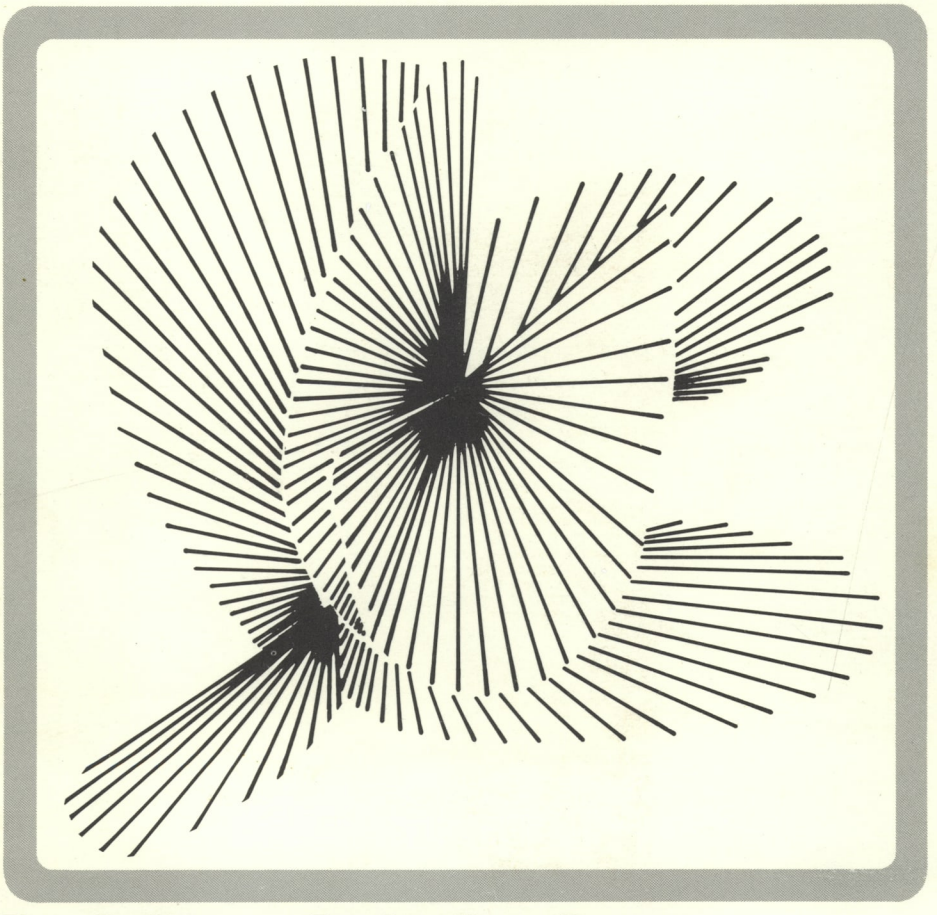
MARCH 1981

LIFELINES®

VOLUME ONE No. 10



The Software Evaluation Group: The Configurable Business System

The Osborne Packages: General Ledger

BASIC Comparisons: SBASIC

The CP/M Users Group Volume 48, Catalogue And Abstracts

ZOSO

LIFELINES®

Volume I No. 10 March

Editor-in-Chief: Harris Landgarten
Managing Editor: Jane Mellin
Contributing Editors: Ward Christensen
Zoso

CONTENTS PAGE

Opinion	Editorial Comments	2
	Zoso	4
	A Reader's Comment	6
Features	The Osborne Packages: General Ledger by Martin McNiff	7
	The Software Evaluation Group: Configurable Business System by Ed Paulette	9
	BASIC Comparisons: SBASIC Version 5.3, Part 1 by Bill Burton	13
The CP/M Users Group	The CP/M Users Group News	2
	CPMUG 86 The CP/M Users Group Volume 48 Catalogue and Abstracts	15
Product Status	Bugs	17
	Bug Fixes	17
	New Products	17
	New Versions	18
	Version List	20
Miscellaneous	OOPS	12
	Tip Contest	6

'SBASIC is a trademark of Topaz Programming

'CP/M is a trademark of Digital Research, Inc. The CP/M Users Group is not affiliated with Digital Research, Inc.

Lifelines, Volume I, Number Ten. Published monthly. Single copy price:$2.50 domestically, including U.S., Canada, Mexico. Single copy price for all other countries:$3.60. Subscription (12 issues) price:$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. $, 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., N.Y., 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 pending at N.Y., N.Y.

Editorial Comments

In the early days of microcomputing, the only available high level software development tools were BASIC interpreters. Since everyone needed BASIC to run application programs, the interpreter's authors were in a position to profit from the distribution of programs written in their language. This situation, however, was not quite so equitable for application developers. They were forced to provide source code to end users, thereby exposing themselves to plagiarism as well as theft. So most professionals concentrated on developing system software or tools and the few applications that were distributed were either very expensive or poorly done.

After the introduction of CBASIC by Compiler Systems, the programming inventives began to shift. CBASIC is a pseudo-compiler; it produces an intermediate (INT) file which requires a runtime package (CRUN) in order to execute. Thus programmers could distribute INT files without fear of plagiarism while the end user was forced to purchase CBASIC for the use of the runtime package. An explosion in CBASIC applications resulted which continues today despite the subsequent release of other more powerful compilers. Why haven't software authors embraced Microsoft's BASIC-80 compiler or Digital Research's PL/I-80 ? Both are powerful code generators which allow the implementation of faster, more complex programs. The answer lies in economics.

True compilers turn source code into directly executable binary COM files. While the distribution of COM files presents no problems for the programmer, it leaves the compiler writer out in the cold because the end user has no need for his product. In an attempt to protect their interests, Microsoft and DI have licensed the code generated by their respective compilers. This means that you have a silent partner in any software distribution venture that includes BASCOM or PL/I-80 generated code. Rather than pay high royalty payments, authors have stuck with CBASIC. The big losers have been software consumers.

A solution to this unfortunate circumstance may be found by examining the appeal of the pseudo compiler. In essence, by purchasing CBASIC, an end user is paying a one time library license fee directly to Compiler Systems. The program author avoids the bookeeping involved in royalty payments and the end user is spared from paying for the same library routines over and over again. The ultimate solution for compiler writers may be the distribution of a pseudo-product which I will name 'LINKLIB'; a program which marries a special version of the linking loader to the appropriate subroutine library. The software user would purchase LINKLIB directly from the compiler developer so that he could run packages written in that language. This would constitute a one time license fee. The software vendor would distribute his product in relocatable module form requiring purchasers to LINKLIB it into executable form. The benefits are: no bookeeping; no royalty payments; better application programs; many LINKLIB sales. By the way, LINKLIB should probably sell for around $100 and be easy to use (i.e. LINKLIB PROGNAME). I think the idea will work. What do you think?

In response to last month's editorial, Micropro has informed me that they are in the process of fixing the bug I mentioned. They say that in most cases the overflow of the key buffer will only result in a severely scrambled screen and not in the destruction of the file. The exact effect seems to be system dependent. Furthermore, I heartly commend Micropro for reporting to us a fatal bug in Wordstar 2.25 along with its fix. See this month's bug section for the appropriate information.

Finally, Bill Burton has reported to me that in conversation with Microsoft last week he asked when the version of BASCOM containing CHAIN with COMMON would be released (it was promised for last October). He was informed that all programmers had been pulled off the project and that no delivery date could be set. Don't hold your breath. See you next month.

Harris Landgarten

The CP/M Users Group News

We've made such a point in the past of saying "don't order a disk until it's officially announced", that some of you have become "order-shy". So, here's the scoop. When you see a catalogue published in Lifelines, that means the Volume is available. (Incidentally, the contents of catalogs published in other magazines may not be "official" and therefore may not be correct.)

This month, Volume 48, a collection of BDS-C programs and .COM files, is available. See the catalog and abstract printed in this issue of Lifelines. Leor Zolman, BDS-C author, is to be praised for his liberal attitude toward the distribution of his run-time library as it exists in .COM files produced by his compiler: no strings attached.

Misc Disk

The "junk" disk I previously mentioned has now been given the more "favorable" name of "MISC", thanks to Chuck Douds. We are soliciting material for it, material which is perhaps useful to a limited audience, or requires a bit of "hacking" to make it work. Typically, these programs are original ones, made for a particular hardware environment, but which might be of interest to others. Another reason for the MISC disk is to try to get people to submit programs which they might otherwise consider "not up to par" for a regular CP/M Users Group disk. When submitting a program for the "MISC" disk, so indicate.

Bugs

I apologize for any trouble they might have caused in my CV.ASM from volume 40, which I'll be putting in a future utility disk.

Crystal Ball

What's in store for the future? Here's a glimpse into The CPMUG crystal ball, a flavor of what's to come. Perhaps you'll realize you have something you might like to add. I'll also be publishing .ASM files of some of my own programs which were distributed only in .COM form recently.

STAGE-2: I am working on a "general purpose macro processor", "STAGE-2", submitted by Richard Curtiss. This is written about in "Implementing Software for Non-Numeric Applications" by William M. Waite. The language is powerful, yet a bit cryptic. Problems with one of the important sample programs are delaying the releasing of this volume.

SPEED, BVIOS: Bob Van Valzah has just completed updating his superb "SPEED" program as well as his very compact BIOS, "BVIOS" for CP/M 2.2.

DU: When we get another of the popular utility disks completed, it will likely contain the super extensions to my DU program that Ron Fowler did; he made it determine what version of CP/M it is running under, and dynamically find the disk parameters. This works for 1.4 and 2.x. Early tests show it to be very good. Popped unmodified into a Tarbell 51-sector system, it worked perfectly.

VALIDATING DISKS: We are working on a scheme to allow you to verify each file on a CPMUG disk. This will probably take the form of a program on each disk, which will store the full CRC of each file, and allow it to be calculated later. Thus, if you suspect a bad file, you will be able to check it yourself. Keith Petersen has written CRCK which will compute the CRC on one or more files and optionally write a file of all CRCs, so this is what we will most likely use.

RATFOR Jim Mills is currently cataloging and abstracting a RATFOR disk. This is RATional FORtran, a "C"-like pre-processor for FORTRAN programs.

PICTURES: We have received a number of pictures on disks, i.e. the kind you send out to your printer. We are approaching this one cautiously because of potential copyright problems.

BDS-C: Volume 48, announced in this issue, is only the first of what I hope to be many "C" volumes. Just a flavor: Jack Wierda has some super catalog and modem programs in BDS-C; Paul Krystosek has an intriguing set of cryptography programs; I have a trivial curly-brace-checker for those of you who are like me and can't always get them balanced; There will be many more.

ADVENTURE: We are toying with the idea of coming out with an "adventure" disk. It will probably not contain the original adventure, but instead other peoples' contributions to this simulation or role playing gaming. We solicit your original contributions to this disk. Possible contents: a general-purpose adventure interpreter (in assembler) compatible with the two excellent Scott Adams adventures, "Pirate", and "Adventure Island". An additional surprise (well, it won't be any more if I tell about it) is the ability to write your own adventures in a "high-level" adventure language, compile them with an adventure compiler written in MBASIC, then run them with an assembler-written interpreter. All the pieces aren't in place for all this, yet, but we encourage your contributions to such a disk. Incidentally, the "GROW" program published in Creative Computing will also be on this disk. We have obtained their permission to do so. (In general, Creative Computing will allow the programs they publish to be distributed by The CPMUG, providing permission is obtained on a program-by-program basis, in writing. We thank them for this.)

CDOS: Trevor Marshall from Australia has contributed some CDOS utilities. He also contributed some Wordstar customization information.

There are so many more, I can't mention them all. Keep the contributions coming in.

Ward Christensen

CPMUG 86

A new chapter of The CP/M Users Group is forming, for 8086 software. A first disk is being collected. This group is looking for volunteer reviewers, as well as contributions. Let us have your thoughts on what you could bring to such an organization. You can write to CPMUG 86, 1651 Third Avenue, N.Y., N.Y. 10028.

ZOSO

Club Papagallo
Nuevo Laredo, Tamaulipas MEXICO
January 28, 1981

Well here we go again. We're well into the New Year, aren't we? Have your hangovers gone away? Mine certainly hasn't.

OLD BUSINESS

As a few of you apparently see it, this column has suffered from my occasional reluctance to address computer related topics. I concede that such criticism may be valid; so here's the New Year's resolution: I will write only relevant stuff, and do it ever so tastefully! I'm even going to include a hardware fix! Of course, topical reportage was not my only resolution; I also decided that smoking, drinking, cussing, unclean thoughts, and unlawful behaviour had to go as well. My intentions as always are the best, but as the years creep up on me, I'm realizing that I have promised myself most of the same S.O.S. (not a distress call) before...

On page 8 of the February issue is a request from Mr. Carl Warren that I publically acknowledge that we are not the same person. Fair enough, Sir; we are indeed not the same person. In truth we have never even met. However, I must add that if 'a number of people' have suspected that you might be the culprit behind these [Zoso] columns, then I am reasonably certain that I would like you... Also, Mr. Warren, I have always enjoyed reading your columns and editorials, and hope to some degree that the feeling is mutual (if not, I'm trying to improve - see last paragraph).

One of the safe approaches to germane subject matter (i.e. when I'm too lazy or preoccupied to come up with anything new) has been to chronicle how various companies measure up (or don't) in terms of after-sale support. Perhaps this is a good topic with which to 'test the waters'. Let's proceed.

NEW BUSINESS

I recently purchased a California Computer Systems 2810 CPU and was disappointed to discover that this well designed board (and it is!) chose not to work in conjunction with my (unmodified revision 4) Imsai front panel. I have made five or six telephone calls to CCS trying to get whatever technical help might have helped me resolve this problem. On two of these occasions, I was connected to very pleasant and helpful (but not helpful to the point of success) tech consultants; none of the other calls were returned. It might well be a rgued that all of my current troubles with the 2810 are really Imsai's fault, but thrashing the dead is not my style.

Microsoft allegedly promised a new version of their BASIC Compiler, implementing CHAIN with COMMON on or around October 1, 1980. As of late January, it has yet to surface. At least one user of this product (Bill Burton, who reviewed it in an earlier issue), took the time to write Microsoft, in care of the Compiler Product Manager, to mention that the random number generator of revision 5.2 was faulty and that the WAIT statement **of that same version still** did not work (see Oct. 80 Lifelines page 13). I happen to know that Mr. Burton has been trying to get them to fix that WAIT thingy since before it was officially released. Now, more than five months after this letter was sent, Mr. Burton has yet to receive the courtesy of a reply, and for that letter to have gone unacknowledged seems pretty sleazy to me, since his (Mr. B's) review of the BASIC Compiler was pretty damned (oops!) flattering, and in large part the upbeat tone of that review seemed to be based upon hollow promises that a version implementing CHAIN with COMMON would be 'here' last October. I for one am not surprised.

Amongst the items offered by Supersoft Associates of Champaign, Illinois is a package of diagnostic utilities. I bought this last summer, and encountered some [minor] problems. These guys advertise a 'technical hot-line' which is only supposed to be answered when a technician is present. I made three calls to this number, and was told on each occasion that the person who could help me was momentarily away from the 'hot-line' phone, but that person would get back to me promptly. I'm still waiting, but I no longer care. Good luck Supersoft, you won't have me tying up your 'hot-line' (nor order line) any more.

One of the more interesting [full page] ads which Supersoft has placed (check Byte, January or February 1981), is for SSS FORTRAN and an optional RATFOR package. Remember TDL (later Xitan), well, at one point, they also offered FORTRAN, and by mysterious coincidence, that also was SSS FORTRAN. (Note: SSS is an acronym for Small Systems Services of Urbana Illinois). Now I just happen to have TDL/SSS FORTRAN, and it is a very promising item (as of the latest release I have seen, circa 1978). I reluctantly abandoned TDL/SSS FORTRAN, back then, in favor of Microsoft's implementation. This was because without the 'in line code generator', which the (then floundering) TDL had advertised but never released, SSS FORTRAN was too damned (oops again!) slow. Does the code generator exist now? Does the new SSS FORTRAN compiler produce code which requires TDL type linkage editors? Is it the same item? If so, has it been improved? Is the manual still that same skinny thing that tells you (on page 5) that you should obtain the Cress, Dirksen and Graham book on FORTRAN IV with WATFOR and WATFIV, Prentice Hall, 1970, as a "tutorial for TDL FORTRAN" (notice the quote marks?)... And finally, as authors of what seemed three years ago to be an unusually good product, what has SSS been doing in the meantime? If you have the 'dime and the time' to call Supersoft, I would recommend asking all the above questions and perhaps more.

That brings us to the subject of RATFOR. Read the book, Software Tools, by Kernighan and Plauger, Addison-Wesley, 1976. RATFOR is neat; it is also in the public domain (i.e. you can get it free). One fine implementation

thereof (for Microsoft FORTRAN) is available on one of the earlier CPMUG disks. If demand warrants, I will devote part of a future column to how to get an even better RATFOR for TDL/SSS FORTRAN (at the public domain price).

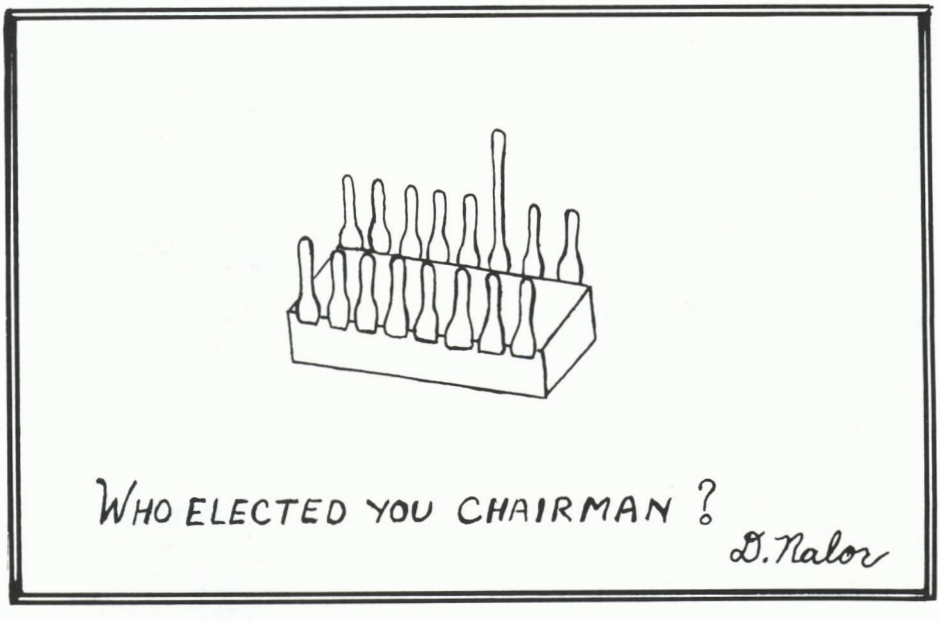
MicroPro has given us WordMaster and WordStar, which incidentally I consider to be superb products. I use both and like both. Some time back, I noticed that there were two or three theoretically valid command sequences which would cause WordMaster (1.06) to self destruct. In the ensuing months I have forgotten all but this one:
*-BB0Pcr *-BB1Pcr *-BB2Pcr etc.

These mean: go to the end of file then return to beginning of file and display the first (or second or third) page/screen. This does to WordMaster 1.06 what Kryptonite does to the Man of Steel. I tried to be a nice guy and alert MicroPro to this problem, certainly not for my own benefit, as I already knew what commands to avoid. My reward for the trouble and expense I invested in the several (unreturned of course) calls I made to them was a chance to hear some snippy broad refuse to discuss the nature of my (nay, their) problem or connect me with anyone higher-up until I recited my serial number over the phone. It goes something like this... Hey, Pal, your house is on fire... I'm sorry, that'll just have to wait until you find your disk and read me the serial number... Get the picture? Fourth of July freaks who enjoy explosions in March, are urged to try the above commands to WordMaster 1.06; just don't bother calling MicroPro unless you have the unrestricted use of a WATS line or are using a blue box.

I bought Microsoft's EDIT-80 when it was first introduced a few years ago. It was a strange item indeed. Amongst its quirks was an adamant refusal to accept lower case commands, a rather major flaw in a text editor. I no longer use this item, but have noted that they just recently got around to revising it. Incidentally, I examined the original release with DDT, and smack dab in the middle was the message from one of their early BASICs asking if I wanted to delete the transcendental functions. If any of you own 1.0 versions of this binary brontosaurus, check it out for yourselves; a cut and paste job if ever I saw one. To be fair, I must mention that it may well be worth it to buy EDIT-80 just for the very good file compare program, FILCOM.COM, (which for reasons I can't fathom is included with an otherwise forgettable editor). Speaking of FILCOM, it seems reluctant to display differences between two binary type files (INT, Pcode, IRL and especially REL). The reasons for this most likely reflect some of Microsoft's direst concerns, just as any of the several ways to get around this problem might very well reflect your indifference to what is best for Microsoft.

Here's the hardware stuff I promised... Several of my friends have purchased the QT Systems Clock/Calendar+ board (with battery backup), which on the surface appears to be a winner. The only problem is that it won't run in certain systems. In good conscience, I can't pass myself off as a digital engineer, but nonetheless I may still be of some help. Oh yes, where was I... The symptoms of incompatibility exhibit themselves as ports from which only the values 6,7 and infrequently 3 can be read. This results in the board returning things like 67:73:67 PM as the time. As I write this, the factory does not yet have a fix available, and seems to be unaware of the problem, but in keeping with my New Year's resolution, the fix appears below (thanks to a few good old boys from the Midwest)...

The fix: Remove U6, a 7410, and bend up pins 11 and 13. Take a 74LS02 and remove pins 1-6. Place 74LS02 over U6 (in piggyback fashion) and solder pin 7 of 74LS02 to pin 7 of U6. Solder pins 14 of both chips together in like fashion. Lift pins 8-12 of 74LS02 clear of their counterparts on U6. Connect pins 11 and 13 of U6 to pin 13 of 74LS02. Connect pins 8 and 9 of 74LS02. Connect pins 10 and 11 of 74LS02. Reinsert U6 (and the 74LS02) in the original U6 socket. Run a wire from pin 12 of 74LS02 through any convenient hole to bus pin 77 on solder side of board. Finally, run a wire from pins 8 and 9 of 74LS02, which were shorted together in an earlier step, through any convenient hole in the board, and connect to either pad 11 or 13 on solder side of board beneath U6 (these pads are already shorted). Voila! This fix not only provides a bit of high-tech sculpture, it gets this board to work properly in obstinate systems.



TIPS CONTEST

The best is the best and this is the tip we selected as winner for our March issue. The others submitted were of a more trivial nature and did not really meet the criteria of originality, utility and economy we use in judging. There must be at least 10,000 of you reading Lifelines; please send in your tips!

"I decided to implement backspacing in CP/M 1.4. CP/M 2.2 has it, and it is nice. Instead of being confused by the deleted characters echoed by DEL/RUBOUT editing during CP/M command input, you have a line that 'looks' right. With this modification, when you press your terminal backspace key, CP/M 1.4 interprets it as a 'charcter delete', and echoes the 08H backspace to your console out routine.

Good points: The BIOS is not modified. It only effects the CP/M BDOS function 10, read console buffer, so it won't impact other programs, and no 'hack' have to be inserted to avoid WordMaster, etc.

Bad points: no longer expands tabs into proper spaces: echoes instead 'uparrow I'. However, I find I never am keying a 'TAB' into CP/M, anyway. Also, if you accidentally press DEL/RUBOUT, a 7F is stored in the input buffer, and echoed to your terminal. Depending upon the terminal or software used, it may or may not show up. It doesn't send a fancy 'space backspace space' so the deleted characters may still appear on your screen, depending on your hardware. You could change your BIOS to either store a space, or send space-backspace-space. Consider these shortcomings before making the changes. I have been happy with them. I now 'never" press DEL/RUBOUT except under WordMaster."

Here is the actual patch. Do the following to a back-up copy of your CP/M, in memory, as it would have been loaded via SYSGEN:

```
;
;Filter: pass BS, instead of tab
;
AT 1259 (WAS) CPI 9, NOW CPI 8
;
;Test for BS, not DEL
;
AT 1301 (WAS) CPI 7F, NOW CPI 8
;
;Modify test to check if we have
;backed to the beginning of the line, so it
;doesn't clobber A.  Also, NOP loading of the
;character to be echoed.
;
AT 1306 (WAS) MOV A,B (NOW) INR B
AT 1307 (WAS) ORA A   (NOW) DCR B
AT 130B (WAS) MOV A,M (NOW) NOP
```

Ward Christensen

A Reader's Comment

February 4, 1981

Dear Mr. Landgarten,

I read in February 1981 Lifelines that you will have an article in the next issue on the Osborne General Ledger Program. I thought I would share with you some of our experiences with that code.

Once operational, the General Ledger is a very useful and easy to use utility. However, we found several problems in setting it up and getting it going. We purchased this in "source on disk" form and there were a number of typographical errors; not surprising since the supplier probably retyped the code from the listings in the boook.

More serious were program errors in the Osborne/McGraw-Hill book. The programs as listed are poorly commented and are prime examples of unstructured code which is difficult to follow. The following source code changes should be noted:

1. The "Posting Update" will not work unless the codes of lines after statement 6000 in program G/L 030.BAS is changed File NO%=1: GOSUB 700 should read Y9=1: GOSUB 700.
2. In the report program G/L 040.BAS line 6420 should read IF L7%=9 THEN LINE COUNT%=60:GOSUB 825\ or else the "extra lines=9" form fee feature will not work properly.
3. In the same report program the line right after 6220 should be moved to two lines above 6460 (i.e., just above GOTO 6220, otherwise the expense total will not print).
4. In the Cash Journal program G/L 080.BAS, the line IF L5% THEN THEN P5=-P5 which occurs seven lines after 6160, must be removed or some totals will not post correctly.
5. Finally, at least when running on a Superbrain, PIP.COM must not be on the general ledger data disk (A). Its presence leads to printer communications protocol problems.

I hope these are a help to you.

Sincerely,
Richard L. Peskin
Professor, Rutgers University

The Osborne Packages: General Ledger

by Martin McNiff

Of the three systems Osborne has published, General Ledger-CBASIC probably is the most widely applicable. There are two reasons why this is so; first, General Ledger accounting systems are all generally similar, much more so than order entry or inventory systems. The second reason is that the Osborne General Ledger interfaces with other software which is already written, and it is relatively easy to interface any CBASIC accounting software to this system.

The General Ledger system contains six programs (not including the menu and two utility programs) which allow you to enter postings manually in a data entry program, or to do so automatically by merging postings in from compatible software (such as the Osborne Accounts Payable or Accounts Receivable systems). The General Ledger prints trial balances and income statements as well as normalized balance sheets and income statements.

ACCOUNT NUMBERING

Your chart of accounts will need account numbers assigned to each account name. In addition, headings, sub-headings, footings and sub-totals will each need an account number. A recommended numbering scheme for the General Ledger system is:

--Assets.....10000.0 to 19999.9
--Liabilities..20000.0 to 29999.9
--Income.....30000.0 to 39999.9
--Expenses.....40000.0 to 49999.9

With each account you put on file in the Account File Maintenance program, you have to specify the account type (regular account, title account, total account or heading account). Title accounts, total accounts and heading accounts are used in the General Ledger Reports program and are explained later in the article. Other things to specify are whether the account is a balance sheet or income statement account and whether the normal balance is debit or credit. Each account has a "total level"; that is, an indicator of up to nine report accumulators for various sub-totals in General Ledger reports. The total level also indents the print position of an account and its amount on the balance sheet or income statement reports. You can also specify how many lines you want to skip after printing the amount on a General Ledger report.

ACCOUNT TYPES

There are four account types in this system: regular accounts, e.g., "Accumulated Depreciation-Vehicles". Regular accounts keep month-to-date, year-to-date, current quarter-to-date and the three previous quarter-to-date totals for the account. You can make postings and dollar modifications to regular accounts.

Title accounts do not hold monetary totals. The account name prints on financial reports and this account's normal balance field sets the sign of accounts which print after it. For instance, if the General Ledger trial balance finishes printing out assets (where debit balances print out as positive) and the report starts printing liabilities, the title account "LIABILITIES" prints to divide the report, and also serves to express credits as positive numbers. If the liabilities title account had the same normal balance field as the assets title account, then all normal credit balance accounts would print with leading minus signs.

Total accounts cause report totals or sub-totals to print. Up to nine "total accumulators" can be used for a General Ledger report. Heading accounts are used only for printing information on the report, for instance, to print "Order and Shipping Expenses" followed by the expense figures on lines below in the report.

GENERAL LEDGER DATA FILES

Aside from a temporary work file, CRT mask file and General Information file, the General Ledger system contains three principal files: the Account file, the internal postings file and the external postings file.

Account File

The General Ledger Account file contains:

--Account Number
--Sub-Account number (used for heading and title accounts)
--Account type code
--Account description
--Report code (Income statement or balance sheet)
--Normal Balance (credit or debit)
--Total level
--Extra lines to advance after printing on report
--Sales account flag (0=no; 1=yes)
--Special Report flag
--Totals MTD, YTD, current and three previous quarters

This file is accessed via binary search, with account number as the key.

Postings

The internal postings file contains entries which come directly from the General Ledger Postings program or the General Ledger cash journal program. Both files merge before updating to the General Ledger, and they are identically structured as follows:

--G/L Account number posted to
--Source code (what system the posting came from; G/L, Payroll, A/P or A/R)
--Date (MM.DD)
--Reference field (six-digit numeric field)
--Posting amount

The Update Program

The General Ledger Update program sorts and merges the two posting files into one before updating the General Ledger. The opening and closing balances for each account print after all postings for an account, in addition to a printed record of each posting merged into the system. Inci-

dentally, users are not required to run this program before any report programs can be run. The system allows you to run any report program as long as you have General Ledger accounts on file.

The Reports Program

This program generates the following reports:

Trial reports: normalized income statement and balance sheet

Income statement: current month vs. YTD, quarterly for current or three previous quarters

Balance sheet: current or three previous quarters

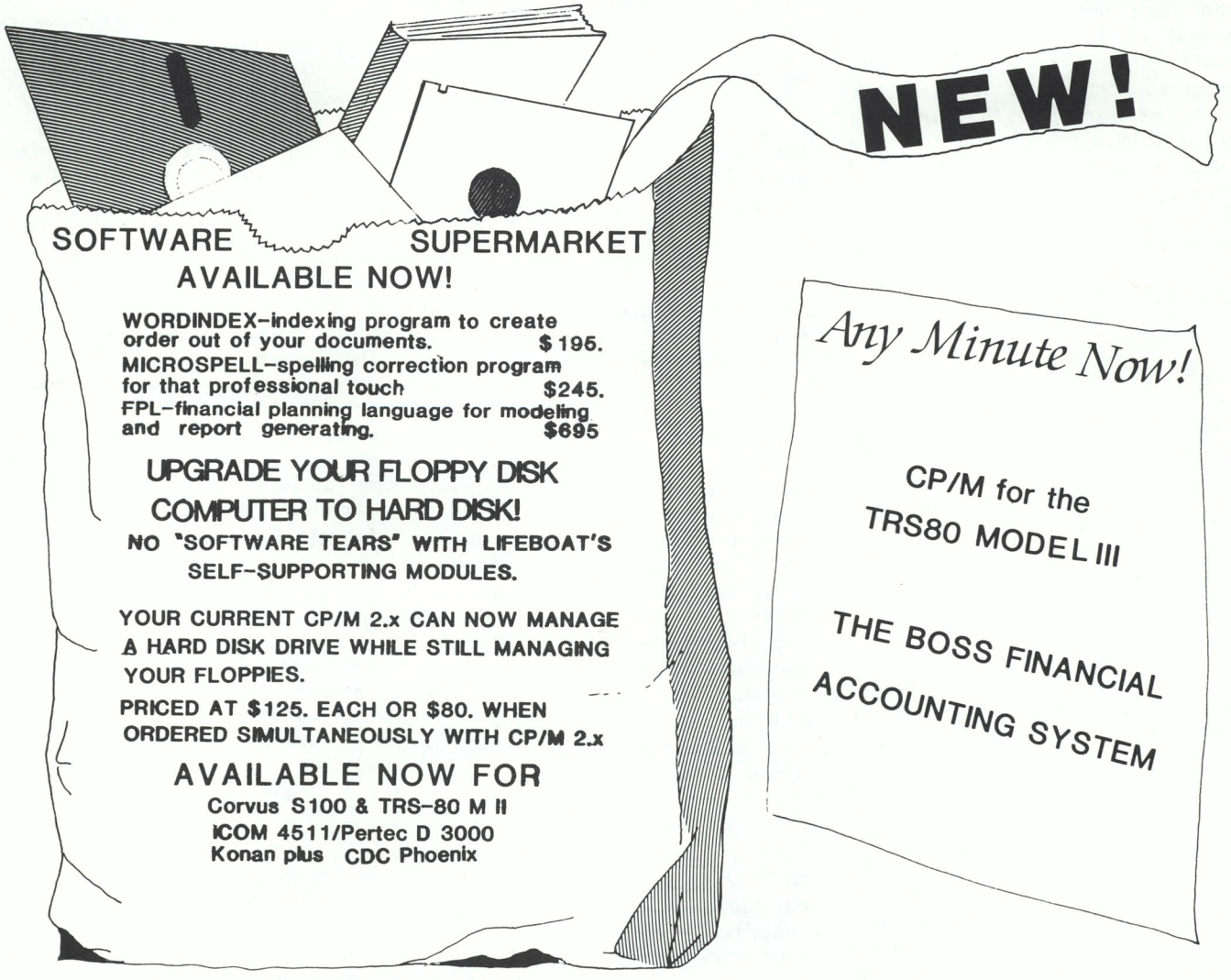
The Reports program also has a provision to move totals at the end of a monthly, quarterly or yearly period.

The Cash Journal Program

This program allows you to enter direct cash receipts and/or cash disbursements to the General Ledger. It is similar, but more powerful than the direct posting program which does not require the operator to stay in monetary balance. In this program you can enter a series of receipts or disbursements with special description fields (e.g., "pay for 8/8/81 UPS" or "return for overpayment") and when you are done, the computer will create postings to the accounts you specified as well as an automatic posting to the cash control account you specify.

Although the description I've given you is kind of a whirlwind tour of the Osborne General Ledger system, it should give you a rough idea of its capabilities. You can create your own chart of accounts using your own numbering scheme, post to the General Ledger manually or automatically, print a wide range of accounting reports and maintain an item-by-item update audit trail, and interface it easily to other software (that is, software you have the source code for). In all, General Ledger is a compact and extremely capable software system. My next article will discuss the Osborne Payroll with Cost Accounting system...see you then.

Advertisement



CALL OR WRITE: LIFEBOAT ASSOCIATES
(212) 860-0300 1651 THIRD AVENUE
TELEX 640693 NEW YORK, NY 10028

THE SOFTWARE EVALUATION GROUP: CONFIGURABLE BUSINESS SYSTEM

by ed paulette

It's obvious that a lot of thought has gone into CBS - the documentation is carefully done with examples of virtually every function that I can imagine attempting with this package. I can imagine attempting quite a lot.

CBS is probably best classified as a data manipulation system using the definitions of our earlier article (Berla & Lehman, Volume I, No.6.). It lacks two significant characteristics of a data base management system: 1) the ability to change the types, order or number of data fields in a dataset without changing the programs which use that dataset; and 2) a clearly enunciated method for representing the structure of the database (the collection of data files or datasets), apart from that implicit in the processing steps which use the data.

It may seem that these are pretty abstract objections, but they significantly affect both the ease of use of the system and the flexibility of the database.

One of the first things I noticed about CBS as I began working through the examples in the manual was its greater speed of operation than I had any right to expect. The system moves from one function to another very quickly. The system has an extensively implemented ISAM file structure which provides rapid access to records when there is an index available on the field you wish to use for access. Every master file has an associated primary index which is automatically maintained during data entry. In addition one can generate any number of additional indices based on any field in the data set. By cascading the production of indices one can sort the database into any desired order. (One index can be built using record orderings for duplicate keys based on a second index.) The technique is slow for large datafiles, but many sort programs alone approach the price for this whole package.

CBS is very easy to install if the user has one of the standard terminals whose control codes are listed in the book. (Another member of our group was unable to get the package to work with his memory-mapped terminal, however.)

The eight-inch single-density CP/M version was conveniently arranged on two disks, one for use when developing new applications or adding to old ones, and a second disk which later serves as the program disk after the application is in operation. One simply mounts a working copy of each disk on drives A and B respectively and proceeds to interact with an application Designer menu to construct the data files and processing steps necessary to implement an application. Even a modest understanding of the system should allow the beginning user to implement her/his first files, to begin entering data and producing reports.

When the development process is over and operation begins, the "Designer" disk in drive A is put aside and replaced by the "Application" disk which had been in drive B. An empty data disk can then be loaded into drive B to hold the data for the application. Thus the new (or even experienced!) user doesn't have to figure out which programs can be erased once an application is up and running.

The manual is organized into two sections: 1) a User's Guide steps one through a series of sample problems which gradually introduce each of the major functions performed by the package; and; 2) a Reference Manual contains a more concise explanation of each program or facility. There are many practical suggestions on how to accomplish common functions which would have been hard to figure out as a beginning user.

Unfortunately, an inexperienced user is going to find it easy to get in over his/her head when designing systems which have several files interacting with one another and where reports need to be produced drawing upon data from several sources. The system is based on the concept of master files and associated transaction files. A transaction file both updates the master and supplements it. This ability to update the contents of one file with fields from matching records in a second file is a very powerful facility indeed. When combined with a facility to perform four function algebraic calculation between fields in a single dataset, there are few data management tasks one cannot accomplish. It just takes a bit of creativity and right angle thinking. Avoid the temptation!

What are the limitations of this system? The most important, I think, is that it is crucial that every processing step and every interrelationship between each datafile be clearly worked out in advance of implementation. It is potentially very difficult to change the design of an application. The information about the structure of the database is distributed throughout the various file definitions, updating and report specifications, and processing steps specified in the menu entries.

This is not a significant problem for the many problems of modest size and relative stability that first time users are often plagued with. A relatively sophisticated mailing list system would be a cinch (but don't try to do market research with it!). The tutorial example in the User's manual provides an example of an inventory system which is somewhat short of complete, but still nearing viability as a real system. Thus, I believe it has a

➡

solid niche of appropriate applications where its cost is easily repaid.

A second limitation is in the amount of editing performed automatically by the system during data entry. The data entry format is fixed so that each field is prompted for in the order of its definition in the datafile. Field names are prompted one per line and do not appear until they are to be entered. Although one can define up to 100 fields, only 17 will show on a 24 line screen at any one time and there is no way to roll forward and backward to look at fields entered on the previous screen or to correct errors which are noticed after a new set of fields begins to appear on the screen. In addition, only the most rudimentary editing is performed on non-Key fields. Date fields are checked to be valid dates (2/29/81 passes); alpha fields are checked for minimum and maximum length and numeric fields for minimum and maximum value. One can implement further checks in additional processing steps, but this must be carefully done to insure valid data.

Those who will be doing a substantial amount of work with mailing lists may consider the optional Mailing label add-on attractive. While the basic package can produce mailing labels with no problem, it can only do so "1-up", that is, a single column of labels at a time. The standard report output is also fixed format so that short first names will be followed by a large blank area before the last name - the label option will both print up to 100 (!) labels across and remove embedded blanks (where specified).

Considerably more significant is a change in the package which is said (by CBS) to be coming in the next release. There will be a major revamp of the Report writer. Currently all the information to be used in producing a report must reside in a single file. That means that one must perform a series of updates to move all the data into the file from which the report will be produced. The report formatting capabilities already available are quite good, but are restricted to what amounts to a formatted listing of the file with three possible levels of sub-totals.

The new version (to be released in April) will allow placement of output fields anywhere on the page, algorithmic specification of calculations, multiple file input to the report and fully nested, "IF .. Then" style conditional selection of data to be included in the reports. (Conditional selection of records is implemented in the current version, but in a much more restricted form.) The changes sound very attractive and could simplify many applications. I expect that the new features will add to the price and I do remain skeptical until the new report module is actually delivered, but watch for it.

The CBS manual has an appendix describing how to use the system on 180K plus mini-disk systems. For relatively small databases, it should be practical to use the system, but even with a modest database there will not be room to implement very many menu options or processing steps - there will be little available room on the program disk! Included with the 5 1/4" version is a "Pause" program which allows swapping disks in mid-jobstream to allow more complicated systems in that environment. Good luck!

Review Summary

Good points:

CBS is well documented, will provide excellent support for data management of relatively simple, but potentially quite large, databases. It requires no prior programming background, no supporting software other than CP/M. And is among the least expensive of the data management products. I encountered no significant bugs during my evaluation.

The ISAM capability aids in producing impressive response of queries on the database once an index on the necessary field exists.

Bad points:

CBS does not ostensibly use any of the common data models and is conceptually hard to grasp when applied to more than moderately complex problems. It is likely to prove inflexible once implemented, due to the embedding of data structure information throughout the specification of the application. The lack of tight editing on data during input may allow invalid data to enter the database unless careful batch editing is done after entry.

Reference section:

See the introductory articles for details on terminology and the evaluation format used:
Lifelines Vol.1, No.4, p.7; Vol.1, No.5, pp.4-6; Vol.1, No.6. pp.12-14, Vol.1, No.7, pp.2-5.

Send suggestions for software evaluations and other correspondence to:
The Software Evaluation Group, c/o Lifelines, 1651 Third Avenue, New York, New York 10028.

Request for information: Does anyone know what happened to the implementation of a relational style database management system called RAINS (formerly MIDAS), developed by Michael Sperling for CitiBank? Last I heard of it was a year ago. If you have any information on the fate of this system, please contact me at the above address. (Ed Paulette)

Ed Paulette is Associate Executive Director of The Michigan Evaluation Resource Center where he is engaged in the implementation and support of financial applications and client data base systems for human service organizations. His address at The Center is 338 South State Street, Ann Arbor, MI 48104

TABLE I
Facts & Figures

Package or Version name:
Configurable Business System, Version 1.2 with Label Option

Price: $395 basic system
$ 80 Label Option

Systems available for:
Evaluated on LB CP/M 2.24 for TRS-80 Model II; said to run on CP/M 1.4 & 2.2, CDOS/CROMIX, MDOS, SDOS, DMA.DOS, and other CP/M compatible op systems.

Required supporting software: None

Memory requirements:
Evaluated in 64K system. Said to require 40K of user memory (TPA + CCP)

Diskette capacity required:
Instructions provided to allow operation with two 180K drives. The second drive would be entirely free for data storage, but application complexity would be restricted - see narrative.

Utility programs provided:
None, but Menu subsystem could be used without the rest of the system.

Record size & type limits:
Internal files are fixed length, binary, direct access with MTREE ISAM indices. Non-alpha data types are automatically aligned to appropriate boundaries which may generate wasted space due to padding. Allowable record lengths are 32, 64 and multiples of 128 bytes up to 2048 bytes. 65535 records may be stored in a single database.

External datafiles can be produced by the report program with record lengths up to 132 bytes in fixed length ASCII "lines". Multiple lines per logical record can be output. ASCII fixed format lines of any length can be input delimited by carriage returns.

The Label option allows (among other things) the production of variable length "lines" and the specification a single character field separator.

Portability:
Transfer to another computer capable of running CBS should be relatively simple (outside of file size constraints), but transfer to another DMS would likely require substantial redesign. Data can be transferred via the external file facility.

User skill level required:
A novice should be able to use the system to operate a prepared application on a day to day basis. A skilled amateur should be able to implement moderately complex applications (such as Problem #2), but beyond moderately complex applications professional level skills will be required.

System upgrade policy:
There is currently no subscription service available. If there is a difference in the list prices between two versions, the new version can be obtained for the difference in the list prices. If there is no difference in list price between two versions, an upgrade costs $25. If upgrade is necessitated by significant bug corrections, a user must only cover the cost of shipping. In each case the original disk must be returned and the charge covers any necessary documentation.

TABLE II
Qualitative Factors

	Rating*
Documentation	
organization for learning	4
organization for reference	4
readability	4
includes all needed information	4
Ease of use	
initial start up	6
conversion of external data	4
application implementation	3
operator use	6
Error recovery	
from input error	5
restart from interruption	4
from data media damage	3
Support	
for initial start up	5
for system improvement	5

* Ratings in this table will be in a 1-7 scale where:
1 = clearly unacceptable for normal use
4 = good enough to serve for most situations
7 = excellent, powerful, or very easy depending on the category

TABLE III
Data Management Capabilities

A. Underlying Data Model

1. Data Types

Alphanumeric, Integer, single precision, double precision, date

2. Relationships

Supports many too many relationships in a somewhat circuitous fashion (through update facility). The overall data model is not well explicated. Lack of a clearly explicated data model makes the system much more difficult to use as a developer.

B. Functions Provided

1.a. Data dictionary maintenance

File definition is part of the datafile itself. Data independance is minimal, processing specifications are part of field definitions.

b. Data reorganization & conversion

File definitions can only be modified in preallocated padding fields without losing data. Files can be reorganized by writing contents in ASCII, rereading and posting.

2.a. Data entry and editing

Prompting one field at a time, simple type and range validity checks, no screen layout capability. Entry and editing functions well integrated and consistent. Retrieval for editing can be done by primary or alternate (pre-defined) keys. Only primary key index is updated during entry and editing.

b. Report generation

(See narrative re new version.) All data must reside in a single file at time of report generation. Can be assembled via multiple posting steps if necessary. Only report function allows case selection based on non-key field contents or boolean comparisons. Displayed or non-displayed fields can be calculated (+,-,*,/). Three levels of totals available as either average, count or sum. Multiple lines per logical record. Report can be written on disk with or without headers (conversion technique).

3.a. Data selection by predicate

Only available in report program except for limited capability through alternate indexes in edit function.

b. Data joining & relating multiple data sets.

Join can be simulated using multiple posting steps - must be carefully planned. Strong, versatile Update function allows relating ¶

c. Calculations on data

Primarily through report writing function. Reports can be written onto disk and used as transactions to update original datasets. Very simple operations can be performed in using the Update function.

4.a. Data independent application interface

None

OOPS

On page 16 of the February Lifelines (Vol. I, No. 9), under the Version List, Datebook is listed as having a memory requirement of 32K. This should be 48K.

In the same section of the same issue it is stated that Selector IV requires 48K; it actually demands a 52K system.

On the second column of page 14 in the same issue, SBASIC is captioned, "by Compiler Systems". It should read "by Topaz Programming".

BASIC COMPARISONS: SBASIC VERSION 5.3 PART 1

—BILL BURTON

Author's note: S-BASIC, created by Topaz Programming of San Diego, has only recently been released, after more than two years of in-house development. I knew nothing at all about S-BASIC, until I was given an evaluation copy just a few days ago. Language compilers are significant and complex products which should only be the subjects of authoritative [sounding] reviews, if the reviewer has a real working knowledge, gained from first-hand experience implementing non-trivial programs. I plan to evaluate S-BASIC in much fuller detail from such a vantage point in a future issue, but this first installment will deal mainly with my observations about S-BASIC derived from a few hours of experimentation and reading the manual. I learned several interesting things during conversations with Bob Goodman of Micro-Ap and Gilbert Ohnysty, the author of S-BASIC. (Micro-Ap is the distribution representative for Topaz). I would also like to acknowledge the courteous cooperation of both these gentlemen, which has been of great help in preparing this introductory article.

S-BASIC is a 'true' compiler which unlike either CBASIC or the BASIC-80 compiler, produces directly executable code (a CP/M COM file upon successful completion of a single step--compiling). By contrast, CBASIC2 produces intermediate code which is processed by an interpretive program; the BASIC-80 compiler produces a REL (relocatable) file which must subsequently be linked with relocatable modules from the BASLIB library before an executable COM file results. The important similarity to note is that both S-BASIC and the BASIC-80 compiler need to expand compiled program statements via library reference. In this context, the relative merits of a 'true' compiler and one which must resolve library calls via linkage in a second step are arguable. The advantage of S-BASIC is that only a single operation is needed to transform acceptable source code into a working program. Although BASIC-80 requires an extra linkage step, it offers its users the capability of designing and manipulating REL type library modules. Either approach could reasonably be the preferred choice for a given environment.

S-BASIC is a most unusual offshoot of BASIC which in many ways is not like BASIC at all. It appears that the designers of S-BASIC have taken some of the features of a full disk BASIC and integrated these with block and procedure oriented control structures resembling those of Pascal, Algol and PL/I. It is probably fair to assume that some users will insist that S-BASIC is an altogether new language, but one especially suited to those who have a background in BASIC.

Learning to program in S-BASIC would probably be fairly easy for [experienced] BASIC programmers, but S-BASIC is so different from other BASICs, including CBASIC and BASIC-80, that converting existing programs to run with S-BASIC might demand as much work as a total rewrite.

The first step in preparing programs for S-BASIC (or virtually any other compiler) is to create an ASCII source program. My first attempt at using S-BASIC consisted of reading the chapter in the manual devoted to compiler operation, and then trying to compile the following simple program which I (incorrectly) assumed would work unmodified:

```
10 FOR X=1 TO 1000
20 FOR Y=1 TO 1000
30 NEXT Y
40 PRINT X,
50 NEXT X
60 PRINT : PRINT
70 GOTO 10
```

This program produced a fatal error message on each of the first six lines and S-BASIC reported endless attempts to compile line 70. At this point, I started to read the manual from the beginning, which is of course what I should have done in the first place.

Upon consulting the manual, it became apparent that the problems in lines 10-50 had resulted from my not having declared the variables X and Y. It is a rigid requirement of S-BASIC that every single variable be explicitly declared as to type, and in the case of string variables, maximum permissable length as well. Acceptable types of variables are:

```
REAL.DOUBLE
REAL
FIXED
INTEGER
STRING
CHARACTER
```

Generally, the above variable types would be declared with the VAR/VARIABLE statement, but they could also be declared by the COM/COMMON, BASED or DIM/DIMENSION statements. The BASED declaration gives the programmer some control of where a variable that can be directly accessed by location will be stored in memory.

I added a line to my test program which declared the variables X and Y as integer, and then attempted to recompile, only to find that lines 60 and 70 were still causing problems. Eventually, I figured out that the problem with line 60 was two statements in the same line. The manual does not imply that multiple statements are legal, but does not specify that they are illegal either. I would suggest a change here, as users with backgrounds in BASIC are apt to presume the validity of standard BASIC constructs (as I did). I made an educated guess that appending an END statement would allow line 70 to be processed correctly (it did). Again, I would fault the manual on this point, as I could find no mention of the need for END statements as the last line of a program, and indeed the sample programs included in the manual do not use the END statement (as the last program line). Author's note: Mr. Ohnysty informed me that the END statement is indeed optional, but this was not my experience.

I was successful getting my test program to compile correctly on the third try. The corrected program (with unnecessary line numbers removed) appears below:

```
     VAR X,Y=INTEGER
  10 FOR X=1 TO 1000
     FOR Y=1 TO 1000
     NEXT Y
     PRINT X,
     NEXT X
     PRINT
     PRINT
     GOTO 10
     END
```

It should be noted that the PRINT formatting with comma of S-BASIC (see line 40 in original example) presumes an infinite console width and does not recognize the physical end of a line. Every value printed by line 40 appears 14 logical spaces to the right of where the one preceding began. This problem, coupled with the fact that S-BASIC has no built-in provision for specifying console or printer width, requires the programmer to either code around the default or accept some rather unattractive formatting. I feel that being unable to specify console and printer widths is a shortcoming. Mr. Ohnysty indicated that this capability is under serious consideration for a future release.

My next experiment with S-BASIC involved comparing speed and size of compiled programs with those produced by the BASIC-80 compiler (which produces physically large programs that execute very fast). For this I adapted a Shell-Metzner sort program to both languages. This program filled an integer array of 1000 elements with the descending values 1000 to 1, and then sorted these values to ascending order. The COM file produced by BASIC-80 was 13K long and executed almost twice as fast as the 9K COM file produced by S-BASIC (with debug code disabled). It is very important to note that the default for the compiler is to include line-referencing debug code unless instructed not to (with the $LINES command at the beginning of the source file). The manual mentions that an overhead of approximately 7 bytes per line is involved with the default. What the manual does not mention is that programs run **much** slower with the debug code enabled. The sort benchmark executed in 25 seconds with debug code disabled, but when the $LINES command was omitted from the source, the program required 90 seconds, more than three times as long!

After having experimented a bit further with S-BASIC, I became convinced that its likeliest use would be in business rather than scientific or recreational applications. This was confirmed in my conversation with Mr. Goodman.

The major requirements of a business oriented language are all fully implemented in S-BASIC. These would include random file I/O, dynamic array dimensioning, BCD representation, PRINT USING and (above all) CHAIN with COMMON. By way of comparison, all of the above are also included in CBASIC2, but only random file I/O and PRINT USING are available with the BASIC-80 compiler.

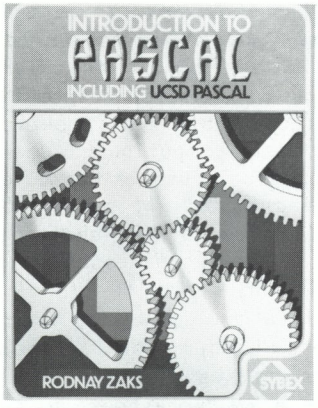
CHAIN with COMMON is important for business applications as finite memory space will support much more elaborate programs. Dynamic array dimensioning is also important because the performance of much good software is dependent on the amount of otherwise unreserved memory which is available. Without dynamic array dimensioning, the programmer must allow for the most demanding (worst) possible case, and usually this means that a lot of memory space is wasted.

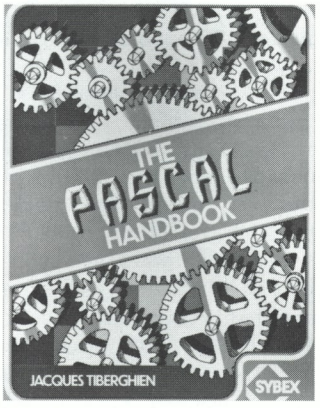
One subtle point which involves a tradeoff is whether to implement ASCII or binary as the storage format for disk records (especially random records). Of the two, the binary format can be more compact on tape or disk, and processing binary records is usually more machine efficient. On the other hand, the ASCII format offers the advantages of greater transportability, and more importantly, information stored as ASCII characters can be maintained, repaired or modified by external programs (typically a text editor or word processing package). Of these two mutually exclusive designs considerations, I have a rather strong personal preference for ASCII storage. Random records written by CBASIC2 are in ASCII format and those written by BASIC-80 and S-BASIC 5.3 are binary. Mr. Goodman and Mr. Ohnysty have both indicated that revision 5.4 of S-BASIC, (scheduled for imminent release), will support either binary or ASCII data storage representation at the user's option. There are many very clever and uniquely useful features to be found in revision 5.3 of S-BASIC, and the choice of data storage protocols will be another valuable addition.

Mr. Goodman mentioned one intended purpose of S-BASIC which had not occurred to me, that being a teaching aid which would allow BASIC programmers to make a smooth transition from BASIC to structured, procedure-oriented languages such as Pascal, PL/I or Algol. This is a point well taken. S-BASIC might prove to be an ideal method by which BASIC programmers can master new [language] skills in minimal time.

It would appear that S-BASIC will compete primarily with CBASIC2. If this is the case, it will be interesting to see what impact S-BASIC will make in a market which to date has belonged to CBASIC2. The factors which may ultimately determine the outcome have little to do with the relative merits of these two languages. CBASIC2 already exists in thousands of installations, and this may prove the greatest factor of all. Both have been conceived as transportable languages but in different ways. CBASIC2 for TRSDOS and UNIX will be released shortly, which means that applications appearing as token coded INT files, developed on any CP/M system will (with minor changes) be suitable for use on [some] Radio Shack and Digital computers. S-BASIC has also been designed for portability but at the source code level. This will require target machine compilers for every different machine. Mr. Ohnysty plans to implement a 'toggle' in S-BASIC which would permit the programmer to specify either portable pseudo-code or absolute target machine code as compile time output.

In the next installment I will continue in greater detail.





SYBEX HAS IT ALL...

INTRODUCTION TO PASCAL (Including UCSD PASCAL) by Rodnay Zaks—Comprehensive, easy-to-use introduction to standard and UCSD PASCALS. A complete tutorial for the novice as well as the experienced programmer, including exercises, advanced concepts, extensive reference appendices.
(Ref. P310) 440 pp., 130 illustr., 2nd Edition, $14.95

THE PASCAL HANDBOOK by Jacques Tiberghien—Unique dictionary of all the features for most existing versions of PASCAL (Standard, UCSD, Jensen-Wirth, OMSI, PASCAL/Z, HP-PASCAL). Each entry includes definition, description, diagram, details of implementation, variations and examples.
(Ref. P320) 340 pp., 220 illustr., $14.95

THE CP/M HANDBOOK (with MP/M) by Rodnay Zaks—Complete guide and reference handbook for CP/M. Includes commands, PIP, files, Editor, reference guide, MP/M commands and programs, troubleshooting, remedial action and more.
(Ref. C300) 336 pp., 100 illustr., 2nd Edition, $14.95

Mail To: SYBEX
Dept. LL31
2344 Sixth St.
Berkeley, CA 94710
Phone Orders: 415/848-8233

PLEASE SEND ME:
☐ The CP/M Handbook (With MP/M) $14.95
☐ Introduction to Pascal (Including UCSD Pascal) $14.95
☐ The Pascal Handbook $14.95

NAME_____ ADDRESS_____
CITY_____ STATE_____ ZIP_____
ADD ☐$1.50/book UPS or ☐75¢/book 4th class mail or ☐$8/book overseas airmail
Total Amount Enclosed_____or charge my ☐VISA ☐MC ☐AM EX
Card No._____ Signature_____
Expiration Date_____ ☐ Please send me your free catalog.

The CP/M Users Group Volume 48: Catalogue and Abstracts

```
DESCRIPTION:    The BDS C Sampler Disk, Compiled by Leor
Zolman, BD Software 11/29/80

-  Manual for BDS-C to give you a "flavor" of the
compiler
-  Good programming examples in C
-  Interesting games to play
-  Good use of H19-H89 (Programs with (H) are H19-H89
only)

Most files are included in .COM form to allow you to
execute them without having BDS-C.  Be careful of
RALLY.COM as it needs to do hard-coded I/O to detect
console OUT status.
NUMBER  SIZE NAME          COMMENTS

         3K  -CATALOG.048  CONTENTS OF CP/M VOL. 48
         6K  ABSTRACT.048  Abstracts of the programs
48.1    17K  BDS-C.DOC     BDS-C users manual (*)
48.2     7K   BDSCIO.H     The standard library used
                           to compile the programs
48.3     8K  BUGS.C        (H) makes things crawl around
                           on your screen
48.4    11K  BUGS.COM
48.5     9K  HLIFE.C       (H) H19/H89 version of 48.9
48.6     8K  HLIFE.COM     (H)  "      "      ".10
48.7    13K  HSTONE.C      (H) "Awari" game program -
48.8    12K  HSTONE.COM     "   "
48.9     9K   LIFE.C       Good implementation of
                           Conway's LIFE algorithm
48.10    8K  LIFE.COM
48.11    5K  MM.C          Mastermind interesting guess-
                           ing game.  Try it and see.
48.12    7K  MM.COM
48.13   11K  OTHELLO.C     Game program
48.14   12K  OTHELLO.COM      "
48.15    7K  PPONG.C       (H) "Polish Pong" bounce the
                           ball off "/" and " "
48.16    7K  PPONG.COM
48.17    8K  PRESSUP.C     Game program
48.18    8K  PRESSUP.COM      "   "
48.19    1K  R2.MAP        (H) map for rally game
48.20   10K  RALLY.C       (H) road rally arcade game
48.21   10K  RALLY.COM     (H) .COM of 48.20, but...
                           Caution!! Hard-coded I/O req'd
48.22    2K  RALLY.MAP     (H) another map for 48.21
48.23   10K  STONE.C       "Awari" game with "very smart
                           computer" opponent
48.24    9K  STONE.COM
48.25    2K  TABIFY.C      Turn multiple spaces into
                           proper CP/M tabs
48.26    6K  TABIFY.COM
48.27    7K TTT.C          Tic-Tac-Toe that "exhaustive
                           ly" tests for best move
48.28    6K  TTT.COM
48.29    3K  VOLUME48.DOC  Leor's own comments

(*) The original was 22K, but "TABIFY" deleted 5K of
spaces!!
```

Leor Zolman's BDS-C is rapidly becoming one of the most popular non-BASIC high-level languages under CP/M, for both the hobbyist and the professional. A nice structured language at a reasonable cost, with good documentation, and good execution speed.

"C" itself it a highly structured language, implementing the "classic" requirements for structured programming: It can either directly or indirectly perform DO WHILE, REPEAT UNTIL, DO CASE, as well as a rich "FOR"-type capability.

With a few exceptions, BDS-C implements "C" as covered in the Kernighan and Ritchie "The C Programming Language" book, which comes with BDS-C. I quote the Lifeboat Ad:
"Supports structures, unions, 2 dimensional arrays pointers, recursion and overlays. Features optimized code generator, variable sized buffers for file I/O, and capability to produce ROMable code. Includes macro package to enable user to produce linkable modules with MAC. Floating point functions, full run-time package and machine code library sources provided. Linker, library manager, and textbook provided. Compiler lacks initializers, statics, floats, and longs."

For the convenience of "experienced" C programmers, BDS-C allows, for example, "register specification" for a variable, which, while not implemented, is syntactically accepted. The most obvious thing missing is a data type of "float", but a floating point subroutine package is included, and a modified "printf" to support it.

Due to the more recent versions having very complete source code for run-time packages and the system libraries, many excellent programs have been "squeezed" off of the single density distribution disk.

This volume, in part, represents a source for these very good source programs, while offering several interesting .COM files for those who do not have the BDS-C compiler.

The Users Group solicits additional BDS-C programs, and hopes to regularly come out with disks containing BDS-C programs.

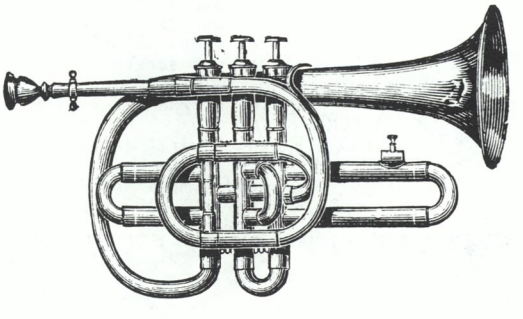
The programs on the disk are primarily intelligent game programs, with some H19/H89 programs which take advantage of this CRT's capabilities, such as graphics. A good utility, TABIFY, is also included. See Leor's own "VOLUME48.DOC" on the disk for comments. Some additional comments about the programs on the disk:

LIFE.COM: Nice implementation. Enjoyable code to read, as are most of the programs. An interesting touch when you decide the configuration is no longer interesting - press your "del" key, and if you don't smile, you're made of stone.
MM.COM: So you want to play a simple game of letter guessing, eh? Try this one. But watch your ego: it can be deflated. MasterMind is a "simple" game that makes you think. The computer generates a random "word" consisting of 4 letters from A-F. You simply "guess" 4 letters at a time, and MM tells you how many are "hits" i.e. the right letter in the right spot, and how many are misses. Thus you deduce the missing pattern. The game goes a bit slow, however (at least at 2MHz). Why? MM is computing how many possible words exist based on the clues it has given you. When this number reaches "1", it says: "You should have it by now". It becomes a real challenge to see how few times you can keep that message from coming out, and is a "real thrill" to "beat it" - especially a couple times in a row. However, having it "know" YOU should "know", but you "missed" catching on for, say 5 turns, makes you feel like a real dummy. I finally made a MM "worksheet" to try to be more efficient. It looks like:

```
                AB ¶ AB ¶ AB ¶ AB
                CD ¶ CD ¶ CD ¶ CD
                EF ¶ EF ¶ EF ¶ EF
Thus I guess:   A    B    C    D
and hear:                          0 hits, 2 misses.
```

I cross off A in the first column, B in the second, etc. You can ONLY do this if there were 0 hits. Still you'll be amused.
OTHELLO.COM: Have you the patience to beat this one?
PPONG.COM: fun H19/H89 game. Deflect the ball with "/" and " ". I was amazed how fast it was on a 9600 baud terminal!
PRESSUP.COM: A challenging game, not quite as bad as Othello.
RALLY: This sounds so interesting I almost bought an H19. If I had actually seen it in operation, I bet I might. Randy Suess ran it and said it was excellent, except that the screen flickered a bit due to lots of reverse video "...but I have an older H19 and the new ones may be better".
STONE.COM: You get to specify how "hard" the computer works to beat you, and if you let it work a while, it's nearly unbeatable.
TABIFY.COM: A nice utility to delete spaces from a file, inserting tabs where appropriate, based on the CP/M convention.
TTT.COM: Hmmm, what could be new in a Tic Tac Toe game? Well, brains, and wit for two things. Ex: it puts its "X", you put your "o", it thinks a while, and says "I've got ya". If it thinks for a while, and after several pieces are on the board, doesn't say "I've got ya", then you MAY be on your way to a rare win, or more likely a "cat" game. Play it and see.
VOLUME48.DOC: Leor's own comments.

Abstracts by
Ward Christensen



BUGS

CP/M on North Star Disk Revision 1.0 Users Notes
In Appendix A, under the heading "Format Information", the type information for CP/M quad capacity reads "D240" where it should read "D112".

CP/M 2 On Micropolis
Version 2.20
A bug in this version causes SYSGEN and COPY to generate unbootable systems under certain conditions. The problem occurs if the destination drive is not A, and if said drive has not already been accessed. A patch sheet to correct this problem is available from Lifeboat Associates. If you need system copies immediately, the best solution is to use A as the destination drive.

WordStar
Version 2.25
The use of the FIND or FIND AND REPLACE commands (ctl. QF and ctl. QA) will cause a fatal error if the "W" (look for whole word matches only) is specified. An update is expected soon. (See the fix below to correct this bug.)

BUG FIXES

CDOS
Versions 2.17 and 2.36
An undocumented feature of Cromemco's CDOS operating system is the ability to toggle Read after Write Verification with versions 2.17 and 2.36. The default is to have Read after Write enabled. Under certain circumstances, it would be advantageous to have this turned off. As an example, XFER will copy files faster if CDOS does no verification (sector by sector), and XFER does it instead (compares only after the file is copied). This can be done as follows:

A.**verify** (Check status)
Read after write
A.**verify no** (Disable)

A.**verify** (Check)
No read after write
A.**xfer/v b:=*.***

..... files copied here .....

A.**verify yes** (Enable)

A.**verify** (Check)
Read after write
A.

WordStar
Version 2.25
The following patch, using DDT, will solve the bug in the FIND commands of this version:

```
DDT  WS OVYL1.OVR
S    1DAF    7F
                6F
     1D8A    1F
                1E
SAVE 107
     WS OVYL1.OVR
```

This problem has been resolved in version 2.26.

CBASIC
Version 2.07
This DDT patch to CRUN will correct defects in the routines for SADD and MATCH functions, permitting T/MAKER to run with this version:

29A4H Change 7EH to 84H
1406H Change 01H to 00H

FPL (Financial Planning Language)
The FPL language, which is implemented in Microsoft BASIC, provides sophisticated modeling techniques for microcomputer users. The available facilities rival or surpass those found on major time-sharing services. Features include:

1-Up to 170 rows and columns on a single work sheet. (Dependent on machine config).
2-Nine character row or column names for readibility
3-Data generatation:
 A-automatic addition of entered items
 B-automatic repetition of values
 C-a row or column will 'grow' by addition, subtraction, multiplication or division by a constant.
 D-a value may be amortized over a specified number of months or years.
 E-a row or column will grow to a target value provided by the user.
4-Data processing:
 A-common algebraic expression are entered by the user for processing.
 B-IF, THEN, ELSE... logic is supported.
 C-cell calculations are allowed.
 D-multiple statements per line are allowed.
 E-BASIC code may be entered inline where special functions are needed.
 F-comments are allowed.
 G-data prompting facilities are provided, allowing non-technical personnel to use previously established models.
 H-the order of processing is completely controlling by the model builder.
 I-external data files may be used.
5-Analytical capabilities:
 A-very powerful WHAT IF, sensitivity analysis.
 B-user may control the display, showing desired rows and columns in any order, by moving the 'window' around the worksheet. Formatting controls are provided.
 B-user may employ all data generation capabilities ineractively, applying the rules to new data and examining the results.
 C-'visible calculator' allows simple changes by calculation - multiply a row or column by a factor, add a constant, zero out a row or column, calculate a cell value.
 D-user may save any subset of his data matrix to disk for later use or consolidation.

E-partial or entire matrix may be saved, or merged with current matrix. Entire matrix or any subset may be processed by arithmetic operators with either constants or another matrix.

6-Report generation:

A-any number or reports may be produced from any number of data files in a single pass.

B-titles and footnotes may be a permanent part of the report or entered at runtime.

C-user may control the order of display, showing any columns or any rows in any order needed.

D-Complete formatting facilities are provided.

7-Totally menu driven.

8- The analytical module uses English like commands

FPL requires a 64K CP/M system, at least 120K of disk storage, Microsoft BASIC version 5.2 and a CRT with cursor addressing and clear screen capabilities.

Spellguard

This package of programs compares documents with its 20,000 word dictionary; mismatched words are presented as possible errors. The user corrects the erroneous words and may at the same time add new words to the dictionary.

WORDINDEX

This product is designed to help users of WordStar maintain large documents; WORDINDEX is activated by dot commands imbedded in WordStar files. Its facilities include the generation of tables of ontents, lists of figures and lists of tables, sorted indices. WORDINDEX also produces index words on two levels, and master and sub-references. Section numbers and page references, whenever the document has been changed. These functions are performed automatically. This program can also be employed to create indices for documents which do not exist in machine-readable form. WORDINDEX includes a user-selectable print control for all headings. On a 2 MHz Z80 computer, it processes between 8-15 pages per minute.

New Versions

OASIS

Version 5.5

This version permits up to fifteen users to access a single printer. With its SPOOLer twenty-six queues, can be defined by the users.

PLINK

Version 3.25

The following bugs have been corrected in this new version:

1-PLINK versions 3.23 and 3.24 would incorrectly attempt to read file CRTDRV.REL when Microsoft COBOL modules were linked.

2-PLINK would stop with an error message when it encountered duplicate symbols. The new version uses the first definition it meets, issues a warning message, and continues. The new version is intended to accommodate software libraries normally containing duplicate symbol names.

3-PLINK now permits the use of INCLUDE and EXCLUDE statements on library files; formerly it went into an indefinite loop.

4-An error #30 no longer results with files produced by Microsoft COBOL version 4.01. However, COBOL segmentation is not supported and error #51 will result from its use.

PLINK version 3.25 has been tested with the following compilers: Microsoft COBOL 4.01, Microsoft FORTRAN 3.31, Cromemco COBOL 3.01, Digital Research PL/1 1.2(index .IRL files not supported), MT Microsystems Pascal MT+ 5.0 (.ERL files for input to the disassembler are not supported.)

Tarbell CP/M Database System

This updated version includes variable length fields, field names of any length, sequential or random files. It requires CBASIC. This system includes utilities for creating and changing files, entering and accessing data. It prints mailing labels and letters. The package includes some non-database programs for inventory control and cross-country flight planning.

TEXTWRITER

Version 3.5

This version encompasses the following bug fixes:

1-When footnotes are too long to fit at the bottom of a first page, they are now correctly continued to the next page.

2-Earlier versions stripped some byte values from the output stream, but now all the byte values are correctly passed to the output device by the OUT command.

3-In index printing, long lines are no longer truncated, but are wrapped around properly.

4-The ctl. T and ctl. V commands now correctly handle NEC Model 5510 and 5520 superscripts and subscripts.

With user-defined control sequences, this version can handle all Centronics Model 737 features, except right margin justification while performing proportional spacing.

WordStar

Version 2.26

These bugs have been fixed in the new version.

1-No dot commands now print.

2-The memory map for Flashwriter II Videoboard has now been corrected to use F1000 instead of V1000.

3-The COPY command no longer writes on an empty sector in the file.

4-User-defined print functions now work on all printers, including daisy wheel, parallel, and serial.

5-After using super- and/or subscripts, a teletype-like printer will now return to the proper line.

6-INSTALL for Diablo 1640 and 1650 now reset the printer on exit to 66 lines per page.

7-The alternate character set for parallel printers now works. (This has not been thoroughly tested.)

The TVI 912/920 installation module has now been optimized, allowing faster screen update. The following terminals are now supported (not fully tested): ADDS 20 and 40, Ann Arbor 6080, Data General D100 and D200, Hazeltine 1420, IBM 3101, TRS-80 Model II with Lifeboat CP/M, Zenith H-19.

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 committment 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

VERSION LIST

2/12/81

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	8080	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	1.3B		CP/M	8080	52K	820/40	Uses IT WORKS run time pkg.
Accounts Payable/Peachtree	10-10-80		CP/M		48K	530/40	Needs Microsoft BASIC 4.51
Accounts Receivable/Cybernetics	3.1		CP/M	8080	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	530/40	Needs Microsoft BASIC 4.51
Accounts Receivable/Structured Sys	1.4C		CP/M	8080	56K	820/40	Uses IT WORKS run time pkg.
ALDS TRSDOS	3.38		TRSDOS	8080	32K	80/25	Is TRS-DOS Macro-80
ALGOL 60 Compiler	4.8C		CP/M	8080	32K	199/20	
ANALYST	1.0		CP/M	8080	52K	250/15	Needs CBASIC2
APL/V80 Compiler	3.2		CP/M	Z80	48K	500/30	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	5.2	5.2	CP/M	8080	40K	325/25	
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	325/25	Includes Versions 4.51 & 5.2
BASIC Utility Disk	2.0	2.0	CP/M	8080		50/35	
BSTAM Communication System	4.4	4.4	CP/M	8080		150/10	
BDS C Compiler	1.42	1.42T	CP/M	8080	32K	145/25	Includes "C" book
Whitesmiths C Compiler	2.0		CP/M	8080	60K	630/30	
BSTMS	1.2	1.2	CP/M	8080	24K	200/15	
BUG / uBUG Debuggers	2.02		CP/M	Z80		129/25	
Cash Register	2.0	2.0	CP/M	8080		805/40	
CBASIC Compiler	2.07	2.17	CP/M	8080	32K	120/15	2.17 is modified vers. 2.07
CIS COBOL Compiler	4.3,1		CP/M	8080	48K	850/50	
CIS COBOL Compact	3.46	3.46	CP/M	8080	32K		
FORMS 2 COBOL Form Generator	1.1,6	1.16	CP/M	8080		200/20	
CBS Applications Builder	1.2		CP/M	8080	48K	395/40	No support language needed
CREAM (Real Estate Acct'ng)	2.3		CP/M	8080	64K	250	CBASIC required
COBOL-80 Compiler	4.01	4.01	CP/M	8080	48K	700/25	
DATASTAR Information Manager	1.1		CP/M	8080		350/35	
Datebook	1.05		CP/M	8080	48K	295/25	Needs 80x24 terminal
DESPOOL Print Spooler	1.1A		CP/M	8080		80/10	
DISILOG Z80 Disassembler	4.0	4.0	CP/M	Z80		65/10	Zilog mnemonics
DISTEL Z80/8080 Disassembler	4.0		CP/M	8080/Z80		65/10	Intel mnemonics with TDL extensions
EDIT Text Editor	2.06		CP/M	Z80		129/25	
EDIT-80 Text Editor	2.0		CP/M	8080		89/15	
ESQ-1	1.0		CP/M	8080		1495/50	
ESQ-1 DEMO	1.0		CP/M	8080		75/50	
FILETRAN	1.20		CP/M		32K	99/20	1-way for TRS-80 Mod I, TRSDOS to Mod I CP/M
FILETRAN	1.30		CP/M		32K	149/20	2-way for TRS-80 Mod I, bwt. TRSDOS & Standard CP/M
FILETRAN		1.4	CP/M		32K	119/20	2-way for TRS-80 Mod I, bwt. TRSDOS & Mod I CP/M
FILETRAN	1.5		CP/M		32K	99/20	1-way for TRS-80 Mod II, TRSDOS to Mod II CP/M
Financial Modeling System	2.0		CP/M		48K	300	
FORTRAN-80 Compiler	3.4	3.36A	CP/M	8080	36K	425/25	
FORTRAN Package	3.38		TRSDOS			80/25	
General Ledger/Cybernetics	1.3C		CP/M	8080	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	530/40	Needs Microsoft BASIC 4.51
General Ledger/Structured Sys	1.4C		CP/M	8080	52K	820/40	No longer needs CBASIC
GLECTOR Accounting System	2.0		CP/M	8080	52K	350/25	Use w/CBASIC2 and Selector III
HDBS	1.04D		CP/M	+	52K	300/35	
HDBS	1.04D		TRSDOS		56K	300/35	Runs only under Radio Shack BASIC
HDBS.SRS	1.03		TRSDOS		56K	150/5	Runs only under Radio Shack BASIC
HDBS	1.04D		APPLE		56K	300/35	
IBM/CPM	1.1		CP/M	8080		175/5	
Inventory/Graham Dorian	1.0	1.0	CP/M	8080		555/40	Needs CBASIC2
Inventory/Peachtree	10-10-80		CP/M	8080	48K	660/40	Needs Microsoft BASIC 4.51
Inventory/Structured Sys	1.0C		CP/M	8080	52K	820/40	No longer needs CBASIC
Job Costing	2.02	2.02	CP/M	8080	48K	805/40	Needs CBASIC2
KBASIC Interpreter	2.03		CP/M	8080	48K	585/45	
KISS File Management System	2.03		CP/M	8080	40K	335/23	
LETTERIGHT Text Editor	1.1A		CP/M	8080	52K	200/25	
LEVEL 3 BASIC / G2			TRSDOS			45	Cassettes
LINKER			CP/M	Z80		69	
MACRO-80 Macro Assembler Package	3.40	3.40	CP/M	8080		149/15	
Magic Wand	1.0		CP/M	8080	32K	395/40	
MAGSAM III	4.1		CP/M	8080	48K	145/25	
MAGSAM IV	1.0		CP/M	8080	48K	295/25	
MAILING ADDRESS Mail List System	8-13-80		CP/M	8080	48K	530/30	
Mail-Merge	2.1		CP/M	8080		150/25	
Master Tax	1.0		CP/M	8080	48K	995/30	Now has 1980 tax forms
MDBS	1.04		CP/M	+	48K	900/35	
MDBS-DRS	1.02		CP/M	+	52K	300/35	
MDBS-QRS	1.0		CP/M	+	52K	300/35	
MDBS-RTL	1.0		CP/M	+	52K	300/35	
MDBS-PKG			CP/M	+	52K	1500/60	Includes all of the above MDBS products
MDBS	1.04D		Apple		56K	900/35	
Microspell	3.2	3.2	CP/M	8080	48K	249/20	
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	
Mu LISP-80 Compiler	2.03		CP/M	8080		200/15	
Mu SIMP / Mu MATH Package	2.03		CP/M	8080		250/20	Is MuMATH 80
NAD Mail List System	3.0C		CP/M	8080	49K	100/20	
Nevada COBOL	1.403	1.403	CP/M	8080	32K	149/25	

S=Standard Version
M=Modified Version
OS=Operating System
P=Processor
$=Price (where two prices are given, the second is the cost of the manual alone)

VERSION LIST

PRODUCT NAME	S	M	OS	P	MR	$	
PASM Assembler	1.02		CP/M	Z-80		129/25	
Pascal/M	3.2		CP/M	8080	56K	175/20	
PASCAL/MT Compiler	3.2		CP/M	8080	32K	250/30	
*PASCAL/MT+	5.1		CP/M	8080	52K	250/30	Also has 32K version
PASCAL/Z Compiler	3.2-1		CP/M	8080	56K	395/25	
Payroll/Cybernetics, Inc.			CP/M	8080		500	Needs RM/COBOL
Payroll/Peachtree	11-7-80		CP/M	8080	48K	530/40	Needs Microsoft BASIC 4.51
Payroll/Structured Sys	1.0C		CP/M	8080	60K	820/40	No longer needs CBASIC
PEARL I	1.01		CP/M	8080	48K	130	Needs CBASIC
PEARL II	1.01		CP/M	8080	48K	350	Needs CBASIC
PEARL III	3.01		CP/M	8080	56K	650	Needs CBASIC
PL/I-80	1.3		CP/M	8080	48K	500	
*PLINK Linking Loader	3.25		CP/M	Z-80		129/25	
*POSTMASTER Mail List System	3.3	3.3	CP/M	8080	48K	150/20	
Property Manager	10-10-80		CP/M	8080		925/40	Needs Microsoft BASIC 4.51
Property Mngemt. Sys.	1.0		CP/M	8080	48K	650	Requires CBASIC
QSORT Sort Program	1.5		CP/M	8080	48K	100/20	
Real Estate Acquisition Programs	2.1		CP/M	8080	56K	500	Requires CBASIC
Residential Prop. Mngemt. Sys.	1.0		CP/M	8080	48K	650	
RM/COBOL Compiler	1.3C		CP/M	8080	48K	750	Supplied with CP/M 2 by Cybernetics
RAID	4.7.3	4.7.3	CP/M	8080	28K	250/25	
RECLAIM Disk Verification Program	2.1		CP/M	8080		80/5	
*SBASIC	5.3h		CP/M	8080		295/35	
*SELECTOR-III-C2 Data Manager	3.23	3.23	CP/M	8080	48K	295/20	
SELECTOR-IV	1.0		CP/M	8080	52K	550/35	Requires CBASIC
SID Symbolic Debugger	1.4		CP/M	8080		105/15	Not available for Superbrain
SMAL/80 Programming System	3.0		CP/M	8080		75/15	Not for CP/M 2.x
STATPAK	1.2	1.2	CP/M	8080		495/30	Needs Microsoft BASIC-80 Vers. 5.2 or above
STRING BIT FORTRAN Routines	1.02	1.02	CP/M	8080		65/15	
STRING/80 bit FORTRAN Routines	1.16		CP/M	8080		95/20	
STRING/80 bit Source	1.16		CP/M	8080		295/NA	
SUPER SORT I Sort Package	1.5		CP/M	8080		225/25	
SUPER SORT II Sort Package	1.5		CP/M	8080		175/25	
SUPER SORT III Sort Package	1.5		CP/M	8080		125/25	
T/MAKER Data Calculator	1.4		CP/M	8080	48K	275/25	Requires CBASIC2
*TEX Text Formatter	2.0		CP/M	8080	36K	105/15	
*TEXTWRITER-III Text Formatter	3.6	3.5	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	1.3	CP/M	8080		95/25	To be used w/Microsoft BASIC-80 5.2
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	
WORDMASTER Text Editor	1.07		CP/M	8080	40K	145/25	
*WORDSTAR Word Processor	2.26	2.26	CP/M	8080	48K	445/40	
MAIL MERGE Printer Overlay	2.1		CP/M	8080	48K	575/40	
WORDSTAR Customization Notes	2.1		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	3.36		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	3.3	CP/M	Z80		95/20	
ZDT Z80 Debugger	1.41		CP/M	Z80		50/10	
ZSID Z80 Debugger	1.4		CP/M	Z80		130/15	

+ 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 Vers. 4.51, and BASIC-80 Vers. 5.xx.

* indicates a new version or new product.

Operating Systems

CP/M for Apple II with Microsoft BASIC	2.0
CP/M for Cromemco System 3 8"	1.4
CP/M 2 for Cromemco System 3 8"	2.2
CP/M 2 for Digital Microsystems	2.21
CP/M 2 for Durango F-85	2.21
CP/M for Heath H8 with HI7 Disk	1.43
CP/M for Heath / Zenith H89	1.43
CP/M 2 by Magnolia for Heath / Zenith H89	2.2
CP/M for ICOM 3812	1.41
CP/M for ICOM 3712 with Altair Console	1.41
CP/M for ICOM 3712 with IMSAI Console	1.41
CP/M for ICOM Microfloppy (# 2411)	1.41
CP/M 2 for ICOM 4511/Pertec D3000 Hard Disk	2.22
CP/M for Intel MDS Single Density	1.4
CP/M 2 for Intel MDS Single Density	2.2
CP/M 2 for Intel MDS 800/230 Double Density	2.2
CP/M for MITS Altair 3202 Disk	1.41
CP/M for Micropolis Mod I - All Consoles	1.411
CP/M for Micropolis Mod II - All Consoles	1.42
CP/M 2 for Micropolis Mod I	2.20
CP/M 2 for Micropolis Mod II	2.20
CP/M for Compal Micropolis Mod II	1.4
CP/M for Black Hawk Micropolis Mod II	1.4
CP/M 2 for Durango F-85	2.22
CP/M for Exidy Sorcerer Micropolis Mod I	1.42
CP/M for Exidy Sorcerer Micropolis Mod II	1.42
CP/M for NYLAC/REX Micropolis Mod II	1.4
CP/M for Vector MZ Micropolis Mod II	1.411
CP/M for Versatile 3B Micropolis Mod I	1.411
CP/M for Versatile 4 Micropolis Mod II	1.411
CP/M for Horizon North Star SD	1.41
CP/M 2 for Mostek MDX STD Bus	2.2
CP/M 2 for Ohio Scientific C3	2.22
CP/M for Sol North Star SD	1.41
CP/M for North Star SD IMSAI SIO Console	1.41
CP/M for North Star SD MITS SIO Console	1.41
CP/M for North Star DD	1.45
CP/M 2 for North Star DD/QD	2.21A
CP/M for Processor Technology Helios II	1.41
CP/M 2 (Cybernetics) for TRS-80 Mod II	2.25
CP/M for TRS-80 5 1/4" (Model I)(Lifeboat)	1.41
CP/M 2 for TRS-80 Model II (Lifeboat)	2.24A

Hard Disk Modules

Corvus Module	1.6
KONAN Phoenix Drive	1.7
*Micropolis Microdisk	1.8
Pertec	1.6

LIFELINES®

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

NEW YORK N. Y.

FIRST CLASS MAIL
U. S. POSTAGE
PAID 28¢
Permit No. 416

FIRST CLASS MAIL