PEEK (65)

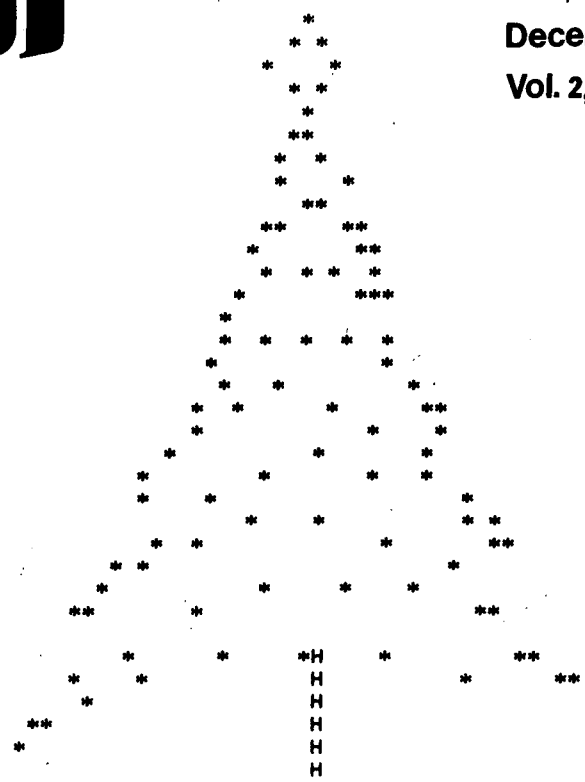
The Unofficial OSI Users Journal

★★ $1.75 ★★

December 1981

Vol. 2, No. 12

P.O. Box 347
Owings Mills, Md. 21117
(301) 363-3267



Column One

Merry Christmas and a very happy and peaceful New Year to all our readers and contributors!

The only thing constant in life is change, and the past month has seen more than its share. We have received a review copy of OS-65U V 1.3, which we will talk about in the next issue; we have had an argument with a faithful contributor stemming from the lengthening deadlines we have had to impose as PEEK(65) continues to grow; even the boolean logic contest has come to an end.

Faithful readers will recall that I was struck by a program line which included something like A=B*(D>9), and offered an unspecified prize for the best use of such "logic in parens."

After several months of hard work, we have been able at last to select a winner. Not too surprisingly, it is Jim Sanders. His entry (see PEEK(65), Oct 81, p.8) demonstrates several possible uses of this sort of logic in the clear and readable form we have come to expect of Jim.

For his hard work and clear writing, we have awarded Jim the best thing we have to offer -- a year's extension of his subscription to PEEK (65)!

A recent article in Infoworld, "Shake-up at Ohio Scientfic Instruments," stated that OSI was planning to redirect its sales efforts toward terminals, virtually abandoning computer sales. Several PEEK (65) readers have read this article and become understandably upset.

IT IS NOT TRUE!!

Paul Warren was kind enough to grant an interview to Infoworld, and mentioned, among other things, that intelligent terminal-type network nodes, designed to work with OSI's time-sharing hard disk computers, would be manufactured in the future. The Infoworld reporter drew some unwarranted conclusions from Paul's statements.

To set the record straight: OSI is **not** abandoning its present users, and **will** continue to manufacture and support both personal and small business computers.

Several readers have responded that they look forward to articles designed for the user rather than the programmer of OSI computers. The first appears in this issue on page 7, an article about an alternative Data Base Management System; I hope this article will be useful to both programmers and users, and we plan to follow it up with many more articles written at a somewhat lower technical level than most of our past material.

I am not really rambling; all of this is related to the theme of constant change. OSI is changing, 65U is changing, PEEK (65) is changing to reflect more closely the needs of our readers as reflected in the responses we receive through the mail and over the CBBS. But through it all, the more we all change, the more we remain the same. Neither OSI nor PEEK(65) will abandon our old friends!

al

INSIDE:

ADD AN 8-INCH FLOPPY TO THE C2-4P/C4P

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

A letter from Ron Biedenbach appeared in the May 1981 issue of PEEK (65) outlining the method of installing an 8" floppy disk drive on a C2-4P. The following details may help anyone who is considering such a system modification.

At the time I added the disk to my system, OSI didn't offer 8" drives for the C2-4P. (The C4P is functionally the same computer with the addition of an expanded I/O board on the back. The changes listed here apply to either machine.) Now, however, a C4P DF model is offered, DF standing for dual (8") floppies. The change came about when Siemens upgraded their disk drives: the older models required -9 volts for the op amps, which the C2/C4 doesn't provide. The current version of the drive, the FDD 100-8, only requires +24 volts for the stepper motor and +5 volts for the logic circuitry, in addition to 115 volts AC for the drive motor. The following steps are required for the upgrade:

Add More Memory

The OS 65D operating system requires just over 12 kbytes of memory, so at least 24 kbytes are necessary in order to provide a minimum workspace. The 502 CPU board supports 8 k of on-board RAM. By adding a 527 memory board, an additional 24 kbytes of memory can be added, for a maximum configuration of 32 k.

The cheap way to go is to order the 527 board with 8 k RAM installed. The board comes fully socketed and low-power 2114 chips are available from other suppliers at less than half OSI's price. (Recommended: A B Computers of Colmar, PA. They usually have ads in Byte and Kilobaud.)

Adding the memory board is simply a matter of plugging it into the backplane. If you specify that it is for a machine with an existing 8 k of memory, it should come addressed to start at $2000, so you can expand your memory in 4 k blocks by simply adding more 2114 chips.

Upgrade the Computer Power Supply

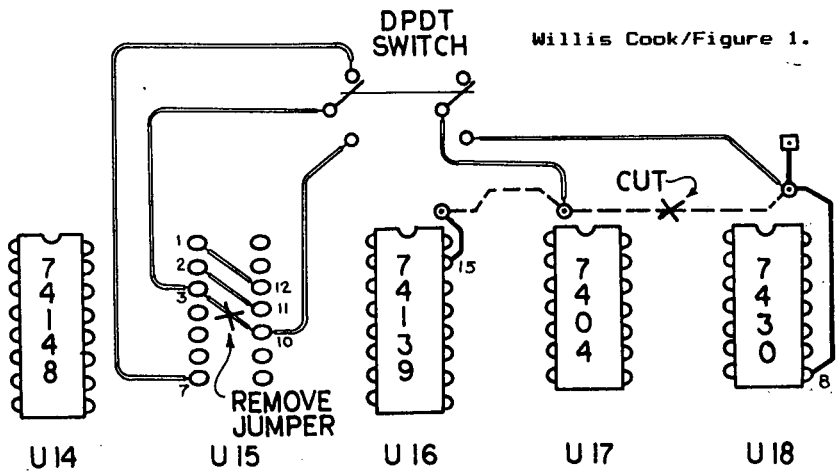
The existing power supply in the C2/C4 is inadequate for the additional memory plus the floppy disk controller board. With 24 k of low-power RAM aboard, my computer draws 4.75 amps at 5 volts.

My solution was to purchase a used 16 amp Lambda power supply and mount it external to the computer. Alternately, you could purchase a second 4.5 amp power supply and mount it in the space provided in the case. In either event, as a precaution, I recommend running a second lead from the new power supply to the opposite end of the +5 volt backplane foil, in order to split the current path. I don't know how much current the foil will carry, but a second lead is much cheaper than any unpleasant alternative.

Modify the 502 CPU Board

Two changes must be made to the CPU board. First, modify the ROM select socket jumpers as shown in figure 1. Located at position U15, the ROM select socket has the following jumpers in place for BASIC-IN-ROM operation: pins 1 to 12, pins 2 to 11 and pins 3 to 10. For disk operation the jumper to pin 10 must be removed and pin 3 wired to pin 7. In order to retain BASIC-IN-ROM capability, pins 3, 7 and 10 should be wired to a DPDT (double pole, double throw) switch as shown in the figure. The second switch pole is used on the second CPU board modification.

The second change involves disabling the ROM BASIC when using the disk. (I'm not sure this modification is required, but my system works with it in place.) It is done by breaking the foil running between U18, pin 8 and U16, pin 15 on the reverse side of the board. After making this cut, run a wire from each side of the cut to the other pole of the DPDT switch such that when BASIC-IN-ROM mode is selected, the two sides of the cut are connected through the switch. In disk mode, the cut should be open. Verify your work to this point by hitting the BREAK key: with the switch in one position, you should see the familiar "C/W/M" prompt; the other switch position will cause the prompt to read "H/D/M" and appear at the top of the screen.



Willis Cook/Figure 1.

Modifications to the 502 CPU board. The dashed line is a foil on the reverse side of the board. As shown, the switch is in the disk position. The other position returns to BASIC-in-ROM operation.

PEEK (65) is published monthly by DBMS, Inc.
Owings Mills, MD 21117.
Editor - Al Peabody
Technical Editor - Dickinson H. McGuire
Asst. 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

Floppy Disk Controller Modifications

The 470 board is the floppy disk controller. The same board is used for both 5 1/4 and 8 inch drives, but the signal timing is different, so be sure to specify when ordering that the board is for an 8" drive.

OSI uses a rather peculiar scheme for connecting the controller to the drive. The signal lines are terminated with Molex connectors, like those used on the backplane. A small auxiliary board then plugs into these connectors. This board has a paddle connector which mates with a paddle plug attached to a 50-conductor ribbon cable that goes to the drive. The easiest thing to do is to order the cable assembly and auxiliary board from OSI when you order the 470 board. Or, you can do what I did and make up your own cable.

The controller board has a prototype area on it that can be used to mount a header connector on. Then you can buy or make a "standard" floppy disk cable consisting of a header connector on one end and a card-edge connector on the other.

Only 13 signal lines are run to the drive. Figure 2 shows the connections between the controller and the drive. Note that a negative voltage is not required on the current model Siemens drives. Positive 5 volts can be supplied to the drive from a separate power supply (you need one anyway, since the drive also requires +24 volts for the stepper motor.) Be sure to ground all odd-numbered pins at the header connector on the controller board to prevent cross-talk between signal lines.

Disk Drive Modifications

The Siemens FDD 100-8 drive is a single-sided unit that can be optionally configured for hard or soft-sector, single or double-density operation. The OSI disk operating system requires a soft-sectored, single-density format. A FM Data Separator option must be installed on the drive for single-density operation. I was lucky that it was included on my drive, but anyone ordering this drive should verify before ordering that it will have this option INSTALLED; otherwise, you will need a scope to set a timing potentiometer. Also, be sure to

Willis Cook/Figure 2.

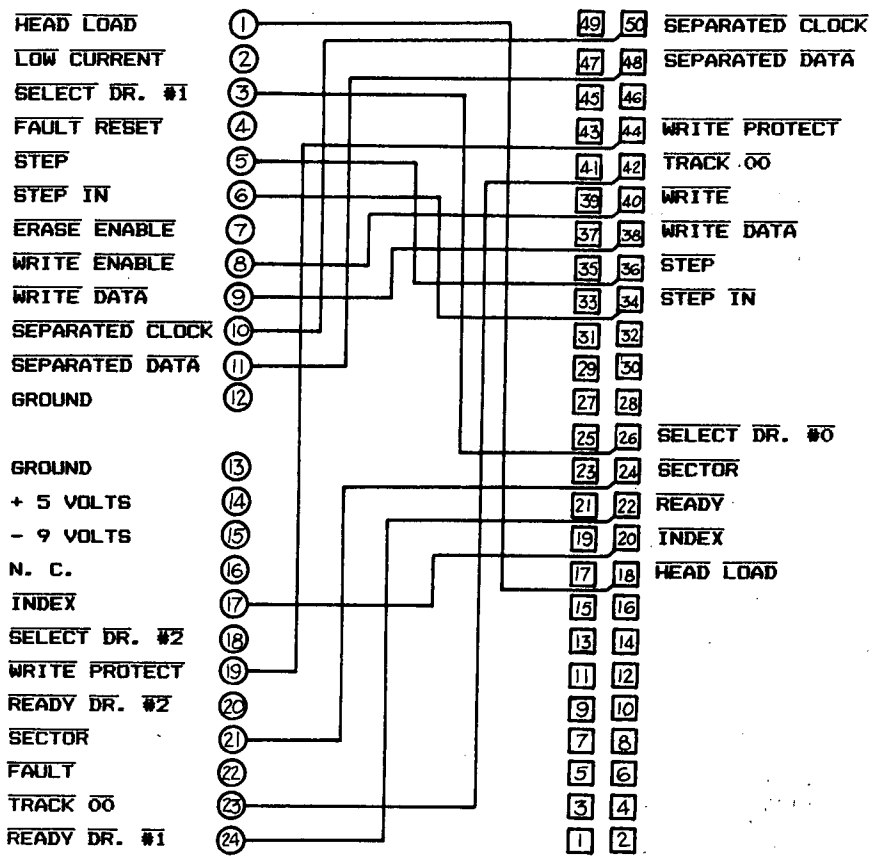


Figure 2. Interconnections between the Molex pin positions on the 470 Floppy Disk Controller board (left), and the card-edge connector on the Siemens drive. All signals are inverted, i.e., an active state is represented by 0 volts.

Willis Cook/Figure 3.

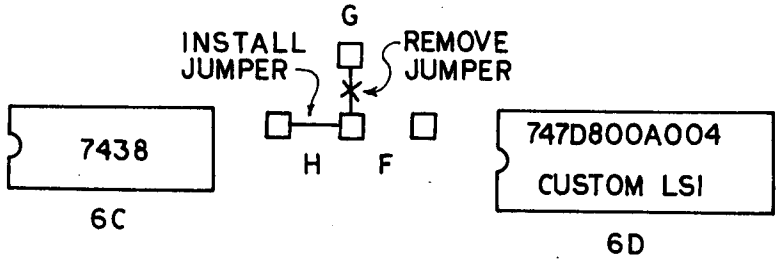


Figure 3. Modification of the Phase Option on the disk drive. This change allows the stepper motor to be energized whenever the drive is selected.

order the Siemens Installation and Maintenance manuals. They contain a complete parts breakdown and an excellent tutorial on the theory of drive operation.

Only one modification is required to the drive. Numerous operating options are selectable via jumpers on the drive logic board. One, called the Phase Option, controls the drive power to the stepper motor which positions the read/write head. As supplied, the stepper motor is energized only when the head is loaded. This won't work with OSI's operating system since the stepper motor is called upon to position the head over the proper track before the head load signal is sent. This is corrected by moving the Phase option jumper from the G pads to the H pads as shown in figure 3. This change allows the stepper motor to be energized whenever the drive is selected. With one drive, it is normally selected all the time, so the stepper motor is continuously energized and gets quite hot. I have a 3" Rotron fan installed in the drive enclosure and it appears to be sufficient to keep everything under control.

Disk Drive Enclosure

When you order a bare drive, that is what you get, so it is up to you to provide whatever case it will reside in. My enclosure consists of a base and back panel of 1/2" plywood with a cover made of lightweight sheet aluminum. The aluminum is available at hardware stores for use as a door kickplate.

The rear panel must have cutouts for the 50-conductor ribbon cable from the computer, a 115 volt AC cord for the drive motor, a +24 volt cable for the stepper motor and a +5 volt logic supply cable. Also, leave room for the fan and some air intake vents.

Disk Drive Power Supply

The Siemens drive requires 1.6 amps at 24 volts DC, 1.0 amp at 5 volts DC and 0.5 amp at 115 volts AC. Several manufacturers offer power supplies designed for one or two 8" drives, so I bought one readymade. As it turned out, my power supply had a bad power transistor in the +5 volt circuit that let 1.5 volt spikes through on top of the 5 volt DC level and burned up the LSI disk controller chip on the drive. Not only did it cost me $100.00 to get diagnosed and fixed, I wasted about two months trying to figure out what was wrong. So, my word of advice: Check the LOADED output of your power supply with a scope before connecting it to your equipment.

Disk Operating Systems

If OSI installs a disk drive on a cassette-based system, they furnish the disk operating system (OS 65-D) with it. If you home-brew a disk system, you must purchase the DOS separately.

OS 65-D is a simple, small, inexpensive system that I have found to be just right for my needs. It comes with two copies of the operating system on floppies and a reasonably complete 77 page manual. An Extended Monitor comes with the DOS. It is an extension of the Machine Code Monitor in ROM. But the Assembler is an optional extra. Be sure and order it if you anticipate doing any Assembly Language programming.

The operating system with all the utility programs (such as CREATE, DELETE and RENAME) requires 26 tracks on an 8" floppy, leaving 51 tracks free. This allows you to have the DOS and all utilities on every disk and still have approximately 160 k bytes of space free.

Conclusions

After about a year of operation I can say that the system has certainly been reliable. After a couple of shake-down clinchers, I haven't had any...no, I'd better not say it...it will break tomorrow!

It wasn't cheap, however. My total cost, including the drive repair and all incidentals, was $900. Still, it was cheaper than buying the double 8" drive system from OSI, and has considerably more capacity than a single 5 1/4" drive. The only problem with a single drive is in copying disks. After you get your system up and running, most copying is file by file and a single drive is no handicap, but in copying an entire disk, it is slow -- three or four minutes. Next month I will give a BASIC program for single 8" drive disk copying without too much hassle.

In disk copying as in a few other jobs, you will find that a single drive system makes you take care of the housekeeping chores that the C3 system running 65-U does automatically. But, hell, you are supposed to be in charge of the computer anyway, not the other way around.

★

CREATE DATE HIDES IN DIREC*

By Al Peabody

Reviewing old issues of PEEK(65) as I periodically do, I recently looked again at a couple of letters concerning the structure of the file directory under OS-65U and realized that there is room in the directory and header for more information. This short note describes one possible use, based on a request from a client; readers are encouraged to think of other uses and submit programs and suggestions.

The situation is this. Bytes 7 and 8 of the file header (created in the 65U program CREATE, then stuffed both into the file header, the first 16 bytes of every file, and into the entry in the directory file DIREC*) contain an encoded version of the file's password. However, if the file is assigned the access rights "R/W" (read/write), these bytes are never used again. The access rights are encoded in byte 9 of the same header, and if they are "R/W" the password is never checked.

Various letters and articles in PEEK(65) have pointed out how insecure the password check is. It is easy, using the program CHANGE, to change the access rights of any file, or to change the password of the file to some known password, meaning that the only real security for files is physical security of the disks which contain the files. Therefore, many people never use passwords on their files. This means these bytes are available for other uses.

The following routine, if included as a subroutine in the program CREATE, allows you to stuff the date of creation of the file into its directory entry:

Continued on page 7

At last!
Software Development TOOLS
for Professional OS-65U Programmers:

FIND:

If you program in OS-65U BASIC, you need FIND, a machine code overlay which resides permanently in the operating system, extending the FIND command to allow searches for variables, literals, statements, commands, functions, and constants such as line numbers.

FIND is an invaluable tool for writing code and debugging programs — especially someone else's! May be used in the immediate mode with any BASIC program in the user's workspace.

COPY & DELETE:

These utilities save you from spending hours manually copying and moving BASIC program code. Both reside in the operating system, allowing use in the immediate mode.

COPY copies program lines character-for-character to a new line number location. Tests to make sure no existing lines will be overwritten.

DELETE removes program lines. Any linerange may be specified, although the DELETE command without a linerange is not accepted (to prevent accidental erasure of a whole program).

Using a single COPY-DELETE command with a linerange performs a MOVE of the block of lines to a new location.

COPY & DELETE are available without EDITV3 for video-based systems.

EDITV3:

Has the usual OSI EDIT features, including Control R, F, P, and Tab, Delete, and Backspace. New features: Control D (erase from cursor to EOL), Auto Upper Case, Bell on All Illegal Characters, Auto -CR- at First-space-closest-to-EOL Flag, Masked Output Flag (prints X's instead of characters for password protection). Underscore and @ symbol are legal characters, replaced with DEL and Control X respectively. Backspace and Delete/Insert work normally. Control T now toggles Insert/Overstrike Character mode, allowing the user to overstrike characters in the middle of a line (without first deleting the old characters and then typing the new). Edit Line command deletes both first space and space between line number and statement, adding one character to editable lines.

Above flags may be set using the calling routine. The Input Editor may a'so be preloaded with a string to be edited, placing the cursor on the appropriate character within the line (for use in BASIC programs). EDITV3 with COPY & DELETE requires no reserved words.

MONITR:

Find out what is going on in your Level 3 system! This simple-to-use program allows you to monitor activity while Level 3 is running. MONITR shows which users are 'up,' which are active, what program they're using, the line number (if running), and their input status. Also enables the user to debug multi-user partitions and programs. Runs from any port, and allows any user to reboot any other user. Designed for use by programmers as well as system operators.

PRICES:

Each program package supplied on an 8-inch flexible disk.

Package 1: FIND $75.00
Package 2: COPY & DELETE (for video-based systems) $75.00
Package 3: FIND, EDITV3 incorporating COPY & DELETE (not for video-based systems) $235.00
Package 4: MONITR w/Talkie $175.00

Brochures available — write or call.

Brown/Collinson Associates
619 "I" Avenue
Lake Oswego, Oregon 97034
(503) 635-5055

Continued from page 4

```
505 PW$="":IF A=3 THEN GOSUB
    XX00:GOTO640:REM GET DATE
XX00 REM GET DATE
XX10  INPUT "DATE(YRMODA);DT$
XX20 REM ERROR CHECK DT$ HERE
     IF YOU LIKE---
XX30  YR$=LEFT$(DT$,2):
     MO$=MID$(DT$,3,2):
     DA$=RIGHT$(DT$,2)
XX40  YR=VAL(YR$):
     MO=VAL(MO$):
     DA=VAL(DA$)
XX50  YR=(YR-80)*16:
     REM MAKE YR HI NIBBLE
XX60  PL=YR+MO:REM PUT YR AND
     MO INTO LOBYTE OF PW
XX70  PH=DA:REM DA = HIBYTE
XX80  RETURN
```

Now the CREATE program will stick your date into the space reserved for the password. Then any program, such as DIR, which reads the directory, can have access to the date the file was created by the following simple routine:

```
XX10  YR=INT(PL/16)+80
XX20  MO=PL-YR*16
XX30  YR=YR+80
XX40  DA=PH
XX50  DT$=STR$(MO)+
      "/"+STR$(DA)+
      "/19"+STR$(YR)
```

...assuming, of course, that the lobyte and hibyte of the password have been read into the variables PL and PH by the program.

This is but a simple example of the kind of information which could be stowed away in the directory and file header. Byte 16 of the directory entry is called the "file header pointer" and apparently points to the last byte of the file header, but also doesn't seem to be used. It may also be available for information storage, though I would be a bit nervous about this until someone tells me what it is actually used for, or will be used for in the next release of OS-65U! It does not seem to be used at present to determine which bytes of a file are transparent to the user, since changing it to a larger number will not cause the system to lose its way when it attempts to load or save files; 16 bytes are always used as the header.

The possibilities seem endless. Please write us a letter if you think of something else to do with these "free bytes."

★ ★ ★

OS-DMX
AN ALTERNATE DBMS

by Al Peabody

Ohio Scientific's Data Base Management System (OS-DMS) has had its ups and downs. Many people use it; many others claim it can't be used in demanding applications such as accounting; still others complain about its simplistic approach to report generation or its inadequate manuals. However, until recently there has been little else to choose from.

Now there is another choice: Digital Technology's OS-DMX. After a couple of days of work with DMX full time, I am happy to report that it is indeed a viable alternative, with a great deal to recommend it.

But let's back up a little. Just what is a "Data Base Management System," anyway, and why would a computer user want one?

Every computer user has a data base. Quite simply, your data base is all the information (data) which is contained in your computer, in whatever form. Even if you only use your computer for games, it contains some data, perhaps in the form of DATA statements in programs you have bought or written. If you use your computer in business, you have a great deal more data -- customer names and adresses, balances due, dates of payments, records of orders, employee names and addresses, etc. It is this data which you must manipulate or manage as you use your computer. Your programs and data files, probably on disks, manage your data base for you, adding things together, redistributing information, generating reports, and so forth.

So, particularly if you are a small business person, you are managing a data base with your computer right now.

A Data Base Management System is designed to do the job simply and flexibly. To this end, the system must have a number of features:

a general file structure, standardized to allow various programs to "find their way around in" the data;

programs which allow you to put data into the files;

programs which allow you to edit files already containing data;

programs allowing you to sort files, rearranging the data in a number of different ways;

programs which can generate reports from the files;

and a system of instructions to teach you how to use the system.

This last item, the instruction system, is perhaps the most important of all: no matter how clever a DBMS is, if you can't figure out how to use it, its utility is, to say the least, limited.

Every DBMS has what might be called a "philosophy," whether explicitly stated or not. As the system is designed, all sorts of questions must be answered and decisions must be made, and the philosophy of the DBMS helps to answer them. Questions like, "shall we assume the user is intelligent or rather dull?" "how consistent shall the 'feel' of the system be?" "How shall we distribute the instructions?" and many others.

Let's have a look at the philosophy manifested in the design of the two Data Base Management Systems available to OSI computer users: OS-DMS and DMX.

OS-DMS assumes the user will be rather dull, at least at first, and uses a Menu and Question system to input information into the computer. All your choices are spelled out on the screen in the form of Menu selections and questions. To generate a statistical report from a data file, for example, you must answer questions such as "What fields will be included in the report?" "Horizontal (H) or Vertical (V) format?" "How many conditions do you want to set (0-4)", etc., etc.

This sort of questioning makes generation of a report rather easy, even for a beginner, but becomes quite tiresome after experience is gained, when the endless questions are no longer needed.

DMX, on the other hand, assumes that the user will be rather intelligent and sophisticated. Rather than presenting many menus and asking many questions, DMX uses a "command language." To include the name in a report, you type INCLUDE NAME; to total up the amount due you type TOTAL AMOUNT; to see which fields are to be included in the report, you type STATUS, and the system shows you; and so forth.

Of course, the question immediately arises, "how do I know the commands I can use?" Good question, with a pretty good answer. DMX comes with a large manual (it weighs over a pound) explaining all the commands and giving some examples. Also, virtually the entire manual is on the disk, in the form of "user memos." At any time when you are confused, just type HELP, and DMX will ask you "Help on what subject?" About the only "subject" you have to remember is COMMANDS. When asked what subject, just type in COMMANDS and you will be shown a list of the available commands. Type in one of the commands, and you will be told how to use the command and what it does. Then you can enter the command, and continue.

What it feels like, is this. At first it is quite confusing. You start up, and get just one menu. It is easy enough to select what you want to do: create a file, input data to a file, edit a file which already has data in it, generate a report, etc. But when you select your option, you get a display like:

Inpt Cmd:

and must figure out what your command is. Invariably, at first,mine was HELP, followed by COMMANDS. Then pretty soon, I learned a few of the simpler commands, and stopped typing HELP so much. By the end of the day, I was a real "pro," typing in commands one after the other and occasionally typing PRINT or STATUS to see where I was. It didn't take me too long in the report program to learn that I could type OUT P followed by PRINT to generate a report to the printer, or just PRINT to have a look at the same report on the screen. Then I could modify the report, in a totally interactive process with the DBMS, before finally printing it to the printer for the final "hard copy."

I could say a great deal more about DMX: it uses standard OS-DMS files, so you can use it with files already created under DMS; it allows you to generate "submit" files of commands, so that reports which must be run repeatedly, for example, can be run by just typing RUN MONTHLY or whatever you have named the file; all the programs use

very rigid and useful programming conventions, with the same line numbers in different programs doing the same things, making it easy for the BASIC programmer to work with them; the manual explains everything, including how to write your own extension programs if you want to, using machine-language routines which are supplied for screen formatting and other goodies. In short, the authors of DMX seem to really care about the user, and to want to help.

Perhaps the nicest thing of all is the price. The list price of the complete DMX system is $1199. You wouldn't have to replace too many DMS modules to pay for that.

Does DMX have no faults? Of course it has some. It is just hard to see them with the tears of joy filling my eyes after reading the manual and realizing how easy it would be to generate powerful programs of my own, using OSI's excellent data file conventions and DMX's terrific programming goodies. For one thing, even though a great deal of HELP is on line, the manual is short on examples. However, Digital Technology's Chuck Sulka assures me that an extensive (hundreds of pages) applications manual with many, many examples is on the way; and in the past, when Chuck has said something was coming, it has followed quickly.

Also, DMX uses a BASIC sort program, which is just too slow. Chuck says a machine language sort is on the way as well, but doesn't promise it as quickly as the applications manual. DMX will be better when machine language sorting is implemented.

All in all, right now DMX is a good buy at the price for the user who must make frequent and extensive use of his data, and who doesn't want or need the very specialized accounting, inventory and other programs produced by Digital Technology and several others.

* * *

GETTING THERE WITH TTL

by Bruce Showalter
857 Cedar
Abilene, TX 79601

Have you ever looked at your computer's innards and wished you could brew a circuit to get some of its wizardry to do something else? Or have you ever wondered how it does things to begin with? For example, how does that 40-pin IC called the CPU select just the right RAM or ROM chips, or the cassette port, or whatever?

Well, a lot of its power lies in software (stored in RAM) and firmware (stored in ROM). But without some hardware to deal with, those instructions end up as little more than theory. So, let's take a look at digital logic and how it works.

First, let's set aside the mathematics of the issue. What I'm dealing with is basic electricity: volts and amps. A TTL compatible device obeys the following rules: +.8 volts or less is LOW or 0; +2.4 volts or more is HIGH or 1. A TTL device never wants to see a negative voltage, or a positive voltage higher than 5.5 (although certain devices can tolerate higher voltage on the output only).

All the 7400 series TTL obey these rules, as do the 74LS series. So, what are the differences, besides code numbers? The difference is in current levels at both the input and output. Regular TTL draws .0016 amps (1.6ma) on each input in the LOW state. Each output can provide .016 amps (16ma) in the LOW state. So you see, each regular TTL output can drive ten regular inputs in the LOW state. Inputs in the HIGH state have almost no current drain. But since the state of the output must go LOW sometime, we don't want to overload it with more than ten inputs to drive.

74LS devices draw .0008 amps (.8ma) per input and supply .008 amps (8ma) per output. A 74LS output can drive ten 74LS inputs or five regular inputs. As you can see, regular TTL can drive 20 LS devices. This is because regular is twice as powerful as LS. By the way, LS stands for Low-power Schottky. A Schottky device is one that uses faster diodes in the IC.

To put the concept of "drive" into more tangible terms, think of these devices as toggle switches. Say your hand has enough strength to flip a bank of ten toggle switches from ON to OFF, all at once. This is what's known as fan-out. The "fanout" of your hand is ten toggle switches. Add an eleventh switch, and you can't quite flip them, they all stay ON. If you keep trying, you sprain your hand.

Of course, other kinds of devices have different ratings. But these are the standards by which others are measured. For example, each input to an S-100 bus circuit board should not exceed the drain of one LS input.

Now let's get down to actual devices. For starters, I choose the 7404. It contains six inverters. Each one takes a TTL input and makes it the opposite. If the input is HIGH, the output will be LOW (and vice-versa). It's like a light switch that you push UP to turn the light OFF, and DOWN to turn ON. It is drawn like fig. 1. Whenever you see a small circle on the input or output of a symbol, that means INVERT. Two inverters in series produce a TRUE output (same as the original input) (fig. 2). This serves as a buffer. That original input can now drive ten more outputs.

Next, consider the OR gate. A 7432 has four of them, each with two inputs. If either input is HIGH, the output is HIGH. Both inputs must be LOW for the output to be LOW. The OR gate is usually drawn like fig. 3 but could be drawn like fig. 4. The latter symbol is an AND gate with inverted output and inputs. You could think of it as in fig. 5.

How about the AND gate? There are four of them in a 7408 with two inputs apiece. If both inputs are HIGH, the output will be HIGH. The output will be LOW if either input is LOW. It is usually drawn as fig. 6. Other ways to draw it are in fig. 7.

The NAND gate is an AND gate with its output inverted. If both inputs are HIGH, the output is LOW. If either input is LOW, the output is HIGH. The 7400 has four. Fig. 8.

Then there's the NOR gate, an OR gate with inverted output. Either input HIGH makes the output LOW. Both inputs must be LOW for the output to be HIGH. Four NOR gates are housed in a 7402. (fig. 9).

Lastly, there are the EXclusive OR and NOR gates, symbolized like so: (fig. 10). The former, type 7486, will provide a HIGH only when its inputs are different. If the inputs are both the same, the output will be LOW. The latter does the same thing, except the output is inverted.

Type 74266 has four EX NOR gates, but they do not go HIGH without pull-up resistors connected from their outputs to +5 volts.

There are other arrangements of all these gates, with three, four, even eight inputs. They are the building blocks of digital logic. From them come flip-flops, counters, shift registers, adders, memories, encoders and decoders.

In your computer, these HIGH and LOW signals are all over the place. When combined and sequenced properly, they turn into video games, financial reports, or whatever. These HIGHs and LOWs are called bits. When eight of them are employed simultaneously, the result is a byte. Each byte represents an ASCII character, a CPU instruction, or other data. RAM and ROM chips store these bytes for use by the system.

But how does the CPU call up these bytes and send them to the right place? This is done by addressing. An address is a 16-bit word. There are 65,536 possible different 16-bit words, and each one is an address for the CPU to use. By "reading" the address lines coming from the CPU, a RAM or ROM can tell if it is being called upon, just like you know to pick up your telephone when it rings.

Consider the 2114 RAM chip used in many OSI Challengers. Each chip can store 4,096 bits, arranged as 1,024 words of four bits each (a four-bit word is a nibble). By wiring two 2114s in a pair, 1,024 bytes can be stored. But 1,024 locations will only use up ten address lines. What do we do with the other six? We use them to choose between the different pairs of RAM chips, ROM chips, or other data locations.

Each RAM or ROM has a Chip Select or Chip Enable. This single input is like an on-off switch. When the appropriate input is applied, the chip is activated. By using suitable decoders, we can take those remaining six address bits and use them to turn on the right chips.

Decoder chips even have Enable inputs, so that they will only decode addresses when they are on. So, bits 14-16 are decoded by X, thus turning on decoder Y. In turn, decoder Y decodes bits 11-13 to turn on RAMs 7 and 8, for example.

But suppose you have a data location - say, an output port - that has only one Enable input and a specific 16-bit address? You could end up using five ICs just to decode that one address. Finding economical ways to decode a single 16-bit word is an interesting study. Let's examine some simpler decoders first, and work our way up to the 16-bit decoder.

EXAMPLE 1: 3-bit address. Most designs call for a 74LS138 decoder. However, a 7442 decoder can be employed. It costs less, and the bit 4 input will serve as an Enable. When bit 4 goes HIGH, all outputs below eight will go HIGH. A 7442 also has a greater fanout capacity than the LS138.

EXAMPLE 2: 4-bit address. The obvious solution is a 74154 decoder. This is required if the address will exceed A16 or 10102. If not, a 7442 decoder will work. The 7442 is preferred because it takes less PC board space and costs less.

EXAMPLE 3: 5-bit to 8-bit address. The most efficient design employs a 7430 (eight bit NAND gate). This serves if all bits are HIGH. Otherwise, some inverters are required. If there are fewer then seven bits LOW, a 2-chip

Continued on page 15.

OSI AARDVARK OSI

NOW MEANS BUSINESS!

WORD PROCESSING THE EASY WAY— WITH MAXI-PROS

This is a line-oriented word processor designed for the office that doesn't want to send every new girl out for training in how to type a letter.

It has automatic right and left margin justification and lets you vary the width and margins during printing. It has automatic pagination and automatic page numbering. It will print any text single, double or triple spaced and has text centering commands. It will make any number of multiple copies or chain files together to print an entire disk of data at one time.

MAXI-PROS has both global and line edit capability and the polled keyboard versions contain a corrected keyboard routine that make the OSI keyboard decode as a standard typewriter keyboard.

MAXI-PROS also has sophisticated file capabilities. It can access a file for names and addresses, stop for inputs, and print form letters. It has file merging capabilities so that it can store and combine paragraphs and pages in any order.

Best of all, it is in BASIC (OS65D 51/4" or 8" disk) so that it can be easily adapted to any printer or printing job and so that it can be sold for a measly price.

MAXI-PROS – $39.95

NEW-NEW—NEW TINY COMPILER

The easy way to speed in your programs. The tiny compiler lets you write and debug your program in Basic and then automatically compiles a Machine Code version that runs from 50-150 times faster. The tiny compiler generates relocatable, native, transportable machine code that can be run on any 6502 system.

It does have some limitations. It is memory hungry – 8K is the minimum sized system that can run the Compiler. It also handles only a limited subset of Basic – about 20 keywords including FOR, NEXT, IF THEN, GOSUB, GOTO, RETURN, END, STOP, USR(X), PEEK, POKE, -,=,*,/, (,), <>, Variable names A-Z, and Integer Numbers from 0-64K.

TINY COMPILER is written in Basic. It can be modified and augmented by the user. It comes with a 20 page manual.

TINY COMPILER – $19.95 on tape or disk

THE AARDVARK JOURNAL

FOR OSI USERS – This is a bi-monthly tutorial journal running only articles about OSI systems. Every issue contains programs customized for OSI, tutorials on how to use and modify the system, and reviews of OSI related products. In the last two years we have run articles like these!

1) A tutorial on Machine Code for BASIC programmers.
2) Complete listings of two word processors for BASIC IN ROM machines.
3) Moving the Directory off track 12.
4) Listings for 20 game programs for the OSI.
5) How to write high speed BASIC – and lots more –

Vol. 1 (1980) 6 back issues - $9.00
Vol. 2 (1981) 2 back issues and subscription for 4 additional issues - $9.00.

ACCOUNTS RECEIVABLE – This program will handle up to 420 open accounts. It will age accounts, print invoices (including payment reminders) and give account totals. It can add automatic interest charges and warnings on late accounts, and can automatically provide and calculate volume discounts.

24K and OS65D required, dual disks recommended. Specify system.
Accounts Receivable. $99.95

*** * * SPECIAL DEAL – NO LESS! * * ***

A complete business package for OSI small systems – (C1, C2, C4 or C8). Includes MAXI-PROS, GENERAL LEDGER, INVENTORY, PAYROLL AND ACCOUNTS RECEIVABLE – ALL THE PROGRAMS THE SMALL BUSINESS MAN NEEDS. $299.95

P.S. We're so confident of the quality of these programs that the documentation contains the programmer's home phone number!

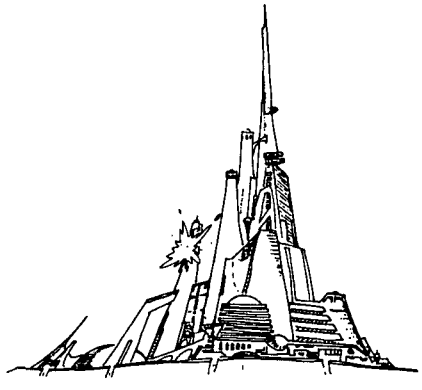
SUPERDISK II

This disk contains a new BEXEC* that boots up with a numbered directory and which allows creation, deletion and renaming of files without calling other programs. It also contains a slight modification to BASIC to allow 14 character file names.

The disk contains a disk manager that contains a disk packer, a hex/dec calculator and several other utilities.

It also has a full screen editor (in machine code on C2P/C4)) that makes corrections a snap. We'll also toss in renumbering and program search programs – and sell the whole thing for – SUPERDISK II $29.95 (5 1/4") $34.95 (8").

AND FUN, TOO!



BOOKKEEPING THE EASY WAY – WITH BUSINESS I

Our business package 1 is a set of programs designed for the small businessman who does not have and does not need a full time accountant on his payroll.

This package is built around a **GENERAL LEDGER** program which records all transactions and which provides monthly, quarterly, annual, and year-to-date PROFIT AND LOSS statements. GENERAL LEDGER also provides for cash account balancing, provides a BALANCE SHEET and has modules for DEPRECIATION and LOAN ACCOUNT computation.
GENERAL LEDGER (and MODULES) $129.95.

PAYROLL is designed to interface with the GENERAL LEDGER. It will handle annual records on 30 employees with as many as 6 deductions per employee.
PAYROLL - $49.95.

INVENTORY is also designed to interface with the general ledger. This one will provide instant information on suppliers, initial cost and current value of your inventory. It also keeps track of the order points and date of last shipment.
INVENTORY - $59.95.

GAMES FOR ALL SYSTEMS

GALAXIAN - 4K - One of the fastest and finest arcade games ever written for the OSI, this one features rows of hard-hitting evasive dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. Specify system – A bargain at $9.95

NEW – NEW – NEW

LABYRINTH - 8K - This has a display background similar to MINOS as the action takes place in a realistic maze seen from ground level. This is, however, a real time monster hunt as you track down and shoot mobile monsters on foot. Checking out and testing this one was the most fun I've had in years! -- $13.95.

NIGHT RIDER - You've seen similar games in the arcades. You see a winding twisting road ahead as you try to make time and stay on the road. NIGHT RIDER uses machine code to generate excellent high speed graphics - by the same author as MINOS.
NIGHT RIDER – $12.95 cassette only

THIEF - Another machine code goody for the C1P cassette only. You must use mobile cannon to protect the valuable jewels in the middle of the screen from increasingly nasty and trigger happy thiefs. Fast action and fun for one or two players. THIEF $13.95 on C1 cassette only!

SUPPORT ROMS FOR BASIC IN ROM MACHINES – C1S/C2S. This ROM adds line edit functions, software selectable scroll windows, bell support, choice of OSI or standard keyboard routines, two callable screen clears, and software support for 32-64 characters per line video. Has one character command to switch model 2 C1P from 24 to 48 character line. When installed in C2 or C4 (C2S) requires installation of additional chip. C1P requires only a jumper change. – $39.95
C1E/C2E similar to above but with extended machine code monitor. – $59.95

Please specify system on all orders

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

VISA

OSI

AARDVARK TECHNICAL SERVICES, LTD.
2352 S. Commerce, Walled Lake, MI 48088
(313) 669-3110

OSI

PEEK(65) INDEX FOR 1981

This is our second index, and a few changes have been made. First of all, it is published in the December issue, rather than waiting for the January issue of the following year.

Also, by popular request, we make no distinction between questions and statements, since questions are often very informative in their own right.

Finally, we have intentionally printed this year's index on the inside pages of the magazine, and have backed up these pages with ads, so that readers wishing to pull the index out and place it with last year's index will not lose any parts of articles or letters.

We have carefully included a reference to all board type numbers and operating systems used in a "second" entry for each article or letter. If you are only interested in 65D, for example, you will find all entries dedicated to 65D together under the 65D heading.

USER GROUP NOTES:

SOUTHERN ONTARIO REGION

Readers of PEEK (65) in the Southern Ontario region might be interested in the Toronto Ohio Scientific Idea Exchange (TOSIE). The group is now five months old and has 37 members.

We meet on the last Saturday afternoon of the month to discuss hardware, software, bugs and projects. The major project of the club at the moment is to produce a new monitor PROM for the superboard. The new monitor is almost finished and will be made available to members at cost.

Membership is $5 a year and $2 at the door to pay for the meeting room rental. Membership includes a subscription to the TOSIE Printout and a 10% discount at Exceltronix. For more information, contact: Crispin Cowan, Newsletter Editor, 665 Oriole Pkwy, Toronto, Ontario, M4R 2C1, (416) 488-4584 or David Cho, Secretary Treasurer, 10 Macey Ave., Apt. 1507, Scarborough, Ontario, M1L 4R4, (416) 698-0032

* * * * * * * * * * * * * * * *

Z-FORTH IN ROM by Tom Zimmer
5 to 10 times faster than Basic. Once you use it, you'll never go back to BASIC! $ 75.00
source listing add $ 20.00

OSI FIG-FORTH True fig FORTH model for 0S65D with fig editor named files, string package & much more $ 45.00

TINY PASCAL Operates in fig-FORTH, an exceptional value when purchased with forth.
TINY PASCAL & documentation $ 45.00
FORTH & TINY PASCAL $ 65.00

SPACE INVADERS 100% machine code for all systems with 64 chr. video. Full color & sound on C2, 4P & 8P systems. The fastest arcade program available. $ 14.95

PROGRAMMABLE CHARACTER GENERATOR $ 99.95
Use OSI's graphics or make a complete set of your own! Easy to use, comes assembled & tested.
2 Mhz. boards $109.95

PROGRAMMABLE SOUND BOARD $ 74.95
Complete sound system featuring the AY-3-8910 sound chip.
Bare boards available. $ 29.95

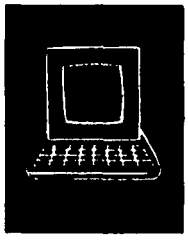
32/64 CHARACTER VIDEO MODIFICATION $ 39.95
Oldest and most popular video mod. True 32 chr. C1P, or 32/64 chr. C4P video display. Also adds many other options.

ROMS!!!
Augment Video Mod with our Roms. Full screen editing, print at,selectable scroll, disk support and many more features.
Basic 4 & Monitor $ 49.95
Basic 3 $ 18.95
All 3 for $ 65.00

65D DISASSEMBLY MANUAL. by Software Consultants
First class throughout. A must for any 65D user. $ 24.95

NUMEROUS BASIC PROGRAMS, UTILITY PROGRAMS AND GAMES ALONG WITH HARDWARE PROJECTS. ALL PRICES ARE U S FUNDS. Send for our $1.50 catalogue with free program (hardcopy) Memory Map and Auto Load Routine.

CHARGEX VISA MasterCharge



OSI Software & Hardware

3336 Avondale Court
Windsor, Ontario, Canada N9E 1X6
(519) 969-2500

3281 Countryside Circle
Pontiac Township, Michigan 48057
(313) 373-0468

progressive computing

Continued from page 10.

solution is possible with one 7430 and one 7404. Otherwise, the better choice would be two 7485s in cascade. A 7485 compares two nibbles for equality. However, the address will result in a HIGH output, rather than a LOW. If a LOW output is required, a 7430 and one or two 7404s would be cheaper than two 7485s and a 7404.

EXAMPLE 4: 9-bit to 16-bit address. The choices here resemble those of the previous example. If less than three bits are LOW and the output must be LOW, two 7430s and a 7402 are preferred. Two of the NOR gates are used to combine and invert the outputs of the 7430s. The two remaining NOR gates can serve as inverters for any two input bits. If the output must be HIGH and less than four bits LOW, the 7430s and 7402 are still first choice. Otherwise, three or four 7485s in cascade would be more efficient.

There is also the alternative of cascading 74154s. This turns out to be expensive, both in terms of dollars and PC board space.

When designing a circuit, it is helpful to consider all combinations of gates available in different ICs. If a decoder design is part of a larger circuit, some unused gates from one can be used in the other. Also remember that it may be more economical to combine gates and inverters for specific tasks rather than to buy a specialized IC. Occasionally, those specialized devices don't even exist.

Right now, let me explain another type of logic condition: tri-state. A tri-state output can be HIGH or LOW as usual, or it may be neither. Such an output seem to disappear, as far as other inputs are concerned. It is as if we disconnect the output from the circuit. With this trick, we can hook up two or more outputs to one input. Only one output is ON at a time.

This is what happens when a RAM or ROM is disabled. Its outputs are cut off from the circuit. So, no matter what information appears on its address lines, the chip never communicates any data. That's how memory chips can share a common data bus.

A tri-state buffer looks like this (fig. 11). When the Enable input is HIGH, the output will either be HIGH or LOW, depending on the input. Drawn like this (fig. 12), it means that the Enable input must be LOW to activate the output. These buffers come four to a chip in the 74125 and 74126, and six each in the 74365 and 74367. Tri-state inverters (fig. 13) are found in the 74366 and 74368.

Now as you may recall, your computer's CPU handles bytes, which are eight bits at a time. So, a single IC with only six buffers won't do the job. Two ICs are required. But that leaves you with four unused devices left over. Seems like a waste to me. It also seems like a waste to use an entire 7400 when I only need one more NAND gate, or an entire 7404 when I only need one more inverter.

And why hasn't anybody made any gates with bi-level inputs, such as (fig. 14)? Well, here's my list of useful but non existent devices ...

A four-pin IC with one inverter (saves wasting a whole 7404).

Four 6-pin ICs: one AND gate, one OR gate, one NAND gate, one NOR gate (saves wasting a 7400, etc.).

An 18-pin IC containing eight buffers, featuring LS inputs and regular outputs (to boost the drive of the data lines).

An 18-pin IC with eight inverters (as opposed to six).

Four 6-pin ICs (1 AND, 1 OR, 1 NAND, 1 NOR), each with one active-HIGH and one active-LOW input (to save having to invert one input).

A 40-pin IC containing 16 tri-state buffers, each with a fanout of 20 (to interface the address bus).

A 16-line to 4-line priority encoder in a 22-pin IC (to convert one of 16 keystrokes into a binary nibble).

A 1,096 by 8 bit static RAM in a 22-pin IC (as opposed to two 1,096 by 4 versions).

A programmable decoder, giving one active-HIGH and one active-LOW output in response to a 16-bit input, in a 22-pin IC (to activate a single device from a specific address).

I have specified 22-pin ICs where suitable because they are .2" narrower and .25" shorter than a 24-pin IC, thus saving PC board space. If any of you chip makers in Silicon Valley are interested, I can give you more specs, and even series type numbers, for any of the above.

For those of you who wish to begin applying some of the ideas in this article, I recommend you obtain a copy of National Semiconductor's TTL DATABOOK and/or Don Lancaster's TTL COOKBOOK. This article is built from information contained in those two books.

FIG 1

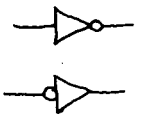


FIG 2

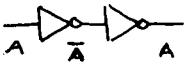


FIG 3

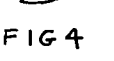


FIG 4

FIG 5

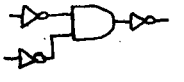


FIG 6

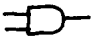


FIG 7

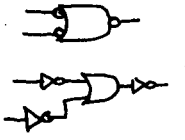


FIG 8

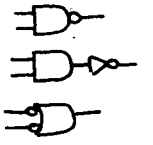


FIG 9

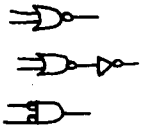


FIG 10

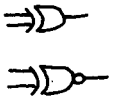


FIG 11

FIG 12

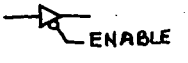


FIG 13

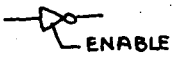
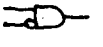


FIG 14



Editors Note:

Some of your useful but non-existent devices do exist.

Such as 18 IC containing eight buffers is called an Octal Bus Transceiver with outputs capable of sinking 100 ma or more.

OSI uses a 1K X 8 static RAM on their 48K static RAM board. There is also a 2K X 8 static ram that is plug compatible with the 2716 EPROM.

For other applications such as special decoders there are things called programmable logic arrays.

Brian

★

LETTERS

ED:

I am sending you some code and some instructions relating to the transfer of files between OS-65U and the WP-2 environments. I believe the instructions are quite straightforward. The problem as previously set forth in PEEK (65) is that the control symbols for the indirect file procedure in 65U are different than the ones used by WP-2. The POKEs to WP-2, as set forth in the following instructions for transferring WP-2 files to 65U, change the beginning of file symbol (for WP-2) to '<ESC> [' and the end of file symbol to ']'. Note that ONLY the end of file symbol used by WP-2 conforms to the end of file symbol used by OS-65U.

The cleanest and safest way to use these routines (I have blown my entire hard disk system a couple of times) is NOT to use them on the hard disk; but instead, utilize a special OS-65U diskette from which and to which the files are transferred. This disk should contain (among with DIREC*) a SCRATC file of 24000 bytes and the stock OSI version of BEXEC* which should be resequenced with line numbers in multiples of 10 (to allow the WP-2 transfer (which should also be in multiples of 10), to override the residue of BEXEC* left in memory when the OS65U diskette is booted). Line 360 and lines 400 through 500 of the following listing should be added. (Lines 480 and 490 are not really needed -- they are a part of the stock BEXEC*).

OS65U -- WP-2 TRANSFER ROUTINES

WPTRA1

#I: ROUTINE FOR TRANSFERRING OS65-U FILES TO WP-2

BOOT 'OS-65U -- WP-2 TRANSFER' DISK (this is a special disk -- see COMMENTS at end).
After 'Messages':

1 a.) Load file to be transferred.
b.) LIST{line specs} [<CR> (the line specs are optional)

When listing is complete enter:

c.)] (which displays ']]')
d.) <CR> (which displays '?SN ERROR')
e.) Entering <CTRL> X X will display the file stored in Indirect Memory and assure that the transfer is ok.

Now BOOT the WP-2 Disk and enter:

2.)
```
WH9000
EXXX
A*}BA

        OSI 9 DIGIT BASIC }
COPYRIGHT 1977 BY MICROSOFT } appears on screen
------ BYTES FREE          }
                           }
OK                         }
```

Continue by entering:

3.)
```
POKE 9550,27 (chr$ for '<ESC>
[')
POKE 9571,93 (chr$ for ']')
DISK!"ASM"   (see NOTE)

{Each input followed by (CR).}
```

The command mode symbol '.' should now be on the screen.

4.) <CTRL> X which will list the file into WP-2 memory.

If the display does not stabilize then too much data has been moved. Rerun move less data -- try not to move more than about 10k (OS65U) or 4pages (WP-2) at one time.

5.) !PU SCRATC
The file can now be manipulated in WP-2.

#II ROUTINE FOR TRANSFERRING WP-2 FILES TO OS-65U

Prior to transferring a file from the WP-2 environment to the OS-65U environment it is necessary to prepare the file. All ']' should be changed to some other symbol (']' is used as the end of file marker and if the file text contains ']' the listing will stop at that point). All other changes should be completed before starting the transfer routine.

1.) Boot WP-2 Disk or can be used when in the command mode.
2.) Enter:
```
WH9000
EXXX
A*}BA

        OSI 9 DIGIT BASIC }
COPYRIGHT 1977 BY MICROSOFT } appears on screen
------ BYTES FREE          }
                           }
OK                         }
```

```
POKE 9550,27  (chr$ for '<ESC>
[')
POKE 9571,93  (chr$ for ']')
```

```
                                        DISK!"ASM"    (see NOTE)

                                    {Each input followed by (CR).}
3.)  The command mode symbol '.' should now be on the screen.
4.)  Load the file to be transferred.
5.)  Enter:                  P <ESC> <CR> which will display
                                      'P[' and the file will
                                      be listed to memory
                                      starting at $9000.
6.) When the listing is complete ENTER:
                             ] <CTRL> ] <CR> which displays ']'.

7.) Entering <CTRL> X will display the file from Indirect
    Memory.
8.) BOOT 'OS-65U -- WP-2 TRANSFER' DISK and after 'Messages'
9.) Enter:                 <CTRL> X X <CR>  which will load
                                      the file into working
                                      memory.
                                      Note that X is depressed
                                      twice!  and NOT hard as
                                      suggested in a prior PEEK
                                      (65)
10.)                                  SAVE"SCRATC"  The WP-2
                                      file is now in OS-65U
                                      and can be treated
                                      accordingly.
```

After the POKEs the WP-2 system is inoperable. The file transfer should be made at this point. Any lines in a WP-2 file longer than 71 will be truncated by BASIC as the transfer is being made.

```
10 REM *** BASIC EXECUTIVE ***
20 REM *NET 6/79
30 IF PEEK(13316)=1 THEN POKE 13404,44: REM OMIT CD-23 FAULT
   CLEAR
40 D$="74": IF PEEK(13316)=1 THEN D$="23"
50 REM
60 REM SETUP CONSOLE INPUT DEVICE
70 I% = PEEK (11664)
80 REM
90 REM LOCKUP SYSTEM
100 FLAG 21: REM INPUT ESCAPE
110 POKE 2073,96: POKE 14639,0: REM CTRL C,O
120 POKE 11668,2^(I%-1)
130 REM
140 REM SETUP CONSOLE OUTPUT DEVICE
150 J% = PEEK (11665)
160 POKE 11686,2^(J%-1)
170 REM
180 PRINT: PRINT
190 FR=INT(10*PEEK(11946)/56+.5)/10
200 PRINT "OS-65U V1.2 -";FR;"MHz"
210 PRINT "1/22/80"
220 REM
230 REM SETUP FOR MASTER HARD DISK SYSTEM
240 IF PEEK(9832) < 8 THEN POKE 61438,0: POKE 61439,0
250 REM
260 REM GET USER FUNCTION
270 GOTO 360
280 REM NETWORK SUPPORT PARTITION USER # *NET
290 NS = 16
300 IF (PEEK(14948)=76)AND(PEEK(55381)=NS) GOTO 360:REM *NET
310 PRINT: INPUT "FUNCTION "; FU$
320 IF FU$ = "DIR" THEN RUN "DIR"
330 IF FU$ = "PDIR" THEN RUN "DIR",60
340 IF FU$ <> "UNLOCK" GOTO 450
350 REM
360 REM UNLOCK SYSTEM
370 REM
380 FLAG 22: REM INPUT ESCAPE
390 POKE 14639,255: POKE 2073,76: REM CTRL C,O
400 REM .... Enable Indirect Files
410 POKE 14721,24: REM .... CNTRL-X
420 POKE 14646,91: REM .... '['
```

```
430 POKE 14677,93: POKE 14688,93: REM .... ']'
440 POKE 11667,144: REM .... MEMORY (9000 HEX)
450 PRINT: PRINT "SYSTEM OPEN
452 PRINT
460 PRINT"INDIRECT FILES ENABLED FOR USE WITH WP-2 WITH HIGH ";
462 PRINT"MEMORY SET AT PAGE 144
463 PRINTTAB(46)"(9000 HEX)
464 PRINT
468 PRINT"ENTER '<CTRL> X X <CR>' TO TRANSFER FILE FROM HIGH";
470 PRINT"MEMORY TO LOW MEMORY
472 PRINTTAB(61)"(6000 HEX)
474 PRINT
480 PRINT: PRINT PEEK(132)+PEEK(133)*256-24576; "BYTES FREE":
    PRINT
490 IF (PEEK(14948)=76) AND (PEEK(55381)=NS) THEN RUN"NETWRK":
    REM *NET
500 NEW
```

Paul Loughridge, Jr.
Kamuela, HI 96743

★ ★ ★

ED:

Thanks for publishing PEEK (65). I have subscribed for 15 months and my level of understanding has about doubled from 5% to 10%. I think I am like a lot of your readers: I do not want to program in Assembly language but need to use some of the POKEs and X=USR(X). I have read both Carlson's manual (1st edition) and Software Consultants' Disassembly Manual. OSI would have been much more successful if they had written similar manuals several years ago and included them as part of the equipment package. Still there is a great need for an expanded version of these manuals. I know a few applications of PEEK, POKE, and USR(X) which I have never seen published anywhere but got them from the OSI dealer.

The college were I am teaching bought a C2/4P and C2/8P several years ago on my recommendation. The C2/4P has been used fairly well (no problems). The C2/8P (with 8" disk) was used by me only for several weeks. The entire time was consumed in writing appropriate software, which still is not available anywhere. Just when the writing was finished, the computer quit. We are 100 miles from the dealer. He came in, worked on it, took it to the shop, shipped it to Ohio, and returned it 4 months later. After a few weeks use, it quit again. Five months later it was returned in working order. Subsequently, I experienced some problems with software which were "fixed" by getting a new disk. Now I will speculate on the cause of these problems.

The first failure was due to a blown condenser. Subsequent failures I believe were all software problems. We, like virtually everyone else, have severe problems with voltage spikes, momentary low voltage or no voltage, and longer term high or low voltage. The college has ordered a large voltage regulator for all computer equipment. When the regulator arrives I will try again to use the C2/8P. I believe that every time the voltage dropped or rose above a certain level, random errors were written on the disk. The next time the disk was used, the errors were either not noticed, minor problems, or fatal to the disk. Does anyone have any real factual information to support my theory? Is the problem high voltage or low voltage? Does the disk receive errors when the power is cut off in the wrong sequence? Why is it possible to cut off the power in the wrong sequence? What is the right sequence?

Also it seems to be almost impossible to load anything from tape into the C2/8P. Therefore some good programs written for the C2/4P were totally rewritten.

We have access to 4 mainframe computers (we are part of the University System), but they are sometimes not available due to telephone line problems, too many users, etc. Therefore, we are considering buying several/many microcomputers instead of more terminals for the mainframe computers. However OSI has such a bad reputation at my college that we have started buying Apples. I would like to store data on the mainframes but use the OSI microcomputers to write the data. I do not know if I can transfer my files. I have been able to print from the C2/8P to paper with a DEC II printer. I have not been able to get anything out of the C2/4P although it is supposed to be possible. Theoretically, one can add a socket and a 1488 chip. Pin connections are supposed to be: #1 -9vdc from line 24 of bus (this must be furnished by a battery or external power supply). #2 to pin 6 of the 6850 chip of the 502 board. #3 output to printer or other computer. #7 ground (27th or 28th line of the bus). #14 +5 vdc (to 25th or 26 line of bus) (5 vdc may not be enough so a separate +12 vdc supply be needed)

Does anyone know if this will work? What about handshaking lines to the other computer? Some of our lines to the other computers are set for 300 baud and some for 1200 baud so we should be able to make something work. Trouble is all of the computer hardware people are at the other end of the telephone lines and are busy with their own projects or problems.

Thanks for reading my long letter. You can reprint any or all of it. If I can find time, I will write an article about my software. It is somewhat unique and educational. Aloha.

Bruce Hughes
Kahului, HI

Bruce:

Do let us at least have a look at your programs. We'd love to have an article.

AL

* * * * *

Bruce:

Taking your questions one at a time. Your questions about power being supplied to the computer are questions that every dealer should be able to answer for you and provide you with possible solutions.

Voltage spikes, whether up or down, typically have no effect on the data that is on the disk, with the following exceptions. If the spikes are of sufficient amplitude and duration, the following can happen:

1. The contents of RAM can be altered.

2. If they occur during a disk read, the read data could be changed.

3. If during a disk write, the write data could be changed.

The above are the subtle problems that can occur with voltage spikes. The really bad problems are, #1 the machine locks up, usually indicating RAM has been altered and/or #2 the machine has various errors, either disk errors or interpreter errors.

In answer to the disk receiving errors on power down or power up, most of the time this happens during power up, but could happen during power down. However!! To my knowledge, NO and I repeat, NO computer manufacturer will guarantee data integrity on floppy disks that have been left in the machine and have had the power turned off or turned on. It is also an excellent practice to have backup copies of all disks just in case.

Printing from the C2-4P can be accomplished out of its printer port by just adding the -9V power supply and several parts that may or may not be on the 502 CPU board. Before doing any modifications I suggest that you consult your dealer for a copy of the Sams manual on the C4P and a competent technician.

For computer to computer communications, you should consult the manufacturer of the "mainframe" computer as to their handshake requirements. If it is strictly RS 232, there may not be any.

If you have further questions, contact us here at PEEK (65). If we can't answer your questions then maybe some of our readers can.

Brian

* * * * *

ED:

In response to my letter in the July issue, you asked for the particulars on how the disk boot in OS65D stores the message RUN"BEXEC* in the buffer. Your wish is my command, but what follows may be more than you want to know on the subject:

1. When you hit the reset button, the monitor ROM jumps to FFA0. The routine there clears the screen, initializes the ACIAs for I/O and prints the message "H/D/M".

2. When you select "D" the routine goes to FF00 via a subroutine call. The code there initializes the disk PIA, homes the disk stepper motor to track 0, and loads eight pages of RAM from track 0, starting at address 2200.

3. Returning from the subroutine call to address FFED, the ROM routine then jumps to the code at 2200 which has just been loaded from disk.

4. The routine at 2200 then proceeds to load the kernel section of OS65D, beginning at 2A00 and ending at 2EC1.This is the code which stores the initial message RUN"BEXEC*(CR) in the buffer.

5. After all this is finished, the message OS-65D V.30 is displayed and the routine then jumps to 2AE6 in the kernel, which loads and runs BASIC from a cold start and executes the command in the buffer.

Upon reviewing the foregoing procedure, I suddenly realized that a simpler and better non-destructive boot would result from simply changing the jump to BASIC to a jump to the kernel. This would preserve all the BASIC (or Assembler) source code without the necessity of any reconstruction of 3179 through 317F, and can be done as follows:

```
A*CA 0200=01,2
A*GO 0200
- DISKETTE UTILITIES -
SELECT ONE:
1) COPIER
2) TRACK 0 READ/WRITE
? 2
- TRACK ZERO READ/WRITE UTILITY -
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p
               PAGES WITH gggg AS
               THE LOAD VECTOR
E - EXIT TO OS-65D
COMMAND? R4200
- TRACK ZERO READ/WRITE UTILITY -
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p
               PAGES WITH gggg AS
               THE LOAD VECTOR
E - EXIT TO OS-65D
COMMAND? E
A*EM
EM V2.0
@42C5
42C5/E6 51
:EXIT
A*CA 0200=01,2
A*GO 0200
- DISKETTE UTILITIES -
SELECT ONE:
1) COPIER
2) TRACK 0 READ/WRITE
? 2
- TRACK ZERO READ/WRITE UTILITY -
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p
               PAGES WITH gggg AS
               THE LOAD VECTOR
E - EXIT TO OS-65D
COMMAND? W4200/2200,8
- TRACK ZERO READ/WRITE UTILITY -
COMMANDS:
```

digital technology

BUS-II LEVEL I BOOKKEEPING & ACCOUNTING SYSTEM
The BUS-II turn-key multi-client accounting package is the leading OSI business software package. BUS-II Version 3.1 includes five principle modules:

	Inst. Price	Ref. Price
GENERAL LEDGER	$1200	$599
ACCOUNTS RECEIVABLE (a)	1000	599
ACCOUNTS PAYABLE (a)	1000	599
ORDER ENTRY W/ INVENTORY (a)(b)	1000	599
PAYROLL	1200	799

The Accounts Receivable, Accounts Payable, and Order Entry W/Inventory are completely interactive with the BUS-II General Ledger. Two optional specialized packages (completely interactive) are also available.

CPA EXTENSIONS (see below)
POINT-OF-SALE TERMINAL W/INVENTORY (see below)

The BUS-II CPA EXTENSIONS Package provides special features for accountants and bookkeepers. The POS-1 Point-of-Sale Terminal package enables the operator to use the computer system's video terminal as an on-line "electronic cash register".

Note: BUS-II V 3.1 operates on floppy-disk or hard disk-based systems running the OS-65U V 1.2 operating system (LEVEL I, II, or III). Multi-client use can accommodate any number of client companies on floppy disk systems or hard disk systems with H/D/E (required for hard disk use). BUS-II LEVEL I files are limited in size for floppy disk back-up; floppy disk operation continues in case of hard disk failure.

BUS-II LEVEL II (EXPANSION TO BUS-II LEVEL I)
BUS-II LEVEL II is designed for much larger businesses. Expanded file size and special operations allow virtually unlimited numbers of accounts and transactions. BUS-II LEVEL II requires BUS-II LEVEL I. Minimal back-up is data cassette (tape) or floppies—although multiple Winchester disk operation is recommended (provides ability to continue computerized bookkeeping functions in case of hard disk failure). H/D/E Hard Disk Executive is required.

	Inst. Price	Ref. Price
GENERAL LEDGER	$ 600	$ 399
ACCOUNTS RECEIVABLE	600	399
ACCOUNTS PAYABLE	600	399
ORDER ENTRY W/ INVENTORY	600	399

POS-1 POINT-OF-SALE TERMINAL (a)(b)
POS-1 is an on-line multi-store point-of-sale terminal program with integrated inventory designed for cash register emulation. POS-1 controls cash drawer and ticket printer (or system printer). Automates taxable or nontaxable sales, cash transactions, and credit sales (with verification operations). POS-1 is interactive with the BUS-II V 3.1 BOOKKEEPING & ACCOUNTING SYSTEM.

POS-1 Inst. Price $2400 Ref. Price $1199

CPA EXTENSIONS PACKAGE (a)
CPA EXTENSIONS is designed for public accounting firms. A number of special operations are provided: a "bankers" Balance Sheet and Profit and Loss statement with summarization and consolidation options, Statement of Changes in Financial Position, Statement of Changes in Components of Working Capital, Cash Flow Analysis, Departmentalized Sales Analysis, Comparative Income Statement, Budgetary Analysis, Asset Depreciation Schedule (compatible with TAXMAN-1040), and Loan Amortization Schedule. CPA EXTENSIONS is interactive with BUS-II V 3.1 BOOKKEEPING & ACCOUNTING SYSTEM.

CPA Extensions Inst. Price $ 3600 Ref. Price $1599

TAXMAN-1040 PERSONAL INCOME TAX PREPARATION
TAXMAN-1040 is designed for tax practitioners and public accountants. TAXMAN-1040 is the leading tax package for OSI microcomputers—the package has been installed on OSI, Hewlett-Packard, DEC and IBM systems. Designed and supported by CPA tax experts. This package automatically prepares FORM 1040 and 28 schedules. Individual state tax option available. Support includes annual forms and tax table revisions. Purchasers of 1980 TAXMAN will automatically receive 1981 revisions at no extra charge.

TAXMAN-1040 Inst. Price $3600 Ref. Price $2399

TAXMAN-1120 CORPORATE INCOME TAX PREPARATION (a)
TAXMAN-1120 (under development) is a corporate tax preparation package designed to work in conjunction with TAXMAN-1040 to provide full-service tax accounting functions. TAXMAN-1120 requires BUS-II G/L. Individual state tax option available; support includes annual forms and tax table revisions. Purchasers of 1980 TAXMAN will automatically receive 1981 revisions at no extra charge.

TAXMAN-1120 Inst. Price $3600 Ref. Price $2399

OS-DMX DATABASE MANAGEMENT SYSTEM
Command-oriented OS-DMS compatible database management system. OS-DMX operates under the OS-65U V 1.2 operating system (LEVEL I, II, or III). Features such as control files, extensive operating commands, and the innovative HELP feature, in addition to Digital Technology's exclusive on-line documentation, make this one of the most usable—as well as powerful—systems available for microcomputers. OS-DMX may be used instead of, or in addition to, OS-DMS Nucleus, Query, Sort; OS-DMX will replace virtually all of the specialized OS-DMS modules—and in most applications will provide greatly improved performance.

OS-DMX Inst. Price $2000 Ref. Price $1199

ECR-1(P) ELECTRONIC CASH REGISTER POLLING (c)
ECR-1(P) provides cash register polling and control (including cash register reprogramming) in conjunction with OSI microcomputers. Cash register polling is an alternative to on-line operation which allows the use of regular preset-total style electronic cash registers (with RS-232 communications). Versions are currently available for MKD BANTAM II and certain NCR cash register systems.

ECR-1(P) Inst. Price $1600 Ref. Price $799

ECR-1(C) DATA CASSETTE POLLING (c)
ECR-1(C) provides data cassette polling, allowing multi-store cash register polling. ECR-1(C) is recommended when diverse store locations make telephone line communications prohibitively expensive.

ECR-1(C) Inst. Price $1600 Ref. Price $799

SALES-1 SALES ANALYSIS (c)
SALES-1 is an OS-DMX-based sales analysis package for use in conjunction with OS-DMX, ECR-1(P), or ECR-1(C). Breakdown is provided by key-hit, family group, etc., indicating totals and percentages of sales. OS-DMX is required; ECR-1 is recommended; manual stand-alone operation is optional.

SALES-1 Inst. Price $1600 Ref. Price $799

INV-1 RESTAURANT INVENTORY & MENU EXPLOSION (c)(d)
INV-1, used in conjunction with OS-DMX, ECR-1 and SALES-1, provides a detailed breakdown of sales by family group and menu components. Provides managerial information detailing waste, pilferage, menu costs, stock levels, reorder levels, percentage-of-sales and percentage-of-cost from menu explosion. OS-DMX required; ECR-1 and SALES-1 recommended; manual stand-alone operation optional.

INV-1 Inst. Price $1600 Ref. Price $799

H/D/E HARD DISK EXECUTIVE
Digital Technology's implementation of H/D/E is the answer to AMCAP's HDM. Digital Technology's H/D/E provides user functions not found on HDM or similar products: ability to copy from any user "system" to another; automatic recovery in case of "back-up to floppy" or "restore from floppy" utility failures, allowing the user 3 options: (1) ignore error, (2) abort to menu, (3) try again; use of both "A" and "B" floppy drives to back-up hard disk files; and automatic back-up diskette initialization. H/D/E operates on any OSI Winchester disk system from 7 - 80 megabytes. Re-use of hard disk space is provided. Superior to AMCAP's hard disk manager in every respect (and Digital Technology software does not self-destruct). NOTE: H/D/E is required when installing any Digital Technology business applications packages on OSI hard disk systems.

H/D/E Inst. Price $800 Ref. Price $499

H/D/M MULTI-USER MANAGER (g)
H/D/M (under development) is Digital Technology's multi-user extensions to OS-65U. Replaces T-MUM by AMCAP. Need we say more?

H/D/M Inst. Price $1200 Ref. Price $499

DIGITAL TECHNOLOGY, INC. software is sold through OSI Dealers worldwide. For detailed product information call (714)270-2000. For the name of your nearest OSI Dealer call (toll free) OSI's "hot line" 1-800-321-6850.

digital technology
inc.
P.O. BOX 178580
SAN DIEGO, CA 92117
(714) 270-2000

REQUIREMENTS
(a) BUS-II LEVEL I G/L req'd
(b) BUS-II LEVEL I A/R req'd
(c) OS-DMX req'd
(d) ECR-1 recommended
(e) C3 CPU W/56K RAM & OS-CP/M or Lifeboat Associates CP/M req'd
(f) SYNCHRONOUS INTERFACE ASSY req'd
(g) H/D/E Required

```
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p
               PAGES WITH gggg AS
               THE LOAD VECTOR
E - EXIT TO OS-65D
COMMAND? E
A*
```

BASIC overlays the code which the cold start routine loads starting at 2200; hence, the usual BEXEC* boot destroys it before you can get a glimpse of it. Once you've installed the non-destructive boot, however, it's still there for your inspection using the EM.

I haven't seen the disassembly you referred to, so I can't comment on whether it is deficient in this respect, I am indebted to Professor Tom Berger of the U. of Minnesota for the disassembly I used to follow the tortuous trails of OSI's software.

Sidney Sosin
Glenview, IL

* * * * *

ED:

As a recent subscriber, I think you are doing a great job. I have most of the back issues and must be learning something as more things are making some sense. I have a C1P, 610 expander board with 32K RAM and a 5-1/4" floppy.

In reference to the X*(A>9) operator. I think it should be considered a legitimate operator. The literature does describe it as such. The statement for the Hex converter is ingenious. It directly replaces 1FA>9 THEN A=A-7 in my two line Hex to Dec and frees the line so that it is now a 1 line operation:

```
10 A=ASC(H$)-48 : A=A+7*(A>9)
   : D=D*16+A : H$=MID$(H$,2)
   : IF H$<>"" THEN 10
```

Also I don't think A=A+(A>X) should be dismissed lightly. For example: A=A+(A>100) replaces the IF statement IF A>100 THEN A=A-1 and frees the remainder of the lines: (used to limit incrementing variables).

What are the significant differences between OS65D V3.1, V3.2, V3.3? In your journal does the reference 65D imply commonality or does it imply V3.2? I have V3.1 and haven't had much use from it. I get angry and revert to the basic machine. All listing etc. are ill-fitted to the 24X24 display. My really big mad with OSI is that they advertised the C1P as fully expandable when in reality it is a red-headed step child. Would it be worthwhile for me to purchase the V3.2 disassembly manual from "Software Consultants", or should I consider buying V3.3?

When I received OS65 V3.1 from "Cleveland Consumers", I could not get the "check-sum" loader of the Ext Mon to work and spent some time trying to find it, without success. Ofcourse my letter to "Cleveland Consumers" wasn't answered. When I received the back copies of "PEEK (65)", I found a note to change two addresses and it made the "check-sum" work. However, the operator to terminate the SAVE, still does not.

Harry Hawkins
Burton, SC 29902

Harry:

Reference 65D, I am not very familiar with the differences between V3.1 and 3.2, etc. Readers, who can help? The Software Consultants manual is very nice, but we need more information on V3.3 to make a decision...

AL

* * * * *

ED:

You were correct in stating that OSI's BASIC will respond with "0" to a false logical statement but to a true statement it will return with a '-1'. After all, the opposite of a binary number with all zeros is one with all ones.

The coding I am submitting was written in a package that required that I frequently check input strings for three 'controlcharacters':

'\' to return to the last input
'/' to skip to the next input
'N' "NO"

I found a cozy little solution using 'logic in parens' and 'FN' statements. In the example below, calling 'FNA(ASC (X$))' will take the ASCII value of the first character of X$, compare it with the ASCII value of the control characters and return either a: 0 (no match); 1 (X$='/'); 2 (X$='\'); or 3 (X$='N'). The program will then transfer to one of the three lines listed in the 'ON GOTO' statement or will (in the case of FNA=0) continue with the next statement. The statement 'ON ABS(X$="")GOTO 200' insures that 'ASC(X$)' will not fall-out with a FC error.

ABS(X$="")GOTO 200' insures that 'ASC(X$)' will not fall-out with a FC error.

```
10  DEF FNA(X)=ABS((X=47)+(X=
    92)*2+(X=78)*3)
100 REM  FIRST INPUT
110 ...
190 REM SECOND INPUT
200 INPUTX$: ONABS(X$="")GOTO
    200: ON FNA(ASC(X$))GOTO
    100,400,300
210 ...
300 REM X$="N"
310 ...
400 REM THIRD INPUT
410 ...
```

R. David Keays
Anaheim, CA

* * * * *

ED:

I was reading the September issue of PEEK (65), digesting all of the good stuff I've come to expect, when I came across a paragraph from Ross Votaw's letter. I couldn't beleive it. Someone has a fix for the DISK GET problem in 65D!

I have a C8P-DF system, bought in March 1980. Shortly thereafter I bought the MDMS database program which brought the problem to my attention. It's more than annoying and time consuming, it is very hard on the disks when a disk read is done for each record in a random access file. I have extensively modified MDMS by adding a sorting capability (primary & secondary sort keys), decimal control on output, record delete, file delete, a few changes to the statistical program, and a range capability to the conditional. I am now VERY pleased with MDMS except for the DISK GET bug. Could you possibly get this fix published in PEEK (65)?

Speaking of bugs, I enjoyed "STRETCH" Manley's OS65D3 column. However, I checked for every bug mentioned and couldn't find any of them in my version of 65D.

Keep up the good work and I hope you can publish the DISK GET fix.

Jack Eddington
Pittsburgh, PA 15219

Jack:

We will certainly publish the

listing as soon as we can get it. Meanwhile, contact Aardvark for the issue of Aardvark Journal in question!

Al

* * * * *

ED:

Put me down as an C28P owner, not at all uninterested in problems of 65U owners, but hoping you will always make clear which machine and which system the letter or article is dealing with. Ideally, if the pertinent differences could be annotated each time, by footnote or otherwise, it would be a tremendous help.

In particular, since my motivating interest is to implement home control systems for my severely disabled wife, it would be useful to know whether the Home Control environment is ever considered when some of the printed suggestions are edited. And will the long promised Process Control Basic ever appear, and if it does, will it supplement, or be compatible with Home Control?

Questions like this may be trivial to many personal and business uses, but to those of us who need Votrax, ACTL, and telephone interface uses, the capability of using all of these in all programming is critical.

Enjoy your publication, and keep it up.

Bruce Stowe
Hamden, CT 06514

Bruce:

As far as we know, OSI has no plans in the immediate future to release the Process Control Basic programs. When it does become available, we will publish it in PEEK (65).

Al

* * * * *

ED:

How do I get my Epson MX 80 printer to print block graphics? I have an OSI - OEM C III with four drives and a Heathkit H19 Terminal. Neither I or my local dealer (in Vancouver) knows how.

Garnet Lloyd
Kelowna, B.C.

Garnet:

OSI strikes again! The reason you cannot get the Epson to print its graphic characters is that OSI's Centronics Compatible Interface is only a 6 bit interface and not an 8 bit interface. Since the Epson 80 needs 8 bits to decode the graphics characters, you will not get them to print using OSI's Parallel Printer Interface.

Brian

* * * * *

Garnet:

One way around the problem would be to use the serial rather than the parallel interface for the MX-80. It would require that you feed your MX-80 through the 550 (Ca-10X) board, with a hardware handshake which your dealer can install for you. Then you must buy the serial interface card option and do a little wiring, explained in the very nice instructions which come with it from Epson. I have done the job and it works fine.

Al

* * * * *

ED:

The following is a listing relating to a benchmark on a suggested alternate coding for the code suggested by Corky Kirk for padding a string with spaces, which is:

```
'35020 X$=" "+X$: IF LEN(X$)<6
       THEN 35020'
```

versus the code I use which is:

```
'36012 SP$="
       ": REM .... Length to
       max required
'36020 IF LEN(X$)<6 THEN
       S$=LEFT$(SP$+S$,6)
```

which runs about 3 1/2 times faster. I didn't have a stopwatch so pardon the 'beats'. Anyway, it's the idea that counts.

Paul Loughridge, Jr.
Kamuela, HI 96743

* * * * *

ED:

Thanks to PEEK's July 1981 issue, I've realized that those awkward characters of our Superboard can be customized by replacing the respective ROM with an EPROM. I own a fair amount of software but still a great number of characters have never been used, and some I just don't like! For example, if the figure of the man CHR$(240) were bigger, so that his hand reached to the side of the box, you could place objects, such as weapons or bats, in the adjacent POKE. But being a novice, I can't imagine how I would do this myself. Is there someone out there who would create and sell to me an EPROM with a set of customized characters?

David Whipp
Salt Lake City, UT 84192

* * * * *

ED:

In the September 1981 issue of PEEK (65), you mentioned in your article "The Beginning Assembler - Part III" a smart terminal emulator SMARTER-M by Phil Lindquist, with data storage capablities. I am looking for such a program for use with my C1P (with two disk drives).

I would appreciate it very much if you would let me know where this program is available.

John E. Young
Woodbridge, IL 60517

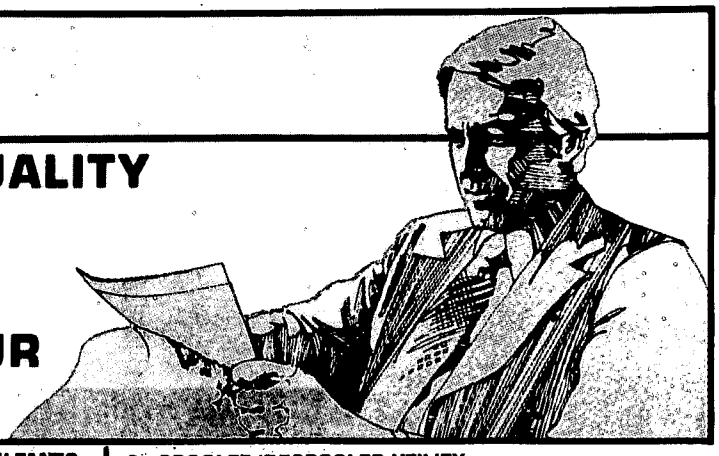
For anyone interested, contact Phil Lindquist at, 8892 Cooley Lake Rd., Union Lake, MI 48085.

Al

* * * * *

AD$

IF
YOU NEED QUALITY
THEN
YOU NEED OUR PRODUCTS



That's right. The kind you get from SOFTWARE CONSULTANTS... it's efficient, cost-effective, and flexible. Next time you dump some old programs, remember to load up with the hardworking software you get from SOFTWARE CONSULTANTS.

We're an independent vendor dedicated to producing top quality software just for you , the OSI users, at the most reasonable prices around. As we've told you before, most of our products are modifications and/or extensions of existing OSI packages. Some of our items completely replace OSI's, and all of our products are easily extensible for your custom applications.

The many responses we've recently received have shown us just how badly many of you needed a quality source for OSI software. So, to keep up with your requests for more "good OSI stuff," we've been working overtime on lots of new products. We'll be telling you about each one of them as soon as they're ready, so be sure to keep us in mind for all of your upcoming projects.

We just don't have enough room in these ads to fully describe our products to you, so please call or write us for the latest copy of the free SOFTWARE CONSULTANTS product catalog. It'll give you all the facts on our current line and full details as each new product is introduced.

Remember, for fine OSI software at the most reasonable prices around, it's SOFTWARE CONSULTANTS. Take a quick look at this list and ask yourself...can you really afford to keep using any one else's OSI software?

1. **OS-65D V3.2 DISASSEMBLY MANUAL**
   A super-complete manual that has it all -- 50 pgs. of dissassembly listings, complete and clear comments on most every line, 10 pgs. of computer generated cross reference listings, and more! Praised by many OSIers who couldn't believe it 'til they bought one. A deal at $25.95.

2. **REF COMMAND UNDER BASIC**
   A complete, cross reference utility that'll find and list any BASIC line number, variable or numeric constant. It's available under 65D or 65U and comes on 5¼" or 8" floppies. This one will save your sanity, and cut out hours of wasted time. Yours for $31.95.

3. **SPOOLER/DESPOOLER UTILITY**
   A useful utility that feeds backed-up data to your printer for normal output, and leaves your screen free for other work by intercepting data bound for your printer and temporarily storing it on hard disk. Written in super fast machine language. Interfaces with serial and parallel printers.

4. **FIG FORTH UNDER OS-65U**
   So far, the only one that's running under 65U at a price within reach. You get all the pluses, like terminal oriented editor, lots of printer and terminal tools, and more. Under multi-user, runs BASIC simultaneously, too. Unbelievable at $89.95.

5. **VIDEO ROUTINE**
   This convenient program really spices up your video system with little niceties like 24 separate control codes, horizontal and vertical plotting, and many variable screen parameters. Software extensions are available to connect this with the graphics resolution booster. The routine alone is $25.95, with extensions, $29.95.

6. **GRAPHICS RESOLUTION BOOSTER**
   An ingenous piece of hardware that'll increase the graphic resolution per character by 8 times. With this device, your circles will be rounder and your curves smoother. A slick addition to your video-based system. Priced at $49.95, or, with the video routine and extensions, $79.95.

Unlike the majority of other software vendors, we offer our customers copies of source code (on floppies) for any of our products they've purchased. For a nominal fee of $12, covering our material, postage, and handling, we'll send you the source code you choose. Simply write it in as you fill out the order coupon. Of course, dealer inquiries are invited for all of our products.

SOFTWARE CONSULTANTS
HARDWORKING *software*

Enclose your check with coupon and mail to: SOFTWARE CONSULTANTS, 6435 Summer Ave., Memphis, TN 38134 U.S.A. Phone us at 901/377-3503 for rush orders. TN. residents add 6% sales tax. Dear Software Consultants:

Sounds great. Here's my order. I want:

_____DISASSEMBLY MANUAL(S) @ $25.95 ea.
_____CROSS REFERENCE UTILITY(S) @ $31.95 ea.
Disk size 5¼"_____or 8"_____
OS-65D_____or OS-65U_____
_____SPOOLER/DESPOOLER UTILITY(S) @ $69.95.
Std. parallel interface_____
Serial interface w/430 board or equivalent_____
Serial interface w/CA10X board_____
_____FIG FORTH(S) @ $89.95 ea.
_____VIDEO ROUTINE(S) @ $25.95 ea.
With software extensions @ $29.95 ea.
_____SOURCE CODE @ $12.00 ea.
For_____
_____GRAPHIC RESOLUTION BOOSTER @ $49.95 ea.
With Video Routine and extensions @ $79.95 ea.
_____SOFTWARE CONSULTANTS Product Catalog Free

NAME _____

ADDRESS _____

FIRM _____

CITY/STATE _____ **POSTAL CODE** _____

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:

PLEASE SEND PEEK (65) FOR ONE YEAR (12 ISSUES).

All rates quoted in and should be submitted in U.S. dollars.

( ) $15.00 Enclosed. U.S. (Maryland residents add 5% sales tax.)
( ) $23.00 Enclosed. Canada and Mexico, 1st class, surface.
( ) $35.00 Enclosed. South and Central America. Air Mail.
( ) $35.00 Enclosed. Europe. Air Mail.
( ) $40.00 Enclosed. All other. Air Mail.

NAME_____ STREET _____

CITY_____ STATE _____

ZIP CODE_____ COUNTRY_____

Please send the following back issues. I enclose:

( ) $2.00 ea. U.S. Surface. (Maryland residents add 5% sales tax.)
( ) $2.50 ea. Canada and Mexico. Surface.
( ) $3.00 ea. South and Central America. Surface.
( ) $3.00 ea. Europe. Surface.
( ) $3.50 ea. All other. Surface.

Vol 1. 1980

( ) Jan #1, "Welcome"
( ) Feb #2, "A month ago"
( ) Mar #3, "Peek continues"
( ) Apr #4, "We are OSI fans"
( ) May #5, "The continued growth" (mistakenly labled #4)
( ) Jun #6, "This column should"
( ) Jul #7, "Several times recently"
( ) Aug #8, "A few minutes ago"
( ) Sep #9, "Of course."
( ) Oct #10, "Publishing PEEK(65) is"
( ) Nov #11, "As you can see"
( ) Dec #12, "This issue marks"

Vol 2. 1981

( ) Jan #1, "First,"
( ) Feb #2, "This month's Peek"
( ) Mar #3, "Last month"
( ) Apr #4, "First things"
( ) May #5, "A careful"
( ) Jun #6, "A letter to"
( ) Jul #7, "The dam is"
( ) Aug #8, "In this month's"
( ) Sep #9, "It was one of"
( ) Oct #10, "The way I write"
( ) Nov #11, "Who are you?"