PEEK (65)

The Unofficial OSI Users Journal

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

$1.75

MARCH 1984
VOL.5, NO.3

INSIDE

Column One

This column has lately begun to look like the Isotron, Inc. PR department. Please be assured this will not long be the case. However, since we are all very curious as to the future, survival, plans and prospects of OSI/Isotron, for the nonce we will continue reporting what we hear. Herewith this month's batch of announcements and information:

During the month of March, two new machines will be announced as a part of the national advertising campaign which will soon start: each will be a 3-user machine, one running OS-65U level 3, the other a multiprocessor Turbodos machine. Old Peekers will remember that we very much like the idea of multiprocessing since it means that each user has his/her own CPU, sharing only the expensive stuff like hard disks and printers.

Both these machines will run DMS. There will also be other software bundled as is the practice of the industry today. However, these machines will go beyond the usual practice by bundling hardware has well as software. The very reasonable prices will include one terminal and a letter quality printer as well as the software. The prices are not finalized yet, but should be very competitive.

In case the national ad campaign isn't enough to stir up some interest, we are told Isotron will be at Comdex/Spring with a 35 ft (!) booth. That should get some attention.

So what, you ask, does all this have to do with your C1P? Simply that if the company had folded, you would be left to your own devices (and those of PEEK(65)) for support and encouragement. With the company still in business, there is at least some support and assistance from headquarters..

Now to the stuff I really like to write about, what is in this month's issue, and what it means.

If there was ever any doubt that PEEK(65) is the hackers' bible, this issue should put it to rest. There is hardly a line here that other magazines would publish, with a few notable exceptions. "Too technical," the editors would say:"Too limited in audience." Translate that to mean that if 3ØØ,ØØØ mindless video-game players don't understand it at a quick scan, it won't sell enough copies of the magazine to print it.

Here at PEEK(65) we feel differently. We can read various other 2.7 lb. computer magazines for non-technical information about computers we don't own (and do read them!), but our particular corner of the information business is to spread technical information among users/enthusiasts with OSI computers. We have fun doing it, and make our living at other things...

It has not always been so. For a long time there, we tried very hard to follow the lead of the company and become a business-computer magazine. However, you our readers told us by your response to our calls for articles and your letters that you were more interested in the technical stuff. So be it. We will publish what you want to read!

Of course, that means if you change your collective minds and decide you want more articles on business systems, we will print them. In short, we are running a service here, and will print what you want to read. This month, obviously, the interest is in hardware articles and program listings, the kind of stuff you just won't find anywhere but good ole PEEK(65).

If you disagree with this mix of articles, write to us. We are at your service, and will print what you want to read.

al.

'OLD' FOR OSI BASIC-IN-ROM

By: L. Z. Jankowski
Otaio RD1 Timaru
New Zealand

'OLD' enables the recovery of BASIC programs which have been inadvertently NEWed, or lost as a result of a crash of zero-page. 'OLD' can be placed in BASIC 4, or alternatively can be loaded and run in RAM.

Ever typed NEW and wished you hadn't? Or, POKEed into RAM and wiped zero page? No need to despair, use 'OLD'!

'OLD' is a machine language program that can be placed either in RAM, or in EPROM in a new BASIC 4. (See listing). If the former choice is taken, routines STORE and CHECK are not required.

BASIC 4 code is full of superfluous messages and contains code for running a non-existent serial port! Available RAM that could be put to better use ranges from $BE39 to $BF2C. Some of this RAM space is required for the Cold Start messages which if shortened, add to available RAM. Over 200 bytes are freed! See PEEK(65), Aug. 82 issue, for a fuller explanation.

'OLD' is simple to use. If recovery from NEW is required, enter the Monitor and run the code from the point at which the routine RUNOLD begins. If zero-page has crashed and 'OLD' is in BASIC 4 then do the following: COLD start and type OLD in response to MEMORY SIZE. Answer TERMINAL WIDTH as desired and the BASIC program is ready for LISTing.

If the code for 'OLD' has been placed in RAM, then answer MEMORY SIZE with a number: 8192 for 8K of RAM, 16384 for 16K of RAM, etc.. Then, enter the Monitor and run the code which begins with the routine RUNOLD. Jump to WARM start is automatic. If PRINT FRE(X) is now required, enter CLEAR first to speed up garbage collection.

```
 10 0000              ; 'OLD' routine for OHIO BASIC 4.
 20 0000              ; by L.Z. JANKOWSKI.
 30 0000              ;
 40 0000              ; Destruction of zero-page contents assumed.
 50 0000              ; 32K RAM system assumed. If not, change
 60 0000              ; contents of $85 & $86 to suit.
 70 0000              ; Alternatively, change lines
 80 0000              ; 150 and 160.
 90 0000              ;
100 BEA4                        * = $BEA4
110 BEA4                       ZP = $F0
120 BEA4                   WSTART = $A274
130 BEA4              ;
140 BEA4 85F0         STORE  STA ZP      Jump here from $BD0A - 'O' hit.
150 BEA6 A900                LDA #$00    Set RAM size
160 BEA8 A080                LDY #$80    to 32K,
170 BEAA 4CBABD              JMP $BDBA    and bypass RAM size check.
180 BEAD              ;
190 BEAD A5F0         CHECK  LDA ZP      Jump here from $BE36
200 BEAF C94F                CMP #$4F    'OLD' required?
210 BEB1 D035                BNE WARM    No.
220 BEB3              ;
230 BEB3 A900         RUNOLD LDA #$00    Yes, 'OLD' required.
240 BEB5 85F3                STA ZP+3
250 BEB7 A903                LDA #$03
260 BEB9 85F4                STA ZP+4
270 BEBB A005                LDY #$05
280 BEBD              ;
290 BEBD 20EBBE       LOOP   JSR ZERO    Search for first null.
300 BEC0 F002                BEQ ADD     Found it?
310 BEC2 D0F9                BNE LOOP    No.
320 BEC4              ;
330 BEC4 C8           ADD    INY         Yes. Lo-byte of pointer to
340 BEC5 8C0103              STY $0301   next line of BASIC.
350 BEC8 A5F4                LDA ZP+4    Hi-byte
360 BECA 8D0203              STA $0302   goes here.
370 BECD              ;
380 BECD 88                  DEY
390 BECE 88                  DEY         Search for end of BASIC,
400 BECF 20EBBE       LOOP2  JSR ZERO    signified by 3 nulls.
410 BED2 D0FB                BNE LOOP2
420 BED4 20EBBE              JSR ZERO
430 BED7 D0F6                BNE LOOP2
440 BED9 20EBBE              JSR ZERO
450 BEDC D0F1                BNE LOOP2
460 BEDE              ;
470 BEDE 98           FOUND  TYA         Reset pointers.
480 BEDF 18                  CLC
490 BEE0 6902                ADC #$2
500 BEE2 857B                STA $7B     End of BASIC pointer.
510 BEE4 A5F4                LDA ZP+4
520 BEE6 857C                STA $7C     Pointer hi-byte.
530 BEE8 4C74A2       WARM   JMP WSTART WARM start jump.
540 BEEB              ;
550 BEEB C8           ZERO   INY
560 BEEC C000                CPY #$0
570 BEEE D002                BNE ONE
580 BEF0 E6F4                INC ZP+4
590 BEF2 B1F3         ONE    LDA (ZP+3),Y
600 BEF4 60                  RTS
```

Placing the program in BASIC 4 requires three changes to be made to BASIC 4 code.

(1) Contents of BD83 are changed from 41 (A for Author) to 4F (O for OLD). If the 'OLD' option is taken, the code branches to BD0A. This is where the second change is made.

(2) At BD0A, change A9 4E A0 to 4C A4 BE. Address BEA4 is the entry to STORE. Finally,

(3) At BE36, change 6C 01 00 to 4C AD BE. Address BEAD is the entry to CHECK which, if completed successfully, leads to RUNOLD and complete recovery of the lost BASIC program.

The change in (1) intercepts COLD start. The second change enables the jump to STORE that sets the OLD flag on zero page. The jump from STORE to $BDBA bypasses the RAM test which would normally destroy any BASIC in RAM.

The third change is the crucial one. When BASIC arrives at $BE36, it has reset all pointers as if there was no BASIC program in RAM. The jump at $BE36, originally to WARM start, is now intercepted and forced to jump to CHECK. If CHECK discovers that the flag on zero-page has not been set then the jump to WARM start is taken immediately.

If the flag is set, then RUNOLD takes over. The first search made, is for the null marking the end of the first line of BASIC. The second search is for the three nulls marking the end of the BASIC program. When this search is successful, the appropriate pointers are calculated and placed in their addresses on zero page. The BASIC program has now been recovered and the jump to WARM start is made.

'OLD' can only recover 'undamaged' programs. If, on

Copyright ©1984 by PEEK (65) Inc. All Rights Reserved.

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

Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsements of the product or products by this magazine or the publisher.

LIST, you see a screen full of garbage, then you have been successful in not only destroying zero-page contents, but have also run amok in RAM!

INSTALLING A NON-OSI SINGLE SIDED 40 TRK DISK DRIVE ON THE C1P

By: David L. Kuhn
109 Shaw Avenue
Lewistown, PA 17044

There are two issues of the now defunct Aardvark Journal, that contain articles on installing non-OSI disk drives on your C1P. For those of you that have all the back issues, refer to the February 1982 and the October 1981 issues. I am writing this article for those who already have an OSI 610 expansion board, or the equivalent. If you do not, buy or steal the February 1982 issue of the 'Journal'. In that issue, there are plans for building a disk controller.

For the last three years, I had already been running a disk drive on my C1P. About a month ago my standard MPI drive started to 'go West' on me. I tried cleaning the heads and correcting the disk rotational speed. That helped but I still ended up with disk errors creeping up on me. I figured the drive needed to be realigned. A friend said that he could probably fix it, but until I got the drive to him, I wanted to experiment with other drives. So I picked up an almost new Siemens FDD 100-5B that was left over from a Heath H89 computer when it was upgraded with better drives. The price was right (<$100) and it is a 40 track drive that is almost OSI-MPI compatible.

The two differences, and they are not small, is that the Siemens drive doesn't have a DATA SEPARATOR and the track-track step speed is slower. The latter difference can be handled by changing the software. The first one though, I thought, was a tough one. I remembered that I read an article in one of the issues of the 'Journal' that dealt with adding a 35 track Shugart SA400 drive. My final work is a combination of those two issues and careful planning. These two problems were overcome.

My solution to the DATA SEPARATOR is really independent of what kind of drive we are dealing with. I used a circuit similar to the one shown in the 'Journal'. The two I-C's used are a 74LS221 and a 74H00. The 74LS221 is a dual monostable multivibrator. The circuit uses only 1/2 of the chip. You can use the 74LS121 if you can get it. I didn't have immediate access to one, so I used the dual chip version, which for me was easier to get. NOTE: The pin outs of the chips are different, but function similarly. I used the 'H' version of the 7400 for the same reason.

To build the DATA SEPARATOR, you can wire it together on a piece of small perfboard using wire-wrap wire or some small hookup wire (I used wire split out of a scrapped 40 wire flat computer jumper cable). Miniature push-in terminal pins that you can buy at a Radio Shack store can act as a hookup spots when you are ready to install the board. Double check all your wiring and then you are ready to hook it up!

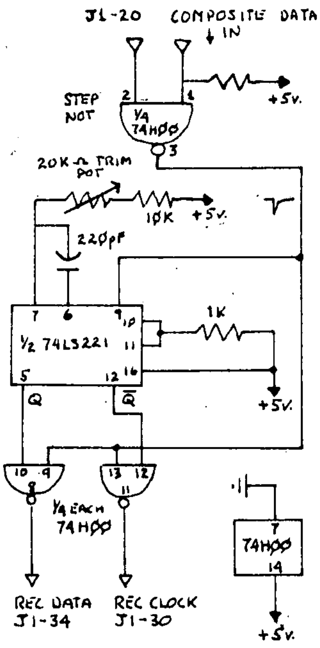
Find a place fairly close to the J1 connector on the disk drive and mount the separator. Wire the ground and +5v lines to the separator by soldering your wires to the power connector on your drive. Watch out! There is ground, +5 volts and +12 volts at this connector. Pin 4 of the power connector is usually +5v and pin 3 is usually ground, but please double check with your drive manual and drive power supply. Next, follow the copper land coming from PIN 32 of J1. On the Siemens FDD 100-5B, it doesn't go anywhere, but if it does on your drive, cut it. Follow the land coming from PIN 30 of J1 back about 1/2 inch from the connector, and cut it. Scrape a little of the paint insulation off each side of the cut land. On the connector side of the cut, solder a wire COMING from the RECEIVE CLOCK OUTPUT of the data separator to this land. On the side of the cut coming from the drive circuitry, solder a wire from this point TO the composite data input of the separator. On the land coming from PIN 20 of J1, solder a wire GOING to the STEP NOT INPUT of the data separator. Solder the wire FROM the RECEIVE DATA OUTPUT of the data separator to the land coming off of PIN 34 of J1. PIN 34 of J1 is the spare pin that OSI uses. The land from PIN 34 of J1 should not go anywhere on your drive. Hardware modification is complete!

Once the circuit was built and installed, I adjusted it by putting a write protected disk in the drive, and then tried booting the system while adjusting the pot on the DATA SEPARATOR. There is a wide range adjustment of the pot that will allow the separator to work. Center it in the middle of that range.

Ahh, but how could the Siemens drive boot a disk when it has a slow step speed? I found that for a short period it will step beyond its rated speed. I wouldn't trust it to do it all the time, but for me it did work at the faster OSI step speed long enough to boot OS65D V3.3. When it did boot the first time, OS65D3.3 went directly to the KERNEL. It shouldn't have done that! Amazingly enough, the KERNEL commands still worked!!! I tried booting it again, and this time it booted to the BEXEC*. You should then immediately EXIT BASIC to the KERNEL. Give the command: 'EM'. This will put you into the Extended Monitor. If the Extended Monitor doesn't load, keep trying until it does. On my system it worked the first time, but with some slower drives it may take a couple of tries. Immediately after entering the EM, type 'EXIT' to return to the KERNEL. Type command: 'CA 0200=06,4' if you are using OS65D V3.3 or type command: 'CA 0200=13,1' ONLY if you are using OS65D V3.1. Then type 'GO 0200'. The Track Zero/Copy Utility will appear. If you are using V3.1 a menu will be the first item displayed, select option 2. Version 3.3 of the DOS doesn't have this menu and goes directly to the Track Zero utility. When you are at the Track Zero Utility, type: R4200. The drive will hum. Exit to the KERNEL. Type: 'RE EM' (You are once again in the extended monitor). Type: '@46A3' (The '@' sign is the 'shift-P'). The monitor should then respond with: 46A3 08. Type: 20 then a <CR> (carriage return). Type: @467B <CR>. The monitor should respond with: 467B 31. It may also respond with: 467B C7. If C7 isn't in location 467B, put it there by typing: C7 and then <CR>. Type 'EXIT'. You will then again be at the KERNEL. Type: 'CA 0200=06,4' (CA 0200=13,1 for V3.1). Get to the Track Zero Utility and type: 'W4200/2200,8' to store the newly modified track zero on your disk. Do this with a back-up disk if possible and make sure it IS NOT write protected. That should give you a modified OS65D disk. Use this disk to modify your other ones by using the Track Zero Copy Utility. With this disk you could even modify a PICO-DOS disk! It seems that PICO-DOS uses the same boot track that OS65D does. Keep in mind that you still need one of the versions of OS65D to modify PICO-DOS as the simpler DOS does not have a KERNEL or Extended Monitor.

If you can't get your disk to boot due to the drive not stepping fast enough, you will either have to get together with a friend that owns a C1PMF or make sure that the drive you are buying has a fast step speed (somewhere around 6ms). The faster drives are becoming more popular and less expensive. If you have a friend that owns a C1PMF then you will not have any problems modifying your diskettes on his/her system.

SCHEMATIC OF DATA SEPARATOR



INTERFACE SIGNALS

SIEMENS MODEL EDD 100-5B DRIVE		OSI 610 BOARD EQUIVALENT DISK CONTROLLER
J1-2	SPARE	
J1-4	SPARE	
J1-6	DRIVE SELECT 0	
J1-8	INDEX/SECTOR	J3-17
J1-10	DRIVE SELECT 1	J3-3
J1-12	DRIVE SELECT 2	J3-18
J1-14	DRIVE SELECT 3	J3-
J1-16	MOTOR ON	J3-
J1-18	DIRECTION SELECT	J3-6
J1-20	STEP	J3-5
J1-22	COMPOSITE WRITE DATA	J3-9
J1-24	WRITE GATE	J3-8
J1-26	TRACK 00	J3-23
J1-28	WRITE PROTECT	J3-19
J1-30	RECEIVE CLOCK*	J3-10
J1-32	SIDE SELECT**	
J1-34	RECEIVE DATA*	J3-11

ALL ODD PINS ARE GROUND

* - AFTER MODIFICATION.

**- CUT, NOT USED!

★

EXPANDING THE C1P/SBII

PART 3

By: David Tasker
111 Bass Highway
Tasmania, Australia 73Ø3

Adding the 1st 8K continued

8K STATIC RAM BOARD PARTS LIST

INTEGRATED CIRCUIT SOCKETS
2 x 14 pin.
7 x 16 pin.
16 x 18 pin.

INTEGRATED CIRCUITS
I.C. Number- 1,2,3.. 74LS367 or 74LS365 or 8T97 or 8Ø97.
I.C. Number- 4.. 74LS138.
I.C. Number- 5.. 74155.
I.C. Number- 6.. 74ØØ.
I.C. Number- 7.. 7412.
I.C. Number- 8.9.. 8T28.
(8T26 may be used if also fitted on C1P)
(8T26 for C4P)
I.C. Number- 1Ø thru 25.. 2114 Static RAM. 45Øns or faster. RAMS increment in pairs e.g. 1Ø and 11.

CAPACITORS
Disk Ceramic preferred for physical size. Polyester may be used.

Capacitors. C1-C6,C9 all Ø.Ø47 or Ø.1uf Disc.
Capacitors C7.
3.3uf Tantalum.
Capacitor C8.
47 uf 16v. Electrolytic.

RESISTORS R1-R4.
All 1Kohm, 1/4 watt.

The printed circuit board comes coated with an anti-tarnish coating which acts as a soldering flux aid. It is not necessary to clean the board. Except that when the board is fully assembled you will have to clean the copper area which plugs into the Motherboard edge connector. To clean, hold the board in such a way that the copper is facing up and the component side of the board is supported only underneath the edge connector area, e.g., place the board at the edge of a table or workbench. A non metallic scourer pad is preferable to steel wool to clean the edge connector area as steel wool tends to scratch heavily but. more importantly, also leaves fine particles of steel which must be carefully cleaned away from both sides of the board.

Avoid touching the cleaned area with fingers as this will cause tarnishing. Once testing of the board is complete, give this area a final rub over with your cleaning pad and once plugged into the Motherboard try to avoid too many insertions and removals.

ASSEMBLY INSTRUCTIONS

Begin by inserting as many straps on the board as possible. Do not yet insert the 7 straps which run zig-zag between the RAM Integrated circuits (I.C.'s 1Ø to 25) as these are best put in after the 18 pin RAM sockets are inserted. If you work from the edge connector end of the board, left to right, you will have the board orientated to the diagrams. There are two component placement sheets but one of them does not show any sockets at all. This sheet is straps, resistors, and capacitors only.

You can use bare wire for all straps if you like but it is a good idea to alternate bare with insulated wire where there are many straps running along side one another.

Make certain the straps underneath the integrated circuits are in.

Insert the 4 Resistors.

Do not insert the capacitors until AFTER the sockets as these will forever be in the way each time you turn the board over to solder.

Once you have inserted all the sockets, check carefully that there are no solder bridges between tracks. If you have a multi-meter you could check for shorts between tracks particularly in the RAM area where tracks and soldering are very close together. You could also check continuity of the finer tracks with your meter.

BEFORE PLUGGING IN INTEGRATED CIRCUITS

If you have a multi-meter or logic probe you could insert the board onto the bus and check that +5 volts (1) and

ISOTRON, INC.

INCREASES SOFTWARE SUPPORT

ISOTRON has opened two regional OSI Software Support Centers, increasing access time to its Support Staff.

Our skilled people are anxious to answer questions regarding our Ohio Scientific software products including: OS-65U and TurboDos operating systems, KeyWord, KeySort, KeyBasic and utility programs.

FOR ASSISTANCE CALL:

From USA, Eastern and Central time zones, Europe, Africa, South America, Central America, the Caribbean and Eastern Canada:

ISOTRON SUPPORT EAST (616) 451-8435
(8:30 A.M. - 5:00 P.M., EST, Monday through Friday)

From USA, Mountain and Pacific time zones, Australia, New Zealand, Asia, Far East and Western Canada:

ISOTRON SUPPORT WEST (503) 796-1018
(9:00 A.M. - 5:00 P.M., PST, Monday through Friday)

THIS SERVICE IS ANOTHER INDICATION
THAT ISOTRON IS COMING THROUGH FOR YOU.

ISOTRON
140 SHERMAN ST.
FAIRFIELD, CT 06430
(203) 255-7443
TLX-756436

earth (Ø) are connected to the respective supply pins of the I.C. sockets.

INSERT CAPACITORS

Insert all capacitors and pay attention to the polarity markings of C7 and C8. Tantalum capacitors are usually marked with one lead positive (+), however, if not, then the longest lead is positive. Electrolytic capacitors have the negative lead marked.

ZIG - ZAG STRAPS

Insert the seven straps that run between the 18 pin sockets. You can do this just after the insertion of the sockets. Insert and solder the strap at one end, then using the sockets as corner posts, run the strap to the other hole. Do not pull the strap too tight as they may cut in under the socket and could cut into one or more of the socket pins.

TESTING

Check once more for any shorts on the board. Insert all I.C.s at this point. It is a good idea to only insert the first pair of RAM chips to start.

BOUNDARY SELECTION

Each 8K memory board can be divided up into two 4K memory blocks. These two memory blocks may be anywhere within the lower 32K of your computer's memory. If you need the RAM in the upper 32K, then refer to the circuit diagram and the note regarding address line A15.

From the strapping sheet which shows the RAM selection table, select the straps which suit your application. In most cases, the 4K blocks will follow each other. For example, if you own an Ohio C1P or 4P, then these computers have provision on their main boards for the 1st 8K of memory. If this memory board is the first one used for expansion. then both 4K blocks will follow on from the computer's memory and thus will provide the second 8K of RAM (total of 16K).

RAM TEST

Make sure all power is turned off. Insert the memory board onto the bus. If you have a memory board which has the 40 pin expansion socket. then make sure you refer to the assembly sheets for that section of the board. You will either be connecting this 40 pin socket directly into the computer or cutting the expansion off and plugging both into a motherboard.

Turn on the computer. If you hit the BREAK key, the computer should respond with the usual D.C.W.M?. If it does not then you have a fault on your new board. There are three most likely areas for trouble at this point and during the subsequent testing stages.

1. Address lines open, shorted, or buffers reversed.

2. Data lines open, shorted, or buffers reversed.

3. Control lines are faulty, e.g. RW or Ø2 wrong. DD (data direction) reversed.

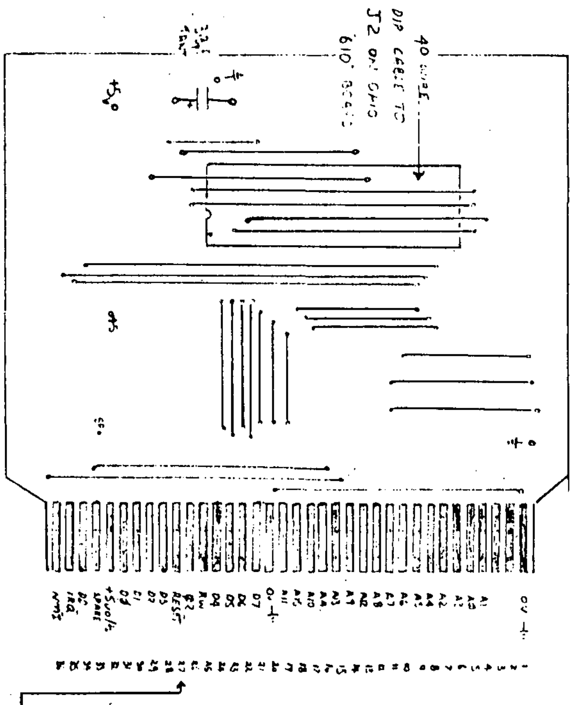
Check that the buffers are correctly inserted. With a logic probe or oscilloscope. check that the address lines and data line are changing from Ø to 1 at a fast rate. Check that DD is high (1 means write to memory).

If you have D.C.W.M.?, then proceed with a cold start in BASIC. If all is well and you have only one pair of RAMS inserted, you should have 8447 BYTES FREE. For the complete 8K RAM inserted you should get 15615 BYTES FREE.

Next month, the Motherboard.

More schematics on page 7

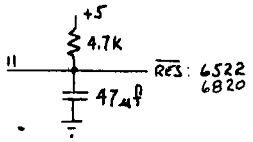
★



Connect via pin 11 J1 to BREAK key on computer (required for IO65)

Reset line is optional and is a hardware reset for input/output board.

For power on reset, connect this line as:

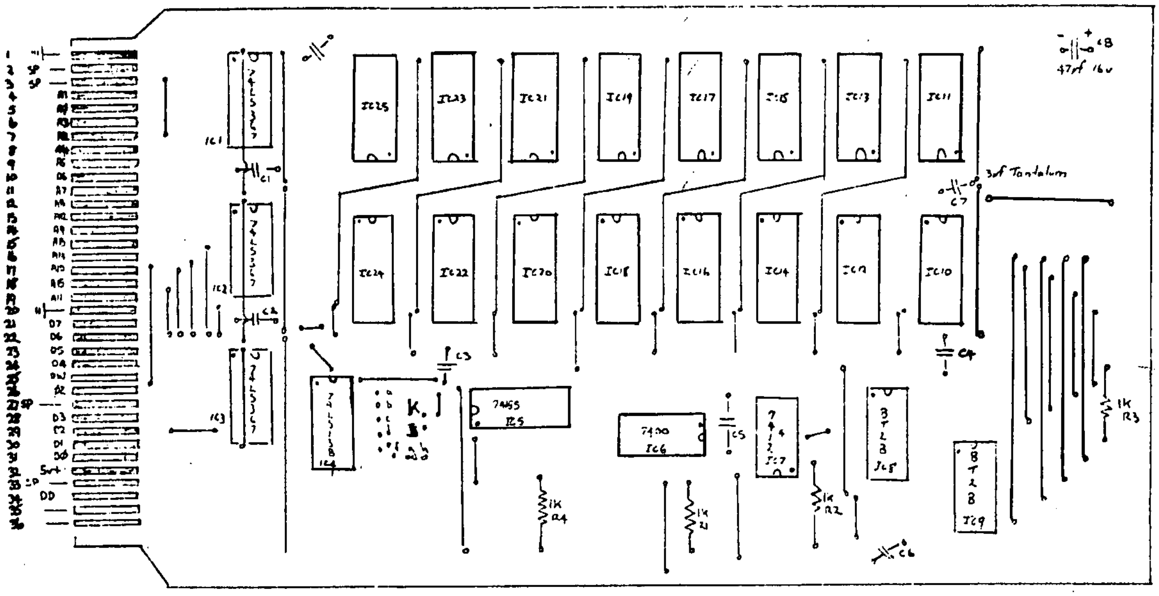


610 TO MOTHERBOARD

LAYOUT - PCB.

DT 1/81

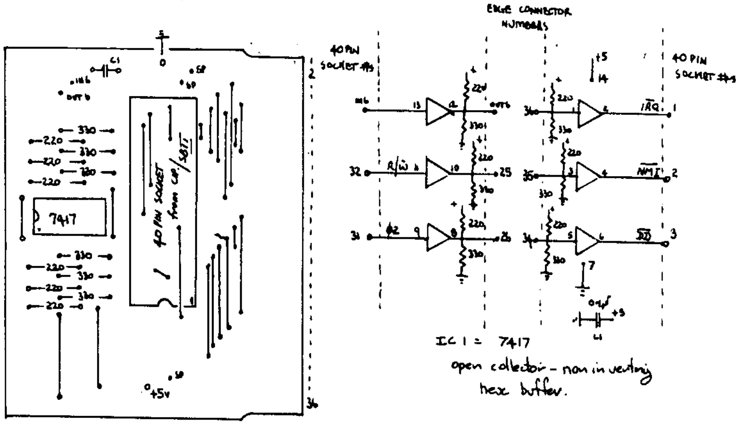
16 x 2114 STATIC RAM MEMORY INTEGRATED CIRCUITS



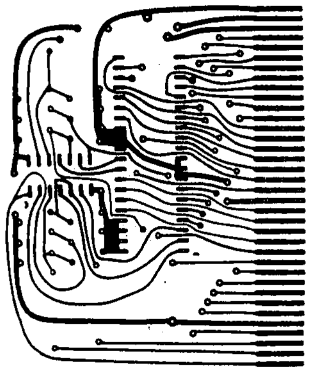
8K - 4K BOUNDARY RAM CARD, 2 mhz STATIC RAM BOARD
* 7412 for OSI Boards with 1K pull up Resistors R1,2,3
capacitors C1-C6 .047 pf OR .1pf (100uf) as required.

40 PIN - RIBBON EXPANSION - C1P/SBII - TASKER MOTHERBOARD

D. Tasker 1/81



PC PATTERN AS VIEWED FROM COMPONENT SIDE



THIS CARD IS NORMALLY PROVIDED AS AN INTACT PART OF THE 1st USED 8K - 2114 RAM CARD, i.e., it is electrically & physically joined at the 36 way edge connector. This enables an 8K MEM card to be used on the C1P/SBII via the 40 pin expansion socket without the need for a motherboard.

WHEN additional boards are required, i.e., a motherboard to be added, then a cut is made through the edge connector to separate the 40 pin socket section as shown.

NOTE: When used on a motherboard - all component sides of boards face towards the address buffer ICs on the M/Bd. This card plugs into Slot "A" only.

★ ★ ★

XREF: BASIC FILE CROSS REFERENCE GENERATOR
PART TWO

by:Rick Trethewey
8 Duran Court
Pacifica, CA 94Ø44

```
10; XREF: BASIC FILE CROSS REFERENCE GENERATOR
20; PART 2
30;
40 READ   LDA #$00         LOAD LSB
50        STA ADRLX        GIVE TO 65D
60        LDA #BUFFER/256  LOAD BUFFER MSB
70        STA ADRHX        GIVE TO 65D
80        LDA #$01         INIZ
90        STA SECT         SET 65D TO SECTOR #1
100       JSR LOAD         LOAD HEAD
110       JSR CALLX        READ SECTOR
120       JMP UNLOAD       UNLOAD HEAD AND QUIT
130;
140 XRQ   LDA STTK         GET FILE START TRACK
150       STA TRAKX        GIVE TO 65D
160       JSR SWAP         * DOS CONTEXT *
170       JSR SEEKX        MOVE HEAD TO TRACK
180       JSR READ         READ IN 1ST TRACK OF FILE
190       JSR SWAP         * LANGUAGE CONTEXT *
200       LDA BUFFER       FETCH FILE START LSB
210       SEC
220       SBC #SRCSTR      SUBTRACT SOURCE START
230       STA SOF          SAVE HEADER OFFSET LSB
240       LDA BUFFER+1     HANDLE MSB TOO
250       SBC #SRCSTR/256-9
260       STA SOF+1        NOTE 3.3 FIX ABOVE
270       BEQ XRQ3         OFFSETMSB = 0 ?? ==>
280       SEC
290 XRQ1  LDA SOF+1        FETCH OFFSET MSB
300       SBC SRCSIZ       SUBTRACT ONE TRACK
310       BCC XRQ2         LESS THAN 1 TRACK ? ==>
320       STA SOF+1        NO! SAVE RESULT!
330       INC TRAKX        SHOW MOVE TO NEXT TRACK
340       BNE XRQ1         AND LOOP!
350 XRQ2  JSR SWAP         * DOS CONTEXT *
360       JSR SEEKX        SEEK PROPER TRACK
370       JSR READ         READ SECTOR INTO BUFFER
380       JSR SWAP         * LANGUAGE CONTEXT *
390       LDA TRAKX        GET RESULTANT TRACK #
400       STA STTK         SAVE AS NEW STTK
410 XRQ3  LDA SOF          GET OFFSET LSB
420       CLC
430       ADC #BUFFER      ADD BUFFER ADDRESS LSB
440       STA INDEX        GIVE TO INDEX
450       STA SOF          SAVE FOR PASS 2
460       LDA SOF+1        GET OFFSET MSB
470       ADC #BUFFER/256  ADD BUFFER ADDRESS MSB
480       STA TXTPTR+1     GIVE TO TXTPTR
490       STA SOF+1        SAVE FOR PASS 2
500       LDA PUTPTR       SAVE LAST FREE ENTRY AS
510       STA REFBOT       REFERENCE TABLE BOTTOM
520       LDA PUTPTR+1     HANDLE MSB TOO
530       STA REFBOT+1
540       LDA #$00         INIZ
550       STA TXTPTR       CLEAR TXTPTR
560       JMP XREF         JUMP TO HANDLING CODE
570;
580 GETCHR LDY INDEX       FETCH BUFFER PAGE INDEX
590       LDA (TXTPTR),Y   FETCH CHARACTER
600       INY              BUMP INDEX
610       BEQ GETC1        PAGE? ==> GETC1
620       STY INDEX        NO! SAVE INDEX
630       RTS              AND QUIT
640 GETC1 INC TXTPTR+1     BUMP TXTPTR MSB ON PAGING
650       STY INDEX        RESET BUFFER INDEX
660       LDY TXTPTR+1     FETCH NEW MSB
670       CPY BFENPG       AT END OF BUFFER?
680       BEQ GETC2        YES! ==> GETC2
690       RTS              AND QUIT
700 GETC2 PHA              SAVE FETCHED CHARACTER
710       TXA              PUT X IN ACC.
720       PHA              SAVE IT TOO
730       LDA #BUFFER      FETCH BUFF. ADDR. LSB
740       STA TXTPTR       RESET TXTPTR
750       LDA #BUFFER/256  FETCH BUFF. ADDR. MSB
760       STA TXTPTR+1     RESET TXTPTR MSB
770       JSR SWAP         * DOS CONTEXT *
780       LDA TRAKX        FETCH CURRENT TRACK #
790       CMP ENDTK        E.O.F. ?
800       BEQ GETERR       YES! ERROR! ==>
810       INC TRAKX        NO, BUMP IT ONE
820       JSR SEEKX        MOVE HEAD TO TRACK
830       JSR READ         READ IN TRACK
840       JSR SWAP         * LANGUAGE CONTEXT *
850       PLA              RETRIEVE X
860       TAX              PUT IT BACK
870       PLA              RETRIEVE TEXT CHARACTER
880       RTS              AND QUIT
890;
900 GETERR JSR STROUT      SHOW ERROR TO USER!
```

Continued

From Gander Software *A New Standard of Excellence*

FINANCIAL PLANNER

Get "What If" answers for up to 10 displayed problems in:

- Loan/Annuity Analysis
- Annuity 'Due' Analysis
- Present/Future Value Analysis
- Sinking Fund Analysis
- Amortization Schedules
- Interest Conversions

HARDWARE REQUIREMENTS: 48K OSI, 8" floppy, serial terminal system, OS-65U v. 1.2 or later.

FEATURES: package allows configuration to almost all non-ANSI terminals, AND user specification of printer port.

PRICE: $300.00 (User Manual, $25.00, credited toward Planner purchase). Michigan residents add 4% sales tax.

DEALERS: This program, of great value to lawyers, bankers, insurance people, and real estate people, will help you sell hardware! Inquiries invited.

A POWERFUL TOOL FOR EVALUATING ALTERNATIVES!

The first four programs all: allow you to solve a named variable after changing another variable, let you net the difference between any displayed problems, provide selective saves to disk, give you very informative printouts based on the problems solved, and much, much more.

The "Amortization Schedules" program provides more flexibility than any other schedule known to GANDER. It lets you deal with balloon payments, early pay-offs, annual payment increases (by percentages or dollars), keeps a running total of your entire transaction to pay off, schedules payments by both month and year, and reports YTD totals based on user selected calendar OR fiscal years.

"Interest Conversions" lets you key in any nominal rate and reports the true effective rate for compounding semi-annually, quarterly, monthly, daily, and continuously, and allows the print out of interest tables (your choice of rate and increments). It also includes a simple calculator, which can be used without disturbing other problems displayed, and which contains three separate user addressable memories.

Finally, to aid planning, the Menu program will generate a calendar for any month/year between 1901 and 2399, and accurately accounts for leap years!

GANDER SOFTWARE

3223 Bross Road
"The Ponds"
Hastings, MI 49058
(616) 945-2821

"It Flies"

```
 910        .BYTE CR,LF
 920        .BYTE 'END OF FILE ERROR',CR,LF,0
 930        LDA DEFAUL+1    GET CONSOLE DV #
 940        STA OUFLAG      MAKE IT CURRENT
 950        JMP WARM        RETURN TO MENU
 960;
 970 GETLIN JSR GETCHR      GET CHAR. FROM FILE
 980        STA NLAL        SAVE AS NEXT LINE LSB
 990        JSR GETCHR      GET CHAR. FROM FILE
1000        STA NLAH        SAVE AS NEXT LINE MSB
1010        CMP #$00        WAS IT $00?
1020        BNE GETLI1      NO, CONTINUE ==> GETLI1
1030        LDA NLAL        YES! CHECK LSB
1040        BEQ GETLI3      $00 ALSO ? ==> E.O.F.!
1050 GETLI1 JSR GETCHR      GET CHARACTER FROM FILE
1060        STA LNLO        SAVE AS LINE # LSB
1070        JSR GETCHR      GET CHARACTER FROM FILE
1080        STA LNHI        SAVE AS LINE # MSB
1090        LDX #$00        INIZ TXBUFF INDEX
1100 GETLI2 JSR GETCHR      GET CHARACTER FROM FILE
1110        CMP #SP         WAS IT A <SPACE> ?
1120        BEQ GETLI2      YES! IGNORE! ==> GETLI2
1130        STA TXBUFF,X    NO, SAVE IN TXBUFF
1140        INX             BUMP TXBUFF INDEX
1150        TAY             CHECK LAST CHARACTER
1160        BNE GETLI2      NOT $00? ==> GETLI2
1170        STA ONFLAG      YES! CLEAR "ON" FLAG
1180 GETLI3 RTS             AND QUIT
1190;
1200 XREF   JSR GETLIN      GET LINE OF TEXT
1210        LDA NLAH        CHECK FOR E.O.F.
1220        BNE XREF1       NO, CONTINUE ==>
1230        LDA NLAL        MAYBE, CHECK LSB
1240        BNE XREF1       NO ==>
1250        JMP DISLIN      YES! GO TO DISPLAY
1260 XREF1  LDY #$00        INIZ
1270 XREF2  LDA TXBUFF,Y    LOOK AT CHARACTER
1280        BEQ XREF        E.O.L.? ==> LOOP!
1290        CMP #GOTOTK     "GOTO"?
1300        BEQ INSERT      YES! INSERT LINE #
1310        CMP #GOSBTK     "GOSUB"?
1320        BEQ INSERT      YES! ==>
1330        CMP #THENTK     "THEN"?
1340        BEQ INSERT      YES! ==>
1350        CMP #ONTK       "ON"?
1360        BNE XREF3       NO! ==>
1370        INC ONFLAG      YES! SHOW "ON" IN PROGRESS
1380 XREF3  INY             BUMP POINTER INDEX
1390        BNE XREF2       AND LOOP!
1400;
1410 INSERT INY             BUMP POINTER 1
1420        LDX #$00        INIZ
1430        LDA TXBUFF,Y    YES! LOOK AT NEXT CHARACTER
1440        CMP #'0         CHECK FOR LINE #
1450        BCC XREF2       NO! => RETURN
1460        CMP #'9+1
1470        BCS XREF2
1480 INS1   LDA TXBUFF,Y    FETCH CHARACTER
1490        STA INBUF,X     TRANSFER TO INBUF
1500        CMP #'0         LOOK FOR END OF NUMBER
1510        BCC INS2
1520        CMP #'9+1
1530        BCS INS2
1540        INY             BUMP TXBUFF INDEX
1550        INX             BUMP INBUF INDEX
1560        BNE INS1        LOOP 'TIL DONE
1570 INS2   STY TMPPTR      SAVE CURRENT TXBUFF INDEX
1580        JSR DECHEX      CONVERT LINE # TO BINARY
1590        JSR FNDLIN      INSERT LINE # REFERENCE
1600        LDY TMPPTR      RETRIEVE LINE # INDEX
1610        LDA ONFLAG      CHECK FOR "ON"
1620        BNE INS3        YES! ==> INS3
1630        JMP XREF2       NO! RETURN TO MAIN LOOP!
1640 INS3   LDA TXBUFF,Y    FETCH LAST CHARACTER SEEN
1650        CMP #',         COMMA ?
1660        BNE INS3-3      NO! GO BACK!
1670        LDX #$00        YES! INIZ XFER INDEX
1680        INY             BUMP FETCH POINTER
1690        BNE INS1        AND LOOP!
1700;
1710 FNDLIN LDA #SRCSTR     INIZ
1720        STA PUTPTR      SET TO TABLE TOP
1730        LDA #SRCSTR/256
1740        STA PUTPTR+1
1750 FNDLI1 LDY #$00        INIZ
1760        STY TFLAG       CLEAR TEST FLAG
1770        LDA (PUTPTR),Y  LOOK AT LINE # LSB
1780        CMP RESLO       SAME AS REFERENCE?
1790        BNE FNDLI0      NO! => FNDLI0
1800        INY             YES! CHECK MSB TOO!
1810        LDA (PUTPTR),Y  FETCH MSB
1820        CMP RESHI       COMPARE TO REFERENCE
1830        BNE FNDLI0      NO MATCH ==>
1840        INC TFLAG       SHOW MATCH!
1850 FNDLI0 LDY #$02        POINT TO # OF ENTRIES
1860        LDA (PUTPTR),Y  FETCH # OF REFERENCES
1870        STA NR          SAVE IT
1880        LDA PUTPTR      GET CURRENT POINTER
1890        STA OLDPTR      SAVE IT
1900        LDA PUTPTR+1
1910        STA OLDPTR+1
1920        INY             BUMP POINTER
1930        TYA             PUT IT IN ACC.
1940        CLC
1950        ADC PUTPTR      ADD TO POINTER ADDRESS
1960        STA PUTPTR      SAVE IT
1970        BCC FNDLI2      WATCH FOR PAGING
1980        INC PUTPTR+1    BUMP MSB ON PAGING
1990 FNDLI2 LDA NR          FETCH # OF REFERENCES
2000        ASL A           * 2
2010        STA T2          SAVE IT
2020        LDA #$00
2030        ROL A
2040        STA T2+1        HANDLE MSB
2050        LDA T2          FETCH LSB
2060        CLC             MUST CLEAR CARRY 1ST!
2070        ADC PUTPTR      ADD POINTER LSB
2080        STA PUTPTR      SAVE IT
2090        LDA PUTPTR+1    FETCH MSB
2100        ADC T2+1        ADD ANY CARRY
2110        STA PUTPTR+1    SAVE MSB
2120        LDA TFLAG       CHECK MATCH FLAG
2130        BEQ FNDLI6      NO MATCH? ==>
2140        JSR MOVEUP      MAKE ROOM FOR ENTRY
2150 FNDLI5 LDY #$00        INIZ
2160        LDA LNLO        GET REF. LINE # LSB
2170        STA (PUTPTR),Y  PUT IN TABLE
2180        INY             BUMP INDEX
2190        LDA LNHI        GET MSB
2200        STA (PUTPTR),Y  SAVE IT TOO
2210        LDY #$02        INIZ
2220        LDA (OLDPTR),Y  GET OLD NR
2230        CLC
2240        ADC #$01        +1 !
2250        STA (OLDPTR),Y  SAVE UPDATED FIGURE
2260        RTS             AND QUIT
2270;
2280 FNDLI6 LDA REFBOT      GET TABLE BOTTOM ADDRESS
2290        CMP PUTPTR      SAME AS PUTPTR?
2300        BNE FNDLI7      NO! LOOP! ==> FNDLI7
2310        LDA REFBOT+1    MAYBE, CHECK MSB
2320        CMP PUTPTR+1    SAME?
2330        BEQ FNDERR      YES! ERROR! ==>
2340 FNDLI7 JMP FNDLI1      CONTINUE! ==>
2350 FNDERR JSR STROUT
2360        .BYTE CR,LF
2370        .BYTE '?US ERROR #',$00
2380        JSR NUMOUT
2390        JSR STROUT
2400        .BYTE ' IN LINE ',$00
2410        LDA LNLO
2420        STA RESLO
2430        LDA LNHI
2440        STA RESHI
2450        JSR NUMOUT
2460        JMP CRLF
2470;
2480 DISLIN LDA #SRCSTR     RESET TO TABLE TOP
2490        STA PUTPTR
2500        LDA #SRCSTR/256
2510        STA PUTPTR+1
2520        JSR STROUT
2530        .BYTE 'LINE NUMBERS',CR,LF,LF,$00
2540 DISLI2 LDY #$00
2550        STY COUNT       CLEAR # LINES
2560        LDA (PUTPTR),Y  GET LINE # LSB
2570        STA RESLO       SAVE IT
2580        INY
2590        LDA (PUTPTR),Y  GET MSB
2600        STA RESHI       SAVE IT
2610        INY
2620        LDA (PUTPTR),Y  GET # OF REFERENCES
2630        STA NR          SAVE IT
2640        LDA PUTPTR      GET CURRENT POINTER
2650        CLC
2660        ADC #$03        ADD HEADER OFFSET
2670        STA PUTPTR      SAVE IT BACK OUT
2680        LDA PUTPTR+1    GET MSB
2690        ADC #$00        ADD ANY CARRY
2700        STA PUTPTR+1    SAVE IT BACK OUT
2710        LDA NR          GET # OF REFS.
2720        BEQ DISLI6      0 ? ==> DISLI6
2730        JSR STROUT      DISPLAY TEXT
2740        .BYTE 'Line # ',$00
2750        JSR NUMOUT      DISPLAY LINE #
2760        JSR CRLF        DO <CR><LF> PAIR
2770        JSR CRLF        DO ANOTHER
2780 DISLI3 LDY #$00        INIZ
2790        LDA (PUTPTR),Y  FETCH REF. LINE # LSB
2800        STA RESLO       SAVE IT
2810        INY             BUMP INDEX
2820        LDA (PUTPTR),Y  FETCH REF. LINE # MSB
2830        STA RESHI       SAVE IT
2840        JSR NUMOUT      DISPLAY REF. LINE #
2850        LDA PUTPTR      GET CURRENT POINTER
2860        CLC
2870        ADC #$02        ADD REFERENCE LENGTH
2880        STA PUTPTR      UPDATE POINTER
2890        LDA PUTPTR+1    FETCH MSB
2900        ADC #$00        ADD ANY CARRY
2910        STA PUTPTR+1    SAVE IT TOO
2920        INC COUNT       BUMP OUTPUT COUNT
2930        LDA COUNT       FETCH # PRINTED ON LINE
2940        CMP #$0A        DONE 10 ?
2950        BCC DISLI4      NO ==> DISLI4
2960        JSR CRLF        YES! DO A <CR><LF>
2970        LDA #$00        INIZ
2980        STA COUNT       CLEAR COUNTER
2990        BEQ DISLI5      AND SKIP A BIT
3000 DISLI4 LDA #SP         GET A SPACE
3010        JSR OUTCH       PRINT IT
3020        JSR OUTCH       TWICE, IN FACT
3030 DISLI5 DEC NR          SHOW WE DID A REFERENCE
```

Listing continued

```
3040        BNE DISLI3      LOOP 'TIL DONE
3050        JSR CRLF        DO FINAL <CR><LF>
3060        JSR CRLF        DO TWO
3070 DISLI6 LDA REFBOT      GET BOTTOM ADDR.
3080        CMP PUTPTR      SAME AS PUTPTR?
3090        BNE DISLI7      NO! CONTINUE! ==>
3100        LDA REFBOT+1    MAYBE, CHECK MSB
3110        CMP PUTPTR+1    SAME ?
3120        BEQ DISLI8      YES! GO TO VARIABLES
3130 DISLI7 JMP DISLI2      NOT DONE! ==> DISLI2
3140 DISLI8 JMP VAR         GO DO VARIABLES NOW
3150;
3160 MOVEUP LDA REFBOT      GET TABLE BOTTOM ADDR.
3170        STA FETPTR      GIVE TO FETPTR
3180        LDA REFBOT+1
3190        STA FETPTR+1
3200 MOVE1  LDY #$00        INIZ
3210        LDA (FETPTR),Y  FETCH A CHARACTER
3220        LDY #$02        + 2
3230        STA (FETPTR),Y  MOVE IT UP TWO BYTES
3240        LDA FETPTR      GET FETPTR
3250        CMP PUTPTR      AT INSERTION POINT ?
3260        BNE MOVE2       NO ==>
3270        LDA FETPTR+1    MAYBE, GET MSB
3280        CMP PUTPTR+1    SAME ?
3290        BEQ MOVE4       YES! ==> MOVE4
3300 MOVE2  LDY FETPTR      NO, FETCH LSB IN Y
3310        BNE MOVE3       NOT $00? ==> MOVE3
3320        DEC FETPTR+1    YES! DECREMENT MSB
3330 MOVE3  DEY             DECREMENT LSB
3340        STY FETPTR      SAVE IT BACK OUT
3350        JMP MOVE1       AND CONTINUE
3360 MOVE4  LDA REFBOT      GET BOTTOM ADDR. LSB
3370        CLC
3380        ADC #$02        ADD 2
3390        STA REFBOT      SAVE IT
3400        BCC MOVE5       WATCH FOR PAGING
3410        INC REFBOT+1    BUMP MSB ON PAGING
3420        LDA REFBOT+1    FETCH RESULT
3430        CMP MAXMEM      AT MEMORY TOP?
3440        BEQ MOVE5       =, OK ==> MOVE5
3450        BCS OMERR       PAST! ==> ERROR
3460 MOVE5  RTS             AND QUIT
3470;
3480 OMERR  JSR STROUT      TELL USER
3490        .BYTE CR,LF
3500        .BYTE 'OUT OF MEMORY',CR,LF,$00
3510        LDA DEFAUL+1    GET CONSOLE DV#
3520        STA OUFLAG      MAKE IT CURRENT
3530        JMP WARMNS      RETURN TO MENU
3540;
3550 VAR    LDA STTK        LOAD 1ST TRACK OF TEXT
3560        CMP TRAKX       COMPARE TO CURRENT TRACK
3570        BEQ VAR1        YES! NO RE-READ! ==>
3580        STA TRAKX       GIVE IT TO 65D
3590        JSR SWAP        * DOS CONTEXT *
3600        JSR SEEKX       MOVE HEAD TO TRACK
3610        JSR READ        READ TRACK
3620        JSR SWAP        * LANGUAGE CONTEXT *
3630 VAR1   LDA SOF         GET START OF TEXT
3640        STA INDEX       GIVE TO INDEX
3650        LDA SOF+1
3660        STA TXTPTR+1
3670        LDA #SRCSTR
3680        STA REFBOT      RESET BOTTOM OF
3690        LDA #SRCSTR/256 REFERENCE TABLE
3700        STA REFBOT+1
3710        LDA #$00
3720        STA TXTPTR      RESET TXTPTR TO BUFFER
3730        STA TYPE        INIZ INITIAL TYPE
3740        JSR STROUT
3750        .BYTE CR,LF,'VARIABLES',CR,LF,LF,$00
3760;
3770 XVAR   JSR GETLIN      GET A LINE OF TEXT
3780        LDY #$00        INIZ
3790        LDA NLAH        CHECK FOR E.O.F.
3800        BNE XVAR1
3810        LDA NLAL
3820        BNE XVAR1
3830        JMP DISVAR      DONE! PRINT TABLE
3840 XVAR1  LDA TXBUFF,Y    LOOK AT CHARACTER
3850        BEQ XVAR        E.O.L.?? ==> XVAR
3860        CMP #REMTK      REMARK?
3870        BEQ XVAR        YES! SKIP TO NEXT LINE
3880        CMP #'"         STRING LITERAL?
3890        BEQ XVAR3       YES! ==> XVAR3
3900        JSR CASECK      CORRECT ANY LOWER CASE
3910        CMP #'A         CHECK FOR LETTER
3920        BCC XVAR2       NO ==>
3930        CMP #'Z+1
3940        BCC GETVAR      YES! ==>
3950        CMP #FNTK       FUNCTION ?
3960        BNE XVAR2       NO ==>
3970        LDA #$10        YES! SHOW FUNCTION
3980        STA TYPE        SAVE IT
3990 XVAR2  INY             BUMP TXBUFF INDEX
4000        BNE XVAR1       AND LOOP!
4010 XVAR3  INY             BUMP TXBUFF INDEX
4020        LDA TXBUFF,Y    FETCH NEXT CHARACTER
4030        BEQ XVAR        E.O.L.? ==> XVAR
4040        CMP #'"         FIND TRAILING QUOTE?
4050        BNE XVAR3       NO! LOOP! ==> XVAR3
4060        BEQ XVAR2       YES! BUMP & LOOP ==>
4070;
4080 GETVAR STA VARNAM      SAVE AS 1ST CHARACTER
4090        LDA #SP         GET A SPACE
4100        STA VARNAM+1    CLEAR 2ND CHARACTER
4110 GETVA1 INY             BUMP INDEX
```

```
4120        LDA TXBUFF,Y    GET NEXT CHARACTER
4130        BEQ GETVA2      E.O.L.? ==> GETVA2
4140        CMP #'%         INTERGER ?
4150        BEQ SETINT      YES! ==>
4160        CMP #'$         STRING ?
4170        BEQ SETSTR      YES! ==>
4180        CMP #'(         ARRAY ?
4190        BEQ SETARR      YES! ==>
4200        CMP #$80        TOKEN ?
4210        BCS GETVA2      YES! ==>
4220        CMP #'0         CHECK FOR NUMERIC
4230        BCC GETVA2      NO ==>
4240        CMP #'9+1       CHECK AGAIN
4250        BCC GETVA0      YES! ==>
4260        JSR CASECK      CORRECT FOR LOWER CASE
4270        CMP #'A         WATCH FOR END OF NAME
4280        BCC GETVA2
4290        CMP #'Z+1
4300        BCS GETVA2      END! ==>
4310 GETVA0 LDX VARNAM+1    CHECK 2ND CHAR.
4320        CPX #SP         CLEAR?
4330        BNE GETVA1      NO! LOOP!
4340        STA VARNAM+1    YES! SAVE 2ND CHAR.
4350        JMP GETVA1      AND LOOP!
4360 SETINT LDA #$01        SHOW INTERGER
4370        .BYTE SKIP2
4380 SETSTR LDA #$02        SHOW STRING
4390        ORA TYPE        COMBINE WITH CURRENT
4400        STA TYPE        SET TYPE
4410        BNE GETVA1      AND LOOP!
4420;
4430 SETARR LDA TYPE        GET CURRENT TYPE
4440        ORA #$80        ADD ARRAY TYPING
4450        STA TYPE        SAVE AND FALL THROUGH
4460 GETVA2 STY TMPPTR      SAVE TXBUFF INDEX
4470        LDA #SRCSTR     INIZ LOOK-UP POINTER
4480        STA PUTPTR      RESET TO TOP OF TABLE
4490        LDA #SRCSTR/256
4500        STA PUTPTR+1
4510;
4520 GETVA3 LDA PUTPTR      CHECK FOR END 1ST
4530        CMP REFBOT
4540        BNE GETVA4      NO ==>
4550        LDA PUTPTR+1
4560        CMP REFBOT+1
4570        BNE GETVA4      NO ==>
4580        JMP NEWVAR      YES! NEW! ==>
4590 GETVA4 LDY #$00        INIZ
4600        STY TFLAG       CLEAR MATCH FLAG
4610        LDA (PUTPTR),Y  FETCH 1ST CHARACTER
4620        CMP VARNAM      SAME AS CURRENT?
4630        BNE GETVA5      NO! ==>
4640        INY             YES! CHECK 2ND
4650        LDA (PUTPTR),Y  FETCH 2ND
4660        CMP VARNAM+1    COMPARE IT
4670        BNE GETVA5      NO ==>
4680        INY             YES! BUMP INDEX
4690        LDA (PUTPTR),Y  ALSO CHECK TYPE
4700        CMP TYPE
4710        BNE GETVA5      NO! ==>
4720        INC TFLAG       YES! SET MATCH FLAG!
4730 GETVA5 LDY #$03        INIZ
4740        LDA (PUTPTR),Y  FETCH # OF REFS.
4750        STA NR          SAVE IT
4760        ASL A           * 2
4770        STA T1          SAVE IT
4780        LDA #$00        CLEAR
4790        ROL A           ROTATE ANY CARRY IN
4800        STA T1+1        SAVE AS MSB
4810        LDA PUTPTR      FETCH PUTPTR
4820        STA OLDPTR      SAVE FOR LATER
4830        CLC
4840        ADC #$04        ADD HEADER OFFSET
4850        STA PUTPTR      SAVE IT
4860        LDA PUTPTR+1    FETCH MSB
4870        STA OLDPTR+1    SAVE IT FOR LATER TOO
4880        ADC #$00        ADD ANY CARRY
4890        STA PUTPTR+1    SAVE MSB
4900        LDA PUTPTR      REFETCH NEW LSB
4910        CLC
4920        ADC T1          ADD NR * 2
4930        STA PUTPTR      SAVE IT
4940        LDA PUTPTR+1    FETCH MSB
4950        ADC T1+1        ADD MSB OF NR * 2
4960        STA PUTPTR+1    SAVE IT
4970        LDA TFLAG       FETCH MATCH FLAG
4980        BEQ GETVA3      CLEAR TRY NEXT! ==>
4990        LDA NR          CHECK NR AGAIN
5000        BEQ GETVA8      0! INSERT!
5010        STA T1          SAVE IT
5020        LDA OLDPTR      GET OLDPTR
5030        CLC
5040        ADC #$04        ADD HEADER OFFSET
5050        STA FETPTR      SAVE RESULT
5060        LDA OLDPTR+1
5070        ADC #$00
5080        STA FETPTR+1
5090 GETVA6 LDY #$00        INIZ
5100        LDA (FETPTR),Y  FETCH REFERNCE
5110        CMP LNLO        TO CURRENT LINE#
5120        BNE GETVA7      NO! ==>
5130        INY             YES! BUMP INDEX
5140        LDA (FETPTR),Y  CHECK MSB
5150        CMP LNHI
5160        BNE GETVA7
5170        JMP GETVA9      MATCH! SKIP ENTRY!
```

Listing continued

```
5180 GETVA7 LDA FETPTR      GET POINTER
5190        CLC
5200        ADC #$02        ADD LINE REF. OFFSET
5210        STA FETPTR
5220        LDA FETPTR+1
5230        ADC #$00
5240        STA FETPTR+1
5250        DEC T1          DECREMENT CHECK COUNTER
5260        BNE GETVA6      LOOP 'TIL DONE
5270 GETVA8 JSR MOVEUP      CREATE SPACE IN TABLE
5280        LDY #$00
5290        LDA LNLO
5300        STA (PUTPTR),Y
5310        INY
5320        LDA LNHI
5330        STA (PUTPTR),Y
5340        LDY #$03
5350        LDA (OLDPTR),Y  FETCH OLD NR
5360        CLC
5370        ADC #$01        ADD 1
5380        STA (OLDPTR),Y  SAVE IT
5390 GETVA9 LDY TMPPTR      RETRIEVE TXBUFF INDEX
5400        LDA TXBUFF,Y    REFETCH LAST CHARACTER
5410        BEQ GETVAA      E.O.L.?
5420        INY             NO! BUMP INDEX
5430        LDA #$00        INIZ
5440        STA TYPE        CLEAR VARIABLE TYPE
5450        JMP XVAR1       AND CONTINUE
5460 GETVAA STA TYPE        CLEAR VARIABLE TYPE
5470        JMP XVAR        AND GET NEXT LINE
5480;
5490 NEWVAR LDY #$00        INIZ
5500        LDA VARNAM      GET VARNAM 1ST CHAR.
5510        STA (REFBOT),Y  SAVE AT END OF TABLE
5520        INY             BUMP INDEX
5530        LDA VARNAM+1    GET 2ND CHARACTER
5540        STA (REFBOT),Y  SAVE IT
5550        INY             BUMP INDEX
5560        LDA TYPE        GET VARIABLE TYPE
5570        STA (REFBOT),Y  SAVE IT
5580        INY             BUMP INDEX
5590        LDA #$01        INIZ
5600        STA (REFBOT),Y  SHOW 1 ENTRY
5610        INY             BUMP INDEX
5620        LDA LNLO        GET LINE # LSB
5630        STA (REFBOT),Y
5640        INY             BUMP INDEX
5650        LDA LNHI        GET LINE # MSB
5660        STA (REFBOT),Y  SAVE IT
5670        LDA #$06
5680        CLC
5690        ADC REFBOT      ADD TO REFERENCE END
5700        STA REFBOT      SAVE AS NEW BOTTOM.
5710        LDA REFBOT+1    HANDLE MSB
5720        ADC #$00
```

```
5730        STA REFBOT+1
5740        JMP GETVA9      AND LOOP!
5750;
5760 KIND   .BYTE $00,'%$'
5770;
5780 DISVAR LDA #SRCSTR     INIZ
5790        STA PUTPTR      TO TOP OF TABLE
5800        LDA #SRCSTR/256 ALSO HANDLE MSB
5810        STA PUTPTR+1
5820 DISVA1 LDA PUTPTR      GET CURRENT POINTER
5830        CMP REFBOT      COMPARE TO END
5840        BNE DISVA2      NO ==>
5850        LDA PUTPTR+1    MAYBE, CHECK MSB
5860        CMP REFBOT+1
5870        BNE DISVA2      NO ==>
5880        JSR CRLF        YES! DO CLEAN-UP
5890        LDA DEFAUL+1    GET CONSOLE DV#
5900        STA OUFLAG      MAKE IT CURRENT
5910        RTS             AND QUIT
5920 DISVA2 LDY #$00        INIZ
5930        STY COUNT       CLEAR COUNT
5940        LDA (PUTPTR),Y  FETCH 1ST CHARACTER
5950        STA VARNAM      SAVE IT
5960        INY             BUMP INDEX
5970        LDA (PUTPTR),Y  FETCH 2ND CHARACTER
5980        STA VARNAM+1    SAVE IT
5990        INY             BUMP INDEX
6000        LDA (PUTPTR),Y  FETCH TYPE
6010        AND #$10        MASK TO 'FN' TYPE
6020        BEQ DISV21      NO ==>
6030        STY TMPPTR
6040        JSR STROUT      YES! SHOW IT!
6050        .BYTE 'FN',$00
6060        LDY TMPPTR      REFETCH INDEX
6070 DISV21 LDA VARNAM      GET 1ST CHARACTER
6080        JSR OUTCH       PRINT IT
6090        LDA VARNAM+1    GET 2ND CHARACTER
6100        CMP #SP         SPACE ?
6110        BEQ DISVA3      YES! IGNORE IT! ==>
6120        JSR OUTCH       NO! PRINT IT!
6130 DISVA3 LDA (PUTPTR),Y  FETCH VARIABLE TYPE
6140        PHA             SAVE ON STACK
6150        AND #$0F        MASK TO LOW NYBBLE
6160        TAX             PUT IN X
6170        LDA KIND,X      FETCH TYPE CHARACTER
6180        BEQ DISVA4      F.P.? ==> DISVA4
6190        JSR OUTCH       NO! OTHER! SHOW IT!
6200 DISVA4 PLA             RETRIEVE ORIGINAL
6210        BPL DISVA5      NO SUBSCRIPT? ==>
6220        STY TMPPTR      YES! SAVE Y HERE
6230        JSR STROUT      SHOW ARRAY
6240        .BYTE '(X)',$00
6250        LDY TMPPTR      RETRIEVE Y
6260 DISVA5 INY             BUMP INDEX
6270        LDA (PUTPTR),Y  FETCH # OF REFS.
```

```
6280        STA NR          SAVE IT
6290        JSR CRLF        DO SEPARATION
6300        JSR CRLF
6310        LDA PUTPTR      GET PUTPTR
6320        CLC
6330        ADC #$04        ADD HEADER OFFSET
6340        STA PUTPTR      SAVE IT
6350        BCC DISVA6      WATCH PAGING
6360        INC PUTPTR+1
6370 DISVA6 LDY #$00        INIZ
6380        LDA (PUTPTR),Y  GET REF. LINE # LSB
6390        STA RESLO
6400        INY             BUMP INDEX
6410        LDA (PUTPTR),Y  GET REF. LINE # MSB
6420        STA RESHI       SAVE IT
6430        JSR NUMOUT      PRINT IT
6440        LDA #SP         GET A SPACE
6450        JSR OUTCH       PRINT IT
6460        JSR OUTCH       TWICE
6470        LDA PUTPTR      GET CURRENT POINTER
6480        CLC
6490        ADC #$02        ADD LINE ENTRY
6500        STA PUTPTR
6510        BCC DISVA7      WATCH PAGING
6520        INC PUTPTR+1
6530 DISVA7 INC COUNT       SHOW WE PRINTED ONE
6540        LDA COUNT       FETCH IT
6550        CMP #$0A        DONE 10?
6560        BCC DISVA8      O.K., CONT.
6570        LDA #$00
6580        STA COUNT       CLEAR COUNT
6590        JSR CRLF        DO A <CR><LF>
6600 DISVA8 DEC NR          SHOW A PASS
6610        BNE DISVA6      LOOP 'TIL DONE
6620        JSR CRLF        DONE! CLEAN UP LINE!
6630        JSR CRLF
6640        JMP DISVA1      GO TO NEXT VARIABLE
6650;
6660        .END XREF1
```

I H S Computer Services Introduces ALPHA/OMEGA Series Software

ALPHA/OMEGA Business Management System

- ✱ Integrated Accounting System for hard disks -- G/L, A/R, A/P, Inventory, P/R, POS.
- ✱ File locking on all Inventory and A/R functions for multi-user systems.
- ✱ Many advanced features, such as Departments (up to 99), Automatic Billing, Budgeting, Comparison with Previous Year, Detailed Cust/Vend/Inven records.
- ✱ Fully screen formatted -- Uses OSI's CRT File to adapt to any terminal.
- ✱ OS-DMS Type 10 File Structure. Extensive use of Key Files for rapid access.
- ✱ Ideal for almost all types of businesses. Easily integrates with specialized applications.

ALPHA/OMEGA Agricultural Management System

- ✱ Comprehensive System for all fertilizer (liquid and dry) dealers.
- ✱ Comprehensive Crop Management from Lab Analysis to Field History. Complete and attractive reporting system for customers.
- ✱ Communications package for customers who have a computer. Runs on their computer and allows downloading of data over phone line.
- ✱ Completely integrated with Alpha/Omega Business Management System.
- ✱ Includes EPA reporting for restricted chemicals and MORE.

I H S Computer Services Route 1 Box 201B Port Republic, VA 24471
(703) 249 - 4833

C1P CORNER

By: David A. Jones
9226 NW 17 St.
Pompano, FL 33065

Part II of a 2 part series on the 64 character modification.

For cassette and HEXDOS users, the simplest 64 character display to start using is ROM BASIC's output routine located at $BF2D and called from $FF69. When this code is executed, it checks location $FFE0 for the initial cursor position ($65), location $FFE1 for the width minus one of one line ($17) and $FFE2 to determine the model ($00). Programming a new PROM and changing these locations to $40, $3F, and $01 respectively forces the output to be 64 characters per line. The drawback here is you must give up 24 character per line capability and compatibility with any existing C1P software that depends upon the normal print routine.

One could also write a new display driver for 64 characters and keep the old for 24 characters thus retaining unmodified C1P compatibility. The new driver could be stored on tape, (possible but not practical), stored in EPROM (possible and practical if you have access to a PROM programmer), or stored on disk if you have a disk based system (the easiest way to go).

The following code is written for the second approach, EPROM, but can be adapted to either of the other by locating the code in RAM. OS65D users will need to use different addresses for the flags and scroll routine but can eliminate the boot changes. I used the top of memory for this function and modified track zero to skip over this section when checking for the amount of memory present.

To start up in the correct mode and allow for mode changes, the boot routine at $FEF0 must be changed. Originally, there was a screen clear routine in this code as well as one in the monitor at $FE00. Eliminating the first screen clear and making the second a subroutine enables us to clear the screen with a JSR and use the now unused memory locations to zero the new flags and set the screen width.

The input vector, originally $FFBA, is changed to $F9A0 where a check is made to see if the user wants to switch to the enhanced input routine.

```
1000                         ;I/064 (INPUT/OUTPUT 64 CHARACTERS)
1010                         ;12-5-83
1020 00FC=                   TEMP  =$FC
1030 00FF=                   XTMP  =$FF
1040 0201=                   TMP1  =$0201
1050 0202=                   TMP2  =$0202
1060 0204=                   PFLG  =$0204
1070 0207=                   SCFM  =$0207
1080 020A=                   SCTO  =$020A
1090 020B=                   CURS  =$020B
1100 0217=                   CFLG  =$0217
1110 0222=                   USER  =$0222
1120 9800=                   XMON  =$9800
1130 B95E=                   CONV  =$B95E
1140 D7A4=                   CRT   =$D7A4
1150 DF00=                   KYBD  =$DF00
1160 FD00=                   KBRD  =$FD00
1170 FE0E=                   CLEAR =$FE0E
1180 FE93=                   LEGL  =$FE93
1190 FEDA=                   ROLA  =$FEDA
1200 FEED=                   INPT  =$FEED
1210 FF6C=                   OLDOUT=$FF6C
1220 FFEE=                   BASOUT=$FFEE
1230 FFBF=                   ACIA  =$FFBF
1240                         ;
1250 F9A0                    *=$F9A0
1260                         ;
1270 F9A0 2C0302  INPUT      BIT $0203        CHECK LOAD FLAG
1280 F9A3 1003               BPL NSET
1290 F9A5 4CBFFF             JMP ACIA         IF SET
1300                         ;
1310 F9A8 2000FD  NSET       JSR KBRD         GET CHARACTER
1320 F9AB C902               CMP #$02         CONTROL B ?
1330 F9AD F001               BEQ NEWVEC
1340 F9AF 60                 RTS              BACK TO CALLING ROUTINE
1350                         ;
1360 F9B0 ADCFF9  NEWVEC     LDA NEWIN+1      SET NEW INPUT VECTOR
1370 F9B3 8D1802             STA $0218
1380 F9B6 ADD0F9             LDA NEWIN+2
1390 F9B9 8D1902             STA $0219
1400 F9BC AD87FA             LDA NEWOUT+1     SET NEW OUTPUT VECTOR
1410 F9BF 8D1A02             STA $021A
1420 F9C2 AD88FA             LDA NEWOUT+2
1430 F9C5 8D1B02             STA $021B
1440 F9C8 8D00F3             STA $F300        SET SCREEN WIDTH TO 64
1450 F9CB A90D               LDA #$0D
1460 F9CD 60                 RTS
1470                         ;
1480 F9CE 4CD1F9  NEWIN      JMP NEWIN+3
1490 F9D1 2C0302             BIT $0203        CHECK LOAD FLAG
1500 F9D4 1003               BPL KEY
1510 F9D6 4CBFFF             JMP ACIA         IF SET
1520                         ;
1530 F9D9 2000FD  KEY        JSR KBRD         GET CHARACTER
1540                         ;
1550 F9DC C901    CTRLA      CMP #$01
1560 F9DE D00A               BNE CTRLH
1570 F9E0 200EFE             JSR CLEAR        CLEAR SCREEN
1580 F9E3 A900               LDA #0
1590 F9E5 8D1702             STA CFLG         HOMES CURSOR
1600 F9E8 A90D               LDA #$0D
1610                         ;
1620 F9EA C908    CTRLH      CMP #$08
1630 F9EC D003               BNE CTRLP
1640 F9EE 4CA4FB             JMP HEXX         HEX CONVERSION
1650                         ;
1660 F9F1 C910    CTRLP      CMP #$10
1670 F9F3 D015               BNE CTRLX
1680 F9F5 48                 PHA
1690 F9F6 AD0402             LDA PFLG
1700 F9F9 C950               CMP #'P
1710 F9FB D004               BNE SET
1720 F9FD A920               LDA #$20
1730 F9FF D002               BNE STOR
1740 FA01 A950    SET        LDA #'P
1750 FA03 8D0402  STOR       STA PFLG         TOGGLE FLAG
1760 FA06 8DA4D7             STA CRT          29TH LINE
1770 FA09 68                 PLA
1780                         ;
1790 FA0A C918    CTRLX      CMP #$18
1800 FA0C D003               BNE CTRLZ
1810 FA0E 4C0098             JMP XMON         EXTENDED MONITOR
1820                         ;
1830 FA11 C91A    CTRLZ      CMP #$1A
1840 FA13 D003               BNE CURCHK
1850 FA15 4C2202             JMP USER
1860                         ;
1870                         ;
1880 FA18 8D0202  CURCHK     STA TMP2         SAVE CHARACTER
1890 FA1B 8A                 TXA
1900 FA1C 48                 PHA
1910 FA1D 98                 TYA
1920 FA1E 48                 PHA
1930 FA1F AD0202             LDA TMP2
1940                         ;
1950 FA22 C9EF    COPY       CMP #$EF         CTRL /
1960 FA24 D005               BNE RUBOUT
1970 FA26 AD0102             LDA TMP1
1980 FA29 D058               BNE OUT2
1990                         ;
2000 FA2B C97F    RUBOUT     CMP #$7F
2010 FA2D D004               BNE FORWD
2020 FA2F A95F               LDA #$5F         SHIFT O
2030 FA31 D050               BNE OUT2
2040                         ;
2050 FA33 C9EE    FORWD      CMP #$EE         CTRL >
2060 FA35 D00E               BNE BACK
2070 FA37 203CFB             JSR PRINT
```

Listing continued

By not switching unless directed, maximum compatibility with existing software is maintained (control B being the exception and the switch command). If the switch is executed then the input flow is through $F9D1.

When ↑B is chosen, the mode is changed to 64 and the screen editor is enabled. The new display is 28 lines of 64 characters with a 29th line used for status. A "P" shows up here when the printer is enabled, a "T" when in the terminal mode, etc. The cursor positioning portion of this code was derived from Kerry Lourash's "Cursor Control for the C1P" which appeared in the May 1981 issue of Micro. His version was for a 24 character per line display and included more features than the enclosed code. Get hold of a back issue, especially if you're interested in implementing a windowed display.

Note that the default mode is 24, and the switch is made by ↑B. (POKING locations 536, 537, 538, and 539 accomplishes this task inside a program). Also. note that the code has provisions for a parallel printer interface routine for which the code after the

```
2080 FA3A EE0B02           INC CURS
2090 FA3D D03F             BNE OUT1
2100 FA3F EE0C02           INC CURS+1
2110 FA42 18               CLC
2120 FA43 9039             BCC OUT1        BRANCH ALWAYS
2130                       ;
2140 FA45 C9EC     BACK    CMP #$EC        CTRL <
2150 FA47 D009             BNE UP
2160 FA49 203CFB           JSR PRINT
2170 FA4C 2048FB           JSR BACKUP
2180 FA4F 18               CLC
2190 FA50 902C             BCC OUT1
2200                       ;
2210 FA52 C915     UP      CMP #$15        CTRL U
2220 FA54 D013             BNE DOWN
2230 FA56 203CFB           JSR PRINT
2240 FA59 AD0B02           LDA CURS
2250 FA5C 38               SEC
2260 FA5D E940             SBC #$40
2270 FA5F 8D0B02           STA CURS
2280 FA62 B01A             BCS OUT1
2290 FA64 CE0C02           DEC CURS+1
2300 FA67 D015             BNE OUT1
2310                       ;
2320 FA69 C904     DOWN    CMP #$04        CTRL D
2330 FA6B D016             BNE OUT2
2340 FA6D 203CFB           JSR PRINT
2350 FA70 AD0B02           LDA CURS
2360 FA73 18               CLC
2370 FA74 6940             ADC #$40
2380 FA76 8D0B02           STA CURS
2390 FA79 9003             BCC OUT1
2400 FA7B EE0C02           INC CURS+1
2410                       ;
2420 FA7E 2028FB   OUT1    JSR PUTCUR
2430 FA81 A900             LDA #0          NON-PRINTING CHARACTER
2440 FA83 4CB7FD   OUT2    JMP $FDB7       TO BASIC
2450                       ;
2460                       ;
2470 FA86 4C89FA   NEWOUT  JMP NEWOUT+3
2480 FA89 8D0202           STA TMP2        TEMP SAVE
2490 FA8C 48               PHA
2500 FA8D 8A               TXA
2510 FA8E 48               PHA
2520 FA8F 98               TYA
2530 FA90 48               PHA
2540 FA91 AD1702           LDA CFLG
2550 FA94 D013             BNE PRIN
2560 FA96 200EFE           JSR CLEAR       CLEAR SCREEN
2570 FA99 A980             LDA #$80        HOME CURSOR
2580 FA9B 8D0B02           STA CURS
2590 FA9E A9D0             LDA #$D0
2600 FAA0 8D0C02           STA CURS+1
2610 FAA3 2028FB           JSR PUTCUR
```

Continued on page 15

WATCH

THIS SPACE IN APRIL

FOR

NEW PRODUCT

ANNOUNCEMENTS

DBi, inc.

p.o. box 7276
denver, co 80207
(303) 428-0222

D&N MICRO PRODUCTS, INC.

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

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

COMPUTER

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

MODEL 80-1200 $2995
2 8" Single sided drives
MODEL 80-2400 $3495
2 8" Double sided drives

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

MODEL 65-1 $2995
2 8" Single sided drives
MODEL 65-2 $3495
2 8" Double sided drives

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

MEM-CM9 MEMORY/ FLOPPY CONTROLLER
24K memory/floppy controller card uses 2114 memory chips, 1 8K and 1 16K partition. Supports OSI type disk interface

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

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

PRINTERS

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

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

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

HARDWARE
OSI COMPATIBLE

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

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

64K CMOS-MEM	$490
48K CMOS-MEM	$390
24K CMOS-MEM	$250
16K CMOS-MEM	$200
Controller add.	$ 90
2 IO mapped serial ports add. on assembled memory board	$125

Z80-IO 2 IO mapped serial.... $160
ports for use with D&N-80 CPU card
FL470 Disk Controller....... $155
Specify 5¼ or 8" drive

STANDARD CP/M FOR OSI

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

D&N-80-P	$349
CP/M 2.2	$150
64K CMOS-MEM with D&N-80 CPU card	$450

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

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

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

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

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

branch is not shown. My parallel printer interface is unique to my surplus printer so the particular driver is not of general interest but the hook is. You could use the check to branch to any other routine for your unique system. In addition ↑X causes a jump to the extended monitor relocation and residing in EPROM at $9800, again something unique to my system but can be implemented on yours by installing an EPROM there. Refer to my article in Feb 1983 Micro.

Control A clears the screen (POKING 535,0 clears the screen from inside a program the next time an output statement is executed. Control H branches to a hexadecimal to decimal conversion routine. see my article in PEEK(65) June 1982 for comments on the code. Control Z will cause a branch to a user routine located at $0222 (546). Control Q, XON, and control S. XOFF, are also honored, see PEEK(65) Dec. 1980 for more information here.

The source code is divided into 2 sections, thus the separate declarations of labels and variables. Section 1 pertains to the new input/output routines and section 2 to the new boot mechanism.

The jumps at newin and newout are used to allow assembly without defining absolute addresses for the beginning of each of these routines.

To use the screen editor the cursor is positioned by using the control keys. Control U moves the cursor up. control D moves it down and control < and > move it left and right respectively. Control / reads into the input buffer any character the cursor passes over. Either shift O or rubout will erase a character already entered and will move the cursor back one space.

Referring to those back issues again, you'll note that the C1P does not use any of the code from $F800 to $FBFF. I previously suggested that a cassette load routine be located at $F800-F89F (March 81) and a cassette save routine be located at $F8A0-F99F. This leaves $F9A0-FBFF free for our display driver. Not all of this space is required for this implementation so you can add some of your own enhancements.

★

```
2620 FAA6 8D1702        STA CFLG        CLEAR FLAG
2630                    ;
2640 FAA9 AD0402 PRIN   LDA PFLG
2650 FAAC C950          CMP #'P         CHECK FLAG
2660 FAAE D006          BNE CRET        IF NOT SET
2670 FAB0 AD0202        LDA TMP2
2680 FAB3 4CB6FA        JMP CRET        WOULD JSR TO PRINTER ROUTINE HERE
2690                    ;
2700 FAB6 AD0202 CRET   LDA TMP2
2710 FAB9 C90D          CMP #$0D
2720 FABB D014          BNE ERASE
2730 FABD A920          LDA #$20        ERASE CURSOR IF CR
2740 FABF 8D0102        STA TMP1
2750 FAC2 203CFB        JSR PRINT
2760 FAC5 AD0B02        LDA CURS
2770 FAC8 18            CLC
2780 FAC9 6940          ADC #$40
2790 FACB 8D0B02        STA CURS
2800 FACE 4CFCFA        JMP NEWL
2810                    ;
2820 FAD1 C95F   ERASE  CMP #$5F        SHIFT O
2830 FAD3 D013          BNE CCHAR
2840 FAD5 C60E          DEC $0E
2850 FAD7 A920          LDA #$20        ERASE CHAR UNDER CURSOR
2860 FAD9 8D0102        STA TMP1
2870 FADC 203CFB        JSR PRINT       ERASE CURSOR
2880 FADF 2048FB        JSR BACKUP
2890 FAE2 2033FB        JSR PUT1        DISPLAY CURSOR
2900 FAE5 4C1DFB        JMP EXIT
2910                    ;
2920 FAE8 C920   CCHAR  CMP #$20        DON'T PRINT CONTROL CHARACTERS
2930 FAEA 3031          BMI EXIT
2940 FAEC 8D0102 LETTER STA TMP1
2950 FAEF 203CFB        JSR PRINT
2960 FAF2 EE0B02        INC CURS
2970 FAF5 AD0B02        LDA CURS
2980 FAF8 293F          AND #$3F
2990 FAFA D01E          BNE OUT
3000                    ;
3010 FAFC AD0B02 NEWL   LDA CURS        START NEW LINE
3020 FAFF 29C0          AND #$C0
3030 FB01 8D0B02        STA CURS
3040 FB04 C980          CMP #$80
3050 FB06 D00A          BNE BLOCK
3060 FB08 AD0C02        LDA CURS+1
3070 FB0B C9D7          CMP #$D7
3080 FB0D D008          BNE OUT
3090 FB0F 2054FB        JSR SCRL
3100                    ;
3110 FB12 AD0B02 BLOCK  LDA CURS
3120 FB15 D003          BNE OUT
3130 FB17 EE0C02        INC CURS+1
3140                    ;
3150 FB1A 2028FB OUT    JSR PUTCUR
3160                    ;
3170 FB1D 68     EXIT   PLA
3180 FB1E A8            TAY
3190 FB1F 68            PLA
3200 FB20 AA            TAX
3210 FB21 68            PLA
3220 FB22 2089FB        JSR XONF
3230 FB25 4C6CFF        JMP OLDOUT
3240                    ;
3250                    ;
3260 FB28 A9AD   PUTCUR LDA #$AD        LDA
3270 FB2A 8D0A02        STA CURS-1
3280 FB2D 200A02        JSR CURS-1
3290 FB30 8D0102        STA TMP1
3300 FB33 A98D   PUT1   LDA #$8D        STA
3310 FB35 8D0A02        STA CURS-1
3320 FB38 A9BB          LDA #$BB        CURSOR CHAR
3330 FB3A D008          BNE PRINT1
3340                    ;
3350 FB3C A98D   PRINT  LDA #$8D        STA
3360 FB3E 8D0A02        STA CURS-1
3370 FB41 AD0102        LDA TMP1
3380 FB44 200A02 PRINT1 JSR CURS-1
3390 FB47 60            RTS
3400                    ;
3410 FB48 AD0B02 BACKUP LDA CURS
3420 FB4B D003          BNE BK1
3430 FB4D CE0C02        DEC CURS+1
3440 FB50 CE0B02 BK1    DEC CURS
3450 FB53 60            RTS
3460                    ;
3470                    ;
3480 FB54 A007   SCRL   LDY #7
3490 FB56 B962FB MOVE   LDA CODE-1,Y    CODE TO BE RELOCATED
3500 FB59 990602        STA SCFM-1,Y    TO RAM AT $0207
3510 FB5C 88            DEY
3520 FB5D D0F7          BNE MOVE
3530 FB5F A207          LDX #7
3540 FB61 D007          BNE LINE        SKIP CODE TO BE RELOCATED
3550                    ;
3560 FB63 B9C0D0 CODE   LDA $D0C0,Y     START FROM HERE
3570 FB66 9980D0        STA $D080,Y     AND MOVE IT TO HERE
3580 FB69 60            RTS
3590                    ;
3600 FB6A 200702 LINE   JSR SCFM
3610 FB6D C8            INY
3620 FB6E D0FA          BNE LINE
3630 FB70 EE0902        INC SCFM+2
3640 FB73 EE0C02        INC SCTO+2
3650 FB76 CA            DEX
3660 FB77 D0F1          BNE LINE
3670 FB79 A920          LDA #$20
3680 FB7B A03F          LDY #$3F
```

Listing continued

```
3690 FB7D 9940D7  BLANK  STA $D740,Y     BLANK LAST LINE
3700 FB80 88             DEY
3710 FB81 10FA           BPL BLANK
3720 FB83 A940           LDA #$40
3730 FB85 8D0B02         STA CURS
3740 FB88 60             RTS
3750                     ;
3760 FB89 48      XONF   PHA
3770 FB8A A9F6           LDA #$F6        SELECT ROWS 1 & 3
3780 FB8C 8D00DF         STA KYBD
3790 FB8F A9C0           LDA #$C0        SELECT COLUMNS 7 & 6
3800 FB91 2C00DF         BIT KYBD        STOP? (CONTROL S)
3810 FB94 D00C           BNE CONT        IF NOT, CONTINUE
3820 FB96 A9FC    WAIT   LDA #$FC        ROWS 0 & 1
3830 FB98 8D00DF         STA KYBD
3840 FB98 A9C0           LDA #$C0        COLUMNS 7 & 6
3850 FB9D 2C00DF         BIT KYBD        RESUME? (CONTROL Q)
3860 FBA0 D0F4           BNE WAIT        IF NOT, KEEP LOOPING
3870 FBA2 68      CONT   PLA
3880 FBA3 60             RTS
3890                     ;
3900 FBA4 A924    HEXX   LDA #'$
3910 FBA6 20EEFF         JSR BASOUT
3920 FBA9 A204           LDX #4
3930 FBAB 20EDFE  INNN   JSR INPT
3940 FBAE 20EEFF         JSR BASOUT
3950 FBB1 2093FE         JSR LEGL
3960 FBB4 86FF           STX XTMP
3970 FBB6 A200           LDX #0
3980 FBB8 20DAFE         JSR ROLA
3990 FBBB A6FF           LDX XTMP
4000 FBBD CA             DEX
4010 FBBE D0EB           BNE INNN
4020 FBC0 A6FC           LDX TEMP
4030 FBC2 A5FD           LDA TEMP+1
4040 FBC4 205EB9         JSR CONV
4050 FBC7 A90D           LDA #$0D
4060 FBC9 20EEFF         JSR BASOUT
4070 FBCC A90A           LDA #$0A
4080 FBCE 20EEFF         JSR BASOUT
4090 FBD1 60             RTS
4100                     ;
4110                     ;
4120 0000=               WSTART=$0000
4130 0130=               NMI   =$0130
4140 01C0=               IRQ   =$01C0
4150 BD11=               CSTART=$BD11
4160 FC00=               DSKIN =$FC00
4170 FCA6=               INIT  =$FCA6
4180 FE00=               VM    =$FE00
4190 FF69=               OUTPUT=$FF69
4200 FF8B=               LOAD  =$FF8B
4210 FF96=               SAVE  =$FF96
4220 FF9B=               CTRLC =$FF98
4230                     ;
4240 FE00                *=$FE00
4250                     ;
4260 FE00 A228    VM     LDX #$28
4270 FE02 9A             TXS
4280 FE03 D8             CLD
4290 FE04 200EFE         JSR CLEAR
4300 FE07 84FF           STY $FF
4310 FE09 84FB           STY $FB         CLEAR ACIA FLAG
4320 FE0B 4C43FE         JMP $FE43
4330 FE0E 48      CLEAR  PHA             CLEAR SCREEN
4340 FE0F A9D0           LDA #$D0
4350 FE11 85FF           STA $FF
4360 FE13 A000           LDY #0
4370 FE15 84FE           STY $FE
4380 FE17 A920    C1     LDA #$20
4390 FE19 91FE    C2     STA ($FE),Y
4400 FE1B E6FE           INC $FE
4410 FE1D D0FA           BNE C2
4420 FE1F E6FF           INC $FF
4430 FE21 A5FF           LDA $FF
4440 FE23 C9D8           CMP #$D8
4450 FE25 D0F0           BNE C1
4460 FE27 68             PLA
4470 FE28 60             RTS
4480 FE29 EA             NOP
4490                     ;
4500 FEF0                *=$FEF0         I/O VECTORS
4510                     ;
4520 FEF0 A0F9           .WORD INPUT
4530 FEF2 69FF           .WORD OUTPUT
4540 FEF4 9BFF           .WORD CTRLC
4550 FEF6 8BFF           .WORD LOAD
4560 FEF8 96FF           .WORD SAVE
4570 FEFA 3001           .WORD NMI
4580 FEFC 00FE           .WORD VM
4590 FEFE C001           .WORD IRQ
4600                     ;
4610 FF00 D8      BOOT   CLD
4620 FF01 A2FE           LDX #$FE        SET STACK POINTER
4630 FF03 9A             TXS
4640 FF04 A00A           LDY #10
4650 FF06 B9EFFE  PSET   LDA $FEEF,Y     PRESET I/O VECTORS
4660 FF09 991702         STA $0217,Y
4670 FF0C 88             DEY
4680 FF0D D0F7           BNE PSET
4690 FF0F 20A6FC         JSR INIT        INITIALIZE ACIA
4700                     ;
4710 FF12 A014           LDY #$14        CLEAR ALL FLAGS
4720 FF14 A900           LDA #0
4730 FF16 990302  ZERO   STA $0203,Y
4740 FF19 88             DEY
4750 FF1A 10FA           BPL ZERO
4760                     ;
4770 FF1C ADE0FF         LDA $FFE0
4780 FF1F 8D0002         STA $0200       PRESET ORIGINAL
                                         CURSOR LOCATION
4790 FF22 200EFE         JSR CLEAR       CLEAR SCREEN
4800 FF25 8D00F7         STA $F700       SET NARROW SCREEN
4810 FF28 EA             NOP
4820 FF29 EA             NOP
4830 FF2A EA             NOP
4840 FF2B EA             NOP
4850 FF2C EA             NOP
4860 FF2D EA             NOP
4870 FF2E EA             NOP
4880 FF2F EA             NOP
4890 FF30 EA             NOP
4900 FF31 EA             NOP
4910 FF32 EA             NOP
4920 FF33 A000           LDY #0
4930 FF35 B95FFF  BMSG   LDA TEXT,Y
4940 FF38 F006           BEQ INPCH
4950 FF3A 20EEFF         JSR BASOUT
4960 FF3D C8             INY
4970 FF3E D0F5           BNE BMSG
4980 FF40 20A0F9  INPCH  JSR INPUT
4990 FF43 C94D           CMP #'M
5000 FF45 D003           BNE WARM
5010 FF47 4C00FE         JMP VM
5020 FF4A C957    WARM   CMP #'W
5030 FF4C D003           BNE COLD
5040 FF4E 4C0000         JMP WSTART
5050 FF51 C943    COLD   CMP #'C
5060 FF53 D003           BNE DISK
5070 FF55 4C11BD         JMP CSTART
5080 FF58 C944    DISK   CMP #'D
5090 FF5A D0A4           BNE BOOT
5100 FF5C 4C00FC         JMP DSKIN
5110                     ;
5120 FF5F 44      TEXT   .BYTE 'D/C/W/M ?'
5120 FF60 2F
5120 FF61 43
5120 FF62 2F
5120 FF63 57
5120 FF64 2F
5120 FF65 4D
5120 FF66 20
5120 FF67 3F
5130 FF68 00             .BYTE $0
5140                     ;
```

THE NEW CHALLENGER PERSONAL COMPUTER

By: Bruce Showalter
857 Cedar
Abilene, TX 79601

The Superboard II/C1P generated no small amount of enthusiasm when it was introduced in 1979. Even today it has a loyal following. Its big brother, the C4P, is an even more praiseworthy machine.

But times change. The competition learned by the C1's example and went hard after the low-end user market. The result was that the Challenger's market was choked off by the flood of ZX-81s, TRS-80s, VIC-20s and Ataris.

The subsequent generations of OSI management have elected to continue pursuing the small business market. They evidently don't want to be burned at the personal market level again. But they don't have to, they can learn from the competition, just as the competition learned from them. Let's proceed, therefore, to propose a future version of the Challenger personal computer, the C-65.

Almost every owner of the C1s has modified the hardware. This points to obvious inadequacies which should have been corrected by the designers early on. Their attempt to recover the fumble is indicated by the revised edition of the S.11/C1P. So, let's build in the most desired features from the outset.

One of the most popular features of the Challengers was their changeability. The hardware hackers loved 'em. With that in mind, let's take a page from the Apple hardware design and build with multiple plug-in boards. We can use either the KIM-44 boards or the OSI-48 boards. If we choose the latter, the size should be about half the original. The only exception would be the video circuit, about which more later. The purpose of using smaller boards is to divide the computer into modules. Each module can be owner-customized or replaced entirely. Implicit details include sockets for each and every IC and use of LS-TTL throughout the system.

The first module is the CPU board. It consists of the microprocessor and suitable line buffers. There is also a clock circuit which can be easily modified or by-passed altogether. This design permits the owner to substitute another CPU of his own choosing. The CPU module is configured for a 1-mhz clock, but there is also a 2-mhz signal available for those who desire it. The WAIT circuit divides the clock speed in half. The CPU Reset line is connected to an R-C circuit which produces a Power -On-Reset pulse.

The MONITOR + BASIC module comes next. 2716 EPROMS are used to facilitate other operating systems and languages. We could use 2732 or 2764 EPROMS, but the 2716 is already well established. By putting firmware on a board by itself, the owner can change language and OS by merely switching boards.

Next, we have the KEYBOARD ADDRESS module. This consists of merely the decoders and buffers. A cable joins this board to the keyboard itself. This module will most likely remain unchanged, unless a new location in the memory map is desired. As with the previous Challengers, we use a polled keyboard. However, significant improvements are made. A hexadecimal numeric pad is added. The RUBOUT key is relabeled CLS to incorporate that function. Since REPEAT is programmed into the OS, we relabel that key for BACKSPACE. And we replace the LINE FEED key with CAPS LOCK. ESCAPE becomes CANCEL (Shift P). More will be said about the keyboard when we discuss the OS.

The CASSETTE + RS-232 I/O is built on the next module. Not much is different from the C1 circuitry, except that the RS-232 interface is installed. Note that the ACIA clock is derived from an on board source, rather than by dividing down a master clock signal. There are provisions for changing the ACIA clock from 48ØØ hz to 96ØØ hz. Preferably, a high speed cassette I/O (such as a VIC Rabbit) could be installed in place of the existing circuit. This would probably require a entirely new module.

Choice of the VIDEO module may be left to the purchaser, since tastes vary on this subject. Some prefer an 8Ø-column display, while others are satisfied with 64 or 48. Some prefer color and hi-res grahics. Perhaps the best bet would be to offer a 54Ø video board (less keyboard/analog inputs) and have second source vendors provide alternates (such as the Orion SEB). The buyer could elect to omit the 54Ø module from the C-65 package at the time of purchase if he didn't want that version.

RAM modules would be fairly standard. I'd recommend CMOS 2K x 8 chips, with each module holding 16K.

The DISK module would be sold like before, either as an expansion option or in a full fledged disk system. As with the video modules, second-source vendors could provide alternate configurations.

The foregoing implies a burden on OSI to offer hardware documentation and licensing to other vendors. This omission in the past, I believe, destined OSI's failure in the personal computer market. Software and alternate hardware from second-source vendors have contributed immensely to the success of Apple, Commodore, TRS, Atari, and IBM. Texas Instruments' recent failure in this market lends even more support to this argument.

Before we leave our discussion of hardware, we should take a look at the bus. Neither the KIM-44 nor the OSI-48 busses are completely satisfactory to me. A table gives the revised C-65 bus definitions I propose. No attempt is made to allow for 8Ø-type or 68-type processor lines. We'll leave CP/M to OSI's line of small business machines.

Now it's time to discuss the Operating System. For the machine-code hackers, a versatile MONITOR is a must with an assembler/editor. The keyboard format, we touched on earlier. To continue, we interpret either Left SHIFT, Right SHIFT, or SHIFT LOCK exactly the same: all character keys input their upper case symbol. Otherwise, the case depends upon whether CAPS LOCK is depressed. This key shifts only the alpha characters into upper case. All other characters are lower case. None of the following keys are affected by case: RETURN, SPACE, BACKSPACE, CONTROL, CANCEL. The RETURN function scrolls the display, but doesn't send a LINE FEED command to the ACIA.

DISK DRIVE RECONDITIONING

WINCHESTER DRIVES

FLAT RATE CLEAN ROOM SERVICE.
(parts & labor included)

Shugart SA1002	5meg	$390.00
Shugart SA1004	10meg	$450.00

FLOPPY DRIVE FLAT RATES
Parts & Labor Included (Missing parts extra)

8" Double Sided Siemens	$170.00
8" Single Sided Siemens	$150.00
8" Double Sided Remex	$225.00
8" Single Sided Shugart	$190.00
8" Double Sided Shugart	$250.00
5¼ M.P.I. Single Sided	$120.00
5¼ M.P.I. Double Sided	$150.00

ONE WEEK TURN AROUND TYPICAL
You'll be notified of —

1. The date we received your drive.
2. Any delays & estimated completion date.
3. Date drive was shipped from our plant.
4. Repairs performed on your drive.
5. Parts used (# and description).

90 day warranty —
Write or call for detailed brochure
We sell emergency parts
Phone: (417) 485-2501

FESSENDEN COMPUTERS
116 N. 3RD STREET
OZARK, MO 65721

Introducing
SCRIBE
WORD PROCESSOR

OS-65U 1.42 < Floppy/Hard Disk, Level 1 or Level 3

and DENVER BOARDS

✱INTERFACED TO OS-DMS FILES
✱AUTOMATIC WRAP AROUND
✱COMPLETE EDITING CAPABILITIES
FULL CURSOR CONTROL
INSERT & DELETE TEXT
SEARCH/SEARCH & REPLACE
✱USER FRIENDLY MANUAL
✱AND MUCH MORE

IHS COMPUTER SERVICES
Route 1 Box 201B Port Republic, VA 24471
(703) 249-4833

$195.00

BASIC-IN-ROM could stand some improvements. Right away, we fix the Garbage Collector. CLS is a must. Next, we incorporate a GET or INKEY statement. Another feature I consider especially useful is the handling of NULLS. I recommend that the number of NULLS be stored in RAM. So when we boot up, NULL = 1Ø. But with a POKE, NULL could be set to anything from zero to 255. One more useful statement is PRINT AT. There are others which I will leave for you, the readers, to recommend.

KIM-44 TO C-65 BUS CONVERSION

PIN#	KIM-44	MODIFICATION	C-65
1	GND		GND
2	SYNC	DISCONNECTED	
3	RDY	REDEFINED	WAIT
4	IRQ		IRQ
5	-15		-15
6	NMI		NMI
7	RST		RST
8	D7		D7
9	D6		D6
1Ø	D5		D5
11	D4		D4
12	D3		D3
13	D2		D2
14	D1		D1
15	DØ		DØ
16	BDSEL	DISCONNECTED	
17	+15		+15
18	DMA	REDEFINED	EXCLK
19	+8	DISCONNECTED	
2Ø	+8	DISCONNECTED	
21	+5		+5
22	GND		GND
A	GND		GND
B	AØ		AØ
C	A1		A1
D	A2		A2
E	A3		A3
F	A4		A4
G/H	A5		A5
J	A6		A6
K	A7		A7
L	A8		A8
M	A9		A9
N	A1Ø		A1Ø
P	A11		A11
R	A12		A12
S	A13		A13
T	A14		A14
U	A15		A15
V	Ø2_		Ø2_
W	R/W		R/W
X	Ø2	DISCONNECTED	
Y	+5		+5
Z	GND		GND

OSI-48 TO C-65 BUS CONVERSION

PIN#	OSI-48	MODIFICATION	C-65
1	WAIT		WAIT
2	NMI		NMI
3	IRQ		IRQ
4	DD	DISCONNECTED	
5	DØ		DØ
6	D1		D1
7	D2		D2
8	D3		D3
9	D4		D4
1Ø	D5		D5
11	D6		D6
12	D7		D7
13	D8	DISCONNECTED	
14	D9	DISCONNECTED	
15	D1Ø	DISCONNECTED	
16	D11	DISCONNECTED	
17	RST		RST
18		REDEFINED	EXCLK
19	A19	DISCONNECTED	
2Ø	A18	DISCONNECTED	
21	A16	DISCONNECTED	
22	A17	DISCONNECTED	
23	+12	INCREASED	+15
24	-9	INCREASED	-15
25	+5		+5
26	+5		+5
27	GND		GND
28	GND		GND
29	A6		A6
3Ø	A7		A7
31	A5		A5
32	A8		A8
33	A9		A9
34	A1		A1
35	A2		A2
36	A3		A3
37	A4		A4
38	AØ		AØ
39	Ø2_		Ø2_
4Ø	R/W		R/W
41	VMA	TIED HIGH	+5
42	VMA-Ø2	REDEFINED	Ø2
43	A1Ø		A1Ø
44	A11		A11
45	A12		A12
46	A13		A13
47	A14		A14
48	A15		A15

READER PROFILE

ED:

I think it's time I wrote and supported this super journal.

My contribution is a description of a uniquely expanded C1P which I bought in January 79 for $330. Gradually, and I do mean gradually, I played with hanging stuff onto the expansion port.

Now, as depicted in the figure, I've totally designed, engineered, and built a 44 pin bus card rack system, and it works!

Technically:

- the adapter / driver board connects the 40 pin DIP port to a 40 pin IDC header. This card generates signal DD and decodes slots in the rack.

- a digital I/O card uses two 6522s to do general purpose stuff.

- a complex sound generator card occupies its own slot where a 6821 drives into a GI AY-3-8910. An ASCII keyboard will also be connected to the I/O Port of the AY-3-8910.

- the magic of the system is an 80 column card based on the 6545 CRTC. I've dedicated 16K of CPU mapped memory to video. Scrolling is accomplished from a 20 key keypad hung on the digital I/O card. (Not used by OS 65D, though.)

- the floppy disk controller is a copy of the 470 design.

The system:

- runs C4 / C8 software
- has 2 8" Siemens SS / SD drives
- capability of 48 K memory, accomplished by disabling BASIC.
- printer OKIDATA 82A
- front panel switches:
  * 300 or 1200 baud
  * BASIC in ROM on-off
  * CPU clock 1 or 2 Mhz
  * Functional C4 or C1 enable switch
- US Robotics "Password" modem

The superboard has been totally modified to provide for more address decoding. I installed the video mod from Progressive Computing and I enabled BASIC device #4 (parallel printer port).

I'd like to say that this has been a lot of fun, sweat, and tears. I've had to rediscover the meaning of much engineering design work. The more remarkable thing is that I've never had a formal course in electronics.

Coming shortly, I'll be adding a card designed by a Rockwell engineer. The card allows any 6502 machine to run CP/M. I will report my progress on that.

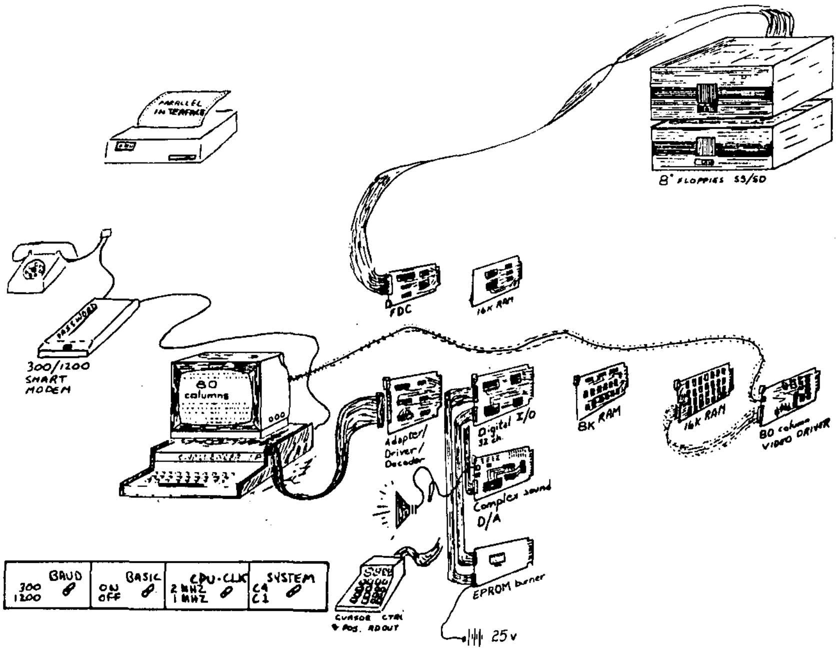
OSI REPAIRS

BOARD LEVEL SERVICE ON C2, C3 & CD SERIES, INCLUDING POWER SUPPLIES, 8" FLOPPY DRIVES, CIRCUIT BOARDS (GOLD UPGRADE)

ONE WEEK TURN-A-ROUND TYPICAL

SOKOL ELECTRONICS, INC.
474 N. POTOMAC ST.
HAGERSTOWN, MD 2174Ø

(3Ø1) 791-2562



I need to acknowledge my beautiful wife who has tolerated the money, time, and boring computer conversations. She can be considered a computer widow, but I really appreciate her.

I know y'all would like to know how much I've spent. Well, not including the printer or modem, I think it's been under $1600.

Gene W. Anderson
Sunnymead, CA 92388

LETTERS

ED:

Our firm runs a three user, ten megabyte Denver Board OSI (a converted C2) with OS65U V.1.43 operating system.

We have been trying to use WP 3.3 word processor, but unfortunately the computer"freezes" immediately after the response to the question:

"Do you want Device 8 to be set up for a serial printer?"

We have entered two new lines numbers 211 and 212 as stated in the September issue of PEEK, but this does not make any difference!

Could you, or one of your readers, please help?

John S. Spry
Wellington, Australia

John:

See "Bug Fix" further on this issue.

Peek(65) Staff

* * * * *

ED:

This is the first program I have submitted to PEEK(65). I felt California should be better represented. I have a C2-4PMF, but the program should certainly run on a C4 and I have been told it will run on a C1, but I cannot confirm this.

The program, re-written for OSI, came from Projects in Machine Intelligence by D. Heiserman. When first run, you are asked for two inputs in ASCII code. The *, 42, makes a nice creature (yes, I have withstood the impulse to title the program Creature Features). An entry of 32 for the trail will give you an invisible trail, while 161 will give a graphic symbol which makes a nice trail. The border is then drawn and the area within the borders is filled with 64 killable obstacles, randomly selected and placed. The creature or Alpha then zips around the screen at random, encountering obstacles and the border itself. Upon an encounter, there is a 50% chance the Alpha will effect a kill and continue on in the same direction. In the case of the border, the creature

NEED OSI COMPATIBLE PRODUCTS?

We sell SCIENCE AZTEC'S full line of OSI compatible PC Boards & Systems.

- 8550 Communications Interface (replaces 550)
- 8590 HD Controller (replaces 590/525)
- 8592 HD Interface
- 8588 Active Terminator
- 919 9 Slot back plane with active/passive terminators
- 68000 Boards to be announced shortly
- BD 14" 80 MB Hard Disk, with controller, interface & cable
- 8470 Floppy Disk Controller (replaces 470)
- 8472 8" or 5¼" Floppy Controller - IBM Format
- 8516 GT 3.3 Mhz CPU, 64k Static DMA
- 8528 Up to 4 partition GT memory, Static RAM

Prices available for Disk Subsystems as well as complete Computer Systems

BECTERM

12 Trans-Canada West Levis, Quebec G6V-4Z2 418-837-5894

can try to take a bite out of the border - it can put a dent in the border, graphic symbol 153, but can never completely escape. After the program runs for awhile, you will find most of the obstacles gone and the borders badly chewed. If you use a visible trail and the Alpha becomes cornered by its own trail, it becomes rather nasty and chews its way to relative freedom. And the Alpha has no qualms about turning cannibal if it encounters a like creature. If you use a trail code number between 128 and 154, the trail becomes impenetrable and the Alpha soon becomes hopelessly entrapped by its own trail. In all cases, the program will run until it is stopped with a CTRL'C'.

Possible modifications include adding color to the program and varying the number of obstacles. The subroutine in line 1040 can be used to display scoring - how many times is the border hit, how many moves are "good", how many kills, etc.. For persons more interested in the theory of machine intelligence and psychology, I would refer them to Mr. Heiserman's book Tab #1391.

If there is much interest in this program, I have a number of other programs available. The Beta programs display a learning response and could apply to the programming of a robot.

I enjoy PEEK(65) very much and am looking forward to the articles on I/O as this area is giving me some problems, I think more so since I have an older C2. I would like to connect a modem and I also have an ADM-2 display terminal I would like to use. At one time there was a local OSI User's Group, but it has fallen on hard times.

```
30   REM**KILLER ALPHA DEMO,OSI
     V3.3**40   PRINT!(28)
50   PRINT"STRIKE 'R. SHIFT'
     TO START..."
60   IF PEEK(57100)<>3 THEN
     N=RND(8):GOTO 60
70   INPUT "SELECT A CREATURE
     CODE";CT
80   PRINT:INPUT "SELECT A
     TRAIL CODE";TT
90   PRINT!(28):PRINT"YOUR CREATURE
     LOOKS LIKE THIS--";CHR$(CT):
     PRINT
95   PRINT "ITS TRAIL LOOKS LIKE
     THIS--";CHR$(TT):PRINT
100  INPUT "IS THAT WHAT YOU WANT
     (Y/N)";S$
105  IF S$<>"Y" THEN 70
107  REM**ALPHA MAINLINE,OSI V3.3**
110  PRINT!(28):GOSUB 1000
112  D$="KILLER ALPHA DEMO":D=55188:
     GOSUB 1040
115  CP=54328+INT(5*RND(8))-2+64*
     INT(5*RND(8))-2:POKE CP,CT
120  FOR N=0 TO 63
125  TP=53400+INT(1500*RND(8)):
     IF PEEK(TP)<>32 THEN 125
126  IF TP>55039 THEN 125
130  POKE TP,INT(33*RND(8))+33:
     NEXT N
135  CI=INT(5*RND(8))-2
136  CJ=INT(5*RND(8))-2:IF CI=0
     AND CJ=0 THEN 135
140  GOSUB 2000
145  IF NOT(CX=32 AND CY=32)
     THEN 165
150  POKE CP,TT
155  CP=NP:POKE CP,CT
160  GOTO 140
165  KC=INT(2*RND(8)):IF KC=0
     THEN 135
170  IF PEEK(NP)>=128 AND PEEK(NP)
     <=154 THEN POKE NP,153:GOTO 140
175  POKE NP,32:GOTO 140
1000 REM**BORDER,OSI V3.3**
1005 F0=53376
1006 F1=53439
1007 F2=54976
1008 F3=55039 40
1010 FOR N=F0 TO F1
1011 POKE N,128
1012 NEXT N
1015 FOR N=F2 TO F3
1016 POKE N,135
1017 NEXT N
1020 FOR N=F0 TO F2 STEP 64
1021 POKE N,149:NEXT
1025 FOR N=F1 TO F3 STEP 64:
     POKE N, 149:NEXT
1030 POKE F0,128:POKE F1,128
```

Continued

From Gander Software *The Ultimate Personal Planner*

TIME & TASK PLANNER

30 DAY FREE TRIAL — IF NOT SATISFIED, FULL REFUND UPON RETURN

- "Daily Appointment Schedule"
- "Future Planning List" - sorted
- "To Do List" - by rank or date
- Work Sheets for all Aspects
- Year & Month Printed Calendar
- Transfers to Daily Schedule

A SIMPLE BUT POWERFUL TOOL FOR SUCCESS

Put the two most effective success techniques to work for you — every day of every year. Just five to ten minutes a day allows your mind and dreams to take charge of your life.

Set Your Goals: To reach a goal, you have to know where you are going. Just enter your goals or future appointments and let your computer remind you.

Set Your Priorities: Success depends upon doing first things first. Assign priorities (1-99) to your "To Do" list, let the computer keep them ranked by date or priority, and then get to work. When the time comes, the computer will help you transfer items to your choice of time on the daily Appointment Scheduler.

Technicalities - Appointment Scheduler: 18 time slots per day (you define) for 60 days. To Do List: 60 items ranked by date or priority. Future Planning: 60 long range items, date sorted; days to event or days overdue. Transfer to Scheduler: just tell it the date and time. Printed Calendars: Year on a page and one month box planning; any month, any year. System uses both Julian and Gregorian calendars to handle dates from 1910-2399 and produce day of the week. Screen and menu driven; DMS Keybase compatible files. Detailed 38 page manual. Simple installation; FD to Multi HD. Files for 5 users=5,400 appointments. Unlimited Warranty.

HARDWARE: 48K OSI, 8" floppy or hard disk, serial terminal system, OS-65U v. 1.3 or later.

FEATURES: package allows configuration to ANSI standard and almost all non-ANSI terminals. AND user specification of printer port.

PRICE: $150.00 (User Manual, $25.00, credited toward TTP purchase). Michigan residents add 4% sales tax.

DEALERS: Your inquiries are invited. This program should be on every 65U machine, including your own. At dealer prices, you could bundle this superior package as a sales incentive.

GANDER SOFTWARE
3223 Bross Road
"The Ponds"
Hastings, MI 49058
(616) 945-2821

"It Flies"

Bug Fix for the WP-3 and WP-65Ø2 Word Processing Programs when using DBI's Denver boards.

WP-3 CHANGES

CREATE FILE - BASIC 4 25Ø88 N PASS

WP-3 Program (WP-3)

```
24   IF PEEK(16317)=5 THEN GOSUB 4ØØØ
115  IF PEEK(16317)=5 THEN GOSUB 4Ø1Ø :GOTO 14Ø
215  IF PEEK(16317)= 5 GOTO 36Ø
1Ø7Ø IF PEEK (16317)=5 THEN GOSUB 4Ø2Ø :GOTO 1Ø8Ø
1Ø75 POKE8778,Ø:POKE8779, 152:X=USR(X)
4ØØØ FLAG 52,3,Ø
4ØØ1 FLAG 52,5,Ø
4ØØ2 FLAG 52,6,Ø
4ØØ3 FLAG 52,8,Ø
4ØØ4 RETURN
4ØØ9 REM *** SET DV#8 FOR PARALLEL PORT ***
4Ø1Ø FLAG 57,Ø,4,8
4Ø11 RETURN
4Ø19 REM *** SET DV#8 TO SERIAL PORT ***
4Ø2Ø FLAG 57.Ø,1,8
4Ø21 RETURN
```

WP-3 Program (BASIC)

```
8Ø  IF X=5 THEN A$="BASIC4"  : REM MULTIPROCESSING BASIC
```

WP65Ø3 CHANGES

```
5 IF PEEK(65535)=254 THEN POKE 26885,76:POKE  26886,213:POKE
26887,1Ø4
```

WP65Ø2 CHANGES

```
5 IF PEEK(65535)=254 THEN POKE 26876,76:POKE  26877,2Ø4:POKE
26878,1Ø4
```

Continued from page 2Ø

```
1Ø31 POKE F2,135:POKE F3,135
1Ø35 RETURN
1Ø4Ø FOR Y=1 TO LEN(D$):POKE D+Y,ASC
       (MID$(D$,Y,1)):NEXT:RETURN
2ØØØ REM**SEARCH AHEAD, OSI V3.3**
2ØØ5 NP=CP:CX=32:CY=32
2Ø1Ø SI=SGN(CI):SJ=SGN(CJ):AI=ABS
     (CI):AJ=ABS(CJ)
2Ø15 IF AI=Ø THEN 2Ø3Ø
2Ø16 AI=AI-1
2Ø19 IF SI>Ø THEN NP=NP+1
2Ø2Ø IF SI>Ø THEN 2Ø25
2Ø21 NP=NP-1
2Ø25 CX=PEEK(NP)
2Ø3Ø IF AJ=Ø THEN 2Ø45
2Ø31 AJ=AJ-1
2Ø34 IF SJ>Ø THEN NP=NP+64
2Ø35 IF SJ>Ø THEN 2Ø4Ø
2Ø36 NP=NP-64
2Ø4Ø CY=PEEK(NP)
2Ø45 IF NOT (CX=32 AND CY=32)
      THEN RETURN
2Ø5Ø IF AI=Ø AND AJ=Ø THEN RETURN
2Ø55 GOTO 2Ø15
```

Robert Jents
El Sobrante, CA 948Ø3

* * * * *

ED:

First, let me thank you for the software listings. I had no idea there was so much available.

In the November issue, Frank Glandorf mentioned that the locations for the comma and

MEM *PLUS*

BARE	$ 75	32K	$300	52K	$415
16K	$200	40K	$350	56K	$440
24K	$250	48K	$390	64K	$490

MEM+ Options Include:

- OSI compatible floppy disk controller — add $85
- RTC — Real Time Clock — day, date and time with lithium battery backup — add $85
- Centronics parallel printer interface with software for OS65D and OS65U — add $65
- High reliability sockets for memory chips — add 15%
- RTC only (OSI CA-20 replacement) — $195

All boards feature solder mask, silkscreen, **gold-plated** edge connectors and a one year warranty.

07514 51893

Generic Computer Products

High Resolution Color Graphics

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

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

Pricing:

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

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

To order, or for more information, contact:
Fial Computer
5221 S.W. Corbett
Portland, Oregon 97201
(503) 227-7083

Introductory **SPECIAL** ***$9990.00****

BRAND NEW **REGULAR $12,990.00**

3 USERS-80 Mega Bytes

WITH DUAL FLOPPIES

- Configured for Time-Share @ 2 MHZ
- Includes: 2 Serial Printer Ports with Handshake and Improved Cooling
- Optional expanded bus allows up to 16 users with 1 floppy drive

Also available with 3 Multi-Processor Denver boards with 64K each user and Centronics Parallel Printer Port at $10,990.00

*Dealer Discounts Available

Ask for our Super Prices on 5, 10 & 20 MB 8″ Hard Disk Systems

Announcing!!

★ **10 Megabyte Removeable Disk Backup Unit.** On line as Dev "F". Uses any std file or system copy utility. ORDER NOW .... $3,495.00.

★ **NEW! 11 SLOT BACKPLANE** replaces any std 8 slot bus. Daisy chainable. .... Only $79.00.

LOOK!!

POWER SUPPLIES . . .

PS-1 (9137) +5 +12 -9	$ 79.00
QS-24 +24 volt/2.5 amp for floppy	$ 39.00
PS-35 +5 volt/3 amp for floppy	$ 20.00

DISK DRIVES . .

Siemans FDD 100-8 8″ s/s (new)	$199.00
Siemans FDD 100-8 8″ s/s (used)	$149.00
Shugart SA 801 8″ s/s (new)	$299.00

WOW

SPARE PARTS BONANZA!!

CA-9 CENTRONICS PARALLEL PRINTER INTERFACE w/CABLE (WOW!)	*$ 99.00*
CA-9D Diablo 12 bit Parallel interface w/ribbon cable	*$ 79.00*
CM-10 8K Executive Memory for level 3 or CP/M (WHILE THEY LAST!)	*$ 69.00*
510c CPU w/6502-Z80-6800 2 mhz (New)	*$199.00*
CM-3a 520 Rev C 16K Static Memory 2 mhz	*$ 79.00*
CM-3 520 Rev B 16K Static Memory 1 mhz	*$ 49.00*
CM-6 535 48K Dynamic Memory @ 1 mhz	*$199.00*
8-SB Std 8 slot backplane populated (Removed from new comp)	*$ 49.00*
542 Polled keyboard w/num pad, case and cable	*$ 69.00*
540 Rev B Color Video Board	*$ 99.00*

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

SPACE-COM International

22991 LaCadena Drive, Laguna Hills, California 92653

ORDER TODAY *(714)* **951-4648** *Some Quantities Limited* colon string terminators has been swapped. Well, those weren't the only ones! In the V3.3 reference manual, on page 21, is a table for the values to be poked for random file operation. The locations are 12042 and 12076. The table shows 12042 as the location for the number of records per track. This is wrong! 12076 is the right one. In using these pokes, the order is important. Poke 12042 first, then 12076. The values listed in the table are correct.

I recently bought and tried to use a modem on my C4P MF without any success. After fighting the program supplied by OSI (by the way, it works), I saw an ad by Aurora Software for an intelligent terminal program. I called and, while talking to them, I mentioned my problem. They told me that some C4s had the modem plug wired differently than others and to check that pin 5 was wired to ground (it wasn't). Maybe this will help someone else. I haven't received the program from Aurora yet. If no one else writes in about it, I'll let you know how it works.

Now, questions. Does anyone out there use a D&N Micro Z80 cpu card in their C4 or C8 system? I am interested in upgrading my system to something closer to a standard. Does anyone know of an 80-column board for video systems? Orion Software was the only one I had heard of, and they're out of the business.

Norman Thorsen
Poulsbo, WA 9837Ø

Norman:

To the best of our knowledge, D&N Proxy Z8Ø boards support video systems, but must be ordered with a video EPROM.

Readers, how about the 8Ø-column question?

Peek Staff

* * * * *

ED:

I am writing about the article that appeared in the Feb. 1984 issue by Guy Vanderwaeren. As the author of the article in MICRO that was referenced, I would regard it as only courteous to give the full reference ("Building a Parallel Printer Interface", MICRO 53, #1Ø (Oct 82), p. 23, by Rolf B. Johannesen.) With regard to Mr. Vanderwaeren's modification, I have only a couple of comments. First, with regard to the EPROM, if your system has a disk, the EPROM is obviously unnecessary, since the print routine can be made to reference the printer port you have built, and you can then forget it. Even with only BASIC-in-ROM, my printer code requires only 21 bytes and I don't regard this as so onerous that I would add an EPROM in preference to POKing this in each time I run. Secondly, there is a misunderstanding of the way the PRINT routine works in the C1P. When the C1P is first turned on, the user is asked first for the amount of memory, then the terminal width. If no number is entered to the width question, the value used by BASIC defaults to 72, and this determines the number of characters sent to the printer port before the return-line-feed is sent. At turn-on, another number may be entered - I have entered a number as large as 15Ø and had the program run correctly. Since BASIC already counts the number of characters before sending a return-linefeed to the printer port, it is quite unnecessary for the programmer to do this again. All of this is quite independent of the screen display routine, which does indeed send a return-linefeed (but only to the screen) after every 24 characters.

Rolf B. Johannesen
Rockville, MD 2Ø853

AD$

Send for free catalog, Aurora Software, 37 South Mitchell, Arlington Heights. IL 60005. Phone (312) 259-315Ø.

* * * * *

WANTED: C4PMF System. Prefer C4P DMF with 48K. Also mailing and word processing software. Send description to: Norman Thiel, 2Ø21 Grismer #37, Burbank, CA 915Ø4 or call 818-847-4664(day) 818-954-Ø549 (evening).

* * * * *

FOR SALE: LIKE NEW OSI C8PDF; 2-8" drives, 48K. Includes UCSD PASCAL & FORTRAN, Word Proc's WP65Ø2 V1.2 & V1.3, WP-2 & WP-3, 65U V1.2, 65D V3.2 & V3.3, DAC, Plot Basic, Home Cont, OS-DMS Nucleus. Sort, Educa, Inv-I, Inv-II, A/R, A/P, G/L, Purchasing, Query, Bill of Materials, Quote Estimation. 5 Game, 8 MDMS, includes Planner & Plotter, 3 Educational. and 8 Misc. Program Diskettes, over 1ØØ disks total. Also, many manuals for above, Tech Notes, PEEK(65)'s, video 1ØØ Monitor & disk caddy. $25ØØ or best offer. Jim Jansen, 12 Cedar, Great Falls, MT 594Ø5, (4Ø6) 727-211Ø.

* * * * *

EPROMS: *BASIC3: fully corrected garbage collector, eliminates string bug. *BASIC1 /BASIC4: one key 24/48 video on Series 2 C1P/SB, one key screen clear, true backspace, more. *MONITOR: corrected keyboard, screen editor, BASIC shorthand, terminal, machine code dump, more. $15 ppd. each EPROM. SASE for information. SOFTWARE SOLUTIONS, BOX 3753, Seattle, WA 98124.

* * * * *

FOR SALE: 32K OSI Challenger C3 with dual 8" drives. Okidata CP110 Printer with Centronics Interface. ADM-3A Terminal and Standard OSI Software. $1500 for everything. (313) 736-3923 leave message.

* * * * *

C2-OEM (two cases) with 48K RAM, dual 8" floppies, includes Centronics interface. RS-232C board, OS-65D, OS-65U, plus miscellaneous software. Almost brand new. No documentation. Must sell. $1500 (includes shipping) or offer. Rick Brown. 316 California #712, Reno, Nevada 89509, 702-322-9936.

* * * * *

FOR SALE: C8P with 48K. Includes Anadex DP-8ØØØ bidirectional printer and interface, many diskettes, joysticks. Well maintained --- all records. Available immediately. $96Ø. 86Ø-4915 Reston, VA evenings.

* * * * *

FOR SALE: ADM-3A Lear Seigler CRT/Modem with acoustic coupler. Hardly used. Over $13ØØ invested. $4ØØ. Reston, VA 86Ø-4915 evenings.

* * * * *

SUPER SALE: C1PMF w/24K, full documentation, Sams Manual. Cassettes & Diskettes Software & OS-65D. Computer in excellent condition, BAD IC in drive. $5ØØ complete. 12" mon/tv if you pay shipping. Oscar Frontera, Box 3517 Mayaguez, P.R. ØØ7Ø9. (8Ø9) 834-195Ø.

* * * * *

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 $1.65	$______
Postage & Handling	$ 3.50
TOTAL DUE	$______

POSTAGE MAY VARY FOR OVERSEAS