy an MPI B51 drive from Jade. After that, you can't imagine how delighted I was to read about the manual I am going to get from Jade in the Dec. '82 PEEK (65). I guess I pretty well agree with Mr. Pye.

Stephen P. Rydgig
Collinsville, IL 62234

* * * * *

ED:

This is about my 3rd note with DMS 9/79 enhancements, but I write them as inspirations come forth and I must have hit a hot streak.

I feel it is annoying when you do a "label search" in "EDMAFL" in order to change the contents of several records, when, after each record has been updated, you have to again go through the whole routine to find the next applicable record.

For the sake of clarification, suppose I need to change something in 3 records where the "key" content in E.G. FIELD #3 is "MEDIUM" so you ask for "L" (label search) and record #56 comes up and you make your change. When I get the prompt "is this correct" I say CR (if "Y" then go back to menu) and then, with this simple enhancement, I get the next record (E.G.#144) and so on.

The solution is very simple. I've tested it on an old data base which I use for program testing and it works AOK.

The first change I made is in line 5360 in the "label/contents search" section. This line is the prompt to "no record not found, continue search/yes record found" branch. I've merely added a condition to it (to indicate that this routine is active).

```
5360 CTR=1: IF LEFT $(U$,K1)=
  "Y"GOTO 910
```

Then already existing is the following code (also a mod found in a previous issue of PEEK(65):

(AFTER THE INPUT SECTION)

```
1282 INPUT"IS THIS CORRECT";A$
1283 IF LEFT $(A$,K1)="Y"
  THEN CLOSE 1 etc. etc.
```

--But in the use of this enhancement we'll reply with a CR--

```
SO--:
1284 IFA$="A"THENGOSUB 62000:
1400 INPUT"DO NEXT RECORD";
  A$GOTO 1400
1405 IFA$=""OR A$="Y"THEN 1290
```

Here's where the 2nd change goes. (ADDED)

```
1401 IF CTR=1 THEN GOTO 1411
1411 IF A$=""OR A$="Y"THEN
     5090
```

```
1412 GOTO 1410
     (1410=CLOSE:OPEN F$(2),
     PW$,1:FPTR=1: GOTO 700)
```

Line 5090 branches back to the search (find command). The RPTR is still at the last updated record so the search continues.

Fred Schaeffer
Jamaica, NY 11435

* * * * *

ED:

In the Dec 82 issue of PEEK(65), David Jones was wondering what all the fuss was about concerning the BREAK key on OSI polled keyboard systems. About a year ago, I was looking for the reason why my disk wouldn't operate properly when I accidentally hit the BREAK key then jumped back into DOS via $2547 to save my program.

What I found was that on the 505 CPU/disk controller board the BREAK key was wired to both the 6502 and the 6821 reset pins meaning that whenever you hit the BREAK key you reset the CPU and also the 6821, trashing whatever was in the PIA's registers at that time, thus the disk controller didn't know where the head was, etc.

Now, if you don't have the 505 board, but have a separate board with the disk controller on it such as a 610 board (for the C1P) or a 470 board or a D&N CM9 board with a disk controller you won't experience the above problem. On

these boards, the 6821 PIA is reset with a RC network so that when the power is first applied, the PIA will reset but then stay out of reset until the power is turned off then on again.

Incidentally, the 6850 ACIA is reset differently in that there is no external reset pin on the chip. Instead, to reset the ACIA, you must send it a reset control code.

One of my recent projects has been designing and building a 6809 CPU board for the OSI bus. The board is now completed, and if I can find some free time, I will write the bootstrap loader and get it into an EPROM. If anyone else has been involved in a project using the 6809, I would be glad to hear from them thru PEEK(65) of course.

Jeff Easton
Brookfield, IL 60513

* * * * *

ED:

I was surprised to see the letter that I had written to PEEK (65) several months ago, reprinted in your "LETTERS" column. I meant what was said in the letter, but after receiving a personal response from PEEK (65), I never expected to see the letter published. Good for you! As your note stated, you expect to get both positive and negative criticism. I am pleased to see that you have the courage to print both kinds.

I must take exception to one remark in your response: I never accused PEEK (65) of "complicity" in any changes at M/A-COM. PEEK (65) is in my opinion exactly what your motto says, "The Unofficial OSI Users Journal". If I made any accusations, they may have been that you printed what you had been 'told' and not what you had 'seen'.

Enough of this, let's declare a truce. If you check your subscription renewals, you will see that mine has just been renewed. I like PEEK (65) and look forward to receiving my copy each month.

If I have any complaint, it is, there is not enough information on C2/4 systems. In retrospect, I am partially at fault for this omission. Perhaps I should be contributing some of my efforts to PEEK (65). Actually, I am not alone in this regard. There must be lots of C2/4 hackers who are doing interesting things with their systems but are too busy to share with other PEEK (65) readers.

Harry B. Pye
Lansdale, PA 19446

Harry:

Thanks for your comments, and we look forward to receiving articles from you, also from all the other C2/4 hackers out there.

Al

* * * * *

ED:

In response to Guy Vanderwaeren's letter in the December issue both project one and project two are covered in my upcoming article in the February issue of MICRO (The Advanced System Journal) formerly MICRO (The 6502 Journal). I don't feel the need for DIP switches is worth the extra effort though. Also, rather than try to use the 2708 and 2716's, 2732's are the better choice. Two IC's will utilize the entire 8K available from 8000 to 9FFF and no -5v is required as with the 2708.

To use the RS232 port, you don't have to give up the cassette port and the 600 board is ready to accept an SPDT switch via J3 without complicated rewiring. Refer to the schematic and cut W12 when installing the switch.

David A. Jones
Miami, FL 33165

* * * * *

ED:

Rockwell International has introduced a new CMOS microprocessor, the R65C02. This chip is claimed to be both software and pin compatible with the 6502. The CMOS version uses very low power and is available in 2, 3 or 4 MHz. Of greatest interest to OSI users is the fact that the R65C02 implements an additional 59 op codes over the 6502. The new instructions include pushing and pulling the X and Y registers onto the stack, increment and decrementing the accumulator and a branch always. A group of 16 new instructions allow any bit in page zero to be set or reset. Another set of 16 new instructions allow branching dependent on any bit in page zero. The STZ command stores a zero in any byte.

A new addressing mode "INDIRECT" has been added to several existing instructions. The JMP command now has an INDEXED ABSOLUTE INDIRECT mode JMP (IND),X.

All of these operations can be performed on a 6502, however, may require two or three separate instructions.

I have not yet located a source of these chips in single units. I intend to install one in my OSI system soon. However, the next step will be to modify the assembler to make use of the expanded instruction set.

Earl Morris
Midland, MI 48640

* * * * *

ED:

This is to inform you and PEEK(65) that the OSI Special Interest Group Bulletin Board is now up and running on CompuServe Inc.'s MicroNet utility. In most cities, CompuServe is only a local phone call away, thus giving OSI users from across the country inexpensive access to an information service dedicated just to them. In addition to being a bulletin board, the OSI SIG has databases for software and exchange of articles, and best of all, a special subset of CompuServe's Citizen Band Simulation program that allows many people to communicate directly with each other. I hope to schedule regular meetings using the Conference system as the SIG develops. To get the OSI SIG, enter "GO PCS-125" from any exclamation point prompt in DISPLA. Thanks for your help.

Richard L. Trethewey
Pacifica, CA 94044

BUG FIX FROM OSI
Planner Plus V 4. XX

In order to suppress the printing of a $ sign on money fields, the location 41227 may be poked with the value 32 in the program PLN$.

AD$

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 OS-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 $1.50 $_____

Postage & Handling $ 3.00

TOTAL DUE $_____

POSTAGE MAY VARY FOR OVERSEAS