

MAY 1981

# LIFELINES<sup>®</sup>

\$2.50

VOLUME ONE NO. 12



A Brief Review of

**PASM, BUG /uBUG, PLINK and EDIT**

Comments On **SSSFORTAN**

**BASIC** Comparisons: **SBASIC**, Part 3

**The CP/M Users Group** Volume 49,

Catalogue and Abstracts

**ZOSO**

---

---

---

---



# LIFELINES®

Volume 1 No. 12 May

Editor-in-Chief: Harris Landgarten

Managing Editor: Jane Mellin

Production Assistant: K.Gartner

CONTENTS		PAGE
Opinion	Editorial Comments	2
	Zoso	14
	Reply	17
Features	A Brief Review of PASM, BUG/uBUG, PLINK, and EDIT by Tom Cochran	3
	Comments on SSSFORTRAN by Trevor Marshall	5
	SBASIC Comparisons, SBASIC Version 5.3h Part 3 by Bill Burton	8
The CP/M Users Group	Catalogue and Abstracts, Volume 49	12
Product Status	New Products	21
	New Versions	20
	Bugs and Fixes	20
	Version List	24
Miscellaneous	Tips and Techniques	18
	Tip Contest	5
	Stop	6

Lifelines, Volume 1, Number Twelve. Published monthly. The single copy price is \$2.50 domestically, including the U.S., Canada, and Mexico. The single issue price for copies sent to all other countries is \$3.60. A one year's (12 issues) subscription is priced at \$18.00, when destined for the U.S., Canada or Mexico, \$40 when destined for any other country. All checks should be made payable to Lifelines Publishing Corporation. Foreign checks must be in U.S. dollars, drawn on a U.S. bank; checks, money orders, VISA and MasterCard are acceptable. All orders must be prepaid. Lifelines is published by Lifelines Publishing Corp., 1651 Third Ave., New York, N.Y. 10028, Telephone: 212-722-1700. Please send all correspondence to the publisher at the above address. Postmaster, send change of address to the above address. Application to mail at 2nd Class postage rates is pending at New York, N.Y.

The listed products, mentioned in this issue, are trademarks of the companies named after them: SBASIC-Topaz Programming; PLINK, PASM, EDIT and BUG/uBUG-Phoenix Software Associates, Ltd.; SSSFORTRAN-SuperSoft Associates; Z80-Zilog Inc.; M80-Microsoft; MDS-Intel; DDS-Program-of-the-Month Corp.; DEC-Digital Equipment Corporation; CP/M, MAC, DDT, SID and ZSID-Digital Research, Inc. The CP/M Users Group is not affiliated with Digital Research, Inc.

# Editorial Comments

What was the overriding theme of the West Coast Computer Faire? Not Apple, not Tandy, not Commodore; it was CP/M that dominated this year's edition of the world's largest (30,000 + attended) end-user computer show. It seems that all new microcomputer hardware being introduced, both domestic and foreign, is delivered with CP/M; no surprise considering the enormous software base.

I had the pleasure of participating in a CP/M - MP/M panel discussion with Gary Kildall (Digital Research), Bruce Kendall (Corvus Systems) and Sol Libes (S-100 Microsystems). I am told that this was the most widely attended session of the show. Gary unveiled the future plans for CP/M; namely, CP/M 3 and MP/M 2. New features will include: time and date, passwords, user 0 public, file and record lock outs, a type ahead buffer, a video ED, and more. Has Digital finally recognized the widespread use of powerful CRT terminals? CP/M 3 will contain terminal definition information in the BIOS (BDOS calls for set cursor?). Unfortunately, the end of terminal configuration programs is not in sight, since CP/M 2 and CP/M 1.4 will continue to be supported. Perhaps, some day CP/M will resolve the terminal attribute incompatibility problem at the operating system level just as disk format differences have been resolved. This seems a more likely solution than terminal manufacturers adhering to a published standard. The cost of these enhancements will be approximately 2k of additional overhead. Delivery was quoted at 60 to 90 days and no pricing information was announced. Other new products from Digital include PL/I-86 and a 8080 to 8086 intelligent translator. No delivery dates were announced for these products. Curiously, Gary seemed to indicate that CP/NET was for the moment a back burner project. He said that networking would not be economically feasible until the price of a node fell to several hundred dollars from the present \$2000 cost. I know many people who would argue that point but I'll leave that for another editorial.

The hardware hit of the show was the introduction of the Osborne Computer. This portable sewing machine size, desk top microcomputer contains 64k of memory, 2 mini floppies (single density), CRT screen, three interfaces (RS232, Parallel, and modem), and full keyboard. Furthermore, it is delivered with CP/M, Wordstar, Microsoft BASIC-80, CBASIC, and Supercalc (a CP/M Visicalc look alike). The whole package is priced at \$1800 dollars, and, needless to say, orders were being entered at a brisk pace. The most criticized oddity of this machine was the 52 x 24 CRT which acted as a window to a 128 x 24 logical screen. Horizontal scrolling is accomplished with manually activated control characters. Later models will contain 80 x 24 screens and double density disk drives (double density will be available as an update). It was rumored that the models shown were prototypes, but the machine appeared to execute the software flawlessly. Delivery should begin in June. Could this be the beginning of a desk top computer price war? Many more desk tops will be introduced at the NCC next month. I'll keep you posted.

Among the new software being shown was a product from England called "The Last One". This BASIC program, which generates BASIC code from human oriented input, is being touted as the "end of computer programming". These boasts were impressed upon attendees in a Madison Ave Hype fashion. What's more, the authors of the program claimed to have committed one million dollars to promotion (for a program generator?). Since the authors refused to give evaluation copies of the program to anyone, I am unable to tell you whether or not this product is real. In any case, I found the marketing method more befitting a 'Rock album' than computer software. "The Last One" will be watched very closely.

Harris Landgarten

# A Brief Review of PASM, BUG/uBUG, PLINK, and EDIT

## by Tom Cochran

Most of the software products recently announced by Phoenix Software Associates Ltd. (PSA) are relabeled or revised versions of software originated for Technical Design Labs (TDL), which later became Xitan. When these programs first appeared about three years ago, they were among the best software utilities available for the 8080 and Z80[1] processors. One of the program authors, Neil Colvin, apparently has helped start PSA to provide continuing support for these great products.

Because I was an early owner and fan of the predecessor TDL software and hardware products, I was anxious to see how much the programs have been enhanced. I must admit that I was slightly disappointed because PLINK was the only program to receive any significant changes. But I must also say that these products are well worth your consideration and evaluation.

The PSA software packages include a Z80 macro assembler (PASM), a linkage editor (PLINK), a Z80 debugging aid (BUG), and a powerful character-oriented text editor (EDIT). These programs were written using Z80 instructions and will run only on Z80-based computer systems. CP/M[2] or another compatible operating system is also required. In the following paragraphs I will attempt to provide a brief overview of the products and a comparison, where possible, to other products I have seen or used.

### PASM

One of the best early macro assemblers was the TDL Z80. The newer PSA macro assembler also uses an extension of the same popular instruction mnemonics originated by Intel for the 8080. This feature allows the use of the PSA macro assembler to process programs which are coded with both 8080 and Z80 instructions. All those instructions common between the 8080 and Z80 use the Intel mnemonic format; those instructions unique to the Z80 have been

structured as logical extensions to that base. The primary benefit of this mnemonic set is the extensive saving of time and effort when re-assembling or modifying assembler programs from other sources, such as The CP/M Users Group.

There are many assembly time options which can be used to specify some of the following possibilities:

- \* Intel or PSA hex format object files;
- \* absolute or relocatable addressing;
- \* unique but powerful macro processing capabilities;
- \* assembly-time assignment of variables;
- \* numerous control options of listings, titles, etc.;
- \* support for linking many program modules;
- \* multiple program assemblies in one execution;
- \* op-code synonyms to translate op-codes of other assemblers;
- \* source library file support;
- \* date and time stamping (operating system dependent).

Some of these features are very useful and worth detailing briefly. The assembly-time assignment of variables is especially advantageous for software developers wanting to set up unique information (such as serial numbers, version notices, or other personalization data) in each assembly process, without requiring the source data to be re-edited. The op-code synonym feature appears to help reduce editing of assembler-dependent op-codes when re-assembling programs written for other assemblers (such as MAC[2], M80[3], or the Intel MDS[4]).

This is a very good macro assembler with many unique features. It is inexpensive enough to warrant your evaluation.

### BUG / uBUG

These debugging aids come in two versions and two operating sizes. BUG is the full package and re-

quires 13K while uBUG is a subset version of BUG with fewer features; it operates in less memory. Both programs are similar in function to other debugging programs like DDT[2], SID[2], ZSID[2] and DDS[5]. However BUG/uBUG do not support symbol files like SID and ZSID.

BUG has five functional modes - byte, word, three-byte, fourbyte, and instruction. These modes are used in conjunction with the different commands to display, modify, and execute programs and data in memory. Commands are provided for the functions listed below.

- \* Display data
- \* Disassemble programs
- \* Modify data
- \* Modify programs
- \* Fill memory
- \* Load COM, HEX, or REL files
- \* Enter ASCII data
- \* Search for data or instruction
- \* Examine and modify breakpoints
- \* Execute programs with registers and flags optionally displayed
- \* Set program traps

The BUG loading process is similar to DDT in its command format but the operation is unique. BUG.COM is simply a loader program which reads and relocates the object file BUG.REL to the highest memory available. Loading just the debugging program takes considerably longer than loading DDT or SID. Both DDT and SID require about three seconds on my system to load and relocate, while BUG requires about 23 seconds to load and relocate. The size of the disk files for BUG is also about twice that of DDT or SID because it resides in hex format.

Some good points of the BUG programs are the searching, breakpointing, and trap-setting functions. These features are not available in DDT and SID. I have often used both BUG and DDT to debug, or to better understand what a particular program is doing. I don't use BUG alone due to a very cumbersome feature of its command syntax. The Display

command is used to list disassembled instructions. Each time you employ the Display command, its format requires a starting address as well as the number of instructions to be listed. Compared to DDT or SID, this is quite annoying.

The Search function permits the scanning of memory. Depending on the current mode, BUG will list the addresses of all bytes, words, three-byte groups or any instructions it finds. The ability to limit the scan with a from and to address range is also a big benefit. A typical request for BUG to do a search would be

```
>Y100,0FFF
```

which initiates the function and limits the scan to memory between hex addresses 100 and FFF. Then by entering

```
CPI OD
```

the resulting list of addresses would be placed where the exact instruction CPI OD existed. This example assumed we were in the instruction mode. The same ability is provided on bytes, words, three-byte, and four-byte groups.

The breakpoint and trap-setting functions are slightly related. Breakpoints in BUG operate as they do in DDT. The exception is that up to seven breakpoints can be set at once in BUG; only two can be set at once with DDT. The significant trap-setting function is especially useful in determining such events as the changing of values within a program, or the loading of an address into the program counter (PC). Up to four traps can be set with each trap performing a different conditional test. The traps only stop execution during tracing operations, since it is only in this mode that registers, program counters, and memory locations can be checked. The types of tests may use Boolean logic to help isolate the exact condition you may want to test. To show a few examples of this function requires some definition of the notations used:

```
<n specifies a register value (i.e. <A, <BC, <B, etc.).  
>n specifies a flag value (i.e. >C returns the bit value of the carry flag).
```

Optionally, the exclamation char-

acter (!) denotes that the register value is prior to the last instruction trace; a quote denotes that the alternate Z80 register is used. A few examples may help to understand this capability.

```
<A ?NE <!A  
says to stop as soon as reg A changes.
```

```
<PC ?EQ 34F  
says to stop when program counter is at hex location 34F.
```

```
<PC ?GE 1000 & <PC ?LE 1296  
says to stop when program counter is between hex 1000 and 1296.
```

Additionally, the operators @, @, and ! are used with an address argument to return, respectively, the word, byte or instruction at specified address. The symbol "." refers to the current program counter value (same as <PC).

```
@100 returns the word at hex location 100.
```

```
\<HL returns the byte pointed by register pair HL.
```

```
!<PC or !. returns the instruction pointed to by program counter.
```

While this is certainly not an all-inclusive list of the possible expressions, it should give you a hint of the power of BUG's conditional trap function.

#### PLINK

PLINK is a program which can take separately compiled modules in object file format and produce a single load (.COM) file that can be executed by the operating system. PLINK can accept both Intel hex format and the PSA(TDL) relocatable format object files as input. Output files can be either the standard load file or another single hex file. Additionally, the Microsoft relocatable format (.REL) files are accepted as input to this linkage editor.

The primary advantages of "linking" several program modules which may be created, compiled, and tested independently (rather than in a large single program) may not be obvious to the nonprofessional programmer. The simplicity of functionally decomposing a desired task into small and unique sub-

tasks or modules greatly improves the programming process. In this way, programs can be well designed with significant reductions in time and effort, by reducing complexities and shortening the debugging and testing phases of a program's development. This is a standard procedure used by nearly all professional programmers; it also reduces the ongoing maintenance of programs.

PLINK is a two-pass linkage editor that reads the input and builds the output on disk. Programs can be created which are possibly larger than the machine performing the link edit. Many separate library files may be accessed with options to select all or only some of the modules of the library for input.

PLINK may be used either interactively or in batch mode. If complex or repeated linkages are required for a particular application, linkage command files (.LNK) can be created to simplify the entering of linkage command statements for a program consisting of many modules.

#### EDIT

The EDIT text editor from PSA is a powerful character-oriented processor which I understand is similar the DEC(6) system's TECO editor. This editor has many features which can assist in the process of editing either large files or massive changes within the files. There are very nice facilities for moving blocks of text, for the iteration of commands, and for invoking command macros. Features such as a number of value and text registers are provided for storing information about the file and for aiding in the moving, copying, and appending blocks of text from one area to another. The text registers may also be used in conjunction with the macro facility to permit complex edit functions from simple input.

While this is the best character-oriented text editor I have used, it has no full-screen video or cursor controlled modes. This is a severe drawback since most hobbyists today seem to prefer the simple and clean video mode editors. (Note: I'm told that PSA has

announced a new version of this editor which does support full screen video mode.)

In summary, I am pleased that Phoenix Software Associates will give new life in the support and hopefully enhancements to some very good software. I was slightly disappointed that there were not more enhancements to these programs over my earlier TDL versions, but I trust that remains in the future.

One last comment should be made about the quality and quantity of documentation. I would compare PSA's documentation with Digital Research's on CP/M. I believe the documentation is fairly complete and sufficiently detailed to answer most questions, but it is very difficult to read. Most of it is written much too technically for many amateurs, both beginning and experienced, to easily follow and comprehend.

As mentioned earlier, I feel these software products are very good and worthy of your consideration and evaluation using your own criteria. I personally am looking forward to new releases and enhancements.

## Tip Contest

Lifelines just wants to let new subscribers know that the Tip Contest is still under way. Send your tip and/or technique to Lifelines Tip Contest, 1651 Third Avenue, New York, N.Y. 10028. Each month the best entry will be selected and published. The winner will not only attain great renown, but will also receive a check for \$50. You can't lose by trying, so send your entries to us. We're sorry, but we can't return your entries to you.

A hint to tip-writers: readers are asking us for information on the "nuts and bolts" of CP/M. We intend to oblige, but hope any subscribers who are hoarding special CP/M morsels (or thinking about writing some) will mail them in.

# Comments on SSSFORTRAN

## Trevor Marshall

As a user of Microsoft FORTRAN (Cromemco licensed version) for several years, my interest was aroused by recent advertisements for a 'new' FORTRAN being marketed by SuperSoft.

Although buying unknown software is a risky business, most of us are aware of the limitations of the Microsoft product. So these advertisements, promising solutions to many of the problems, spurred me to action. Having been assured that the product was not really new, but a mature product just newly released, I ordered the combined FORTRAN/RATFOR package.

My experience with this product has not yet been sufficient to discover all its features, but I have been so impressed that I think it imperative that other readers hear about SSS FORTRAN.

This critique logically splits into two sections: that describing the software itself; and that describing the documentation.

The FORTRAN compiler is described as being totally ANSI (1966) FORTRAN compatible, including complex variables, and having many extensions. The extensions are primarily in the area of string handling. For instance a filename extension can be changed thus:

```
CHARACTER*14 INNAME
INTEGER PNTR,UNIT,MODE,DEV
C Unit 1 is the console
WRITE (1) " What is the input filename [B:name.ext]..?"
READ (1) INNAME
C Open the input file for reading
CALL IOREAD(UNIT,MODE,DEV,INNAME)
C Now change the extension to .PRN, first locate a pointer to the '!'
PNTR=ITABLE(INNAME,1,'.')
CALL PUTCHR(INNAME,PNTR,'P')
CALL PUTCHR(INNAME,PNTR+1,'R')
CALL PUTCHR(INNAME,PNTR+2,'N')
C Now open the .PRN file for writing
CALL IOWRIT(UNIT,MODE,DEV,INNAME)
```

Parsing a filename suitable for the file I/O routines of MFORTRAN is a much messier business. (I can't help feeling that one of the other routines in the string library would put all three 'PRN' characters at once, but I can't decipher the documentation, more of that later.)

Another major limitation of MFORTRAN is its inability to place (reasonable) arithmetic expressions in array subscripts or subroutine calls (eg PNTR+1 above). SSSFORTRAN allows them.

To cut a long story short, here is a summary of the features of SSSFORTRAN which appeal most to me---

- 1) HEX format variables can be input and output via the Zn format field.
- 2) Free format input statements are allowed.
- 3) Logical I/O formats are supported.
- 4) Subroutine call may be by value or location.
- 5) Multiple statements can be placed on a line, BUT I have not yet been able to place a comment on the same line as code. Use of columns 72-80 for (short) comments is not allowed.
- 6) X.AND.!FBCO! (logical masking) is supported on INTEGERS (note the non-standard use of !FBCO! to denote Z'FBCO' (OFBCOH in INTEL).

(continued next page)

- SSSFORTRAN distinguishes between LOGICAL, INTEGER\*1 and CHARACTER variables.
- 7) Core to core I/O is supported.
  - 8) Format free I/O does not waste a full sector for each of the output statements.
  - 9) Runtime errors are less numerous (but adequate), though many I have received are not yet in the manual.
  - 10) Runtime errors reference the LUN in which the error occurs.

Core requirements were much less using LOADER than LINK-80. The loader is only 4K long, using a non-standard .REL format.

Perhaps the most significant enhancement relates to the ease of compilation/debugging of the source code. When an error occurs a diagnostic cross-referencing 200 error messages (printed in the manual) is output to the console.

Compilation speed was comparable with MFORTRAN and the compiled code was slightly smaller (37K vs 40K for the author's 6805 cross-assembler). The .COM file ran at about the same speed as the MFORTRAN version.

There are, of course, several problems yet to be solved by SSSFORTRAN.

REWIND and BACKSPACE directives are not really supported, merely placing a call to a user-written routine 'REWIND' or 'BACKSPACE'.

Use of a non-standard .REL format precludes the use of a DISK TO DISK linker such as PLINK, reducing the maximum size of programs which can be compiled. Too little information about the .REL format is available to contemplate writing a disk linker.

The .REL libraries are difficult to use, and could perhaps be merged.

A Z80 assembler suitable for producing non-FORTRAN linkable modules has not yet been released (it may have been when this article gets to press). Let's hope it supports ZILOG format (or that somebody writes a ZILOG to INTEL translator).

The documentation, although it is 71 pages long, does not do justice to the compiler. Many runtime errors I have received are not documented. Paragraph 3 on page 56 refers the reader to another section which does not exist. The string handling routines include an obvious (but fatal) typographical error and are described less succinctly than the (familiar) library function routines.

It should not be assumed that the user of such documentation is familiar with WATFIV FORTRAN. If the contents of a book are necessary to use the compiler it should be included, or easily available.

Finally, two pleas to compiler writers:

PLEASE allow variables to be longer than 6 characters, even if only 6 are significant. The days of programming in cipher should be long gone (e.g., PNTR is needed instead of POINTER).

PLEASE allow some mechanism for the placing of frequent comments throughout the source.

Much of the criticism of FORTRAN readability directly results from these two restrictions.

It can be argued that RATFOR eases comment placement, but conversely it also manages to mess up any FORTRAN passed through it.... still, they have supplied the source to the RATFOR precompiler.... one day I might get round to fixing the incompatibility.

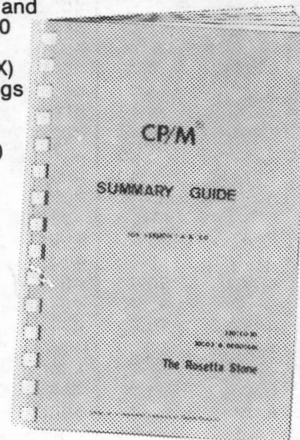
In summary, the purchase of SSSFORTRAN has enabled me to contemplate the implementation of many programs I could not have previously compiled.

# STOP

...And renew your subscription, if it began last June or July. Are you sure you already sent in your renewal order? Don't forget, or you may find yourself missing out. Be sure to enclose your mailing label (or a facsimile of it) so that Lifelines may process your renewal efficiently. We're looking forward to a great year, with all kinds of reviews and tips in store for you.

## CP/M SUMMARY GUIDE

Tired of fanning through your CP/M manuals or writing notes that remind you of the commands, functions and error codes? Well it's about time you ordered our CP/M Summary Guide! Spiral bound and handy to hold, our guide is a 60 page booklet summarizing the features of CP/M (Ver. 1.4 & 2.X) and 2 totally alphabetical listings of the commands, functions, statements and error codes of MICROSOFT BASIC-80 Ver. 5.0 and CBASIC™.2. Areas summarized are in table form and include all direct and transient commands plus MAC™, DESPOOL™ and TEX™. Our booklet is a much needed supplement to any of the literature currently available on CP/M and has been recommended by Digital Research.



P.S. Over 4000 users can't be wrong!

Ask your local computer store for our guide or send \$6.95 plus \$1.00 (postage and handling) to:

THE ROSETTA STONE, P.O. BOX 35, GLASTONBURY, CT 06025 (203/633-8490)

Name \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

CP/M™, DESPOOL™, MAC™ are registered trademarks of Digital Research. CBASIC™ is a registered trademark of Compiler Systems.



# MICROPROCESSOR SUPPORT I.C.'S

## WE GUARANTEE FACTORY PRIME PARTS

We are going to become the largest supplier of prime microprocessor support I.C.'S. We guarantee that our I.C.'S are purchased from manufacturer authorized distributors. This is the only way to deliver prime parts at the lowest possible prices. Our commitment is to offer the best price and the fastest delivery to our customer. We give many thanks to our valued customers who have helped us grow.

**NEC 16Kx1 DYNAMIC RAM 200 N.S.**  
These are prime 4116's from one of the best MOS RAM manufacturers in the world.

**4116** **200ns**  
**8 for \$25.00** **32 for 96.00**  
Only prepaid orders on this special.

8080A	CPU	4.95	2708	EPROM 1Kx8	4.95
8085A	CPU	8.95	2716	EPROM 2Kx8	8.95
8086	CPU	99.95	2732	EPROM 4Kx8	21.00
8088	CPU	44.95	4118	STATIC 1Kx8	15.00
Z-80	CPU	10.50	4164	200ns 64Kx1 Call	
Z-80A	CPU	12.95	Z80B	CPU	21.00

**NEC 1Kx4 STATIC RAM 250 N.S.**  
These are prime low power static ram's NEC for the finest in MOS MEMORY.

**2114L** **250ns**  
**8 for \$25.00** **32 for \$96.00**  
Only prepaid orders on this special.

Z80-P10	7.75	8289	49.95	4050	.69	4531	.99	74C925	6.95	74LS107	.45	74LS244	1.95
Z80A-P10	9.75	4000	.35	4051	1.10	4532	1.25	74LS00	.35	74LS109	.45	74LS245	4.95
Z80-CTC	7.75	4001	.35	4052	1.10	4539	.99	74LS01	.28	74LS112	.49	74LS247	1.10
Z80ACTC	9.75	4002	.35	4053	1.10	4543	1.99	74LS02	.28	74LS122	.55	74LS248	1.10
Z80-DMA	22.25	4006	1.39	4055	3.95	4553	3.50	74LS03	.28	74LS123	1.19	74LS249	1.69
Z80A-DMA	27.75	4007	.29	4056	2.95	4555	.75	74LS04	.39	74LS125	1.35	74LS251	1.79
Z80-S10/0	24.95	4008	1.39	4059	9.95	4556	.75	74LS05	.28	74LS126	.89	74LS253	.98
Z80A-S10/0	29.95	4009	.49	4060	1.39	4581	1.99	74LS08	.39	74LS132	.79	74LS257	.98
Z80-S10/1	24.95	4010	.49	4066	.75	4582	1.01	74LS09	.39	74LS136	.59	74LS258	.98
Z80A-S10/1	29.95	4011	.35	4068	.35	4584	.55	74LS10	.28	74LS138	.89	74LS259	2.95
Z80-S10/2	24.95	4012	.29	4069	.35	4585	.99	74LS11	.39	74LS139	.89	74LS260	.69
Z80A-S10/2	29.95	4013	.49	4070	.49	4702	9.95	74LS12	.39	74LS145	1.25	74LS261	2.49
3205	3.45	4014	1.39	4071	.35	74C00	.39	74LS13	.47	74LS148	1.49	74LS266	.59
3242	10.00	4015	1.15	4072	.35	74C02	.39	74LS14	1.25	74LS151	.79	74LS273	1.75
8155	11.25	4016	.59	4073	.35	74C04	.39	74LS15	.39	74LS153	.79	74LS275	4.40
8185	29.95	4017	1.19	4075	.35	74C08	.49	74LS20	.26	74LS155	1.19	74LS279	.59
8185-2	39.95	4018	.99	4076	1.29	74C10	.49	74LS21	.38	74LS156	.99	74LS283	1.10
8202	45.00	4019	.49	4078	.35	74C14	1.65	74LS22	.38	74LS157	.99	74LS290	1.29
8205	3.45	4020	1.19	4081	.35	74C20	.39	74LS26	.39	74LS158	.75	74LS293	1.95
8212	2.00	4021	1.19	4082	.35	74C30	.39	74LS27	.39	74LS160	.98	74LS295	1.10
8214	3.95	4022	1.15	4085	1.95	74C32	.99	74LS28	.39	74LS161	1.15	74LS298	1.29
8216	1.85	4023	.38	4086	.79	74C42	1.85	74LS30	.26	74LS162	.98	74LS324	1.75
8224	2.65	4024	.79	4093	.99	74C48	2.39	74LS32	.39	74LS163	.98	74LS347	1.95
8226	1.85	4025	.38	4099	2.25	74C73	.85	74LS37	.79	74LS164	1.19	74LS348	1.95
8228	5.00	4026	2.50	4104	1.99	74C74	.85	74LS38	.39	74LS165	.89	74LS352	1.65
8238	5.45	4027	.65	4501	.39	74C85	2.49	74LS42	.79	74LS166	2.49	74LS353	1.65
8243	4.65	4028	.85	4502	1.65	74C89	4.95	74LS47	.79	74LS170	1.99	74LS363	1.49
8251A	5.55	4029	1.29	4503	.69	74C90	1.85	74LS48	.79	74LS173	.89	74LS365	.99
8253	9.85	4030	.45	4505	8.95	74C93	1.85	74LS51	.26	74LS174	.99	74LS366	.99
8255A	5.40	4031	3.25	4506	.75	74C95	1.85	74LS54	.35	74LS175	.99	74LS367	.73
8255A-5	5.40	4032	2.15	4507	.95	74C107	1.19	74LS55	.35	74LS181	2.20	74LS368	.73
8257	9.25	4033	2.15	4508	3.95	74C151	2.49	74LS73	.45	74LS190	1.15	74LS373	2.75
8257-5	9.25	4034	3.25	4510	1.39	74C154	3.50	74LS74	.59	74LS191	1.15	74LS374	2.75
8259A	7.30	4035	.95	4511	1.39	74C157	2.10	74LS75	.68	74LS192	.98	74LS375	.69
8271	60.00	4037	1.95	4512	1.39	74C160	2.39	74LS76	.45	74LS193	.98	74LS377	1.95
8275	32.95	4040	1.29	4514	3.95	74C161	2.30	74LS78	.65	74LS194	1.15	74LS385	1.95
8279	10.80	4041	1.25	4515	3.95	74163	2.39	74LS83	.99	74LS195	.95	74LS386	.65
8279-5	10.80	4042	.95	4516	1.69	74164	2.39	74LS85	1.19	74LS196	.89	74LS390	1.95
8282	6.70	4043	.85	4519	.99	74173	2.59	74LS86	.45	74LS197	.89	74LS393	1.95
8283	6.70	4044	.85	4520	1.39	74174	2.75	74LS90	.75	74LS221	1.49	74LS395	1.70
8284	5.85	4046	1.75	4522	.99	74C175	2.75	74LS92	.75	74LS240	1.95	74LS399	2.95
8286	6.70	4047	1.25	4526	1.15	74C192	2.39	74LS93	.75	74LS241	1.90	74LS424	2.95
8287	6.70	4048	.99	4527	1.75	74C193	2.39	74LS95	.88	74LS242	1.95	74LS668	1.75
8288	25.40	4049	.69	4528	.99	74C195	2.39	74LS96	.98	74LS243	1.95	74LS670	2.29

MAIL ORDERS SHOULD BE SENT TO:  
P.O. Box 21432 Seattle, Washington 98111  
Telephone Orders & Inquiries (206) 453-0792  
Minimum Order \$10.00 Add \$3.00 Shipping

**HANLEY ENGINEERING**  
RETAIL STORE  
1644 116th NORTHEAST  
BELLEVUE, WASHINGTON 98005

**FOR THE FINEST IN MICROPROCESSOR SUPPORT I.C.'S**

# SBASIC version 5.3h, part 3 bill burton

Note: One of the most important points mentioned in the last installment involved suggestions from S-BASIC's author for ordering elements of an S-BASIC program (e.g. declarations). The accompanying diagram which appeared last month was in error. The correct version appears below.

DECLARE (in this order):

- 1: COMMON variables
- 2: Variables global to program, procedures and/or functions
- 3: PROCEDURES [and their local variables]
- 4: FUNCTIONS [and their local variables]
- 5: Variables local to program only

<followed by the beginning of the main program which might include:>

6: Static DATA definition

S-BASIC offers a number of useful extensions to standard BASIC. The TEXT command (discussed last month) is just one of S-BASIC's many innovative features. This installment continues with some others.

ECHO OFF/ON - This command is used to toggle echo (display) of console input. If this command is not used all input will be echoed. One obvious use of the ECHO command would be to help conceal a system access code. For example:

```
PASSWORD="Hello"
```

```
PROCEDURE LOGIN
  FLAG=0
  ECHO OFF
  INPUT "ENTER PASSWORD "; TEST.STR
  CASE TEST.STR OF
    PASSWORD : FLAG=-1
  END
END
```

```
BEGIN
  WHILE FLAG=0 DO
    LOGIN
  END
ECHO ON
<rest of program...>
END
```

In this simplified example, the WHILE loop will be executed until the password, 'Hello', is entered exactly. It is also worth noting that the CASE statement may be used like IF-THEN to test a single condition.

COMMENT - Multiline comments may be included in a source program with the COMMENT command, which is a

shorthand replacement for [a series of] REM statements. Everything appearing between a COMMENT and its matching END statement will be ignored by the compiler. For example:

```
COMMENT
  <multiline REMark/Comment>
END
```

MID\$/MID - This intrinsic function (note the '\$' is optional in this and other S-BASIC string manipulation commands) resembles that of the 5.xx releases of Microsoft's BASIC-80 products as it permits substring assignment.

```
VARIABLE A,Z = STRING
VARIABLE B,C = INTEGER
A = "ABCDEFGHIJKLMN"
Z = (SEE BELOW)
B = 4
C = 3
MID (A,B,C) = Z
```

In this example, some or all of the characters comprising string 'Z' will replace those specified in the MID expression. When the above sequence is executed, replacement will begin at the fourth character of string 'A' for up to three characters, or the length of string 'Z', whichever is less.

```
IF Z = "yyyyyy" THEN
  A becomes "ABCyyyyGHIJKLMN"
```

```
IF Z = "." THEN
  A becomes "ABC.EFGHIJKLMN"
```

To duplicate this additional capability of MID/MID\$ in standard BASIC would require more elaborate coding.

CHAIN - Any .COM type files produced by S-BASIC may be loaded and executed by the CHAIN command. In the example below, <expr.> is any valid CP/M .COM filename.

```
CHAIN <expr.>
```

COM/DIM COMMON - These statements are used to assign common storage for chained modules. S-BASIC's chaining is surprisingly fast. If a large program is implemented as several small chained modules which share large common storage, disk activity can appear almost transparent.

EXECUTE - This command allows an S-BASIC program to invoke any .COM type file. One or two arguments are permitted.

```
EXECUTE <arg. 1>[, <arg. 2>]
```

The optional second argument to the EXECUTE command

processes an alternate valid CP/M command line if the program called by the first argument returns to the operating system (typically due to fatal error). Although EXECUTE is generally used only as a part of larger programs, there are some cases where a single EXECUTE statement might be a useful, stand-alone program.

Possible uses might include:

```
VARIABLE TEST=STRING
TEST="TEST.COM"
```

```
1: EXECUTE "STAT.COM"
2: EXECUTE TEST, ED TEST.BAS
3: EXECUTE TEST, TEST
```

Example 1 would merely call CP/M's STAT from BASIC. Examples 2 and 3 illustrate more sophisticated possibilities. In example 2, if TEST (TEST.COM) returns to the operating system due to unexpected error, S-BASIC would initiate a further edit of TEST.BAS using the CP/M editor. Example 3 demonstrates another approach. Here, one might test the thoroughness of error trapping by trying to make TEST fail in every conceivable way. By noting the conditions under which fatal errors return to the operating system, the programmer can isolate problem areas. EXECUTE may also be used to duplicate the function of CP/M's SUBMIT utility in BASIC.

It is important to note that EXECUTE does not preserve any common storage and therefore should never be used to call any S-BASIC program containing undeclared COM/Common references.

Entering 'ctl.C' as input to programs compiled by S-BASIC, CBASIC or the BASIC-80 compiler forces a return to the operating system, which can result in catastrophic loss of file data. In all of these languages, it is easy to bypass this potential problem if programs are written to process single character input only. Unfortunately, there are always circumstances in which single character input will not suffice. For example:

```
INPUT "Enter Date "; DT$
INPUT "Enter Name "; NM$
INPUT "Enter Code#"; CN
```

In such cases, it is not easy to protect CBASIC or compiled BASIC-80 programs from inadvertent input of ctl.C.

CONTROL.C.TRAP ON/OFF - When OFF, this S-BASIC command assures that ctl.C input will never cause the operating system to be returned by ctl.C entry, regardless of how many characters are permitted by an INPUT statement.

S-BASIC supports three variants of the standard INPUT statement:

1: INPUT1 generates a question mark prompt, but will not advance to the next line after input.

2: INPUT2 will advance to the next line, but will suppress the question mark prompt.

3: INPUT3 skips the question mark and suppresses advance to the next line.

By comparison, only the equivalent of INPUT2 is offered by the LINE INPUT/INPUT LINE commands of the other BASICs.

S-BASIC supports 'I/O channels' for the PRINT and INPUT statements. See figure 1. The dummy channels shown in figure 1 refer to non-existent devices. Input may be terminated either by 'CR' or 'ESC'. If 'ESC' is used, it will be stored as part of the line. Console status may be tested by channel #4 which will return OFFH if a key has been pressed. If no key has been pressed, channel #4 will return 0 and will also set the Carry flag of the PSW.

XLATE translates strings by using each successive character of a source string as an index into a predefined table string. The table string corresponds to each valid ASCII character and therefore must always be 127 characters long. XLATE would most likely be used for translation to upper case, but it also lends itself to data encryption, especially as more than one table string may be defined.

XLATE (source string, table string)

The following illustrates one way in which XLATE can be used for lower to upper case conversion.

```
VAR A = STRING:127
VAR B,C = STRING
VAR J,K = INTEGER
```

```
PROCEDURE MAKE.TABLE.STRING
FOR K=1 TO 127
  J=K
  IF K>96 AND K<123 THEN J=K-32
  A=A+CHR(J)
NEXT K
END
```

```
BEGIN
MAKE.TABLE.STRING
INPUT "ENTER SOURCE STRING ";B
C=XLATE(B,A)
<remainder of program>
END
```

Referring to the above segment, the function performed by the line, C=XLATE(B,A), could be coded in other BASICs as follows:

```
C$=""
FOR K=1 TO LEN(B$)
C$=C$+MID$(A$,ASC(MID$(B$,K,1)),1)
NEXT K
```

Although this accomplishes the same thing as XLATE, it is still an emulative approach which will produce

slower code. For simple tasks such as case translation, speed differential would likely be unnoticeable. Some other uses might require the XLATE function (or equivalent) to be repeated many times. In such cases, differences in execution speed would be apparent.

A sophisticated use of S-BASIC's XLATE command, which implements 'SOUNDEX', can be seen in figure 2 (See COMMENTS in figure 2 for explanation).

S-BASIC's file handling is especially powerful. Up to 32 random or sequential disk files may be active at one time. Random files in S-BASIC 5.3 may only be stored in binary format. (S-BASIC 5.4, scheduled for imminent release, will also support random files stored as ASCII. I feel that the choice of these two formats makes the wait for 5.4 worthwhile).

S-BASIC offers several random file management capabilities not available in other CP/M BASICs. For example, S-BASIC permits partial access to a random record which will not corrupt file or record pointers. In S-BASIC one might access a random record, read one or more fields within that record, perform computation using data read from the partial record and finally continue file operations beginning with the first unread byte in the selected record. More importantly, any variable type may be placed in the random buffer. Some advantages of these implementations include:

1: S-BASIC's provision for all six legal variable types within the random buffer means that random records need not be represented only as binary strings (as in BASIC-80) or ASCII strings (as in CBASIC). In fact, random files in any of the six valid data types (plus straight ASCII) may all be processed by a single program.

2: S-BASIC's direct access to the random buffer can support very fast file I/O. This is especially evident when random file data is binary numeric. Since S-BASIC can pass any variable type through the random buffer, neither string conversion nor FIELDing are required, (as in BASIC-80).

3: Entire arrays may be passed to and from any random record through the random disk buffer.

These examples should suggest ways in which very efficient file I/O can be coded in S-BASIC.

```
$INCLUDE <FILE name> [<module name>] This compiler option supports powerful library management. The simplest case is when <module name> is not given. <FILE name>.BAS is inserted into the source program and compiled. Thus a long program could appear to the compiler as a series of $INCLUDE statements. If <module name> is given, then only the named module within the specified library file. Modules are framed with the $MODULE and $END.MODULE commands. For example, the module, CIRCLE.AREA, might appear as follows in the library file, LIBRARY.BAS:  
REM THIS IS LIBRARY.BAS
```

```
<some modules>  
$MODULE CIRCLE.AREA  
REM DECLARATIONS IN MAIN PROGRAM  
  INPUT "ENTER RADIUS "; R  
  A = PI*(R@2)  
  PRINT "THE AREA IS "; AREA  
$END.MODULE  
<more modules>
```

This module would be included in a source program like this:

```
VAR A,R,PI=REAL  
PI=3.14159  
$INCLUDE LIBRARY CIRCLE.AREA  
<etcetera>
```

As module names are not symbolic to the compiler, they may be used as names for variables, procedures or functions.

Integer constants may be defined at compile time.

```
$CONSTANT <NAME>=<INTEGER VALUE>
```

It is important to note that \$CONSTANT is used to declare constant values only. These values may not be changed at run-time.

#### SUMMARY

S-BASIC is a 'large' language (like Pascal), and although this review has appeared in three installments, there are many features which could not be covered.

No language is truly all-purpose. S-BASIC is a business oriented language, and as such, might be less than ideal for [scientific] applications which require high volume number-crunching. Also, S-BASIC is a compiler oriented language, and generally compiler-only languages do not suit the learning needs of beginners.

I would fault S-BASIC only in two areas: the way in which compiler errors are reported (not explained in the manual) and the manual itself. A single error in the source can generate multiple compiler error messages. To an untrained eye, these multiple messages may make it difficult to isolate the error. (To a lesser degree, this same problem exists with CBASIC and the BASIC-80 compiler.) Although the manual is good and quite well organized, it has not been written with the inexperienced programmer in mind; more examples and sample programs on the distribution disk would be of help here.

Despite these minor complaints, I am very enthusiastic about S-BASIC. It is an extremely powerful language with many innovative extensions and conveniences. Mr. Ohnysty has borrowed the best from several other languages and given us a remarkably useful product. My (often repeated) reactions to S-BASIC have been "Why hasn't someone done [it] this way before?" and "Why haven't other BASICs included [these] features before?"

The suggested retail price of S-BASIC is \$295. A separate license is required to sell programs compiled by S-BASIC. This license is available for a one time fee of \$750. In my opinion, both of these prices are quite reasonable.

In the months to come, we may expect to see even further enlargements and more features. I have found S-BASIC to be very reliable, and this is not usually the case with new software products. It will be interesting to see what happens to S-BASIC when it has been on the market as long as the other BASICs which have been reviewed in Lifelines. Within the next year, I expect to prepare a follow-up report.

Rating of S-BASIC:

- 1: Ease of programming - - - \*\*\*\*
- 2: Available features - - - \*\*\*\*\*
- 3: Transportability - - - \*\*\*  
(see note)
- 4: Documentation - - - - - \*\*\*\*
- 5: Error handling - - - - - \*\*\*\*  
(see note)
- 6: Reliability - - - - - N/A  
(see note)
- 7: Vendor support - - - - - N/A  
(see note)
- 8: File handling - - - - - \*\*\*\*\*

LEGEND

- \*\*\*\*\* Excellent
- \*\*\*\* Very good
- \*\*\* Good
- \*\* Fair
- \* Poor

Notes regarding the ratings above:

3: It is easy to recode standard BASIC programs to run under S-BASIC; the reverse may not be the case.

5: Downrated slightly because of compiler error reporting. See above.

6: The probable rating would be either Very good or Excellent, but it is not appropriate to rate the reliability of new products.

7: Cannot be determined for new product.

FIGURE 1 - S-BASIC I/O CHANNELS

CHANNEL#	INPUT	OUTPUT
0	Console	Console
1	Dummy	List
2	Dummy	Punch
3	Reader	Dummy
4	Console status	Dummy
5	Keyin	Keyout (Dummy)

FIGURE 2 - 'SOUNDEX' IN S-BASIC

```

COMMENT
THIS FUNCTION IS FOR USE IN SEARCHING DATA RECORDS IN WHICH THE KEY WORD, SUCH
AS A PERSON'S NAME, MAY HAVE BEEN MISSPELLED. THE GOAL IS TO TRANSFORM THE
ARGUMENT (NAME) INTO CODE WHICH GROUPS VARIANTS OF THE SAME NAME. THE FOLLOWING
METHOD CALLED 'SOUNDEX' WAS ORIGINALLY DEVELOPED BY MARGARET ODELL & ROBERT
RUSSEL [CF. U.S. PATENTS 1261167 (1918) AND 1435663 (1922)]. FOR A MORE COM-
PLETE EXPLANATION SEE DONALD KNUTH, THE ART OF COMPUTER PROGRAMMING, ADDISON-
WESLEY, PAGES 391 AND 392.
The 'SOUNDEX' function appears courtesy of Gilbert Ohnysty.
END
REM SNDX.TBL IS USED TO TRANSLATE NAMES INTO SOUNDEX CODES.
VAR SNDX.TBL = STRING:128
VAR SRC.S, OBJ.S = STRING
VAR INDX = INTEGER
REM INSIGNIFICANT CHARACTERS APPEAR AS PERIODS IN SNDX.TBL
SNDX.TBL = \
".123.12..22455.12623.1.2.2.....123.12..22455.12623.1.2.2....." + \
INDX = 0
PROCEDURE PRINT.4
PRINT
PRINT
PRINT
PRINT
END
FUNCTION SOUNDEX (TXT.S = STRING) = STRING
VAR FIRST.CHAR ; FIRST CHARACTER OF NAME
LAST.CHAR ; USED TO AVOID SAME CODE TWICE
= CHAR
REM A CHARACTER OVERLAY ONTO TXT.S
DIM BASE CHAR TXT (80)
VAR ADDRESS, I1, I2 = INTEGER
LOCATION VAR ADDRESS = TXT.S
LOCATE TXT AT ADDRESS
FIRST.CHAR = TXT (1)
IF FIRST.CHAR >= 61H THEN FIRST.CHAR = FIRST.CHAR - 20H
REM CONVERT ASCII INTO SOUNDEX CODES
TXT.S = XLATE (TXT.S, SNDX.TBL)
I1 = 1
LAST.CHAR = TXT (1)
TXT.S = RIGHT (TXT.S, 2)
REM FORM SOUNDEX CODE OF FORM LETTER, DIGIT, DIGIT, DIGIT
FOR I2 = 1 TO LEN (TXT.S)
IF TXT (I2) = ' ' THEN LAST.CHAR = ' ' ELSE BEGIN
IF LAST.CHAR <> TXT (I2) THEN BEGIN
TXT (I1) = TXT (I2)
LAST.CHAR = TXT (I2)
I1 = I1 + 1
END
END
NEXT I2
REM ADD TRAILING ZEROS IF NECESSARY
WHILE I1 < 4 DO BEGIN
TXT (I1) = '0'
I1 = I1 + 1
END
REM FORCE AN END OF STRING MARK
TXT (4) = 0
END = FIRST.CHAR + TXT.S
BEGIN
INDX = 0
WHILE INDX < 7 DO BEGIN
PRINT.4
INDX = INDX+1
END
WHILE INDX < 999 DO BEGIN
PRINT.4
INPUT "ENTER LAST NAME "; SRC.S
OBJ.S = SOUNDEX (SRC.S)
PRINT
PRINT "SOUNDEX CODE IS : "; OBJ.S
END
END

```

# Catalogue and Abstracts Volume 49

## The CP/M Users Group

NUMBER	SIZE	NAME	COMMENTS
		-CATALOG.049	CONTENTS OF CP/M VOL. 49
49.1	4K	4FDCBIOS.DOC	Files for running CP/M
49.2	2K	4FDCBIOS.HEX	2.0+ on CDOS systems
49.3	13K	4FDCBIOS.Z80	using this BIOS and BOOT.
49.4	1K	4FDCBOOT.HEX	
49.5	2K	4FDCBOOT.Z80	
49.6	11K	ABSTRACT.049	Abstracts of the files on this disk
49.7	1K	ASMDEF.RAT	Used by LCAT, part of RATFOR
49.8	6K	ASTRO.FOR	FORTRAN Astronomy
49.9	10K	CDOS.DOC	Comments on CDOS features.
49.10	2K	CDOSCOPY.COM	Full disk copy program
49.11	2K	CDOSCOPY.DOC	for CDOS systems,
49.12	4K	CDOSCOPY.HEX	single or double
49.13	26K	CDOSCOPY.PRN	density, and single
49.14	12K	CDOSCOPY.Z80	or double sided.
49.15	1K	CONSTS.RAT	Definitions of some frequently used constants (RATFOR)
49.16	1K	F77.DOC	FORTRAN 77 Documentation
49.17	1K	F77DEF.RAT	FORTRAN 77 Functions
49.18	2K	F77LIB.FOR	Definitions of some intrinsic FORTRAN 77 functions
49.19	3K	FILTER.Z80	Source code for emulating CDOS calls under standard CP/M.
49.20	1K	FILTER.HEX	
49.21	6K	FILTER2.Z80	Modified version of
49.22	1K	FILTER2.HEX	above.
49.23	2K	INDEX.Z80	RATFOR Subroutine
49.24	1K	INDEXT.FOR	Test for INDEX.Z80
49.25	1K	IOER.Z80	Used by LCAT (RATFOR)
49.26	1K	LCAT.DOC	Used to concatenate
49.27	3K	LCAT.RAT	Microsoft .REL files.
49.28	2K	MINV.DOC	Matrix Inversion
49.29	3K	MINV.FOR	Determinant Calculator,
49.30	3K	MINV.RAT	FORTRAN, RATFOR source code.
49.31	2K	RATCOMNS.RAT	Used by RATFOR
49.32	3K	RATFOR.DOC	Documentation of RATFOR
49.33	40K	RATFOR.FOR	FORTRAN source code for RATFOR
49.34	25K	RATFOR.RAT	RATFOR source code for RATFOR
49.35	17K	RATFOR.REL	Microsoft .REL file, no library routines
49.36	1K	READLB.Z80	Subroutine for SCOPY(RATFOR)
49.37	1K	RFQ.CMD	CDOS equivalent of a submit file. (^=\$)
49.38	1K	SCOPY.DOC	Direct Disk I/O (RATFOR)
49.39	2K	SCOPY.FOR	FORTRAN source code
49.40	1K	SORT1.DOC	Quicksort program (RATFOR)
49.41	2K	SORT1.FOR	FORTRAN Source code
49.42	3K	SORT1.RAT	RATFOR Source code
49.43	1K	TRNSLT.Z80	Used by RATFOR

This volume consists mostly of Rational FORTRAN (RATFOR) files. A letter and abstract by the author (Tim Prince) are presented here; the rest of the abstracts are by Chuck Weingart of the Chicago Area Computer Hobbyist Exchange. Please note that some of the files on this disk rely on Cromemco CDOS calls and are not compatible with "standard" CP/M systems. Almost ALL FORTRAN and RATFOR files require a Z-80 processor. See the abstracts which point these out.

Jim Mills

I have enclosed a disk which has RATFOR.RAT adapted for FORTRAN-80. This gives essentially identical results to your Vol. 24 RATFOR.COM but is somewhat faster and gives people the chance to play with it. The foreign .COM file is hard to work with. I had to write a special program to copy it correctly on my system. The disk also has some useful sample programs.

This disk contains mainly utility programs associated with the Ratfor preprocessor and helps to bring some of the FORTRAN 77 functions into FORTRAN-80. The RATFOR follows the Addison-Wesley code closely but uses BYTE variables as appropriate. Transliteration, which is done inefficiently in the A-W version, is omitted. The matrix inversion-determinant calculator MINV and the address quicksort SORT1 are given as examples of the use of RATFOR.

The INDEX subroutine is given both in FORTRAN (in F77LIB) and Z80 versions. As FORTRAN-80 does not permit the same usage of this function as in FORTRAN 77, the calling sequence is somewhat arbitrary.

LCAT, for concatenating .REL files, calls assembler library functions which may not be available on some systems. However, the .COM file may work anyway. SCOPY calls the READLB subroutine, which will have to be rewritten for systems which do not support the same extended CP/M calls.

Tim Prince

RATFOR is a preprocessor for FORTRAN source programs which facilitates control constructs and text insertions, described in "Software Tools" by Kernighan and Plauger (Addison-Wesley). RATFOR source code is free form, with statements separated by ; or [CR]. RATFOR statements include extensions to FORTRAN such as IF ELSE, WHILE, REPEAT UNTIL, FOR, BREAK, NEXT, and INCLUDE. An earlier version of RATFOR.COM is in CPMUG Volume 24. This new version runs only on a Z80, is faster, has documentation (RATFOR.DOC), source (RATFOR.RAT, RATFOR.FOR), and uses RATCOMNS.RAT and TRNSLT.Z80. The resulting FORTRAN program is

NOTE: Most of the FORTRAN files on this disk will run on Z-80 processors only, some will run only with Cromemco CDOS. Check the abstracts and .DOC files for more details.

compiled with Microsoft FORTRAN-80 or Cromemco FORTRAN, V3.21. It should run on any version of CP/M that supports the FORTRAN and has a Z80.

F77LIB.FOR is a file containing definitions of most of the new intrinsic functions in FORTRAN 77. These functions can be employed by searching the file F77LIB.REL which will be created by compiling F77LIB.FOR. There is also an INCLUDE file CONSTS.RAT which automatically defines some frequently used constants. INDEX.Z80 is a subroutine given both in FORTRAN (in F77LIB) and Z80 versions. They are written in Cromemco Z80 FORTRAN and assembler language, but should be acceptable to Microsoft versions. INDEX.FOR is a test for INDEX. There is some FORTRAN-77 documentation in F77.DOC and F77DEF.DOC.

LCAT.RAT is for concatenating Microsoft .REL files, described in LCAT.DOC. It calls CDOS Z80 assembler library functions which are not be available on CP/M systems. It is written for RATFOR, and uses ASMDEF.RAT and IOER.Z80, written in Cromemco Z80 assembler; Microsoft will also work. There are two possible problems with this routine, both related to the end-of-file. First, some versions of Microsoft FORTRAN did not put an "eof"(1A hex) at the end of the REL library. Second, this program stops if it hits 9E1A Hex, and there is a small chance of that occurring in the middle of a file; don't destroy your old library before checking the output of LCAT.

MINV.RAT and MINV.FOR contain a subroutine which inverts a matrix and calculates the determinant. It does not have any protection against a case where the equations are of wildly different orders of magnitude. There is a test case included in the file. These are included as tests for RATFOR, and the FORTRAN version will run on Microsoft or Cromemco FORTRAN, V3.21, any supported size and version of CP/M or CDOS, respectively. The DOC file includes some, but not all, info needed for use. Look at the test file included for more.

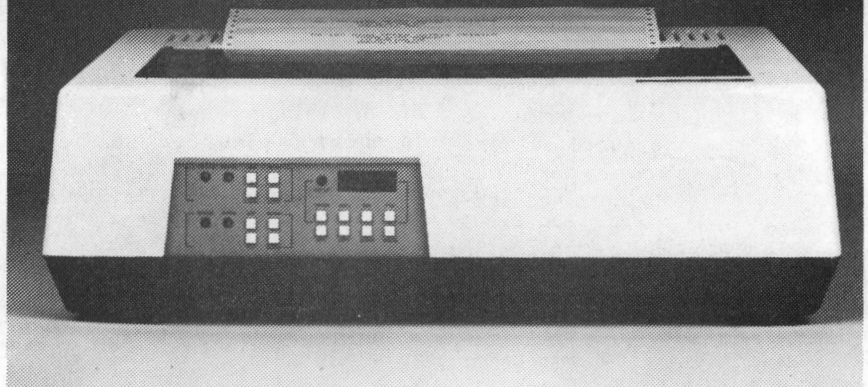
SORTI.RAT, SORTI.FOR are an address quicksort with a test driver program. Unfortunately, although the FORTRAN appears essentially identical to a successful program developed on a large system, the Microsoft version doesn't work yet. They are given as examples of the use of RATFOR. The SORTI.DOC file has some information describing the use of this routine. The FOR file will compile on Microsoft or Cromemco FORTRAN, V3.21, with any supported CP/M or CDOS, respectively.

SCOPY.FOR does direct disk I/O, either disk-to-disk or disk to

console, described scantily in SCOPY.DOC. The FOR file should be usable by any Microsoft or Cromemco FORTRAN, V3.21. It calls the READLB.Z80 subroutine which will have to be rewritten for systems which do not support the same CDOS calls. READLB is written in Z80 assembler; Microsoft acceptable.

FILTER.ASM is source taken from an article by J. Warner in the January, 1980 issue of Dr. Dobbs. It implements two essential CDOS calls, 80H and 86H, to allow some Cromemco software to run under CP/M. This is commented Zilog Z80 source, and can be as-

## datasouth announces... THE TOTAL PRINTER PACKAGE!



The DS180 matrix printer provides the total package of performance features and reliability required for applications such as CRT slave copy, remote terminal networks and small to mid-range systems. Not a "hobby-grade" printer, the DS180 is a real workhorse designed to handle your most demanding printer requirements. And pricing on the DS180 is hundreds of dollars below competitive units.

**High Speed Printing**—Bidirectional, logic-seeking printing at 180 cps offers throughput of over 200 lpm on average text. A 9-wire printhead life-tested at 650 million characters generates a 9x7 matrix with true lower case descenders and underlining.

**Non-volatile Format Retention**—a unique programming keypad featuring a non-volatile memory allows the user to configure the DS180 for virtually any application. Top of form, horizontal and vertical tabs, perforation skip-over, communications parameters and many other features may be programmed and stored from the keypad. When your system is powered down, the format is retained in memory. The DS180 even remembers the

line where you stopped printing. There is no need to reset the top of form, margins, baud rate, etc...it's all stored in the memory. If you need to reconfigure for another application, simply load a new format into the memory.

**Communications Versatility**—The DS180 offers three interfaces including RS232, current loop and 8-bit parallel. Baud rates from 110-9600 may be selected. A 1K buffer and X-on, X-off handshaking ensure optimum throughput.

**Forms Handling Flexibility**—Adjustable tractors accommodate forms from 3"-15". The adjustable head can print 6-part forms crisply and clearly making the DS180 ideal for printing multipart invoices and shipping documents. Forms can be fed from the front or the bottom. If you would like more information on how the DS180's low-cost total printer package can fill your application, give us a call at Datasouth. The DS180 is available for 30-day delivery from our sales/service distributors throughout the U.S.

**datasouth**  
computer corporation

4740 Dwight Evans Road • Charlotte, North Carolina 28210 • 704/523-8500

sembled by Microsoft or CDOS assemblers. The chicken-egg problem is avoided by supplying the FILTER.HEX file to bring up the CDOS assembler. As supplied, Cromemco FORTRAN IV and COBOL won't work with it; more functions must be added. This routine is installed by making it part of the memory image of a program: using DDT, load FILTER.HEX, then load the Cromemco program starting at 200H; then save the combined programs. When it gets control it boosts the function code filter into place at the top of the TPA, then moves the original program down to 100H and jumps to it. FILTER2.ASM is the source for a modified version of FILTER.ASM, primarily minor improvements and a bug fix. More comments are included from the original article in DDJ. The same installation procedures and comments as FILTER apply to FILTER2, including FILTER2.HEX. FILTER2 is from the reviewer, Chuck Weingart.

**Warning:** CP/M and Cromemco CDOS directories are partially incompatible. The first byte of each FCB must be made zero (1.4 mode). These routines do not allow most Cromemco software to run on non-Cromemco equipment. They are supplied here to let the Cromemco owner run his/her legitimately obtained programs under CP/M on Cromemco hardware. It does not appear that any software supplied with versions of CDOS after 2.17 are compatible with CP/M (2.2) with FILTER.

4FDCBIOS.DOC, 4FDCBIOS.Z80, 4FDCBIOS.HEX, 4FDCBOOT.Z80, 4FDCBOOT.HEX, 4FDCBIOS.Z80 and 4FDCBOOT.Z80 are the CBIOS and boot programs for running CP/M 2.x on a Cromemco 4FDC disk controller board, or a California Computer Systems 2422 disk controller, with a 3P + S or Cromemco TUART or similar console port at address 0. Assembly options include memory size, one to four drives, support for 5", 8" Shugart, and 8" Persci drives. Drivers for the Cromemco TUART for paper tape and Cromemco printer board are optionally included. The source is moderately commented, not structured, in Cromemco Z80 assembler; Microsoft can also be used. It is supplied for two 8" Shugart drives and 64K, and 4FDCBIOS.HEX and 4FDCBOOT.HEX are the assembled output files. Since the 4FDC is de-

signed for the Z80, conversion to 8080 would be difficult. 4FDCBIOS.DOC contains detailed documentation.

CDOS.DOC is comments by Trevor Marshall and this reviewer for users of Cromemco hardware and software. Nothing can be guaranteed, but everything is true as far as can be checked. (No attempt is made to help those who want to run the cheap Cromemco programs on CP/M without buying any Cromemco hardware). A must for CDOS users.

CDOSCOPY is the COPY program, CPMUG 1.3, converted to Z80 and extensively rewritten by Trevor Marshall to work on the Cromemco CDOS operating system. The program supports 8" disks, single or double sided, single or double density. It is possible to convert it for 5" disks; see the DOC file for details. As supplied, it would require a 48K system to copy double-density disks, but I haven't personally been able to verify operation that way. The program runs correctly, but unfortunately slower than the COPY program because of CDOS internal operations. A single-sided, single density disk copies in about 5 minutes. Supplied are the source file(.Z80), the output (.HEX), the print file(.PRN), and the program (.COM). This program runs only on a CDOS system, with all Cromemco hardware as required by that system, and assembles only with the Cromemco Z80 assembler.

ASTRO.FOR is a series of FORTRAN routines for astronomical calculations. (From an old SMITHSONIAN ASTROPHYSICAL OBSERVATORY computing note.) Of only specialized interest, for the DECML, IDECML, and KDECML. These are routines for free format reading from FORTRAN. They are used by reading the characters into a LOGICAL\*1 array of dimension of say 80, using a format of say 80A1. Successive calls to DECML as a function then give the real numbers on that line. The variable N is a pointer, the current column of the input buffer. Can be compiled by CDOS or Microsoft FORTRAN IV, though their usefulness can only be judged by an astronomer.

Use of the term Z80 in this application does not imply that this program was written by Zilog.

# ZOSO

Editor's note: The first part of what follows was withheld from Zoso's last column in order that Intertec might reply. The partial column which appeared in the March issue concluded with a fix for the QT clock board. Zoso continues...

To be honest, having finally done it, I think hardware fixes are boring, so I'll try to pick up the pace with a few words about Intertec Data Systems. I'll bet that many of you suspected that sooner or later I'd have something further to say about them.

I have only owned one product from the Augean stables of Intertec, that being the SuperDEC board (a high speed logic board for the Digital DECwriter II), which they no longer make.

For almost three years my DECwriter hummed along at 120 CPS. That is until a few months ago, when my SuperDEC died. I called Intertec to arrange for repairs. They will not fix it, they cannot or will not direct me to someone who might fix it and they offer no solace beyond that, except perhaps for awkward apologies in those charming rice bog drawls which have proven somewhat disarming to the unwise. I really have no time to troubleshoot things which become troublesome, but being desperate, I got out my SuperDEC schematics only to discover that they resembled a parts placement diagram on which disturbed youngsters had been allowed to dull their crayons.

I really shouldn't wallow in self pity about my demised SuperDEC. Far more tragic casualties are the half dozen or so poor souls I have met, all of whose [not inexpensive at \$1995 for the fully stripped 60 CPS model] SuperTerm printers expired in infancy as a result of grossly underdesigned circuitry implemented with 'collector's item' components. For two years or so the Superterm has neither been sold nor has factory



maintenance been readily available. One independent local repairman (Let's call him 'R') had five of those things cluttering his shop for well over a year. All had experienced failure of the power supply board with visible heat damage near the failed components. For at least eighteen months, 'R' was unable to obtain [possibly non-flammable] replacement parts from Intertec. He called them many times and was repeatedly told that the required parts were 'back-ordered'.

Even before these SuperTerms broke down, 'R' (who is also a dealer) was having another set of problems with Intertec. 'R' had prepaid them for ten SuperDEC boards. When NCC was last in New York, a couple of years back, he had been waiting roughly a year for six of these ten to arrive. (The other four did arrive, and of these mine was the one for which no warranty card was returned to the factory - Sorry, guys). Needless to say, 'R' was not happy about this, and he decided to visit Intertec's exhibit at NCC to seek some manner of redress. I tagged along to take notes. We headed straight for the SuperTerms and some prototype SuperBrains, which comprised just about all of Intertec's product line that year. I lurked inconspicuously, eavesdropping on his confrontation with the Intertec representatives (some of whom had rather imposing management titles on their lapel badges). I personally overheard Intertec's effusive apologies for the ostensibly 'misprocessed' orders uttered by Intertec's [highest level] directors. Anyway, 'R' was given a special hot-line number to call (after NCC), and he was promised that everything would be straightened out immediately. Did 'R' ever get satisfaction from Intertec? HA! HA! HA! (which means 'NO' for those of you who don't understand the occasional irony of a hearty chortle).

Oh yes, the five lifeless SuperTerms; I haven't asked lately what became of them, but I know that at least one is still sitting in a corner of 'R's shop under two years worth of dust. This has made me sit and wonder why I've read nothing about Intertec suing (nor

naming) the component supplier whose purported year plus delayed deliveries kept them (Intertec) out of the warranty business. You've got to figure that any reputable company would be mad as hell over a thing like that. That this has not become a front page scandal makes me wonder about bulging envelopes changing hands under the table, but since no such inducements to silence have fallen down my chimney, I might as well start the ball rolling.

I must assume that Intertec did in fact take their warranty obligations seriously. Let me quote from their full page, full color ad promoting the SuperTerm (which in this case can be found on page 31 of the April 1978 Interface Age).

"...SuperTerm's unique "ballistic" printhead design is warranted for an entire year. This means that during the warranty period, should you ever encounter defects in printhead workmanship, Intertec will replace or repair the defective component free!..."

Interesting wording... How does one establish defects in workmanship? It seems like it might be easier to cite defects in performance, if for example despite all the care lavished upon workmanship, a product simply ceased to work. In all fairness, I must mention that I have not personally observed nor even heard of printhead failure in the SuperTerm, which might be the case because all the horrid electricity which makes printheads move and eventually wear out gets cut off before having a chance to do any of the damage covered by the [quoted] warranty. Anyway, later in the same ad:

"...In addition to unparalleled price and performance, every SuperTerm is backed by Intertec's nationwide factory trained service network with over 160 service centers strategically located coast to coast..."

From this it must be safe to assume that some very professional service manuals existed back then. At the bottom of this ad are the addresses of the corporate headquarters in Charlotte, North Caro-

lina as well as two regional distributors located in Maryland and California.

"...Intertec Data Systems, the only company with international sales, service and the revolutionary SuperTerm..."

Revolutionary indeed! Disposable printers, what a novel concept!

On page 159 of the January, 1981 Byte is an ad for Intertec's CompuStar. This item is apparently a much more powerful item than the SuperTerm, again permit me to quote:

"...there's no fuss in getting a CompuStar up and running. Just plug in a Video Processing Unit and you're ready to go... with up to 254 more terminals in the network by simply connecting them together in a "daisy chain" fashion. CompuStar's special parallel interface allows for system cable lengths of up to one mile with data transfer rates of 1.6 million BPS..."

I think that the real fun would be watching 255 well heeled hobbyists stringing up (or perhaps burying) their one mile cables, but that's another story. A single corporate address, in Columbia South Carolina appears at the end of the ad. Aha! That must be it, They switched Carolinas and the movers lost all those service manuals for the 1978 product line. Well, that's probably not it either...

On page 53 of the February 1981 Byte is another full page, full color ad from Intertec, this one for the SuperBrain QD, which reads in part as follows:

"...Our best salesmen are our present users. Not only have SuperBrain users been impressed with the inherent reliability of the system, they tell us that no other microcomputer system available today offers such a unique modular design concept. Just about the only tool required to easily maintain the system is a common screwdriver. And Intertec's total commitment [sic] to product service and customer support, with service outlets in most major cities, insures your original investment

will be a valuable one for many years to come..."

Possibly the 'many years' are those of some other galaxy where inhabited planets orbit their 'suns' once every few Earth days. Otherwise, one would have to presume that the reliability and servicing pictures for current Intertec products has brightened considerably. Still, I have a funny feeling about the SuperBrain QD ad because no prices are mentioned. In my experience, certain companies with arguable track records have grudgingly learned not to announce prices for products that have yet to exist. I am also somewhat curious about just which major cities are blessed with the 'service outlets' to which they refer and whether or not there are still over 160 of them, but frankly I am a lot more curious about where I can get my hands on one of those miracle screwdrivers.

O.K. Let's get serious, we'll have a scavenger hunt... Red team, you've got to bring me the guidance system from a Soviet cruise missile; Greens, you must bring me the Purser's safe from the Titanic; and Blues, you have to find me SuperDEC and SuperTerm service manuals... On your marks, get set, GO!!!

Zoso continues below with current material - ED

New York City  
April 3, 1981

I'm rather sorry the public service segment of my last column was delayed. Since then, I've triple checked every little thing which I wrote, and it's a good thing I did. Remember my friend, 'R'? A recent visit to his shop has unearthed some new facts which oblige me to offer the following retractions:

#### RETRACTIONS

"...one of them (SuperTerms) is sitting in a corner of his shop under two years worth of dust..." This statement as it appeared is misleading, and I sincerely apologize. The offending copy should have read, "...four of them are

sitting in various corners of his shop under more than two years worth of dust...". By the way, I have taken photographs, just in case the proud parents of these moribund 'quadruplets' still have the slightest interest. Additionally, Intertec did finally refund 'R's money 'in full' at about the same time I prepared the original piece. 'R' estimates that at best this process took about three years worth of bank interest and a few hundred dollars worth of phone calls more than it should have. I agree!

At this point I have said more than I ever cared to about failed matrix printers and logic boards for the DECwriter II which can no longer swing the right visas for 'factory serviceland'. I'd like to change the tone by sincerely endorsing another company which offers essentially the same things (only much, much better) which are fully backed by cooperative and reliable factory service. In a word, Datasouth (Computer Corporation of Charlotte, North Carolina). Good Products! Good Company!!!!

In my last column that I broached the topic of problems I've had getting my trusty old IMSAI 8080 to coexist with certain of the newest boards which claim IEEE compatibility - whatever in blazes that might mean. Although it seems too good to be true, help is at hand. All you IMSAI owners should contact W/W Component Supply, Inc. of San Jose, California. They offer a line of totally redesigned IMSAI compatible retrofit components. I have seen a couple of these items and they are fabulous. To give you one example, a 22 slot assembled and tested mainframe, differing from the original IMSAI 8080 only in terms of substantial design improvements, can be had for \$600. This is about 2/3 of what the lesser product originally cost (back when the dollar was worth an honest quarter). In fact, I now own one of their CPA front panels, and my hardware conflicts appear to be over. The CPA costs \$225 (A&T), and was shipped promptly. In consideration of those who might want to add a front panel to non-IMSAI machines, all the goodies and trim (ex-

cepting screws and standoffs) are included. TEI (and perhaps other) mainframe owners, just drop in a CPA and enjoy the light show. I think we've got another winner here.

In the last column, I mentioned the possibility of putting RATFOR (tailored to SSS FORTRAN) into the public domain. Look for this to be released soon by the CPMUG. This will be a very nice version which improves upon the standard as both INCLUDE and REMOTE BLOCKS are implemented. I have persuaded the individual who upgraded this particular RATFOR to release it to CPMUG. As soon as [he] can find a little time to fix up the DOC file, it's yours, and I'll bet that the 'other' RATFOR being offered not so inexpensively, by you know who (in Champaign, IL), will pale by comparison. If this doesn't make my point crystal clear, let's try again... Anyone who pays for RATFOR must have suffered forcep damage of the skull at birth.

This brings to mind another subject. How many 8080/8085 only computers have you seen lately? I haven't seen too many. I think it's high time that application programmers banded together and released Z80 only products. If the vast majority of us own Z80 based computers (and I am convinced that this is the case), then we are frequently getting much less than optimally tuned software; CP/M itself comes immediately to mind. Continuing courtesy to the remaining (Intel) holdouts has become an unwarranted burden. Now don't expect the big software houses to accept this idea just yet, but I have one suggestion which might speed up the process. Everyone who has ever written a good assembly language program for the 8080 should take the time to make at least one super-duper improvement, and then rerelease that program only after adding a few totally meaningless chunks of Z80 code. It's ever so easy and lots of fun, and if you do what I suggest, you will be casting a vote which might actually count, unlike those silly referendums they stage each November. Here are a few of my favorites (expressed in Zilog's dreaded mnemonics):

1: PUSH AF  
LD A,1  
POP AF

2: PUSH BC  
POP 1Y

3: LD R,A

4: PUSH AF  
SRA A  
POP AF

5: EXX  
EXX

Just stick a couple of these guys in parts of your programs which get executed a lot (if applicable, disk I/O is especially recommended), and your improved programs will only be available to 'club members'. Once virtually everyone has joined the Z80 club, some of the major software houses may belatedly show us just how good the Z80 really is! (See important postscript).

Speaking of major software houses, I recently attended the funerals of several fellows who succumbed to asphyxia. Each had tried to hold his breath until Microsoft's BASIC compiler supported CHAIN with COMMON. Sure, the death certificates said something about no oxygen. The real truth is that these guys perished from ignorance. I can't believe it, but they really hadn't figured out that in the case of CHAIN with COMMON, Microsoft had set the stage for breath-holding to become an attention grabber suited only to the very slow of grasp or the blatantly suicidal. Nonetheless, some of the recently demised left gorgeous widows and/or substantial estates behind, and I think it's time to find out if feigned sympathy can improve my lot as much as I think it can.

What else can I say,  
Zoso

POSTSCRIPT: Here's what else I can say: 'Z80' is a trademark of the Zilog Corporation. The term 'Z80 club' is a satirical invention which does not exist to the best of my knowledge. In the unlikely event that my knowledge is incomplete in this regard, I should organization administered or primarily supported by Zilog, or more

accurately, the EXXON corporation, of which Zilog is a debatably profitable subsidiary. To me, any issues addressed by disclaimers and acknowledgements of this sort are nothing more than fuel for the ego engines of corporate attorneys. Nonetheless, I have always liked the Z80, so it has never bothered me before that the ever vigilant protectors of Zilog's several duly registered trademarks ultimately answer to the free world's number one petrovillain while somehow managing guiltless sleep.

Dear Zilog, Please keep your lawyers away from the 'formidable letter' generators. As I see it, the venerable American buffalo has been in the chipmaking business a lot longer than yourselves, and so far neither buffalos nor their representatives have ever imposed on my time or anyone else's with legal excess.

Bye, Bye (really)

Zoso

## REPLY

Thank you for the opportunity to respond to Zoso's statements regarding Intertec and our products.

We were gratified to hear that Zoso's SuperDEC board (an add-on device to enable the Digital DEC-writer to print at 120 characters per second) operated flawlessly for three years. We were not really surprised, however, because this level of performance and reliability has been engineered into every product we make. A true testimonial to our commitment to this industry.

Mr. Zoso is correct on our repair position on the SuperDEC. We no longer repair the device in-house, but it can easily be maintained by a competent technician at any one of the many local computer stores in the New York area who distribute our current range of products. The schematics Mr. Zoso has obtained should be adequate in enabling repair of the device since other customers have not experienced significant problems with the same documentation. We do agree that perhaps the schematics are not as professionally prepared as others; nonetheless [sic] they are adequate.

Like the SuperDEC. we also do not

maintain in-house SuperTerm repair service. But there is a reason. We arranged in 1978-79 for several independent dealers to take over worldwide service for this product before deactivating our in-house facility. To date, this arrangement has been satisfactory. In contradiction to Mr. Zoso's statements, the printer is maintainable and continues to be in widespread use today, particularly in many timesharing environments. And good repair documentation is available on this product. As a good faith gesture, I've enclosed a complimentary maintenance manual (with professionally drawn schematics!) which Zoso may make available to others.

Zoso's suspicion of our nationwide factory-trained service network is also unfounded. We currently maintain a third party on-site maintenance agreement with the Carterfone Corporation of Dallas, Texas under which we provide factory training to their technicians to maintain our products from over 50 service centers nationwide. Earlier, over 160 service centers (as referenced in Zoso's statement) were available to SuperTerm users for a monthly contract maintenance fee. However, since production of the device was ceased in late 1977, the requirement for service from this many locations diminished. Eventually, the service responsibility for the SuperTerm was turned over to other independent vendors.

I would like also to comment on Zoso's remarks about our new CompuStar system. Quite frankly, however, I am not sure what to say since I do not understand the intent or objective of his remarks. The CompuStar is our newest product and it has been well-received in the market thus far. In its first full month of production (March, 1981), it accounted for nearly 20% of our gross sales and is expected to account for as much as 50% of our sales during the next couple of years as we place more emphasis on multi-user microcomputer systems. Prices are mentioned in the CompuStar ad and we felt it would be obvious that additional information could be obtained by calling the manufacturer whose name, address and telephone number appears at the bottom of the ad.

Since we've had many calls, I assume most readers have also come to this conclusion.

Intertec has not grown from a one man organization in 1973 to a \$40 million a year organization with over 200 employees in 1981 by abandoning its customers and failing to provide product support. All of the text books we've read tell us that such tactics do not foster good customer relations nor do they promote additional sales. How, then, have we been able to prosper if we are indeed guilty of these sins? If we are guilty of anything, then it must simply be growing pains. The industry we serve has grown more than 300% for each of those years. With growth as phenomenal as this, one must seriously question those who find fault with our business practices. Nonetheless [sic], I suppose there have been occasions on which some customers may have become unhappy (or even irate) with some of our policies. Perhaps this is inevitable but we would like to think it is not. We are continually striving to provide an enhanced product offering in terms of value and reliability while at the same time strengthening our customer support programs. In fact, an enormous amount of the profits which we have generated in the last couple of years are now being plowed back into the company to bolster our customer service, product support and national service programs. The results of this will first be evidenced at the upcoming National Computer Conference in Chicago on May 4th. We invite Zoso (and others who are unhappy with us) to drop by and talk with us for a while. He (and all the others) will quickly notice that Intertec really is responsive. We do care... about our customers, our reputation and our company.

I've enclosed a full packet of current literature including information on our industry unique CARE™ Membership Service Program so that Zoso can better understand our philosophy of customer support. After reviewing the material, I'm sure you will agree that the customer always has been and still is the No.1 concern at Intertec.

Best Regards,  
 Ron L. Wells, Director of Marketing  
 INTERTEC DATA SYSTEMS

## Tips & Techniques

This tip is not an entry in our contest, but has been released by MicroAp, for use with Selector III-C2. FN.SIZE() makes it possible to determine accurately the size of a data file, whatever the density or CP/M version used. In addition, it corrects the record count when multiple cycle construction of a new INDEX file is under way. FN.SIZE is listed below and may be used in other CBASIC programs and as a replacement for CBASIC SIZE() function. It is for non-commercial use only.

```

33DEF FN.SIZE(DATA.FILES)
  STANDARD.SIZE%=SIZE("CRUN2.COM")
  IF STANDARD.SIZE%>5 AND STANDARD.SIZE%<19 THEN \
    FN.SIZE=SIZE)DATA.FILES):\
  RETURN
  FCB$=""
  FCB$=FCB$+FCB$
  FCBADR%=SADD(FCB$)+1
  POKE FCBADR%,0
  DSN$=UCASE$(DATA.FILES)
  IF MID$(DSN$,2,1)<>"." THEN 200.10
  POKE FCBADR%,((ASC(LEFT$(DSN$,1))-ASC("A")) AND OFH)+1
  DSN%=MID$(DSN$,3,20)
200.10 FOR I%=1 TO 8
  IF LEN(DSN%)<0 THEN 200.30
  IF LEFT$(DSN%,1)=". " THEN 200.20
  POKE FCBADR%+I%,ASC(LEFT$(DSN%,1))
  DSN%=MID$(DSN$,2,20)
  NEXT
200.20 IF LEFT$(DSN%,1)=". " THEN DSN%=MID$(DSN$,2,20)
  FOR I%=9 TO 11
  IF LEN(DSN%)<0 THEN 200.30
  POKE FCBADR%+I%,ASC(LEFT$(DSN%,1))
  DSN%=MID$(DSN$,2,20)
  NEXT
200.30 FOR I%=12 TO 35
  POKE FCBADR%+I%,0
  NEXT
  POKE FCBADR%+36,00EH
  POKE FCBADR%+37,023H
  POKE FCBADR%+38,011H
  POKE FCBADR%+39,FCBADR% AND OFFH
  POKE FCBADR%+40,FCBADR%/256
  IF FCBADR%$0 THEN POKE FCBADR%+40,(FCBADR% XOR OFFFFH)/256 XOR OFFH
  POKE FCBADR%+41,0CDH
  POKE FCBADR%+42,005H
  POKE FCBADR%+43,000H
  POKE FCBADR%+44,0C9H
  CALL FCBADR%+36
  IF PEEK(FCBADR%+35)<0 THEN FN.SIZE=8192 \
  ELSE FN.SIZE=(PEEK(FCBADR%+33)+7)/8+PEEK(FCBADR%+34)*32
  FCB$=""
  RETURN
FEND
  
```

Our winning tip this month comes from Al Rich, an Applied Logician with The Soft Warehouse in Honolulu, HI. Here is his contribution:

As anyone familiar with the operation of computers is well aware, undetected memory problems can ruin your whole day. Memory diagnostic programs abound, but unless you realize that there might be a problem, they won't do you any good. In addition I detest the thought of my computer goofing around doing nothing while waiting for me to type a character.

Both of these problems are resolved on the The Soft Warehouse's system by a continuously running background memory test. It keeps the computer busy when not doing anything else and lets me know of any memory problems developing. What's more, it is totally transparent to users of the system, unless of course there is bad memory!

The following listing is a simple addition to the console input (CONIN) polling routine in the CP/M BIOS. The pointer, TSTPT, is used to point to successive memory locations for checking. After the check is made the location is restored to its original value. If bad memory is found the address and bad bits are displayed on the console in Hex.

The test could be improved with the addition of a millisecond delay before reading back the tested byte. A word of caution: if your system uses interrupts these should be disabled during the test.

Also of interest is the PRTSTR utility routine used by the test routine. I have found PRTSTR to be a most convenient way to print character strings in assembly language programs, since strings do not have to be labelled and located elsewhere. PRTSTR does not destroy any registers. This last feature makes it great for debugging, since "I AM HERE" type messages can be put anywhere within an assembly program, making it easy to follow the flow of control.

```

CONINO:CALL TEST ;A<-- ICONIN (CONSOLE INPUT CHARACTER)
CONIN: IN CSTAT ;A<-- CONSOLE STATUS
      ANI IMASK
      JZ CONINO ;JUMP TO TEST MEMORY IF NO CHAR READY
      IN CDATA ;A<- CHAR
      ANI 7FH ;STRIP PARITY
      RET

TEST:  PUSH H
      LHLD TSTPT ;HL<-- TEST POINT ADDRESS
      INX H ;HL<-- HL + 1
      MOV A,H
      CPI $/100H
      JNZ TEST1 ;JUMP IF BELOW CURRENT LOCATION
      LXI H,0 ;RE-INITIALIZE TEST POINTER
TEST1: SHLD TSTPT ;INCREMENT TEST POINT POINTER
      MOV A,M ;TEST MEMORY @ TSTPT
      CMA
      MOV M,A
      CMP M
      JZ TEST2 ;JUMP IF MEMORY OK AT TSTPT
      XRA M ;A<-- BAD BITS
      PUSH PSW
      CALL PRTSTR
      DB CR,LF,'*** ',0
      POP PSW ;A<-- BAD BITS
      PUSH PSW
      CALL PRTHX ;DISPLAY BAD BITS
      CALL PRTSTR
      DB ' Bits BAD at: ',0
      MOV A,H
      CALL PRTHX ;DISPLAY BAD ADDRESS
      MOV A,L
      CALL PRTHX
      POP PSW
TEST2: CMA
      MOV M,A
      POP H
      RET

PRTHX: PUSH PSW ;PRTHX (PRINT A-REG IN HEX)
      RRC
      RRC
      RRC
      RRC
      CALL PRTHX1 ;PRINT MOST SIGNIFICANT NIBBLE
      POP PSW
PRTHX1:ANI OFH
      CPI 10
      JC PRTHX2
      ADI 7
PRTHX2:ADI '0' ;(ZERO)

OUTPUT: PUSH B
      MOV C,A
      CALL CONOUT ;OUTPUT CHAR TO CONSOLE
      POP B
      RET

PRTSTR: XTHL ;PRINT CHARS IN STRING FOLLOWING CALL
      PUSH PSW ;TO PRTSTR
      CALL PRT
      POP PSW
      XTHL
      RET

PRT: MOV A,M
      INX H
      ANA A
      RZ
      CALL OUTPUT ;OUTPUT CHAR
      JMP PRT

TSTPT: DW 0 ;CURRENT ADDRESS OF TEST POINT
    
```

**FREE!**  
 For the asking  
 our **BIG** new  
**CATALOGUE**  
 LIFEBOAT ASSOCIATES  
 1651 Third Ave. NYC,NY 10028  
 (212)860-0300 TELEX 640693



#### CP/M Version 2.x

PIP can send output to the .PRN device in CP/M 1.4, but apparently causes the system to lock up if tried in Version 2.x.

#### CP/M Version 2.23

When SYSGEN is used to generate a new CP/M disk, printer control characters 01, 02, 28, 29, 30, and 31 become inoperative. These are the expanded printer control characters and those which determine characters per inch.

#### CP/M for TRS-80 Model II

The COPY program (currently version 4.0) supplied with Lifeboat's CP/M does not identify disk errors when copying or verifying.

#### CP/M on Northstar Disk User's Notes

##### DD-Q Version

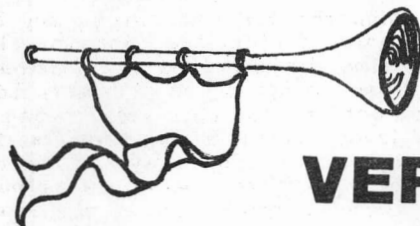
Mode byte does not work as described in the User's Notes Read After Write option on pages 11-12. It works as follows:

- Bit 1 (value 2) = 1 Enables feature on warm boots.
- Bit 0 (value 1) = 1 Enables feature on cold boots.

#### Pascal M

##### Version 3.2

This version will not accept numbers input on the console which would cause an integer overflow error. It backspaces in order to erase the invalid input, but all the numbers are not erased.



## NEW VERSIONS

#### BASIC Compiler

##### Version 5.24

These former problem areas have been fixed--

- 1-multiple floating division by constants is corrected;
- 2-Peek and Poke now allow only numeric constants;
- 3-/T in conversion of negative floating numbers to integers has been fixed;
- 4-double precision integer functions are handled properly;
- 5-the floating point to integer function now matches that of the BASIC Interpreter.

#### BDS C Compiler

##### Version 1.43

Listed below are bugs fixed with this new version.

- 1-Incorrect code is no longer generated when a sub-expression of a binary operation uses && or || operators.
- 2-An arithmetic binary operation where the left argument was a logical expression and the right a binary expression of greater precedence failed to evaluate correctly; now it does.
- 3-Missing commas are diagnosed before and after a format string specification, so they no longer cause wierd code to be generated.
- 4-An illegal constant in a place where a constant is required no longer results in the incorrect "unmatched left parenthesis" diagnostic and an impossibly large line number.

- 5-When the "-w" option is used with CLINK, a terminating control-Z is now put out to the SYM file when its length is a multiple of 128 bytes. As a result, "-y" can now be used to read the SYM file back in.
- 6-The "getc" library function now permits "fgetc" to read in lines from a text file not terminated by a control-Z.

- 7-Mismatched curly braces and mismatched square brackets now result in a more meaningful error message than "Unexpected EOF encountered".

- 8-"Main" is no longer treated as a keyword. "Main" can now be called recursively and its use is not restricted to the first line of a "main" function.

- 9-CLINK no longer crashes because of insufficient room in the reference table to hold all the symbols being read in from a SYM file.

- 10-"Printf", "sprintf", and "fprintf" library functions contained a bug which has now been fixed. Formerly, when a field width specification value was preceded by a '0', 0-fill instead of space-fill occurred.

- 11-CMAC.LIB properly handled such lines as  
exrel call, putchar

but now also handles such lines as  
exrel <lxih>, putchar

- 12-Quoted strings using an open-comment delimiter sequence ('/\*') no longer draw a "string too long" error.

These improvements have been made in the new version:

- 1-The body of a function definition now must be enclosed in curly braces.

- 2-A low-level character I/O function package, called CIO.C, has been added. This creates greater flexibility in console interaction, especially for game applications. The code generated using this facility is not portable to another system using a different hardware configuration than the object computer. Source code portability is maintained between systems equipped with C compilers.

- 3-String constants are now handled better by the code generator. Instead of placing the text where it is used and generating a jump around it, the compiler accumulates up to 50 text strings in a function and places them all at the end of that function.

#### FORTRAN-80

##### Version 3.42

These bugs have been repaired in the new version:

- 1-Multiplication of INTEGER\*4 variables where the right hand side(RHS) is zero no longer gives a non-zero result.
- 2-The I4ZERO constant now functions properly.
- 3-"+32768" and "+32769" INTEGER\*4 values now print out properly.
- 4-An integer assignment like "J=- (K)\*\*K" formerly generated a code equivalent to (-K)\*\*(-K), causing a zero result; this problem has been remedied.
- 5-Existing random access files are no longer deleted and recreated when opened. EOF characters are no longer written out to binary or random files.
- 6-Expressions are permitted (theoretically) over 2000 levels of parentheses, instead of the 14 which previously crashed the compiler.

Some other modifications have been made. The run time disk driver (DSKDRV) allows default file names of the FORTxx.DAT form to work with logical unit numbers up to 99. Lower case letters are permitted by the command line scanner of the compiler (COMSCN). COMSCN also, in the TEKDOS version, can pass linefeed characters on compiler output to the printer.

#### Nevada COBOL

##### Version 1.404

The following new features augment this program:

- 1-figurative constants HIGH-VALUES, LOW VALUES, and the ALL "literal" are implemented;
- 2-the RUN time package and the compiler both look more frequently for a ctrl. C from the console.

Pascal MT+  
Version 5.2

This new version consists of bug fixes, and includes no new features. Listed below are problems which have been adjusted.

- 1-Type checking on VAR parameters has been improved so that items of identical type but improper class are flagged when passed as parameters.
- 2-One-letter file names can now be converted for PRN files by the MTPUS.COM program.
- 3-The Z80 switch problems have been corrected.
- 4-READ and WRITE work on non-text files.
- 5-Logic of the WAIT procedure has been fixed, and the documentation has been altered to say the procedure does not generate inline code.
- 6-The standard FORWARD feature now works properly and is documented.
- 7-The documentation did not formerly specify this fact, but EXTERNAL procedures and functions may not have procedures and functions as parameters.
- 8-The linker now handles a module where the module name and an entry point were identical.
- 9-The READ integer and real routines now work.
- 10-The WRITE real routine now performs as the ISO standard specifies.
- 11-Because the disassembler has been fixed, source code now correctly comes before object code in proper synchronization. Certain statements caused the compiler to optimize on constants located on the line before an END statement (without a trailing semicolon) and resulted in an "error in comrel."
- 12-The debugger breakpoint logic did not work.
- 13-When displaying a pointer to a byte in the debugger, the debugger now properly displays two bytes.
- 14-REWRITE followed by RESET (without an intervening ASSIGN) now is handled correctly.
- 15-INTERRUPT procedure logic now saves/restores PSW and re-enable interrupts.
- 16-INP and OUT formerly did not use inline code with constants. Now an expression to be used must be in parentheses.
- 17-READHEX now accepts lower case hex digits.
- 18-Compiler error handling has been improved, especially in the area of variable/procedure functions.
- 19-The size limit of compilable procedures for the 8080/Z80 is 2K bytes; an error 253 will result if the limit is exceeded.
- 20-For certain users the linker did not work at all under CP/M 1.4. RANDOM file I/O does not operate under 1.4.
- 21-RMAC has problems assembling certain runtime routines with features only found in Microsoft MACRO-80. Write to MT Microsystems if you need to reassemble the run-time package.
- 22-5.2 and later versions will include the runtime source code only; it must be changed for use in a ROM-based environment. The full run-time source code is available for a fee.
- 23-Formatted output of REAL numbers is now handled correctly.
- 24-Hardware and software floating point users formerly had problems reading signed REAL numbers because of a bug in PFLT.
- 25-Floating point over- and underflow are now properly signalled.
- 26-READ and READLN of REALS now works when REAL is the last item on the line.
- 27-The constant folding bug in early releases has been repaired.
- 28-BLOCKREAD and BLOCKWRITE now accept a second parameter of any type without complaint from the compiler.
- 29-BLOCKREAD and BLOCKWRITE crossing extent boundaries now can work even if the record sizes employed cause a record to end on an extent boundary. The user is directed to the OPENX (open

extent) procedure.

30-An obscure problem with certain \$K toggles formerly caused parameter lists of some built-in functions and procedures to be wrongly constructed.

Selector III-C2

Version 3.24

This version includes FN.SIZE() in AUTOUPD.BAS, making it possible to determine accurately the size of a data file, whatever the density or CP/M version used. See page 18 for details.

Selector IV

Version 2.12

These bugs have been fixed in the new version:

- 1-REPIEX (LINE REPORT EXECUTION) no longer produces an SB error when relatively few fields are used to produce labels.
- 2-Formerly when a user converted Selector III-C2 or CBASIC files to Selector IV files (with 3T04) only shorter-than-normal CBASIC or Selector III-C2 source file names could be entered; now longer names are permitted. It is also possible now to EXIT directly to the operating system from 3T04.
- 3-Because of changes in CBASIC versions 2.07/37, REPORT, PAGE, SET and SELECT were creating SS errors. This problem has been solved.
- 4-The INCREMENT function in file conversion definition now displays properly.

Several improvements have been made. Trailing spaces are stripped and fields concatenated faster during label printing; these changes are designed to expedite the printing operation. The user may now specify that the labels be printed any number of columns from the left side of the form--allowing the use of continuous form envelopes.

Ranges, bounded by high and low key values, may be defined for alphanumeric fields. The range may be chosen at execution time and can order file conversion, batch update, line report or page report. Low null or high null directions can be given, to begin selection from the lowest possible key value or to continue to the highest possible key value; of course, designating both high and low values as null results in the key sequence processing of the whole file.



The software described here is available from authors, software distributors and computer stores.

Access/80

by Friends Software

This report writer is aimed at the small businessman. It does not limit record size, number of fields or file size; the number of variables a user may include in a LIST command is not restricted. ACCESS/80 formats tables, and is designed for statistical analysis, file auditing, and reporting. It tabulates fields in spread sheet or table form.

This package permits the creation of library files for storing variable definitions or other often-used data. Standard arithmetic, selective retrieval, and automated subtotalling on sort breaks are supported.

ACCESS/80 requires a minimum of 54K in memory, CP/M; it is for 8080 and Z80 processors.

### Address Management System

by Ivanco

This package is designed to work with small records. It can perform these functions with records: add, delete, index, display, print, compare, alter, and search. In addition, this program can make a unique similarity check; files can be sorted on up to three fields, and a reference or code field is permitted. It requires CP/M.

### Express RDBMS

by Phase One Systems

The multi-user Relational Data Base Management System is an option to the OASIS operating system. It includes a self-prompting screen creator and a program generator. It is designed for the manipulation, comparison and reporting of information accumulated from multiple files; RDBMS includes macro capability.

### PSORT

by Teratek

This is a machine language sort/merge utility designed to work at high speed on files with records of fixed length and aligned fields. It calculates record lengths automatically and allows up to ten keys (ascending or descending) to be specified for sorting. It is especially suited to large Postmaster mail lists; records the size of free memory may be sorted. Sorting attribute files are stored for future use. PSORT has a menu driven machine language program for producing sort parameter files.

### ZDM and ZDMZ

by RD Software

ZDM is a Z80 debugger and monitor intended to replace DDT for Z80 and 8080 debugging and development. The utility uses extended 8080 mnemonics similar to TDL (Xitan) which retain Intel mnemonics. ZDMZ supports all Z80 mnemonics. Both require a Z80 processor. Commands not found in DDT include: display and alternation of the alternate Z80 register and flag set, enabling or disabling interrupts when entering the target program, reading an input port and writing to an output port. There are seventeen commands and thirty functions.

ZDM overlays the CP/M Console Command and displays an error message when its own nucleus is about to be overlaid.

This month Lifelines is featuring new publications.

### The CP/M Primer

by Mitchell Waite and Stephen Murtha

This introduction to CP/M includes explanations of CP/M terminology, hardware concepts of I/O parts, list devices, etc. Simple exercises are presented to aid the novice. (96 pp., \$11.95)

### The Microcomputer Primer, Revised Edition

by Michael Pardee and Mitchell Waite

This book concentrates on the CPU, memories, I/O interfacing, programming, operating and numbering systems. It covers the first of the new Very Large Scale Integration (VLSI) chips. The new edition adds a number of microprocessors to its MPU Comparison Section, such as the 6809, 8089, 8086, Z80, etc. (384 pp., \$11.95)

### Osborne 4 & 8 bit Microprocessor Handbook

### Osborne 16-bit Microprocessor Handbook

by Adam Osborne

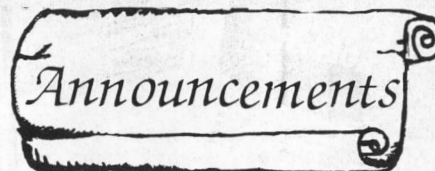
These books are intended to comprise an encyclopedic reference library for designers requiring access to microprocessor technology.

### The Pascal Primer

by Mitchell Waite and David Fox

This is an introductory text intended for beginners and hobbyists. The material is delivered in lesson form and interspersed with quizzes to test the reader's comprehension. The USCD version of Pascal was employed in writing the book.

Chapters included cover decision-making statements, numeric functions, string functions, arrays and sets. Appendices explain the pros and cons of Pascal, and present interfacing assembly language routines. (208 pp., \$16.95)



Lifeboat Associates has announced that it is no longer supporting its former Q3 format. Blackhawk Single Density and Nylac Single Density disks will no longer be produced by Lifeboat; nor will disks be configured for the Rex computer.

### Advertisement

Lifeboat Associates seeks to hire individuals experienced in software for mini- and microcomputers. Knowledge of RSTS/E, UNIX or CP/M required. Openings in various functions associated with marketing, manufacturing, and customer service. Interested persons are asked to send resumes to: Personnel, Lifeboat Associates, 1651 Third Avenue, New York, N.Y. 10028.



"HELLO! Z-80 4MHZ OPERATION WITH NO WAIT STATES."





# Microsystems — the CP/M\* and S-100 User's Journal

**CP/M is the software bus!\***  
**S-100 is the hardware bus**  
**for sophisticated microcomputer users!**

If you are a CP/M user, on any system—S-100, Apple, TRS-80, Heath, Ohio Scientific, Onyx, Durango, Intel MDS, Mostek MDX, etc.—after all CP/M is the Disk Operating System that has been implemented on more computer systems than any other DOS—then *Microsystems* magazine is the "only" magazine published specifically for you!

Or, if you use an S-100/IEEE-696 based computer—and the most sophisticated microcomputer systems available use the S-100/IEEE-696 hardware bus—then *Microsystems* magazine is the "only" magazine published specifically for you!

We started publishing *Microsystems* almost two years ago to fill the void in the microcomputer field. There were magazines catering exclusively to the TRS-80, Apple, Pet, Heath, etc. system users. There were also broad based publications that cover the entire field but no one system in depth. But no magazine existed for CP/M users—nor did one exist for S-100 users.

## The why and what of a software bus

First of all what is a "bus?" And why do we call CP/M "the software bus?\*"

A "bus" is a technique used to interface many different modules. Examples are the "S-100/IEEE-696 Bus" and the "IEEE-488 Bus." These are hardware buses that permit a user to plug a bus-compatible device into the bus without having to make any other hardware modifications and expect the device to operate with little or no modification.

CP/M is a Disk Operating System (DOS). It was first introduced in 1974 and is now the oldest and most mature DOS for microcomputer systems. CP/M has now been implemented on over 250 different computer systems. It has been implemented on hard disk systems as well as floppy disk systems. It is supported by two user groups (CP/M-UG and SIG/M-UG) that have released over sixty volumes containing over 1,600 public domain programs that can be loaded and run on systems using the CP/M DOS. Add to this another 1,500 commercially available

CP/M software packages and you have the largest applications software base in existence.

CP/M is the only DOS for micros that has stood the test of time (seven years) with the highest level of compatibility from version to version. And over the years this compatibility has been maintained as new features have been added.

This is why we say "CP/M is the software bus" and why *Microsystems* magazine is vital to providing CP/M users with technical information on using CP/M, interfacing to CP/M, new CP/M compatible products and for CP/M users to exchange ideas.

## Why support the S-100 bus?

S-100 is currently the most widely used microcomputer hardware bus. It offers advantages not available with any other microcomputer system. Here are a few of the advantages:

*S-100 is processor independent.* There are already thirty different S-100 CPU cards that can be plugged into an S-100 bus computer. Nine 8-bit microprocessors are available: 6502, 6800, 6802, 6809, 2650, F8, 8080, 8085 and Z80. Eight 16-bit microprocessors are available: 8086, 8088, 9900, Z8000, 68000, Pascal Microengine, Alpha Micro (similar to LSI-11) and even the AMD2901 bit slice processor. Take your pick from the incredible offerings.

*S-100 has the greatest microcomputer power.* What other microcomputer system has direct addressing of up to 16 megabytes of memory, up to 65,536 I/O ports, up to 10 vectored interrupts, up to 16 masters on the bus (with priority) and up to 10 Mhz data transfer rate? You will have to go a long way to use up that computing power.

*S-100 is standardized.* The S-100 bus has been standardized by the IEEE (Institute of Electrical and Electronic Engineers) assuring the highest degree of compatibility among plug-in boards from different manufacturers. And, *Microsystems* has published the complete IEEE S-100/696 standard (all 26 pages).

*S-100 has the greatest hardware support.* There are now over sixty different manufacturers of about 400 different plug-in S-100 boards. Far greater than any other microcomputer system.

With all these advantages is it any wonder that S-100 systems are so popular with microcomputer users who want to do more than just play games?

## For the serious computer user.

Each issue of *Microsystems* brings you the latest in the CP/M and S-100 world. Articles on applications, tutorials, software development, product reviews, and lots more, to keep you on top of the ever changing microcomputer scene.

And if you are an S-100 system user using other operating systems (e.g. North Star) *Microsystems* also supports you.

## Get your copy today

Order your subscription to *Microsystems*. Send \$10 for one year (6 issues), \$18 for two years (12 issues) or \$24 for three years (18 issues). If you prefer, call our toll-free number, **800-631-8112** (in NJ 201-540-0445) to put your subscription on your MasterCard, Visa or American Express card. Canadian/Mexican and other foreign surface subscriptions are \$15 and \$25, respectively, per year and must be prepaid. We guarantee that you will be completely satisfied or we will refund your subscription.

Join thousands of *Microsystems* subscribers like Jim Johnstone of Los Altos CA, who said "Microsystems has lived up to your promises and my expectations. Congratulations."

the CP/M\* and S-100 user's Journal  
**MICROSYSTEMS**

a Creative Computing publication

39 East Hanover Avenue  
Morris Plains, NJ 07950, USA  
Toll-free 800-631-8112  
(In NJ, 201-540-0445)

## VERSION LIST

The below software is available  
from the authors, computer stores,  
and distributors.

PRODUCT NAME	S	M	OS	P	MR	\$	
A-NATURAL Assembler Package	1.1		CP/M	8080		330/15	
A3+ Development Package			CP/M	Z80		409/40	
A4 Development Package			CP/M	Z80		299/40	
Accounts Payable/Cybernetics	3.1		CP/M	Z80	64K	500	Needs RM/COBOL
Accounts Payable/Graham Dorian	1.11	1.11	CP/M	8080	48K	805/40	Needs CBASIC2
Accounts Payable/Structured Sys	1.38		CP/M	8080	52K	840/40	W/IT WORKS run time pkg.
Accounts Payable/Peachtree	10-10-80		CP/M		48K	530/60	Needs BASIC-80 4.51
Accounts Receivable/Cybernetics	3.1		CP/M	Z80	64K	500	Needs RM/COBOL
Accounts Receivable/Graham Dorian	1.08	1.08	CP/M	8080	48K	805/40	Needs CBASIC2
Accounts Receivable/Peachtree	10-10-80		CP/M	8080	48K	530/60	Needs BASIC-80 4.51
Accounts Receivable/Structured Sys	1.4C		CP/M	8080	56K	840/40	W/IT WORKS run time pkg.
ALDS TRSDOS	3.38		TRSDOS	8080	32K	80/25	TRSDOS Macro-80
ALGOL 60	4.8C		CP/M	8080	24K	250/20	
ANALYST	1.0		CP/M	8080	52K	250/20	Needs CBASIC2, QSORT/VSORT
APL/V80 Compiler	3.2		CP/M	Z80	48K	500	Needs APL terminal
Apartment Management	1.03	1.03	CP/M	8080		805/40	Needs CBASIC2
ASM/XITAN	3.11		CP/M	Z80		69/20	
Automated Patient History	1.2		CP/M	8080	48K	175	
*BASIC-80 Compiler	5.24	5.24	CP/M	8080	48K	360/35	
BASIC-80 Compiler		5.1	TRSDOS		64K	400/25	TRS-80 Model II only
BASIC-80 Interpreter	5.2	5.2	CP/M	8080	40K	335/35	W/Vers. 4.51, 5.2
BASIC Utility Disk	2.0	2.0	CP/M	8080	48K	75	
*BSTAM Communication System	4.5	4.5	CP/M	8080	32K	200	
*BDS C Compiler	1.43A	1.43AT	CP/M	8080	32K	150/30	W/"C" book
Whitesmiths' C Compiler	2.0		CP/M	8080	60K	630/30	
BSTMS	1.2	1.2	CP/M	8080	24K	200	
BUG / uBUG Debuggers	2.03		CP/M	Z80		129/25	
Cash Register	2.0	2.0	CP/M	8080		805/40	
CBASIC Compiler	2.07P	2.17P	CP/M	8080	32K	125/20	2.17* modified 2.07
CBS Applications Builder	1.2		CP/M	8080	48K	395/40	Needs no support language
CIS COBOL Compiler	4.3.1		CP/M	8080	48K	850/50	
*CIS COBOL Compact	3.46	3.46	CP/M	8080	32K	650/50	
FORMS 1 CIS COBOL Form Generator	1.06	1.06	CP/M	8080		150/20	
FORMS 2 CIS COBOL Form Generator	1.1.6	1.16	CP/M	8080		200/20	
COBOL-80 Compiler	4.01	4.01	CP/M	8080	48K	710/35	
*COBOL-80 PLUS M/SORT	4.01		CP/M	8080	48K	845/55	
*CONDOR	1.10		CP/M	8080	48K	695/35	
CREAM (Real Estate Acct'ng)	2.3		CP/M	8080	64K	250	CBASIC needed
DATASTAR Information Manager	1.1		CP/M	8080	48K	350/60	
Datebook	1.05		CP/M	8080	48K	295/30	Needs 80x24 terminal
DESPOOL Print Spooler	1.1A		CP/M	8080		80	
DISILOG Z80 Disassembler	4.0	4.0	CP/M	Z80		110	Zilog mnemonics
DISTEL Z80/8080 Disassembler	4.0		CP/M	8080/Z80		110	Intel mnemonics, TDL extensions
EDIT Text Editor	2.06		CP/M	Z80		129/25	
EDIT-80 Text Editor	2.0		CP/M	8080		99/25	
ESQ-1	2.1		CP/M	8080		1495/50	Needs CBASIC2
ESQ-1 DEMO	2.1		CP/M	8080		75	
FABS	2.4A		CP/M	8080	32K	195/25	
FILETRAN	1.20		CP/M		32K	99	1-way TRS-80 Mod I, TRSDOS to Mod I CP/M
FILETRAN		1.4	TRSDOS		32K	149	2-way TRS-80 Mod I, TRSDOS & Std. CP/M
FILETRAN	1.4		CP/M		32K	119/20	2-way TRS-80 Mod I, TRSDOS & Mod I CP/M
FILETRAN	1.5		CP/M		32K	99	1-way TRS-80 Mod II, TRSDOS to Mod II CP/M
Financial Modeling System	2.0		CP/M		48K	300	
FORTTRAN-80 Compiler	3.42	3.42	CP/M	8080	36K	435/35	
FORTTRAN Package	3.38		TRSDOS			80/25	
*FPL	1.52	1.52	CP/M	8080	56K	695/30	
General Ledger/Cybernetics	1.3C		CP/M	Z80	48K	500	Needs RM/COBOL
General Ledger/Graham Dorian	1.09	1.09	CP/M	8080	48K	805/40	Needs CBASIC2
General Ledger/Peachtree	10-10-80		CP/M	8080	48K	530/60	Needs BASIC-80 4.51
General Ledger/Structured Sys	1.4C		CP/M	8080	52K	840/40	W/It Works Package
General Ledger II/CPaid	1.1		CP/M	8080	48K	450/30	Needs BASIC-80 4.51
GLECTOR Accounting System	2.0		CP/M	8080	56K	350/25	Use w/CBASIC2, Selector III
HDBS	1.05		CP/M	+	52K	300/35	
HDBS	1.05D		TRSDOS		56K	300/35	Runs only w/Radio Shack BASIC
HDBS.SRS	1.05		TRSDOS		56K	150/5	Runs only w/Radio Shack BASIC
HDBS	1.05D		APPLE		56K	300/35	
IBM/CPM	1.1		CP/M	8080		175	
Inventory/Graham Dorian	1.0	1.0	CP/M	8080		555/40	Needs CBASIC2
Inventory/Peachtree	10-10-80		CP/M	8080	48K	530/60	Needs BASIC-80 4.51
Inventory/Structured Sys	1.0C		CP/M	8080	52K	840/40	W/It Works Package
Job Costing	2.02	2.02	CP/M	8080	48K	805/40	Needs CBASIC2
*LETTERIGHT Text Editor	1.1B		CP/M	8080	52K	200/25	
LEVEL 3 BASIC / G2			TRSDOS			45	Cassettes
LINKER			CP/M	Z80		69	
MAC	2.0		CP/M	8080	20K	120/25	
MACRO-80 Macro Assembler Package	3.40	3.40	CP/M	8080/Z80		159/25	
Magic Wand	1.1		CP/M	8080	32K	395/40	
MAGSAM III	4.1		CP/M	8080	32K	145/25	
MAGSAM IV	1.0		CP/M	8080	32K	295/25	
MAILING ADDRESS Mail List System	12-2-80		CP/M	8080	48K	530/60	
*Mail-Merge	2.26		CP/M	8080		150/25	
Master Tax	1.0-80		CP/M	8080	48K	995/30	Has 1980 tax forms
MDBS	1.05		CP/M	+	48K	900/35	
MDBS-DRS	1.02		CP/M	+	52K	300	
MDBS-QRS	1.0		CP/M	+	52K	300	
MDBS-RTL	1.0		CP/M	+	52K	300	
MDBS-PKG			CP/M	+	52K	1500/60	W/all above MDBS products
MDBS	1.05D		Apple		56K	900/35	
Microspell	4.1		CP/M	8080	48K	249	Needs 150K

S Standard Version  
M Modified Version  
OS Operating System  
P Processor  
MR Memory Required  
\$ Price  
\* Indicates a new version or new product.  
+ These products are available in Z80 or 8080, in the following host languages: BASCOM, COBOL-80, FORTRAN-80, PASCAL/M, PASCAL/Z, CIS-COBOL, CBASIC, PL/I-80, BASIC-80 4.51, and BASIC-80 5.xx.

## VERSION LIST

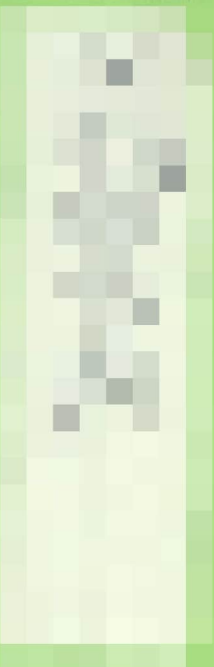
PRODUCT NAME	S	M	OS	P	MR	\$	
Mini-Warehouse Mngmt. Sys.	5.5		CP/M	8080	48K	650	Needs CBASIC
MP/M Operating System	1.1		MP/M	8080	32K	300/50	
MSORT	4.01		CP/M	8080	48K	160/20	
Mu LISP-80 Compiler	2.03		CP/M	8080		210/25	
Mu SIMP / Mu MATH Package	2.03		CP/M	8080		260/30	muMATH 80
NAD Mail List System	3.00		CP/M	8080	48K	115/25	
Nevada COBOL	1.404	1.404	CP/M	8080	32K	149/25	
Order Entry w/Inventory/Cybernetics			CP/M	Z80		500	Needs RM/COBOL
*PAS-3 Medical	1.71		CP/M	8080	56K	995/25	Needs 132-col. printer
*PAS-3 Dental	1.61		CP/M	8080	56K	995/25	Needs 132-col. printer
PASM Assembler	1.02		CP/M	Z80		129/25	
Pascal/M	3.2		CP/M	8080	56K	175/25	
PASCAL/MT Compiler	3.2		CP/M	8080	32K	250/30	
*PASCAL/MT+	5.2		CP/M	8080	52K	500/30	Also has 32K version
*PASCAL/Z Compiler	3.3		CP/M	8080	56K	395/25	
Payroll/Cybernetics, Inc.			CP/M	Z80		500	Needs RM/COBOL
Payroll/Peachtree	11-7-80		CP/M	8080	48K	530/60	Needs BASIC-80 4.51
Payroll/Structured Sys	1.00		CP/M	8080	60K	840/40	No longer needs CBASIC
PEARL SD	3.01		CP/M	8080	56K	900	W/CBASIC2,Ultrasort II
PL/I-80	1.3		CP/M	8080	48K	500	
PLINK Linking Loader	3.25		CP/M	Z80		129/25	
POSTMASTER Mail List System	3.3	3.3	CP/M	8080	48K	150/20	
Property Manager	10-10-80		CP/M	8080		925/60	Needs BASIC-80 4.51
Property Mngmt. Sys.	1.0		CP/M	8080	48K	650	Needs CBASIC
QSORT Sort Program	1.5		CP/M	8080	48K	100	
Real Estate Acquisition Programs	2.1		CP/M	8080	56K	500	Needs CBASIC
Residential Prop. Mngmt. Sys.	1.0		CP/M	Z80	48K	650	
RM/COBOL Compiler	1.3C		CP/M	8080	48K	750	w/Cybernetics CP/M 2
RAID	4.7.3	4.7.3	CP/M	8080	28K	250/25	
RECLAIM Disk Verification Program	2.1		CP/M	8080		80	
SBASIC	5.3h		CP/M	8080		295/35	
SELECTOR-III-C2 Data Manager	3.24	3.24	CP/M	8080	48K	295/25	
*SELECTOR-IV	2.12		CP/M	8080	52K	550/35	Needs CBASIC
SID Symbolic Debugger	1.4		CP/M	8080		120/25	N/A-Superbr'n
SMAL/80 Programming System	3.0		CP/M	8080		75/25	For CP/M I.x
Standard Tax	1.0		CP/M	8080	48K	495/30	Needs BASIC-80 4.51
STATPAK	1.2	1.2	CP/M	8080		495/30	Needs BASIC-80 4.2 or above
STRING BIT FORTRAN Routines	1.02	1.02	CP/M	8080		75/25	
STRING/80 bit FORTRAN Routines	1.22		CP/M	8080		95/25	
STRING/80 bit Source	1.22		CP/M	8080		295	
SUPER SORT I Sort Package	1.5		CP/M	8080		225/40	Max. record=4096 bytes
T/MAKER Data Calculator	1.4		CP/M	8080	48K	275/25	Needs CBASIC2
TEX Text Formatter	1.1		CP/M	8080	36K	105/15	
TEXTWRITER-III Text Formatter	3.6	3.6	CP/M	8080	32K	125/20	
TINY C Compiler	800102C		CP/M	8080		105/50	
ULTRASORT II	3.1	3.1	CP/M	8080	48K	195/25	
Unlock	1.3		CP/M	8080		95	Use w/ BASIC-80
VisiCalc	1.37		Apple	8080	32K	150	
VSORT Sort Program	1.8	1.8	CP/M	8080	48K	175/20	
WHATSIT? Data Manager	2.04		CP/M	8080	44K	175/25	
*WORDINDEX			CP/M	8080	48K	195	Needs Wordstar
WORDMASTER Text Editor	1.07A		CP/M	8080	40K	145/40	
WORDSTAR Word Processor	2.26	2.20A	CP/M	8080	48K	445/60	
*MAIL MERGE Printer Overlay	2.26		CP/M	8080	48K	150/25	
*WORDSTAR Customization Notes	2.2X		CP/M			95	
XASM-18 Cross Assembler	1.30		CP/M	8080		200/25	
XASM-48 Cross Assembler	1.30		CP/M	8080		200/25	
XASM-65 Cross Assembler	1.95		CP/M	8080		200/25	
XASM-68 Cross Assembler	1.96		CP/M	8080		200/25	
*XMACRO-86 Cross Assembler	3.40		CP/M	8080		275/25	
XYBASIC Interpreter Extended	2.11		CP/M	8080		450/25	
XYBASIC Interpreter Extended CP/M	2.11		CP/M	8080		550/25	
XYBASIC Interpreter Extended COMP	2.0		CP/M	8080		450/25	
XYBASIC Interpreter Extended ROM	2.1		CP/M	8080		450/25	
XYBASIC Interpreter Integer	1.7		CP/M	8080		350/25	
XYBASIC Interpreter Integer COMP	2.0		CP/M	8080		350/25	
XYBASIC Interpreter Integer ROM	1.7		CP/M	8080		350/25	
Z80 Development Package	3.3		CP/M	Z80		130	N/A-Magnolia, Superbr'n, mod. CP/M
ZDT Z80 Debugger	1.41		CP/M	Z80		50	N/A-Superbr'n, mod. CP/M
ZSID Z80 Debugger	1.4		CP/M	Z80		130	N/A-Superbr'n, mod. CP/M

Hard Disk Modules	
Description	Version
Corvus Module	1.6
APPLE-Corvus Module	1.1
KONAN Phoenix Drive	1.7
*Micropolis Microdisk	1.9
Pertec	1.6

Description	Version		
Operating Systems		Intel MDS Single Density	2.2
		Intel MDS 800/230 Double Density	2.2
		North Star SD IMSAI SIO Console	1.41
		North Star SD MITS SIO Console	1.41
		MITS Altair 3202 Disk	1.41
		North Star DD	1.45
		Micropolis Mod I - All Consoles	1.411
		* North Star DD/QD	2.22
		Micropolis Mod II - All Consoles	1.42
		Processor Technology Helios II	1.41
		* Micropolis Mod I	2.20A
		by Lifeboat/TRS-80 5 1/4" (Mod I)	1.41
		* Micropolis Mod II	2.20A
		by Lifeboat/TRS-80 Mod II	2.24A
		by Cybernetics/TRS-80 Mod II	2.25
		OASIS for Altos, Bell Ctrls., Billings, CA	
		Computer Systems, Compucorp, Cromemco,	
		Delta, Digital Microsys., Dynabyte, God-	
		bout, GRI, Index Comp. Sys., IBC, In-	
		tertechnique, Kontron, Media Sys. Corp.,	
		Micromation, Doubler, Morrow Thinker Toys,	
		NNC Elect., Onyx, Quay, S.D. Sys., Teletex,	
		TRS-80 Mod. II, Vector Gr., Vorimex, Zilog	
		* Ohio Scientific C3	2.23
		Sol North Star SD	1.41
CP/M for:			
Apple II w/Microsoft BASIC	2.0		
Cromemco System 3 8"	1.4		
Cromemco System 3 8"	2.2		
* Durango F-85	2.23		
Heath H8 w/H17 Disk	1.43		
Heath/Zenith H89	1.43		
by Magnolia-Heath/Zenith H89	2.2		
ICOM 3812	1.42		
ICOM 3712 w/Altair Console	1.42		
ICOM 3712 w/IMSAI Console	1.41		
ICOM Microfloppy (# 2411)	1.41		
ICOM 4511/Pertec D3000 Hard Disk	2.22		
Intel MDS Single Density	1.4		

**LIFELINES**

1651 Third Avenue / New York, N.Y. 10028



*Handwritten mark, possibly the number '4'.*



FIRST CLASS MAIL