Thoughts on the SWTP Computer System

This installment takes a more in-depth look at the Percom LFD-400 disk system.

Phillip Schuman
1627 Woodcutter
Wheaton IL 60187

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

In part 1 of this article, we examined the characteristics of the SWTP MF-68 and the SSB BFD-68 disk systems for the SWTP 6800 system. This month we will conclude this two-part series with a look at the Percom LFD-400.

The Percom LFD-400

The Percom disk system is quite different from the others, both in hardware and in software. The basic system consists of a small single-drive cabinet, power supply, drive, cable, controller, a DOS called MINIDOS and a batch of "technical memoranda," which give patches for adapting existing software to run on the system. At $600 for the package, this is the least expensive SWTP-compatible disk system there is.

But there are various ways to expand the basic system as time goes on—by adding a second drive and by adding more software. When this is done, the system becomes more expensive than a comparable SWTP system and approaches the price of the SSB disk system.

Let's look at the hardware first. The controller is a large board that plugs into the 50-pin main bus of the computer. Unlike the other two disk systems, this one does not use the 1771-01 disk controller IC; it uses an AMI S2350 USRT (universal synchronous receiver-transmitter). The USRT handles the serial-to-parallel conversion and synchronous data-transfer protocol for the disk, but additional hardware and software are needed to handle some of the functions normally handled by the 1771-01 chip in the other systems, such as keeping track of where the head is on the disk and so on. Thus this controller is more complex than even the Smoke Signal controller; it has 27 ICs.

There is room on the controller board for three 2708 EPROMs, which are addressed at memory locations C000, C400 and C800. The controller also uses addresses CC00 through CC06 for I/O. Hence the controller just about uses up the 4K address block from C000 through CFFF, which creates some problems for owners of newer systems who want to use that area for 2716 EPROMs on their CPU board. (This is not as much of a liability as it seems though; it is possible to move the entire board from the C000-CFFF area down to 9000-9FFF and release at least some of the C000 address space for 2716 use.)

The Percom disk uses 10-sector hard-format diskettes. Hard sectors allow more data to be squeezed on the disk than soft sectors: There are 35 tracks with ten sectors each; the resulting 350 sectors each have 256 bytes, so that the total disk capacity is 89,600 bytes, about 18 percent more than the 75,888 bytes of the SWTP soft-sectored format. A hard-sectored disk also does not have to be formatted prior to use, which saves a few minutes for every new disk.

On the other hand, data is always stored on consecutive sectors, starting from the outside of the disk. There is no provision for splitting a file into pieces so they will fit into available sectors. When a file is removed from the disk, it leaves a hole.

Percom offers three different disk operating systems: MINIDOS, MINIDOS-PLUSX and INDEX. Let's look at them one by one.

MINIDOS is the simplest and is included in the price of the basic disk system. It is completely contained on one 2708 EPROM and occupies addresses from C000 to C3FF. It uses addresses 0000-001F for a single FCB and also uses the upper portion of the MIKBUG/SWTBUG RAM (from A07F down) for a stack; this portion of RAM is normally unused by the monitor, so there is no conflict. Moreover, addresses 0000-001F are normally not used by other software, so MINIDOS will work in a system without any need for an additional 4K memory, such as the other disk systems require. It would work perfectly well even on an old SWTP system with only 2K of memory (do you remember that far back?).

Since the entire MINIDOS is contained in an EPROM, there is no booting required—you simply jump to location C000, and there it is. Since SWTBUG has the Z command to do just that, starting up the Percom disk is easy. Moreover, since there is no space taken up on the disk for a DOS or directory, the full 89K bytes on the disk can be used for files.

On the other hand, MINIDOS is a Spartan DOS. It does not maintain a directory of any kind. The 350 sectors are numbered 000-349, and it is up to you to keep a written record of what is where on the disk. In this sense, MINIDOS is like a fast cassette. All it can do is save and load chunks of memory. It has only two commands: S and L. To save a program, you might give MINIDOS the command

```
S 0100 1DB0 0100 1030.
```

This command tells MINIDOS to save the contents of memory locations 0100 through 1DB0 on the disk, starting at sector 030 of drive 1, and assign the program a starting address of 0100. MINIDOS then proceeds to write the contents of those memory locations on the disk and prints out the number of the last sector when done. It is up to the user to keep track of the sectors used, so that the program is not overwritten at some later time.

MINIDOS does not allocate disk file space in any way—it simply starts storing at the sector specified in your S command and takes up as many sectors as needed. It uses consecutive sectors and keeps no directory. This makes it fast—saving or loading programs takes a half to a third of the time of other disks.

But to fit into a 1K EPROM, MINIDOS is simple. You just can't compare it with FLEX or DOS-68. There is little facility for tying it into other programs or storing data files. It is strictly intended for storing programs.

It is not character oriented but transfers whole blocks of memory to and from the disk. It is simply a fast cassette replacement—it will dump 16K of memory to disk in 8 seconds and load it back in 5 seconds.

The next step up is an operating system called MINIDOS-PLUSX. It also comes in a 2708 EPROM and mounts in the second EPROM socket on the disk controller board, occupying addresses C400-C7FF. (This leaves the third EPROM socket empty, and you can use it for your own purposes.) PLUSX is now included in the system price. It includes the 2708, a diskette with a complete source and object listing of PLUSX and four disk utilities. (Supplying source code on diskette is a Percom specialty—they do it for most of their system software. I wish more manufacturers did that!)

In addition to the ROM, PLUSX requires 256 bytes of RAM starting at address A080, but versions are available to run with the area located at 7080 also. (But with the source code, you can reposition this almost anywhere.) For compatibility with other software, though, it is best not to move this area. This essentially requires that you disable the 6810 RAM on the CPU board (easy with the MP-A2 CPU board) and substitute a separate memory board. Since only memory from A000 through A180 is needed, a 4K board with only 1K worth of 2102s would be an inexpensive and easy solution.

MINIDOS and MINIDOS-PLUSX have to be used together; that is, you need both 2708s plugged in at the same time. But MINIDOS is so set up that, when you jump to it with SWTBUG's Z command, it tests to see whether MINIDOS-PLUSX is plugged in. If it is, it jumps to it. Hence, booting still involves only typing a Z into SWTBUG.

MINIDOS-PLUSX uses a file directory that occupies the first ten sectors (first track) on the disk. Each file gets a name of up to six characters; if the first character of the name is an at-sign, the file becomes protected and cannot be erased or written over.

Each PLUSX command is a single letter; the commands are:
I—initialize the diskette directory.
S—save a region of memory to diskette.
A—add ten sectors to the end of the last line.
L—load a file from disk back into memory.
F—list the directory.
R—rename a file.
D—delete a file.
J—jump to an address.
X—exit to MIKBUG or SWTBUG.
M—go to MINIDOS.

In addition, four disk-utility commands are provided on diskette:
BACKUP—copy full tracks from one disk to another.
COPY—copy a file from one disk to another.
CREATE—allocate disk space for a file and place its name in the directory.
PACK—pack the files on a disk to close up empty holes between them.

PLUSX is structured at a more fundamental level than either FLEX or DOS-68. In addition to the more familiar linkages to user software, various subroutines are available for total disk control. A jump table is located at the beginning of the MINIDOS area, but no entry points are provided to the PLUSX directory handler. Thus, the user can use the jump table to access mundane I/O functions but has to piece through the PLUSX software to figure out how to link to it. (Fortunately, Percom supplies the commented source code.)

Space on the diskette is allocated using a "go to the end of the line" scheme. This means that any new space allocations (not old files to be overwritten) will be placed at the end of all previous files on the diskette. If a file is deleted, its space may not be reclaimed until the diskette is packed with the PACK utility. The sectors within a file are chained together, but empty sectors are not.

Although dynamic allocation of disk space is convenient, the fragmenting of disk files into pieces that are spread out all over the disk has one big disadvantage (aside from its slowing everything down). Accidents do happen. Sometimes, because of operator error, sheer stupidity or accidents, something happens to erase part of a disk—a single file, part of the directory or perhaps just a sector or two. With a dynamic-allocation scheme, it becomes a major job to try to rescue the rest of the disk. Just finding out what's where is a major job. But with a single-minded scheme such as Percom's, it's a simple matter to examine disk contents sector by sector and reconstruct at least some of the files. They even provide a DSKMAP program to do just that.

The most advanced Percom DOS is called INDEX; unlike MINIDOS and MINIDOS-PLUSX, INDEX comes on a diskette rather than being in ROM. As supplied, one diskette contains a MINIDOS- or MINIDOS-PLUSX-compatible dump of INDEX, while a second diskette contains the INDEX disk utilities. The INDEX software and a user's manual (no source text this time) cost $100.

INDEX requires 8K of memory at addresses A000 through BFFF; thus the entire lower 32K of memory addresses is available for user programs (except for 0000-001F). Whereas SWTP's FLEX and SSB's DOS-68 are somewhat similar to each other, INDEX is again quite a departure. Just about the only similarity is that it uses dynamic allocation of disk space, like the others.

First of all, INDEX uses the interrupt system of the computer. The console terminal and other I/O devices are handled via interrupts, rather than constantly being polled as in the others. Both the terminal keyboard and the printer or display use a software buffer, so that I/O can be overlapped with computation. This can be timesaving, since input, computation and output can go on at the same time.

A second departure is that I/O equipment is treated the same as disk files, that is, a program having some output calls INDEX, gives it the name of a destination for the output data and then passes it to INDEX. If the destination is a disk file, then INDEX will write it on disk; if the destination is an output device, then INDEX will send it to that device.

Since I/O and files are handled the same way, the COPY command can be used to copy files, send data from one I/O device to another or transfer data between disk and I/O. Thus a program could, without any changes, store all its output on disk, and the COPY command could later be used to get the output off the disk and print it. This also means that it is easy to add new I/O equipment at a later time. As soon as an INDEX-compatible I/O driver is written to go with it, the new I/O device becomes a part of INDEX and can become either a source or destination for data.

Like the other disk operating systems, INDEX maintains a disk directory. But this directory contains much more information than the others. Each file has a version number and a date and a protection flag, which is used by the software to avoid accidental erasure of files. The directory entries are classified into levels, and the user can specify which levels he wants listed when he asks for a directory printout. Thus it is possible to get a printout of only user programs, not system programs. The level number of a program is also used when a disk is being copied.

Since INDEX is interrupt-driven, it can respond to user commands even while a user program is executing. One of its interesting capabilities is that the user can command INDEX to stop outputting without actually stopping a user program running at the same time. Thus the program can continue running, but its terminal output can be turned on and off at will. When off, the program obviously runs much faster.

INDEX has the following commands:
BACKUP—create a backup copy of an entire disk (except certain kinds of temporary files).
CONVERT—convert hexadecimal files of the S113 . . . . Type into pure binary files to save

space and time.
COPY—transfer data from one file or I/O device to another.
COUNT—count the number of lines and characters in an ASCII file.
DATE—change the current date being used by INDEX.
DELETE—delete a file.
DIR—list the diskette directory.
DISKEDIT—examine or modify the contents of a specific disk sector.
DISKINIT—initialize the disk and its directory.
EXAMINE—examine an ASCII file to look for control characters.
FILL—specify the number of fill characters (nulls) sent to the terminal after a line feed or form feed.
HELP—get instructions for using a given command.
RENAME—rename a file or change its protection flag.
SAVE—save memory contents to a file (or I/O device).
SETSTART—set the starting address of a program file.
SETVER—change the version number of a file.
SYSDISK—specify which drive holds the system disk.
USERDISK—specify which drive holds the user disk.
TYPE—list file contents on the terminal.

INDEX is a full DOS, in the same ball park as FLEX or DOS-68, but since it is an extra cost option, it may never become as popular. We have found MINIDOS-PLUS to be quite adequate for most uses, though not up to FLEX or DOS-68 in versatility.

MINIDOS/PLUSX Programs

With the basic disk system or on a $15 user's group disk, Percom provides patches to other people's cassette software. Both source code and object code are provided. This includes:
SWTP 8K BASIC version 2.0 or 2.2—permits saving and loading programs on the disk with SAVE, LOAD and APPEND commands.
Computerware Software Services Super Cassette BASIC version 4.0—same functions as SWTP BASIC patch above.
TSC Text Editor—a patch to implement disk SAVE and READ commands.
TSC Assembler—a patch to permit assembling programs from disk and put the object code back on the disk.
SWTP Cores Assembler—a patch to permit saving and loading source programs on the disk.
SSB Source Code Generator—a patch to permit saving the generated source code on the disk so that it can be edited and reassembled.

The following programs are also provided, either as part of the system or on the same user's group disk:
HEXLDR—for loading S113 . . . . type disk files, produced by the assembler, into memory.
DSKMAP—for examining the header for any disk sector or listing the contents of the sector.
MEMTST—for doing a memory test.
PRINTOUT—for printing out ASCII files on a printer.

There are two other patches, which are supplied separately ($18 each for instructions and a diskette with both source and object codes):
The BASIC Bandaid—patches SWTP 8K cassette BASIC version 2.0, 2.2 or 2.3 to allow saving and loading programs on disk, chaining of programs and sequential or random disk files.
The TSC Touchup—patches the TSC Text Editor and Text Processor (cassette versions). The editor patch is most interesting, since the patched editor is even better than TSC's own FLEX-based editor. The patch improves the editing within a line, allowing the addition, substitution or removal of characters within a line, a process which is awkward in the original TSC editor. It allows a disk-to-disk editing of files too long to fit into memory at one time. The editor can also edit BASIC source programs or data files.

Finally, Percom has written a BASIC interpreter and an assembler of their own (unfortunately, supplied without a source listing):
Super BASIC ($50) is a 12K disk BASIC that supports both sequential and random files. It is quite a bit faster than the SWTP BASICs but retains their BCD arithmetic for maximum accuracy in dollars-and-cents calculations. It provides PRINT USING for formatting, a LINPUT instruction for inputting a complete line of text including commas, array subscripting starting at either 0 or 1, direct execution of any DOS command within a BASIC program and a number of other interesting features. For example, it is possible to PEEK and POKE data into disk file buffers. In this way, it is possible to compress disk files into less than half the space they would need if stored in ASCII.
The assembler ($30) is a disk-to-disk assembler similar in function to the TSC assembler patch above but allows specifying assembly options at time of assembly, rather than requiring them as OPT statements in the source text.

Percom also has several business-oriented packages, including "Finder," a general-purpose filing system ($100), a mailing list program ($100) and a general ledger system ($200). A complete checkbook-balancing program and BASIC utilities for renumbering programs, producing paginated source listings and driving the terminal via the interrupt system are available from Star-Kits, PO Box 209, Mt. Kisco, New York 10549.

Another company providing PerCom-compatible programs is Hemenway Associates, 151 Tremont Street, Boston MA 02111, which has a STRUBAL compiler ($250) for a structured BASIC-like language, a relocating marco-assembler ($80), a relocating loader ($50) and a text editor with editing macros ($40). Microware's A/BASIC is available in a Percom version for $65. Much of Ed Smith's Software is also available in versions to go with MINIDOS, MINIDOS-PLUSX or INDEX. This includes a relocating macro assembler, loader, disassembler, relocater and others.

Listing 1.

```
                    NAM  PERCOM-TSC EDITOR
                    OPT  NOT,NOG
               * ********************
               * * PERCOM TSC/EDIT  *
               * *  FOR PLUSX V3.0  *
               * ********************
               *  LINKAGE POINTS INTO TSC EDITOR
               *
0058           SPCPT1 EQU  $58  TEXT FILE TARGET- BEGIN SAVE
005A           SPCPT2 EQU  $5A  TEXT FILE TARGET- END SAVE
0441           ERROR  EQU  $0441    EDITOR ERROR PROMPT
E07E           PDATA  EQU  $E07E    MIKBUG STRING PRINT
               *
               *   LINKAGE POINTS INTO PERCOM MINIDOS-PLUSX
               *
7080           LINBUF EQU  $7080   INPUT LINE BUFFER AREA
70A0           LINPTR EQU  $70A0   INPUT CHARACTER POINTER
C42D           XINPUT EQU  $C42D     INPUT CHARACTER ROUTINE
C4BD           XFIND  EQU  $C4BD    SEARCH DIRECTORY FOR NAME
70A8           XFILE  EQU  $70A8    ALLOCATION FIELD
70AD           XBEG   EQU  $70AD    DUMP START ADDRESS
70AF           XEND   EQU  $70AF     DUMP  END ADDRESS
70B1           XEXEC  EQU  $70B1    PROGRAM EXEC ADDRESS
C554           XSAVE  EQU  $C554    PLUSX SAVE ROUTINE
C01B           MLOAD  EQU  $C01B    DOS NORMAL LOAD ROUTINE
C01E           MERR   EQU  $C01E    PRINT DOS ERROR CODE
C363           CRLF   EQU  $C363    PRINT CR/LF
0016           MTW    EQU  $0016    DOS TARGET MEMORY ADDRESS
0014           MTA    EQU  $0014    DOS RETURNS NEXT MEMORY LOCATION
0001           MDISK  EQU  $0001    DOS TRACK/SECTOR REQUEST
               *
               *   SAVE TEXT FILE FROM MEMORY TO DISK
               *
13D3                  ORG  $13D3
13D3 CE 14 0C         LDX  #ASK     PROMPT MSG
13D6 BD E0 7E         JSR  PDATA    PRINT IT
13D9 CE 70 80         LDX  #LINBUF  POINT TO OUR
13DC FF 70 A0         STX  LINPTR     LINE BUFFER
13DF BD C4 2D         JSR  XINPUT   GO INPUT FILE NAME
13E2 86 20            LDAA #$20     OVERLAY 'EOL'
13E4 A7 00            STAA 0,X       WITH A 'BLANK'
13E6 DE 58            LDX  SPCPT1   TEXT BEGINNING ADDRESS
13E8 FF 70 AD         STX  XBEG     STORE IN OUR AREA
13EB DE 5A            LDX  SPCPT2   TEXT ENDING ADDRESS + 1
13ED 09               DEX            CHAR
```

Advanced MINIDOS-PLUSX Programming Example

For those interested in assembly-language programming, Listing 1 shows the patches to the TSC Editor to enable it to run under the MINIDOS-PLUSX DOS. As mentioned before, MINIDOS-PLUSX works in conjunction with MINIDOS. Although the basic I/O diskette drivers in MINIDOS have a jump table and specified entry points, the higher-level directory handler in PLUSX does not. Phil had to scan through the handler and piece together parts that could be used to provide the functions needed. (Thank heaven for the source code!)

The routine called XINPUT is used to accept data from the terminal and place it into a line buffer. The routine XSAVE is used to provide a linkage into the save function. The directory can be searched via a routine called XFIND. If an entry is located, its diskette address will be returned to the calling program. This diskette address is then used to inform the I/O loader where to access the file. The loader routine MLOAD will then load a file into memory and return the next memory location after the file.

The save function involves three sequences: prompt for a file name, load parameters indicating the memory region to be saved and then the disk save function. The read function involves four jobs: prompt for file name, search the directory for the name, inform the loader of the file address and load the file into memory.

```
13EE FF 70 AF          STX  XEND     STORE IN ENDING ADDRESS
13F1 CE 00 0E          LDX  #$000E   USE EXEC AS FILE TYPE
13F4 FF 70 B1          STX  XEXEC
13F7 CE 00 00          LDX  #$0000   INDICATE LAST FILE ??
13FA FF 70 A8          STX  XFILE    USED FOR ALLOCATION
13FD BD C5 54          JSR  XSAVE    *** USE PLUSX TO SAVE IT WITH A NAME ***
1400 25 01             BCS  ER1      IF ERROR, REPORT IT
1402 39                RTS
1403 BD C3 63  ER1     JSR  CRLF
1406 BD C0 1E          JSR  MERR     PRINT ERROR CODE
1409 7E 04 41          JMP  ERROR    GO BACK TO EDITOR

140C 0D        ASK     FCB  $D,$A,0,0,0,0
1412 46                FCC  'FILE ?  '
141A 04                FCB  4
               *
               *   READ TEXT FILE FROM DISK
               *
142D                   ORG  $142D
142D CE 14 0C          LDX  #ASK     FILE MSG
1430 BD E0 7E          JSR  PDATA    GO PRINT IT
1433 01                NOP

1447                   ORG  $1447
1447 CE 70 80          LDX  #LINBUF  POINT TO OUR BUFFER
144A FF 70 A0          STX  LINPTR   STORE IT
144D BD C4 2D          JSR  XINPUT   GO GET FILE NAME
1450 BD C4 BD          JSR  XFIND    FIND IT IN DIRECTORY
1453 25 AE             BCS  ER1      MAY NOT FIND IT
1455 EE 00             LDX  0,X      IF FOUND, USE DISK ADDRSS
1457 DF 01             STX  MDISK    STORE IT IN DOS AREA
1459 DE 40             LDX  $40      PICK UP END OF TEXT POINTER
145B DF 16             STX  MTW      STORE IN TARGET AREA
145D BD C0 1B          JSR  MLOAD    ** USE DOS LOAD FUNCTION **
1460 25 A1             BCS  ER1      GO PRINT ERRORS
1462 DE 14             LDX  MTA      POINT TO NEW END OF TEXT

146D                   ORG  $146D
146D 01                NOP
146E 01                NOP
146F 01                NOP
1470 01                NOP
1471 01                NOP
               *
0272                   ORG  $0272
0272 04 41             FDB  ERROR         CHANGE GAP VECTOR- NOT USED
               *
                       END
     NO ERROR(S) DETECTED
```

Conclusions

Several factors must be weighed when considering a disk system for your SWTP computer: why you want the system and how it will be used. Some of the areas to consider are:

- ease of use
- software support
- interface to user software
- access speed
- memory requirements
- cost
- future expansion
- compatibility with other disk systems in your area
- the product's future

The final answer will be a compromise, but we hope that it will be right for you. Order a user's manual for each company and look through it. This will give you the opportunity to look at the fine points of each system before making a decision.

The Smoke Signal system was the first on the scene, but it lacks some of the finer points of a general-purpose DOS. Some of the user information is displayed in hex rather than decimal. Control of the terminal is somewhat lacking . . . or obsolete. The facilities of the monitor and file-management routines make it easy to interface to assembly-language user software. The access time is slow, but this is to be expected with dynamic allocation.

For the BASIC crowd, the SSB/Computerware BASIC works well, and the availability of Computerware's random BASIC is nice, too. All in all, a good DOS, but it needs polish to bring it up to current standards.

The FLEX system from SWTP and TSC may be the best overall DOS for the general user. Moreover, at $900 for a dual-drive disk system, including a versatile DOS and good BASIC, the SWTP disk is a clear winner in the price/value category. The other systems have certain features that FLEX does not, but the layout of FLEX seems better.

We did not like the hardware constraints from SWTP. We also weren't pleased because the bootstrap in SWTBUG didn't always work well and because we had to install a jumper on the motherboard. The disk bootstrap also clobbers memory locations 2400-2547, which destroys any programs already about that. Then there is the problem of what happens if you accidentally push the RESET switch with a disk in the drive but not running. On the SWTP system, you will probably clobber the disk.

FLEX is also quite slow. But a large part of the reason is that, after each write, FLEX does a read of the disk to make sure that it was written right. This is an important feature to anyone concerned about losing his data or programs.

The SWTP Disk BASIC version 3.0, which comes with the system, is also slow, even slower than SWTP cassette BASICs. The Computerware BASIC that comes with the SSB disk runs anywhere from 50 to 80 percent faster. (Percom also has a BASIC that runs about 50 percent faster, but in their case they charge extra for it, so that may be an unfair comparison.) SSB has a FORTRAN compiler, and TSC has a new BASIC, both of which should be faster still, but both of these are expensive options. Moreover, when you consider that editing, compiling and then executing the program require a lot of disk operations and DOS interaction, the total running time for a compiled program is often longer than using an interpreter. A compiler can often be justified only for longer programs that are compiled only once and then run many times. Besides, typical business and home programs are mostly slowed down by the I/O speed, and in that case a compiler would make little difference.

The group at TSC has thought out a user-oriented piece of software, which works well. There are some incompatibilities with the disk BASIC, and the assembler output is not a MIKBUG object file (for those who like to see their programs), but these are surmountable. The FLEX system will probably become the standard, if for no other reason than that it comes from SWTP and TSC.

Above all, the new FLEX system and the big push from TSC to come out with more software is a welcome sign. But it does raise that $900 price somewhat when you consider getting more TSC software.

The Percom system is quite different from the others. Its $600 price for a complete, wired, single-drive system puts it into a completely different ball park. Unfortunately, this price is deceivingly low. It's almost a sure bet that, not too long into the future, many Percom owners will plunk down the extra $400 or so for a second drive. Add to this $40 for MINIDOS-PLUSX, and you're above the price of the SWTP disk, getting close to the price of the SSB, without having as good a DOS.

For assembly-language programmers, the Percom disk is difficult to tie into character-oriented software. The need to squeeze the DOS into 2708 EPROMs has resulted in a DOS not easily adapted to such use. On the other hand, having the DOS in ROM is a tremendous convenience, especially in single-drive systems. It is also much less subject to being clobbered by a wayward program, and then, in turn, clobbering the disk.

Since it supports random disk files, Percom's Super BASIC is better than SWTP Disk BASIC, which doesn't. It is also less expensive than Computerware's Random Disk BASIC. But if all you want are sequential files, then SWTP and SSB provide a sequential disk BASIC for free, while you have to pay for Percom's.

The same kind of reasoning applies to Percom's INDEX operating system. INDEX is potentially much more versatile than FLEX and DOS-68 (although the new FLEX versions may equal it). But this is again an unfair comparison, since FLEX and DOS-68 are included in the price of the system, while INDEX (and the new FLEX) are extra cost options.

Now for the big question: Which is best? Which one to get? We can't quite answer that question for you. (This is an area where the two of use don't quite see eye to eye. Phil has all three disk systems on his computer and feels that the Percom is just a glorified cassette replacement, not in the same class as the SWTP or SSB systems. He thinks the SSB has the best hardware design, while SWTP's FLEX has the best software. Pete, on the other hand, owns the Percom and uses the SWTP disk at work. He has become accustomed to the simplicity of the Percom approach and feels more comfortable with it.)

Some Fixes

Here are several simple fixes for users of the SWTP MF-68 disk.

The current MF-68 uses a DC-2 controller board, but you may still have the older DC-1 controller. SWTP issued an update for that board some time ago. Add a 100 pF capacitor from the WD1771 chip, pin 31, to the ground that runs from pin 8 of IC5 (74LS139).

SWTBUG has a disk bootstrap for the SWTP disk, but it has a slight problem: it does not provide enough time for the drive motor to come up to speed before trying to load from the disk. (This was apparently due to a change in the Shugart drive, after SWTBUG was written.) To give some extra time, type the D before closing the drive door. This will start the motor, and the boot will wait for the door to close before trying to load.

When you do a reset, the head on drive 0 loads (moves up against the disk) and then retracts to track 0 while still touching the disk. It is also touching the disk during idle periods. This is somewhat awkward and, among other things, increases the chance of clobbering the disk when you push reset or if you accidentally turn off the power with the disk still in the drive. A modification is to pull out the 7407 IC at the top of the controller board and bend out pin 6 so it doesn't make contact when you replace the IC in the socket. Now the head will only load on command, and the drive will not go to track zero on a reset.

A note on the FLEX P command, used to direct output from the following command to the printer: It does not save or restore the status of the pause option, which provides a pause when you fill up a screen on a CRT terminal. If you have a printer and use it, then going back to the CRT will cause you to run off the screen. We don't know of a patch for this right now, but perhaps TSC does. ■

EXTRA SPECIAL
Prices & Delivery

PRINTERS

HEATHKIT LINE PRINTER WH-14 (Assm)	$ 825.00
CENTRONICS PRINTER #779 w/tractors & paper tray	$ 995.00
TEXAS INSTRUMENTS PRINTER #810 (Up. & low. case)	$1689.00

TERMINALS

INTERTUBE II BY INTERTEC .... $ 777.00

(Terminal) Z-80 controlled, 80 Char. by 24 lines, 128 upper and lower case, ASCII Char., graphic symbols, reversible video, half intensity video, special 25th display (status) line, 12" screen, operating mode: conversational, message, page, Char. insert/delete, line insert/delete, half or full duplex, keyboard with 18 key numeric pad, full cursor control, RS232 and 20/60 Ma loop auxiliary printer port, 75-9600 baud and a lot more special features all software controlled, no little mini-dip switches to bother with.

LEEDEX VIDEO 100 MONITOR	$ 138.00
NOVATION CAT ACOUSTICAL MODEM (300 baud)	$ 185.00

DISK DRIVES

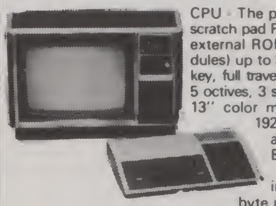
MEMOREX 550 8" Disk Drive Single or Double Density	$435.00
6' Cable for two drives	$ 25.00
Connectors for power supply (2 sets)	$ 4.50
Power Supply	$ 98.00
TARBELL DISK CONTROLLER BOARD (Kit)	$179.00
PERTEC 5¼" disk drives for TRS-80 (Flippy Disk)	$359.00

This drive gives you 40 tracks of storage versus the 35 track Radio Shack drive. The Pertec drive also has the flippy disk feature. When you have filled one side of the diskette, flip it over and use the other side.

Cable	$ 29.50
TRS-80 New Dos Plus from Apparat-drives 40 tracks	$ 65.00
DISKETTES - VERBATIM 5¼" soft sector (TRS-80) Box of 10	$ 28.50
8" Soft sector - FD 1000 Box of 10	$ 38.50

COMPUTERS

TEXAS INSTRUMENTS 99/4 HOME COMPUTER .. $1095.00


CPU - The powerful 9900 16 bit microprocessor with 256 byte scratch pad RAM. **MEMORY** - Total 72K bytes of ROM & RAM, external ROM memory - (Solid state software command modules) up to 30K bytes each, 16K user RAM. **KEYBOARD** - 40 key, full travel, overlay for second functions. **MUSIC & SOUND** - 5 octives, 3 simultaneous tones plus noice generator. **DISPLAY** - 13" color monitor. **GRAPHICS** - 16 colors high resolution 192 x 256, you can define your own characters, create animated displays, charts, graphs and more. Built-in software - 14K byte BASIC interpreter 13 digit, floating point, internal graphics language interpreter, equation calculator & internal 4.4K byte monitor, recorder interface. **PERIPHERALS** - remote controls, solid state speech synthesizer, RS-232 interface.

CROMEMCO

Z-2 (Assm)	$ 845.00
Z-2D (Assm) For rack mounting, 4MZ Z-80, Floppy Disk Controller, Mini-Floppy Disk Drive	$1779.00
Z-3 (Assm) 4MZ - Z-80, Dual 8" Drives, 32K RAM, Printer and RS-232 Interface	$4989.00
Cobol, Fortran, Assmb., DBM, & other Cromemco software	$ 86.00
TUART (Kit) $176.00 TUART (Assm)	$255.00

CALL US FOR OTHER CROMEMCO DISCOUNTS

SUPERBRAIN $2845.00

Superbrain - The most cost effective machine on the market today. Intertec has combined their excellent video terminal with a sophisticated Z-80 computer and has come up with a totally integrated package. This beauty includes a 4MZ Z-80 computer system, 2 double density Mini-Floppies with 320K of disk storage, 64K of RAM memory, CP/M disk operating system with utilities, and a complete 80 character X24 line video smart terminal. This machine boasts two Z-80 processors to handle processing & I/O. A serial and parallel interface for the I/O is included. A S-100 edge connector is provided, so upgrading to 10-300 MB hard disks is available. If you need more info, call us.

EXIDY SORCERER W/16K	$1035.00
W/32K	$1135.00

GODBOUT UNKITS
16K - $249.00 32K - $498.00 I/O Board - $184.00

VISA®

Checks, money orders accepted
Add $2.50 freight charges on orders under 10 lbs.
Over 10 lbs. F.O.B. Cleveland

WE HONOR master charge THE INTERBANK CARD

QUASAR DATA PRODUCTS ✓Q9
25151 Mitchell Dr., No.Olmsted, Ohio 44070(216)779-9387