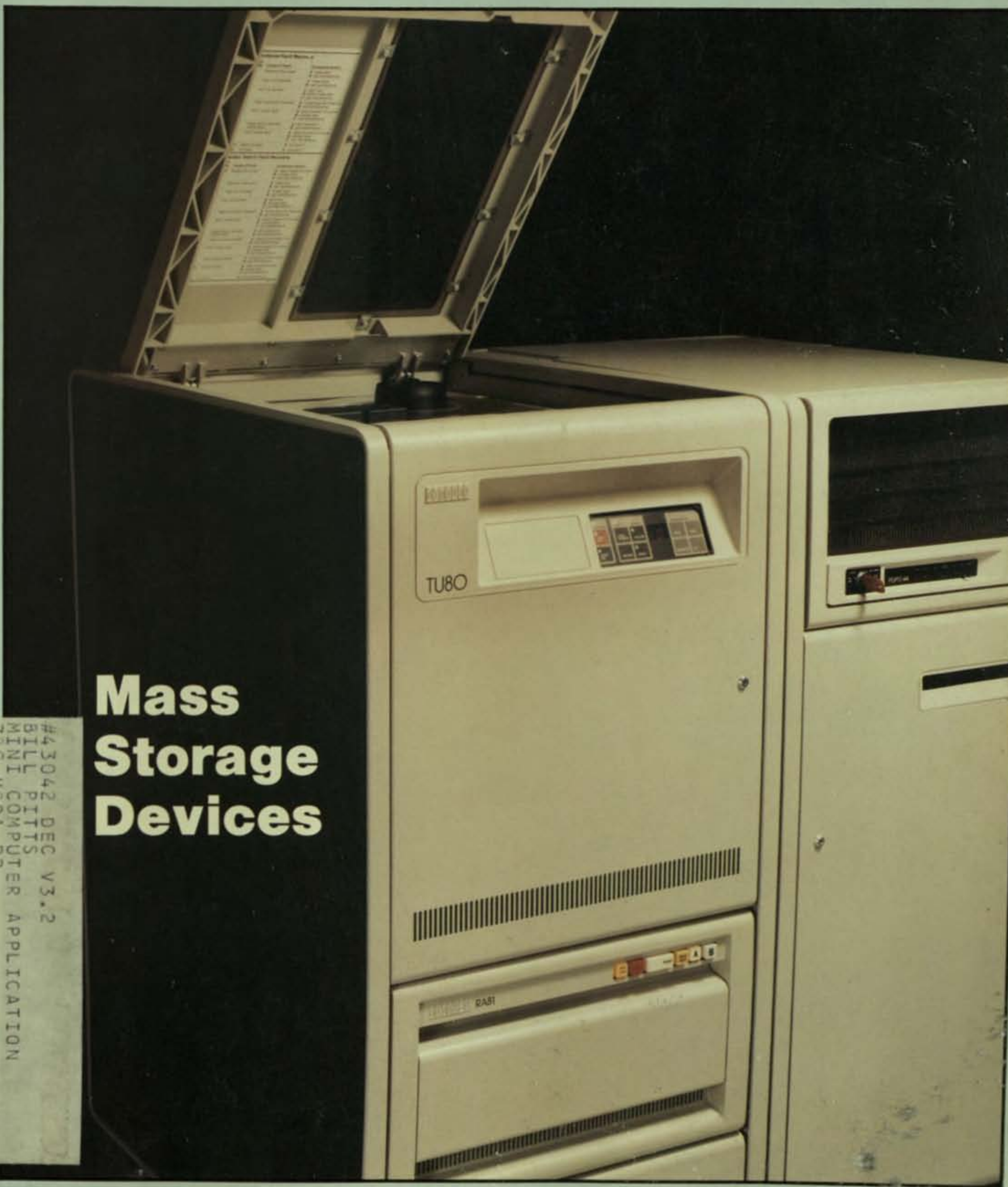
MARCH • 1984

THE DEC* PROFESSIONAL

• VOLUME 3, NUMBER 2 • THE MAGAZINE FOR DEC USERS



Mass Storage Devices

EMULEX TALKS DEC 10

BIG SURPRISES COME IN SMALL PACKAGES...

Response to the Emulex SABRE™ keeps getting better all the time. Originally intended for LSI-11 and LSI-11/23 PLUS, this "do-it-all" storage subsystem is now being welcomed with open arms by users of the just-introduced MICRO/PDP-11.

Why all the excitement? A quick glance at the specs tells all... DEC offers 10 megabytes on a 5¼-inch Winchester, the RD51. SABRE gives you over three times the capacity—31.2 megabytes on a 5¼-inch Winchester—and an access rate that's twice as fast. For backup, DEC offers the RX50 dual 400-kilobyte diskette, while SABRE has a unique 10.4-megabyte 8-inch cartridge drive. Not only is its transfer rate five times faster than the RX50, but it holds 20 times more data.

And how does SABRE stack up to streaming ¼-inch tape? It offers the advantages of a random access device. What's more, unlike floppies or tapes, the SABRE cartridge can double as primary storage whenever you are pressing the upper limit of your fixed-disk capacity. These benchmarks are discussed in detail in a special Emulex report. Write to us for your free copy.

FLEXIBLE FLYER...

The removable part of the SABRE is a flexible disk with Bernoulli heads. It gives you floppy-type economy, with Winchester-type capacities and transfer rates. That makes it perfect for program loading, data exchange, fixed-disk backup and data storage. Couple this to SABRE's other advantages and you have a system that's equivalent to a fully expanded, four-drive RL02 cartridge subsystem. In one-eighth the space. Requiring only one-fourth the power. With significantly increased flexibility.

ENHANCED CARTRIDGE, IMPROVED RELIABILITY...

Shock and vibration resistant, the standard SABRE cartridge is an advanced, lubricated version that has been continuously rewritten and read for over 500 hours without an error. (A special head-cleaning cartridge takes less than five minutes to restore the drive to its Winchester-level reliability specifications.) The cartridge costs only half the price of a DEC RL02 14-inch cartridge.

HAVE IT YOUR WAY...

SABRE is a compact desktop enclosure with its own integral power supply. This makes it an ideal add-on device for end users (dealers and distributors take note!). For OEMs there's the advantage of a single SCSI interface for both SABRE subsystem devices. And OEMs can purchase the SABRE in a variety of configurations, with or without drives. The SABRE bipolar, microprocessor-based host adapter—a single quad-wide board embedded in the QBus backplane—can, in fact, support up to five additional SCSI-compatible peripherals on the same bus. Completely transparent to all the standard LSI-11 operating systems, including RT11, RSX11M and RSTS/E, and nearly all standard UNIX device drivers.

FROM THE EMULEX FILE...

Onward and upward. Emulex revenues for the second quarter of fiscal 1984 were over $15.5 million, more than double those of a year ago. Net earnings were up 154%. Two new buildings have been added to our manufacturing facilities in Puerto Rico, a new sales-and-service headquarters has been opened in England, and a site has been selected in Ireland for a new European manufacturing operation.

EMULEX

3545 Harbor Blvd., P.O. Box 6725
Costa Mesa, California 92626
Toll-free (800) 854-7112, in Calif. (714) 662-5600.

GSA Contract #: GS00K840155575

CIRCLE D107 ON READER CARD

Why should you care that we became the big name in the little end of the concentrator business?

One reason you might care is that we got there by building a family of data concentrators which saved you money and solved your data communications problems:

Micro800/2 Data Concentrators

The world's most popular line of data concentrators. Specifically designed for the user of minicomputers and "dumb" asynchronous terminals, they can pay for themselves in a few months by supporting many remote terminals on one telephone line, while also providing Automatic Retransmission on Error, Satellite Capability, Synchronous Channel Support, a Command Port, and much more . . .

Micro900/2 Multidrop Concentrators

Bringing the benefits of MICOM's data concentration to users whose terminals are widely scattered, so that "dumb" terminals in up to 16 different locations can share a *single* telephone line.

Micro800/2HP Data Concentrators

Specially designed to handle the unique requirements of HP 3000 systems employing HP's ENQ/ACK protocol.

Another reason you might care is that now we can solve your next data communications problems too, with new family members such as:

Micro860 Concentrator Switches

Brand new kinds of products which bring add-on switching, contention, queueing, and centralized management to networks of up to eight data concentrators.

Micro800/X.25 Concentrator PADs

Products which combine the benefits of Micro800/2 Data Concentrators with CCITT X.25-compatible packet assembly to allow asynchronous terminals and computer ports to access public or private Packet Data Networks easily and inexpensively.

And still another reason is that concentrators are only one family of MICOM products. Now we can be the only source you need for minicomputer data communications products from modems to data PABXs to local networks. Thanks to you, we're big in those fields too!

MICOM® MicroComputers for DataCommunications™

MICOM SYSTEMS, Inc. • 20151 Nordhoff Street • Chatsworth, CA 91311 • Telephone (213) 998-8844 • TWX 910/494-4910
Regional Sales/Service • Atlanta, GA • (404) 435-2999 • Boston, MA • (617) 527-4010 • Chicago, IL • (312) 789-2430
Dallas, TX • (214) 258-0774 • St. Louis, MO • (314) 576-7626 • Teaneck, NJ • (201) 836-4000
MICOM-BORER Ltd. • Bel Court • 15 Cradock Road • Reading, Berkshire RG20JT, England • (0734) 866801 • Telex 847135

CIRCLE D111 ON READER CARD

FROM THE PUBLISHERS . . .

WHY CAN'T DEC DO IT?

Carl B. Marbach

While it is easier to spot DEC's failures than successes lately, the failure to deliver quantities of the RA-81 and RA-60 disk drives has particularly affected users and deserves comment.

It has been more than a year since DEC announced the "new generation" in mass storage for DEC computers. Shipments of the RA-81 are far behind and the RA-60 has barely been seen by users. In the meantime Fujitsu (and others) has been producing large numbers of 400 + MB disk drives that are very fast, inexpensive and impressively reliable. While it may not be statistically correct, our neighborhood has had six of these large drives installed on large PDP-11s for almost a year and nobody has ever seen one fail. Even field service knows a good deal when they see one; they are servicing one.

The first quarter fall in earnings reported by DEC shook the financial world but it was no surprise to all those customers waiting for these disk drives. There were many computers that remained unshipped because there weren't any disks to put on them. The RA-81 problems may have been solved, but the RA-60 is still hard to find. This 200 MB removable disk should have replaced the aging RP06. Instead, the RP06 is in great demand (used prices have skyrocketed). All of this proves that a mistake can cost even a big company a lot.

A lot of us have grown used to a DEC with great products at a fair price. We have developed real feelings for their computers. What other computer company has ever had a ten year birthday party for not one, but two (RSTS and RSX) operating systems? When the godfather falters, we all feel the repercussions. It' s time to get the ball rolling and get those disk drives out the door.

When DEC needed more and more terminals, they abandoned the teletypes that got them started and developed the LA-30. It was a good try but it didn't quite make it so they followed it with the LA-36, a classic in its own time. I would still rather have an unbreakable LA-36 than some of the newer models. When CRTs became a necessity, DEC built the VT-05. It was pretty but had some shortcommings and was followed by the VT-50 . . . almost right . . . Then came the venerable VT-52. Its successor was the VT-100 series, and we now have the VT-220 which is the nicest terminal we have ever seen. DEC has the talent to do the same thing with disk drives that it did with terminals.

George Patton used to say "Lead, follow, or get the hell out of my way!" If the guy running the show on the RA-81s and RA-60s can't do the job, find someone who can. You need it and we deserve it.

NUMBER TWO

Dave Mallery

As this is written, DEC is still number two. If the DATAMATION survey comes out before AT&T announces its first computer product, I guess there is still a chance they'll stay number two.

With IBM growing by another DEC every few months, we should simply rank all the companies with sales under 30 billion and forget number one.

Since the opportunity to become number one is about as remote as the nearest galaxy, a more fruitful topic might be how to stay number two.

We are promised a new VAX in the very near future. It is supposed to have 30 percent more speed than a 780. One of the funnier rumors has been that this is simply a 750 with the NOPs removed from the firmware. Could be . . . time will tell. Mr. Olsen was quoted in the papers as having said that the MicroVAX II (for late '84) will have almost the power of the 780 on a single chip. Now that will be one fine personal computer. The super VAX is now slated for mid '85 and will clock in at 5 mips. That will be one fine main frame.

One way to continue your tenure as number two is to deliver faster VAXs as soon as possible.

Rumor also has it that the RA60s are shipping. Should that be the case, this will be a great quarter. The backlog for RA60s is the only thing inside DEC that is growing as fast as IBM.

If you have not yet tried one, try a VT220. One sure way to stay in position is to have the finest terminal ever built. They do.

I also have been raving about DECtalk. This little box can be best described as an "audio printer" that can answer the phone. It can decode your touch tone input and send it to your machine. The ASCII coming back from the machine is "spoken" by DECtalk into the phone line. Call 617-493-8255 for a demo.

It also looks like the 11/73 logjam is about to break. I make this statement based solely on the fact that a neighbor OEM just got one. We are still looking for the quad 11/23 + replacement board announcement. Here is a product that a marketer dreams about. It obsoletes the entire installed base (fortunately purchased, not leased)! Hopefully later this year, DEC will announce an 11/73 based system in a box with an order number.

So, number two is probably a tenable position. DEC has the depth to sustain the magic numbers that Wall Street demands. Forget Big Brother. 1984 has lots to offer for number two. ■

THE DEC* PROFESSIONAL
THE MAGAZINE FOR DEC USERS

*THIS MAGAZINE IS NOT SPONSORED OR APPROVED BY OR CONNECTED IN ANY WAY WITH Digital Equipment Corporation. "DEC" IS A REGISTERED TRADEMARK OF Digital Equipment Corporation. Digital Equipment Corporation is the owner of the trademark "DEC" and is the source of all "DEC" products.

Publishers: R.D. Mallery, Carl B. Marbach

Vice President Finance & Operations: Peg Leiby

Vice President of Advertising: Helen B. Marbach

Managing Editor: James L. Trichon

Assistant Editor: Linda DiBiasio

Director of Advertising & Circulation: Pauline Noakes

Advertising Sales Manager: Hope Makransky

Advertising Sales: Connie Mahon, Eileen P. Weiss

Directors of Production: Martin Grossman, Peg Traina Grossman

Production Assistant: Darwin Au

Staff Assistants: Kathi B. Campione, Harriet S. Coren, Claire Hollister, Hope Miller, Margie F. Pitrone, Donna L. Schmidt, Anne Schrauger

Contributing Editors: Arthur Edward Groulx, Steve Holden, Richard Lowe

Contributors: Josef S. Bleier, Joe Celko, Albert B. Chu, Robert F. Curley, David E. Ellis, Louis J. Finnegan, Rex Jaeschke, Robin C. Johnson, R.M. Kaplan, Jerome A. Martin, Steve Roy, Rick Scherle, Richard Scott, Ron Trellue, Ed Vaccaro, Allen A. Watson

Cartoons: Frank Baginski, Emily Dahlstrom, Johns

Colorado Representative: Jeff Yusem, P.O. Box 8013, Aspen, CO 81612, 303/920-2180

Mid-Atlantic Representative: Ed Shaud, P.O. Box 187, Radnor, Pa. 19087, 215/688-7233

Printing & Binding: Mack Printing

Cover Photo: David Sheppard

Circulation: 80,000 DEC users, worldwide

Editorial Information: We will consider for publication, all submitted manuscripts and photographs, and welcome your articles, photographs and suggestions. All material will be treated with care, although we cannot be responsible for loss or damage. (Any payment for use of material will be made only upon publication.)

THE DEC PROFESSIONAL Magazine, March 1, 1984, Vol. 3, No. 2. ISSN 0744-9216 is published bi-monthly by Professional Press, Inc., 161 E. Hunting Park Avenue, Philadelphia, PA 19124. Subscriptions: no charge for qualified U.S. sites; foreign - $50 (US$) per year. Second Class postage paid at Philadelphia, PA, and additional mailing offices. POSTMASTER: Send all correspondence and address changes to: THE DEC PROFESSIONAL, P.O. Box 362, Ambler, PA 19002-0362, telephone (215) 542-7008.

LETTERS TO THE DEC PRO . . .

Send letters, comments, photos, etc. to: Letters to the DEC PRO, Box 362, Ambler, PA 19002.

I'm responding to Steve Holden's call for opinions about UNIX. Here's mine:

We, at NIKUV, are at the final stages of developing our new and (hopefully) revolutionary set of application programming tools, which are aimed at the commercial applications development market.

When, about a year ago, my ideas were transformed into a business plan/product specs, all signed and approved by my board of directors, I still had a small problem to resolve. I knew the machine: VAX; but what operating system? UNIX, or VMS, that was the question. (Oh great Shakespeare, forgive thy humble servant . . .)

That wasn't an easy one. UNIX has its great advantages: portability, programmer-friendly environment, and so on . . .

But, there are always some "buts":

1. UNIX does not feature ISAM files (or, for that matter, any other file system; I call stream "files" — strings on disk).

2. UNIX means using "C," and maybe PASCAL.

The decision, of course, was VMS; it is unthinkable to write commercial applications without good old ISAM (or a DBMS — if you can afford it). I could always write my own file access system (a thing I have already done for RSTS), but it takes time and is a tedious job; I could try and use somebody else's package, which means to be dependent on him for support, portability in the future, and so on. Besides, the first amendment in any programmer's book says: "Never Shalt Thou Trust Thy Neighbor's Program . . ."

The second "but" is a little more sensitive for all you UNIX/C cultists. "C" is not suitable for commercial applications programming. Our set of tools does require some programming from the programmer, unlike other "code-free" miracle workers in the market who do-it-all for you, coffee at coffee breaks included. As I found out, if I was going to use UNIX, I had better write in "C," or at the most PASCAL. COBOL or BASIC seemed to be out of the question, at least when considering compatibility and portability.

By the way, I think that any business applications programmer is bound to fall in love with DEC's BASIC V2 on VMS. It has the simplicity of BASIC, the structures of PASCAL and PL/1, the data-types and data structures of COBOL and PL/1; all this without the complexity of PL/1, the terribly overworded, almost poetic sentences of COBOL and lack of decent files/data handling in PASCAL. For me, it was love at first sight.

To make it short: I really like UNIX and all those UNIX tools, and I enjoyed writing in "C"; but I wish UNIX were more standardized, supported by DEC, supplied with a standard file access method (maybe even RMS compatible?), and most of all, that DEC would adapt its BASIC V2 to UNIX.

That's it. I hope nobody's feelings got hurt . . .

Eli Meirovitz
Software Manager
NIKUV Computers (1976) Ltd.
Haifa, Israel

Our UNIX Editor replies:

Eli seems to have made his decision for sound commerical reasons, and I hope that my last column made it clear that I am not in favor of using UNIX just because it's a wonderful system.

I would endorse everything he says, but add a rider: if he had chosen to develop his programming tools under UNIX rather than VMS then the UNIX world would have become the richer for it, and the day would have drawn nearer when it was possible to buy a UNIX system with confidence that all necessary software would be available.

I hope that the VMS tools are such a success that a port to UNIX becomes economically feasible. The more people get involved in the UNIX market both as well-informed buyers and well-informed suppliers, the faster that market will develop and the sooner UNIX will be able to stand on its own two feet.

* * *

I am writing regarding your constant complaints of sites selected for DECUS symposia.

I am somewhat new to DECUS. I have only been a member for a little more than a year, and have only attended two symposia meetings to date. However, I fail to see any constructive purpose for your magazine's constant "slams" of the sites chosen for DECUS symposia.

I come to DECUS, as I am sure the majority do, to learn as much as I possibly can and to interact with other DECUS members. I don't come for the scenery, the sun, the fun, etc. Those things are nice extras when visiting another city, but they shouldn't be the sole factor determining DECUS locations. Facilities, lodging, and transportation should be the most important factor for selecting a site.

At the 1983 Fall DECUS in Las Vegas, we were all spending a week in one of the "fun spots" of America. I spent my week attending sessions, even until 10 or 11 PM on Monday and Thursday evenings. I don't feel I missed anything. I gained a lot by spending my time in those sessions. If I had wanted to have fun that week, I would have brought my swimming trunks and towel, and a lot of cash and spent all my time having "fun." When I left Las Vegas all I would have had was a bad hangover, no money, and, hopefully, a few memories. As it was I left with a head full of knowledge, a great deal of satisfaction, and new found confidence.

This is what DECUS is all about and I would like to see worthwhile, constructive criticism from DEC PRO instead of just carte blanche criticism. I enjoy reading your magazine; it's informative and very thorough in its coverage of the wide variety of topics it deals with. I enjoy it very much, but

. . . continued on page 55

CHOOSING A DATA BACKUP METHOD

By Jerome A. Martin, President, First Computer Corporation, Westmont, Illinois

Thousands of businesses have operated for many years without a computer system, but it is almost impossible for any company to exist without some form of records. While no one questions the need for a backup system, many of the factors that should be considered in the selection of the system are often given only brief attention or accepted on only the manufacturer's recommendation.

Reliability, capacity, media, performance, physical size, and cost should all be considered. Of course the importance of the individual criteria will vary with each organization.

As a systems integrator for many types and sizes of businesses, the versatility of a backup system is especially important to us. We require a backup method for our systems and subsystems that will meet our customers' immediate and future needs yet provide an attractive price/performance ratio.

The exploding applications of large Winchester disk drives have mandated our use of magnetic tape backup. While streaming tape drives can provide highly cost effective solutions to the large disk backup problem, most streaming tape drives are not fully utilized and instead use the standard start/stop software drivers available in most operating systems to avoid the complexities of implementing expensive customized streaming software for the drives.

A comparison of streaming and start/stop drives is illustrated in Figure 1.

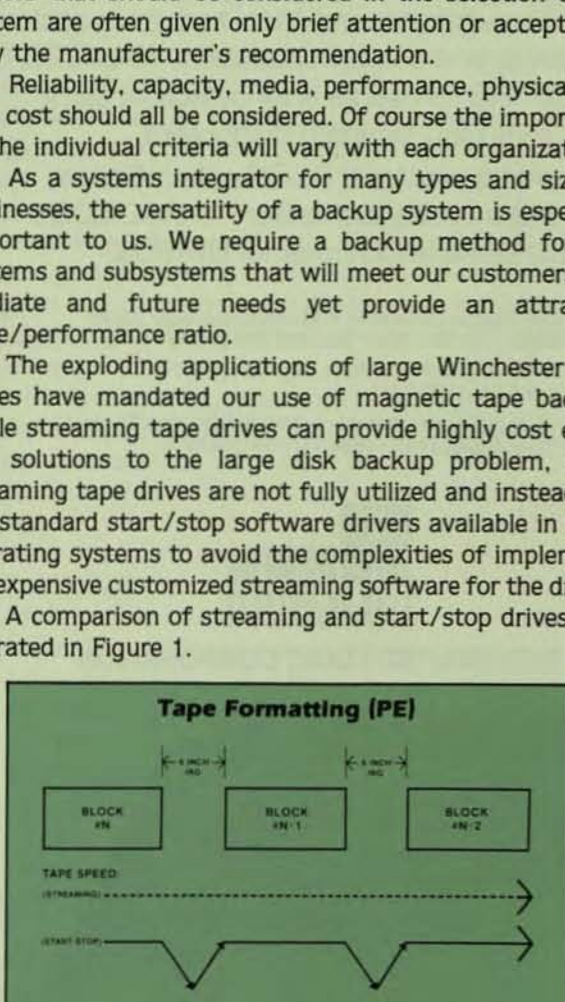


FIGURE 1

The solid line beneath the data blocks illustrates the tape motion for a start/stop drive; the dashed line shows it for a streaming drive. The space between the data blocks shows the Inter-Record Gap (IRG) of 0.6-inch required by ANSI and IBM standards. The length of the data block varies depending on the number of characters written for each block.

If the tape drive is stopped between each data block, it must slow down to a stop and then speed back up to operating speed before the next data block is written. This is the ramp time, and it must be accomplished within the IRG. If the tape drive is not stopped between data blocks — the streaming tape drive — the IRGs must be inserted "on the fly." During this time, the tape drive is not available to the system for reading or writing data.

To establish the price/performance ratios, other differences need to be identified and performance established.

START/STOP TAPE DRIVE PERFORMANCE FACTORS

Four major factors contribute to the performance of the start/stop drives: (See Figure 2)

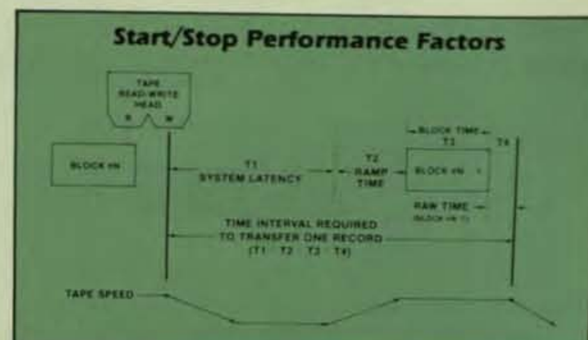


FIGURE 2

System Latency. This is the time it takes for the host system to issue a new command to the tape drive following the completion of a previous command.

Ramp Time. This is the time needed to accelerate the physical tape to operating speed before data can be transferred to — or read from — the tape.

Block Time. This is the time required to transfer the characters in a data block of a given size.

Read-After-Write (RAW) Time. The transfer of a data block is not complete until the data is read back to insure that it was written correctly. A RAW time results because the read head is physically located a short distance from the write head.

The following formulas are used to compute each element of time in a data transfer influenced by the tape drive:

$$\text{Ramp Time} = \frac{\text{Start or Stop Ramp Distance}}{\text{1/2 Tape Speed}}$$

$$\text{Block Time} = \frac{\text{(Bytes per Block/Recording Density)}}{\text{Tape Speed}}$$

$$\text{RAW Time} = \text{(IRG Length/Tape Speed)} - \text{Ramp Time}$$

EXAMPLE. Calculations for a 45ips, 1600bpi start/stop drive writing in 2048 byte blocks are:

$$\text{Ramp Time} = \frac{0.1875"}{22.5\text{ips}} = 8.3 \text{ milliseconds}$$

$$\text{Block Time} = \frac{2048/1600}{45\text{ips}} = \frac{1.28"}{45\text{ips}} = 28.4 \text{ milliseconds}$$

$$\text{RAW Time} = (0.6"/45\text{ips}) - 8.3\text{ms} = 13.3\text{ms} - 8.3\text{ms} = 5 \text{ milliseconds}$$

To find system latency, the average transfer rate of the present system must first be determined where:

$$\text{Average Transfer Rate} = \frac{\text{Total Data Transferred}}{\text{Total Time to Transfer Data}}$$

$$\text{System Latency} = \frac{\text{Block Size} - (\text{Ramp} + \text{Block} + \text{RAW})}{\text{Average Transfer Rate}}$$

EXAMPLE. For a system which can write 10 megabytes of data to tape in 5 minutes and 3 seconds, writing 2048 byte blocks at a tape drive speed of 45ips:

$$\text{Average Transfer Rate} = \frac{10{,}000{,}000 \text{ bytes}}{303 \text{ Seconds}} = 33 \text{ Kilobytes/second}$$

$$\text{Latency} = \frac{2048 - (8.3\text{ms} + 28.4\text{ms} + 5\text{ms})}{33{,}000 \text{ bytes/second}} = 62\text{ms} - 41.7\text{ms}$$

$$= \text{Approximately } 20\text{ms}^*$$

*Typical for file structured backup operations in a disk to tape dump.

Total transfer time for any start/stop tape drive can now be figured by first figuring the average transfer rate for each data block where:

$$\text{Average Transfer Rate} = \frac{\text{Block Size}}{(\text{Latency} + \text{Ramp} + \text{Block} + \text{RAW})}$$

$$\text{Total Transfer Time} = \frac{\text{Total Megabytes Transferred}}{\text{Average Transfer Rate}}$$

EXAMPLE. To transfer 31 megabytes of data using 75ips vacuum column, start/stop tape drive, 1600 bpi reel using 2048 data blocks and having a 20ms system latency:

$$\text{Average Transfer Rate} = \frac{\text{2048 Block Size}}{(20\text{ms} + 5\text{ms} + 17.1\text{ms} + 3\text{ms}}$$

$$= \frac{2048}{45.1} = 45.4\text{kbs}$$

$$\text{Total Transfer Time} = \frac{31{,}000{,}000 \text{ bytes}}{45{,}400 \text{ bytes/sec.}} = 682.81 \text{ seconds}$$

$$= 11.4 \text{ minutes (approximately)}$$

Using the same system parameters, total transfer times for:

45ips Tension Arm Drive = 15.6 minutes

125ips Vacuum Column Drive = 8.9 minutes.

STREAMING TAPE DRIVE PERFORMANCE FACTORS

Introduced to provide cost-effective backup, streaming tape drives require special software to provide commands to the drive within the reinstruct window to take full advantage of the higher average transfer rates. (See Figure 3)

Streaming tape drives usually operate at 25 or 100ips. The reinstruct time at 25ips is about 11ms. At 100ips, the reinstruct time is approximately 3ms.

To maintain streaming, the system must send the next command within the reinstruct window. If the command is not received at this time, the tape will overshoot the point at which it must write the next block. The tape must then be reversed and repositioned. (See Figure 3)

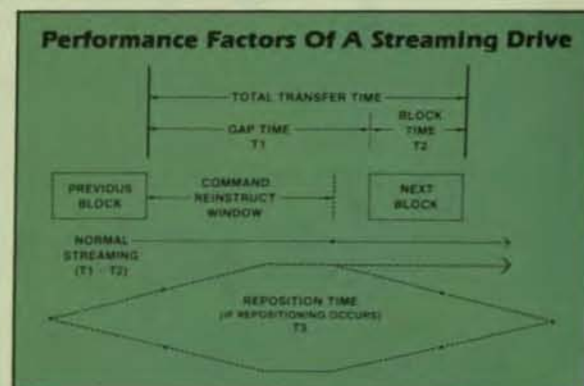


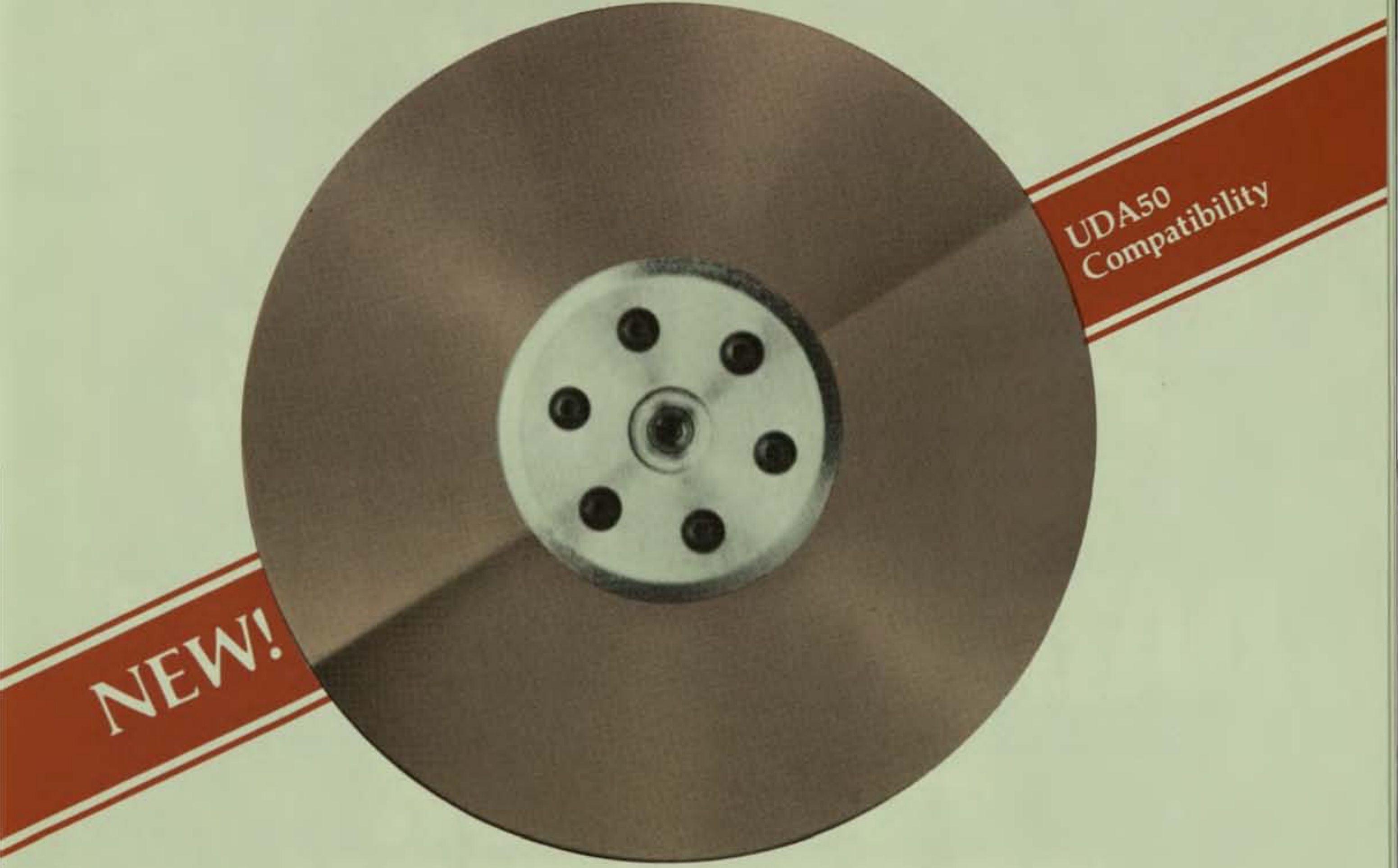
FIGURE 3

Using the same parameters as for the start/stop examples, the 20ms system latency exceeds the 3ms reinstruct time needed to sustain 100ips streaming causing the tape to be repositioned each time a block was transferred. At the 100ips speed, the reposition time is about one second. To transfer 31 megabytes, there would be 15,000 repositions — each one lasting one second which would add more than four hours to the job!

When the streamer is operated at 25ips, the reinstruct time is approximately 11ms. Because the 20ms system latency is greater than the reinstruct time, the streamer must still reposition for each block transferred. But at 25ips, the reposition cycle is much shorter because the overshoot is reduced with the lower speed.

With a 120ms reposition time at 25ips, the total time required to transfer 31 megabytes of data is reduced to 49 minutes. This is still about three times longer than it would take to transfer the data using a 45ips start/stop drive on the same system. But because the cost of the 25ips streamer can be about one-half the cost of the start/stop drive, for applications more economy-oriented than performance oriented, the 25ips streamer can be an attractive option.

DATARAM. Your disk drive connection.



It's easy to interface your disk drive to a DEC computer. When you have connections.

Dataram provides connections to your host LSI-11, PDP-11, or VAX minicomputers for the full range of disk drives — from 5¼" Winchesters to Fujitsu's 1.8 MB/sec. Eagle. Emulations that go all the way from RL02 up to DEC's new UDA50. Cartridge drives, SMD drives, fixed and removable Winchesters. We connect with them all.

Our new single-board UDA50-compatible controller, the S35, is especially exciting. UDA50 compatibility allows you to interface any SMD drive (up to 1.8 MB/sec.) to the UNIBUS of any PDP-11 or VAX minicomputer. Sizing is done automatically and 100% of the disk is utilized —there is no lost capacity.

Dataram's available emulations are listed below. For more details on any of our disk controllers, call (609) 799-0071. We'll help you make the connection you need.

LSI-11	PDP-11	VAX
RK05 RL02 RM02 RM05 RK06 RK07	RK05 RM02 RM05 RK06 RK07 UDA50	UDA50

DATARAM

Dataram Corporation □ Princeton Road □ Cranbury, New Jersey 08512 □ (609) 799-0071 □ TWX: 510-685-2542

CIRCLE D116 ON READER CARD

AT LAST, A HIGH PERFORMANCE, RELATIONAL DATA BASE SYSTEM FOR YOUR PRODUCTION-ORIENTED VAX ENVIRONMENT.

You've come to expect high levels of performance from your DEC™ VAX™ hardware. And you should expect no less from your data base software.

That's why Cincom introduced the ULTRA INTERACTIVE DATA BASE SYSTEM™—a next generation relational technology that combines high performance with practical, ease of use implementation.

New Generation Architecture

The secret behind ULTRA's interactive high performance is its unique architecture. An architecture built on a solid foundation of the very latest in relational subject data base technology. In-line control and security are achieved through the ULTRA In-Line Directory while relational implementation is achieved through the unique ULTRA Logical User View and the new Data Base Management facility. Combined, these components provide interactive synergism between users, programmers and the data they wish to access.

For multiple-machine applications that can range from scientific to high volume transaction oriented business systems, ULTRA provides VAX users the control and consistency needed to build high performance, low overhead, high quality systems that can be readily used by both programmers and by the burgeoning number of end-users. ULTRA also facilitates the latest in subject data base strategies for optimum data storage, retrieval and data manipulation.

Integrated Components

The integrated components which provide this unparalleled power include:

The ULTRA Logical User View. Modern, relational technology to insulate users and application programs from data structures and navigation requirements.

- **The ULTRA In-Line Directory.** Centralized control during program execution. Integration. Security.
- **The ULTRA DBMS.** High performance. Powerful. Faultless non-stop processing. Supports normalized data structures.
- **The ULTRA Intelligent Query and Reporter.** Direct end-user access. Flexible and easy to use.
- **The ULTRA Data Manipulation Language.** So powerful even the most complex processing can be handled using four commands: GET, INSERT, UPDATE, DELETE.

The Ultimate In Interactive Data Base Software For The VAX Environment

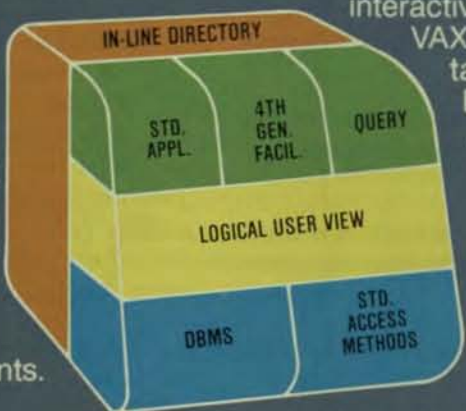
Like no other data base system available today ULTRA enables you to achieve the full potential of your VAX system. As your company evolves, ULTRA keeps right in step. And ULTRA is completely compatible with TIS™, our next generation data base system for the IBM mainframe.

So now you can implement Cincom's "IBM-DEC Connection" to link your strategic and tactical applications. In fact, the compatibility is so close, applications built on your IBM can be modified to run on VAX, or vice-versa. An ULTRA/TIS combination is the perfect IBM-DEC Connection.

For an in-depth brochure on the ultimate in interactive data base software for the VAX environment, simply contact our Marketing Services Department, 2300 Montana Ave., Cincinnati, OH 45211, or call us direct.

800-543-3010

(In Ohio: 513-661-6000)
(In Canada: 416-279-4220)



ULTRA, the architecture of the next generation of interactive data base technology for the VAX environment.

Cincom Systems

The alternative to longer transfer times requires that the system software is modified to reduce latency and eliminate any repositioning. The formula for the average transfer rate of a streaming drive shows:

$$\text{Average Transfer Rate} = \frac{\text{Block Size}}{\text{Gap Time} + \text{Block Time} + \text{Reposition Time}}$$

Eliminating the reposition time from the formula reduces the total transfer time to 19 minutes at 25ips and 4.8 minutes at 100ips. However, the system software modifications required to reduce the system latency usually require that the system be totally dedicated to the tape drive during the data transfer. Therefore, streaming is not ideally suited for multi-user systems which demand that users be given highest system priority.

AN ALTERNATIVE TO START/STOP AND STREAMING

The alternative to either start/stop or streaming drives we discovered is CacheTape from Cipher Data Products, Inc. CacheTape provides a streaming 1/2-inch tape drive at 100ips plus an interface which simulates a start/stop drive. The cache is a 64k random access semiconductor memory maintained under Z8002 microprocessor and DMA control cache memory has independent pointers for the logical and physical transports. The logical port is connected to the system interface, the physical port to the read/write logic of the physical transport. It is programmed to emulate the command protocol and data access times of start/stop drives.

CacheTape is software transparent with standard start/stop operating and utilities software. The 64k cache memory functions as an Electronic Capstan eliminating the need for mechanical tape buffering systems used in the past.

Start/stop performance is improved because:

1. The ramp time for CacheTape is only a short electronic delay.

2. The write time is less for CacheTape because of the high burst transfer rate to RAM memory instead of the tape speed limited transfer.

3. Data transfer is complete when the block is written into cache. This means that the RAW delay does not enter the performance formulas.

On other drives, errors encountered during a write operation result in the tape being automatically backspaced, erased, and the record being re-written on a fault-free area of tape. The CacheTape corrects all single track read errors "on the fly" and automatically backspaces and retries multi-track read errors. (This error recovery is also totally transparent to the host system.)

To figure performance comparisons for CacheTape:

$$\text{Average Transfer Rate to Cache} = \frac{\text{Block Size}}{(\text{Latency} + \text{Access Time} + \text{Block Time})}$$

where:

$$\text{Block Time} = \frac{\text{Block Size}}{\text{Cache Burst Transfer Rate}}$$

Cipher Data Products offers two models of CacheTape. The Model M890 has a Burst Transfer Rate of 120kbs and Cache Access Time of 5.0ms; Model M891 has a Burst Transfer Rate of 384 kbs and Cache Access Time of 0.5ms.

Using the same system parameters as given in the examples for the start/stop drives, figures show that CacheTape can be 62 percent faster than the 75ips drives and 26 percent quicker than the 125ips vacuum column drives. In addition, both CacheTape models have an optional high density 3200bpi mode to provide 92mb of unformatted storage capacity. With the 1600bpi density, interchange in industry standard ANSI 1600bpi is still possible, while large backup jobs can be done at 3200bpi.

The high performance interface characteristics and timing of start/stop through the logical CacheTape emulator, combined with the low cost tape streaming mechanics, give CacheTape very attractive price/performance ratios. Typically CacheTape will cost about 27 to 52 percent less than drives with comparable performance. And the model M891 with 3200bpi requires less time for backup than the new 50ips GCR (6250bpi) transports at about one-half the cost.

As one of the first companies to employ this new technology in our fully integrated mass storage subsystems, reliability was an important consideration. We found that the Electronic Capstan technology and rigid metal substructure prevent common mechanical failures. These and other design features give CacheTape a 5,500 hour Mean Time Between Failure rating. In addition, the CacheTape models are equipped with self-test diagnostics which can be accessed by keying in a special code on the front panel.

CacheTape also fares well in the other criteria we consider as we assemble our systems. The compact 8¾"H X 17"W X 24½"D size and front-loading allow us to mount the units in racks or cabinets. Because it is quiet and requires no computer room cooling, combined with the patented automatic loading and threading features which greatly reduce the training required for the end-users, CacheTape is ideal for the fast-growing "office automation" computer applications.

In the immediate future, we expect CacheTape technology to further improve the new high performance 6250bpi GCR drives. Later, we expect this technology to permit transparent stand-alone intelligent mass storage subsystems creating backup/restore operations and data base management functions which will be totally independent of the host system.

CacheTape and Electronic Capstan are trademarks of Cipher Data Products, Inc.
Figures, formulas, and examples taken from CacheTape Performance Report by Cipher Data Products, Inc.

EXPERIENCING RT-11 VERSION 5.1

By Arthur Edward Groulx, Logicaid Limited, Nepean, Ontario, Canada

Although this article was written in January, by the time it is published in March, RT-11 Version 5.1 will have been released for at least a couple of weeks. Like most new releases of RT-11, this version is immediately installable on your version 5 (or 4 or 3 or 2 or 1) RT-11 system. It is immediately upward compatible with previous RT-11 versions. Besides some maintenance fixes, there is support for new hardware and new software utilities and monitor features. Each of these topics will be dealt with separately.

MAINTENANCE FIXES/ ENHANCEMENTS

As far as we can tell, BINCOM, DIR, FORMAT, KED and SLP are the only RT-11 utilities to escape maintenance fixes or enhancements. This can be determined by doing a binary compare of version 5 versus version 5.1 .SAV files. (However, by the time of release, FORMAT might not escape the software surgeon's knife to support DZ/DW verification formatting.)

Most of the changes fix relatively minor problems in the version 5 software. For example, with RT-11 version 5 PIP, copying a single named file with the SETDATE option, viz.

.COP/SET DL0:NAMED.FIL DL3:

caused half of the largest space on DL3 to be allocated, instead of the size of NAMED.FIL, resulting in some incorrect error situations. IND.SAV has been modified to support greater programmer control over the use of CONTROL-Zs, type-ahead, and TIMEOUT for monitors without timer support. The single-line command editor, SL.SYS has been modified to allow the use of the second-last command entered (through the use of GOLD-uparrow), and not only the last command or one that may have been saved using the GOLD-downarrow key. This is especially useful in situations where you have typed-ahead a new command before the previous command finished. If the previous command resulted in an error, you can now retrieve it with GOLD-uparrow and make the repairs.

NEW HARDWARE SUPPORT

The major development in RT-11 version 5.1 is support for the Professional 325 and 350 computer systems. This development is so important that it will be dealt with separately in section D below. Along with support of this hardware, RT-11 will also run on the Falcon (SBC-11/21 and SBC 11/21 PLUS) processors, and the PDP-11/73. Previously, only the RT-11 foreground/background monitor ran on the SBC-11/21. The 11/73 is the LSI-11 series model which uses the J-11 microprocessor chip.

Only two "new" disk types are supported with RT-11 version 5. The RX50 diskettes on the Professional 300 series computer system, referred to using the mnemonic DZ, and the RD50/RD51 Winchester disk on the Professional 350, referred to using the mnemonic DW.

NEW SOFTWARE

RT-11 version 5.1 contains new software which both responds to some very old outstanding user wish list items, as well as considerably enhancing the superior friendliness and ease-of-use of the RT-11 operating system. Each particular new piece of software will be described separately.

1. VTCOM AND TRANSF

VTCOM is a utility which greatly increases the power of your RT-11 system by permitting your terminal to communicate with both your current RT-11 system, and another "host" system in the field, which conceivably is any computer system, but is probably a VAX/VMS, RSX, RSTS, or RSX/RTEM system.

A typical use of VTCOM is to boot the RT-11 foreground/background monitor, and run VTCOM.REL as a foreground job. You must also use the XL handler (on PDP-11 systems) or the XC handler (on Professional 300 systems) for communication with the "host" system. By judicious use of CTRL-B or CTRL-F at your RT-11 system console, you can instantly switch between your RT-11 system and the "host." You can start a process running in the RT-11 background, and then CTRL-F into the "host" system and do some work with it.

But here's the best part. Using the related TRANSF.SAV program on the "host" RT-11 or RTEM system, you can transfer files, ASCII or BINARY (!) between the two systems. This is a wonderful facility when developing programs to run on the Professional series under RT-11. Instead of having to acquire a 5¼" diskette drive and UDA50 controller for your PDP-11 system (in order to transfer files via 5¼" diskettes), you can transfer the files by wire using VTCOM and TRANSF.

In order to use the VTCOM utility, you need an extra DL11 or DLV11 on the "host" system, or a PDT-11/150 modem port or Professional series communication port. Actually, to transfer files between the Professional and a PDP-11, we have simply disconnected the cable from the "host" terminal and reconnected it to the Professional communications port.

Not only does the system work, it works well! The VTCOM/TRANSF connection compresses files during transfer (to save time), checks each transfer for errors, maintains modem control, and reports any error conditions or compression statistics at the end of each transfer. VTCOM has many more features; there are simply too many to start describing them here. The VTCOM facility is so good, it is in itself worth the cost of the upgrade to version 5.1. If you have a computing environment with more than one processor running RT-11 or RTEM, you will wonder how you lived without VTCOM.

2. SP AND SPOOL

Together, the programs SP.SYS and SPOOL.REL provide a method for transparent spooling of files for print. What is meant here by transparent is that absolutely no application programs need be modified. If an application program creates a print file, then, after the spooler is installed, the application will automatically and immediately begin using the spooling facility.

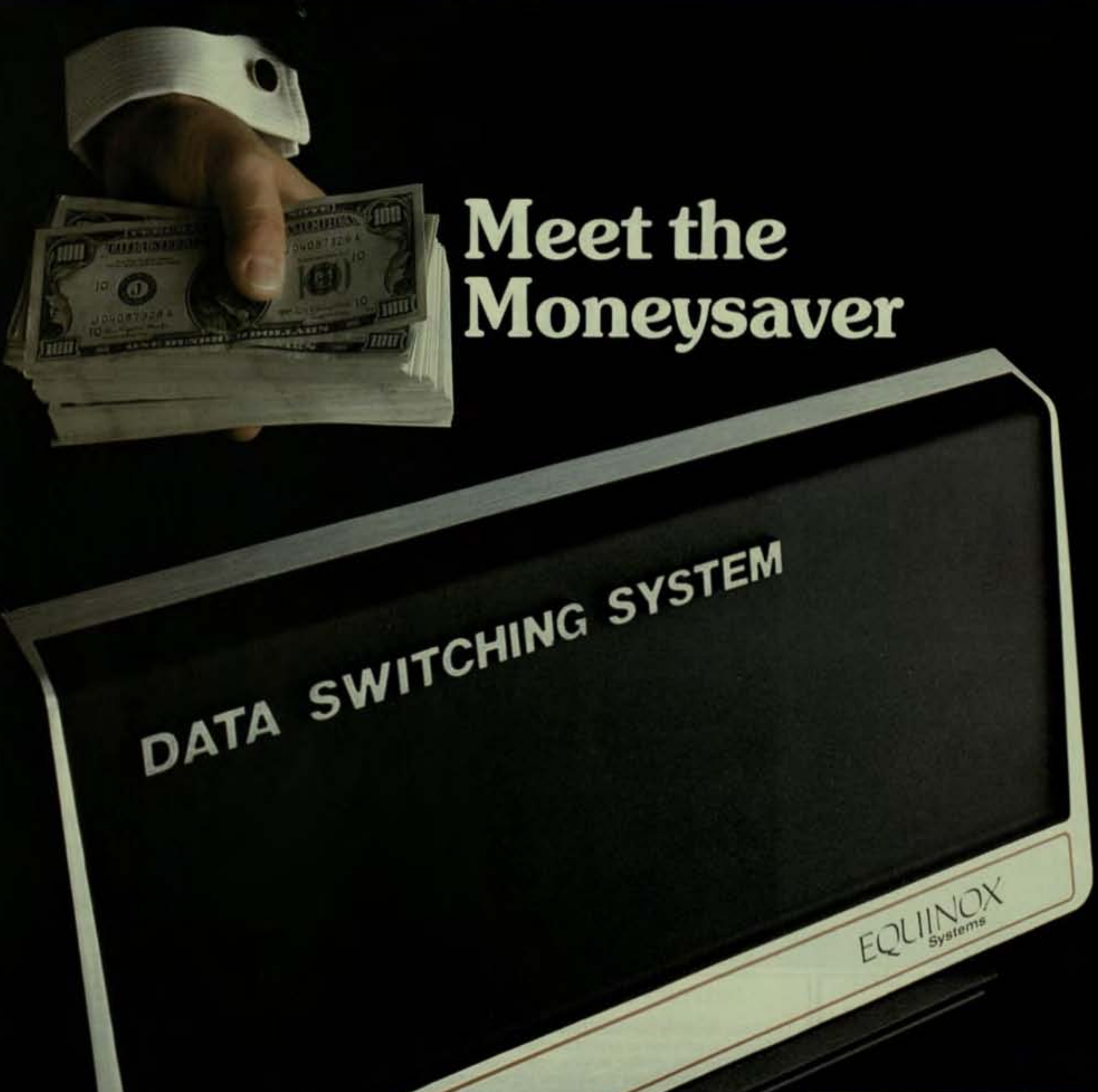
The spooling function must be run under only a foreground/background or extended memory monitor, either as the foreground or system job. The function is initialized using a rather complex set of initial RT-11 keyboard monitor commands. Examples of these complex sets of commands are provided in the manuals; the ideal solution is to create an indirect command file containing the initialization commands and embed them in your startup indirect command file sequence.

The beauty of the spooling function is its (almost) total transparency. Requests to LP (or your current physical printer) are intercepted by the SP handler, which passes the data on to the SPOOL program, which stores the spooled data in a temporary spool file, called SPOOL.SYS, resident on the system disk. The SPOOL program passes the spooled data to the actual printer serially, file by file.

Customization patches are provided to modify SPOOL.REL. The first patch modifies the physical name of the printer. In our PDP-11/34 system, for example, our printer is attached to a DZ-11 and we access the DZ-11 using our YZ handler. Consequently, we patched SPOOL.REL to change "LP" to "YZ7" (the printer is attached to DZ-11 port 7). The second patch allows you to change the size of the temporary file SPOOL.SYS. The default size of SPOOL.SYS is 1000. blocks. This obviously creates problems on systems with a limited amount of disk storage.

Although not published, you can also patch SPOOL.REL so that the name and location of the spool file is other than SY:SPOOL.SYS. Dump the file SPOOL.REL in radix-50 (use the DUMP /RAD50 option or /R switch of DUMP) and the name of the spool file is visible in block 1. Patch the locations containing that name. This is useful if you have a multidisk system, but do not have much room for spooling on the system disk. (WARNING: If you place the temporary spool file on other than the system disk using this patching technique, DO NOT SQUEEZE the disk containing the temporary spool file while the spooler is running or else you will certainly lose data on the nonsystem disk. Unfortunately, RT-11 will let you do this!)

Meet the Moneysaver



Call or send for your free brochure today.

Full featured, less expensive and one-sixth the size of other Data PBX's. The DSS-1 Data Switching System can tie together up to 1,320 terminals and computer ports into an efficient local data network without special cables, or interfacing problems. Even in networks with as few as 50 terminals, the DSS-1 could be a moneysaver.

Better Computer Port Utilization – Ports dedicated to one terminal can now be shared by many terminals ... add terminals without adding computer ports.

Better Terminal Utilization – Any terminal user can select a connection to any computer by simply typing a few keys.
Network Statistics – A running record of all activity lets you identify system problems and needs.
Network Management – Maximizes your network's efficiency with easy on-site, menu-driven reconfiguration.
Easy Installation – Installs with the same connectors and wiring methods as your telephone PBX.
High Reliability – Maximum redundancy and continuous background diagnostics provide virtual zero downtime.

EQUINOX
Systems

Equinox Systems Inc.
16201 SW 95th Avenue
Miami, Florida 33157
(305)255-3500

CIRCLE D264 ON READER CARD

The spooling function also contains a number of SET commands, allowing you to control spooling-in-progress. These include halting all spooling, moving on to the next file, printing of banner pages, and so on.

The size of the temporary file is the major constraint on how smoothly the spooling function works. Printing a single file using the RT-11 keyboard monitor print command, viz.

.PRI OUTPUT

returns control to the background (and the keyboard monitor) in one or two seconds, depending on the size of OUTPUT.LST. What has happened is the file has been copied into the temporary spool file, complete with banner pages, if any. However, should OUTPUT.LST be larger than the temporary spool file, or should you request to print more files than the temporary spool file can hold, the background hangs up until the required spool space is available. Besides the transparency, this is the major difference between the SPOOL/SP and QUEUE method of spooling. With QUEUE, only file names (and not their contents) were passed to the "spooler."

3. SETUP

The SETUP utility permits you to change the characteristics of your VT100-like terminal without putting the terminal into SETUP mode. This is mandatory with the Professional series of computers; the terminal has no SETUP mode and must be customized entirely by the processor.

You use the SETUP facility by typing keyboard monitor commands like

.SETUP LIGHT

which, for example, produces black characters on a white background on a VT100 terminal. What is happening here, of course, is the character string "LIGHT" is being passed to the program SETUP.SAV via the keyboard monitor. In non-Professional series RT-11 systems, the appropriate character sequence is transmitted to the terminal. (LIGHT, for example, sends ESC[?5h to the VT100/VT102 terminal.)

There are a large variety of SETUP options. On non-Professional series systems, you can change most of the characteristics supported by VT102 terminals. On Professional series systems, you can change terminal characteristics, printer characteristics (if you have a printer attached to the printer port), color monitor characteristics (for example, you can display magenta characters on a yellow background), and characteristics of the time-of-day clock embedded in the Professional series hardware.

FROM THE RT-11 EDITOR

RT-11 ON THE PROFESSIONAL SERIES

The Professional series computers contain an LSI-11/23 processor. This means that the processor should be able to run any program that runs on the 11/23. RT-11 is a program running on the 11/23. Conclusion: RT-11 should be able to run on the Professional series.

Unfortunately, there are two major problems. The first problem is that although the Professional processor is an 11/23, the bus is non-standard. The bus used to be proprietary; Digital is now releasing details of its structure. The second problem is that the video display monitor has absolutely no peripheral intelligence of its own. Every character, every pixel of the monitor is controlled by the 11/23 processor using some 11/23 program.

The first problem has been solved successfully by RT-11 version 5.1. RT-11 handlers for the Professional series disk drives attached to the non-standard bus are incorporated and work well. However, the problem of controlling the video monitor with RT-11 has resulted in serious difficulties in using standard RT-11 on the Professional series.

When you press a character on the Professional keyboard, some portion of RT-11 must interpret the character, determine where the character is on the screen, determine the character mode (double-height, double-width, bold, flashing, underscored, etc.) and then must set bits in a special portion of memory to make the video screen glow appropriately. Not only are there a lot of bits to control, there is a lot of work and intelligence required to control those bits. Look inside the VT100 terminal by removing the plate on the back: what is being done by those controller cards you see must now be done completely by the RT-11 software. It is not a simple thing. (As an illustrative example, RT-11 must totally simulate the "flashing" cursor (block or underscore) by setting and resetting bits at the correct time).

In RT-11 version 5.1 running on the Professional series, the part of RT-11 which controls the video display is called the PI handler (for Professional Interface). Whenever a character is to be displayed on the screen, or whenever characters change mode, or the time-of-day clock is modified, RT-11 passes the request on to the PI handler, which interprets the request and executes the requested function. In this mode, PI acts like a giant subroutine to RT-11.

The problem is: PI is TOO giant. Depending on the monitor you are running, the PI handler requires between 7140. and 8888. words of memory. If you are not using the RT-11 extended memory monitor, ALL of this memory must lie between the system disk handler and the resident monitor in the lower 56Kb of physical memory. This memory is subtracted from that available to run background or foreground programs.

To understand the impact of this, let's say you're going to run on the Professional 350 under the RT-11 extended memory monitor (XM). To save memory, you SYSGENed RT11XM without .FETCH support (and without a lot of other stuff, too). You'll want to use SL, the single-line command editor for your keyboard monitor commands; it's one of RT-11's nicest features. So LOAD SL and SET SL ON. Hmmmm. You'll want to load XC because you plan on using VTCOM to transfer some data from a remote "host." So LOAD XC. Now that transparent spooler is really nice. So we'll LOAD SP. You've got some logical disks on your 10 megabyte Winchester — you'll have to LOAD LD. You'll be using some workfiles in extended memory; after all, your Professional 350 is equipped with 512 Kb of memory. And oh yeah, you forgot to load LS as your spooler output device, so LOAD VM and LOAD LS. There, everything's loaded . . . let's get to work.

First you use the SHOW MEMORY command. Oh, oh! The largest available address is 57724 meaning there are

BASIS retrieves information from 1 to 10,000,000 records in nothing flat.

Tired of manually searching through massive files of data? Let BASIS turn your existing files into an automated Information Center.

Imagine a system that retrieves textual information within seconds, the same way a database management system retrieves data. That's BASIS, a proven, interactive retrieval system that does much more than examine titles.

BASIS actually probes to find the key word or phrase you're looking for within any record. Then it displays a list of all records containing the word or phrase you specified. BASIS also displays any or all of the information, and prints hard copy if you want.

BASIS can turn your existing files into an automated Information Center within a few days. *So if you have one or ten million records that need rapid referencing, see what BASIS can do for you.* Call or write for a descriptive brochure today.

Battelle

Software Products Center
505 King Avenue
Columbus, Ohio 43201
Telephone (614) 424-5524

Visit us at DEXPO EAST '84 (April 4-6, 1984) Booth Number 828

BASIS runs on DEC VAX® and DEC® mainframes.
BASIS OA — with DECMATE® word processor link in test site.
UNIX version of BASIS available mid-1984.
DEC, VAX, DECMATE are trademarks of Digital Equipment Corporation.

CIRCLE D175 ON READER CARD

FROM THE RT-11 EDITOR

12010. words of memory left for "user" programs. Well, anyhow, let's transfer that data from the "host" first. You try to run VTCOM.REL and are informed that there is "insufficient memory." In fact, you try to run that application software the nice salesman sold you and find out that none of it runs either ("Start fail" and "JOB STARTUP ERROR" messages appear).

The situation using the RT-11 foreground/background monitor (RT11FB) is only marginally better (about 500. words better). Of course, a few of the problems can be circumvented by loading, unloading, and reloading handlers at the correct time. But, try to imagine your average "professional" user (lawyer, doctor, dentist, etc.) doing this. It makes you laugh.

The problem is the size of PI.SYS. This problem is a direct result of Digital's decision to strip intelligence from the Professional video display. There are two solutions to the problem. The first is a hardware solution: provide an extra serial port and a video terminal with intelligence. This has the unfortunate side effect of raising the price of the hardware and eroding the Professional's precarious market edge.

The second solution is a software solution. Change RT-11 so that the extended memory monitor is completely mapped and resides, with handlers, in the far-off areas of memory. This would increase the size of the job space to a full 64Kb and abolish "low memory" problems on the Professional and all other processors forever.

So why hasn't this been done? First, there is the rush to get RT-11 version 5.1 with "support" for the Professional to market. In less than 10 months since the release of RT-11 version 5, a very small software engineering group has practically accomplished the impossible. Compare the man-years in developing RT-11 on the Professional with developing P/OS on the Professional: you'll see what I mean. Digital wants to move (sell) those Professionals. There are already large numbers of applications running under RT-11. These applications could run on the Professional on RT-11. Consequently, new markets for the Professional hardware would become accessible.

The second reason is a very technical one. To make RT-11 a mapped monitor conflicts with the philosophy that it be genable on all processors, including those without memory management or extended memory. Also, design assumptions that RT-11 runs in the lower 56Kb of memory permeate the monitor source code in subtle ways. Finding the subtleties is a job that simply cannot be rushed. And the result would be two RT-11s: the traditional RT-11 with which we are familiar and a new RT-11 (RT-11 PLUS?) which runs as a mapped monitor (and might even run on a VAX in PDP-11 compatibility mode).

The final reason is a political one. RT-11, running as a mapped monitor on the Professional, would directly compete with the P/OS product. If this were to happen, the implication of the reality would undoubtedly cause heads to roll inside Digital. After all, wasn't RT-11, the original "single user" operating system, the natural choice to run on a new "single user" computer?

What will happen with the Professional and RT-11 is difficult to predict. The Professional with or without RT-11 is a solid, durable, reliable, attractive desktop computer. It's got status! The keyboard is nice to use and easy to touch. The disk is quiet. The processor is fast. But in Canada, where I live, only $1000 more (Canadian) gets me a similarly equipped Micro-11 with a Q-Bus to which a selection of thousands of third party peripherals can be later attached and which runs RT-11 version 5 (or 5.1) with no memory problems whatsoever. My choice is obvious. ∎

Retro-Graphics® and DEC.

Three models now deliver 800-by-480 resolution with DEC ReGIS and Tektronix emulation.

Whether you're doing business charts or technical diagrams, Retro-Graphics enhancements deliver the highest bit-map resolution and widest variety of protocols available on DEC™ displays.

With our 800-by-480 resolution, graphs — both displayed and printed — come out sharply defined. More professional. Without the exaggerated "stair-stepping" experienced on low-resolution terminals.

But most important is our range of graphics emulations.

With one Retro-Graphics model, for instance, you can generate point plots and line drawings with Tek® 4014 emulation.

Another employs user-friendly Tek 4027 commands for drawing and filling bar and pie charts.

And the third provides you with DEC's versatile ReGIS protocol as well as 4014 graphing.

With Retro-Graphics, therefore, you can interface to a large selection of name-brand software. PLOT 10™, DISSPLA®, and DEC's DATATRIEVE™, to name but a few.

Retro-Graphics. Now at 30,000+ installations. And backed by full warranty and comprehensive service programs.

Strong reasons for choosing Retro-Graphics over any other for developing the precise lines that make your graph stand apart.

Contact us today for full details, demonstration, and the name of your local DE/DEC dealer — your "one" source for graphics.

Retro-Graphics and DEC.

A choice of Retro-Graphics models.

Retro-Graphics — a PCB assembly — installs in DEC VT100/101/102/131/132™ VDTs and features 800-by-480 resolution, fast 16-bit microprocessing, and multiple I/O.

NEW
DQ640 SERIES
- Tek 4014 emulation
- Vector drawing and point plotting
- Low cost — under $1,300

GEN.II™
DQ650 SERIES
- Tek 4027 and 4014 emulation
- Primitives drawing, solid or pattern fill
- Multiple character sets
- User-friendly commands

Tek/ReGIS
DQ650 SERIES
- DEC ReGIS and Tek 4014 emulation
- Primitives drawing, solid or pattern fill
- Compatible with DEC's DATATRIEVE and Tek's PLOT 10

DIGITAL ENGINEERING
630 Bercut Drive, Sacramento, CA 95814
(916) 447-7600 Telex: 910-367-2009

CIRCLE D263 ON READER CARD

(**Editor's Note:** Here is the second installment of Rex Jaeschke's series on the constructs of "C." The programs have been tested using the C compiler from Whitesmiths, Ltd. of Concord, Massachusetts, running on a DEC PDP 11/23 with RSX-11M V3.2. The C compiler has the same capabilities across PDP-11 and VAX series processors and operating systems. Throughout the article, Mr. Jaeschke refers to "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie as "K&R.")

LET'S C NOW

LOOPING AND TESTING

By Rex Jaeschke, Washington, D.C.

In the last issue we saw several program examples that used the printf library function. Unfortunately, the Whitesmiths C compiler library does not include this common function. Several other functions that we will use later (including getchar and putchar) are also missing. These three functions are replaced by putfmt, getch and putch. Beware, these functions behave differently from those they replace. In order to use that compiler while following K&R and this column, the user may need to slightly modify program examples. Whitesmiths, Ltd. published a version of printf written in C in their user newsletter several years ago. A copy of this should be available from them.

Like most traditional programming languages, C has several different methods of implementing conditional and unconditional branching and looping. This month we will take a look at several of these methods and comment on their use, particularly where more than one method can be used to achieve the same result.

THE WHILE CONSTRUCT

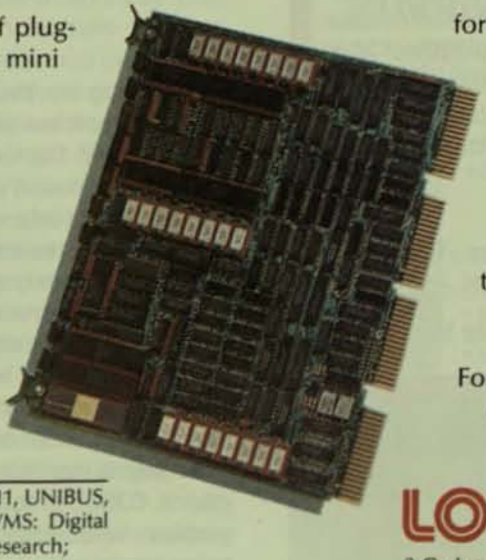
A while loop consists of a condition and a body. While the condition is true (non-zero), the body is performed. Let's look at a simple example. /*---while.c The while construct---*/

```
main ()
{
    int counter;

    printf ("i\ti*i\n");            /* print headers */
    printf ("-----------\n");
    counter = 1;                    /* initialize loop control variable */
    while (counter <= 10) {         /* test loop control variable */
            printf ("%d\t%d\n", counter, counter * counter);
            ++counter;              /* increment loop control variable */
    }
    printf ("-----------\n");       /* print footer */
    return;
}
```

The output produced by this program is as follows.

```
i   i*i
-------
1   1
2   4
3   9
4   16
5   25
6   36
7   49
8   64
9   81
10  100
-------
```

Make your VAX, PDP-11 or LSI-11 speak fluent CP/M & MS-DOS



With our CARDWARE™ series of plug-in boards and software, your DEC mini can now execute the vast library of low cost, user friendly CP/M and MS-DOS software with NO MODIFICATION to either the programs or host computer. In effect, any DEC UNIBUS™ or Q-BUS™ system workstation can become a high performance, personal computer at a fraction of the price.

CARDWARE™ is simple to install, operate and is available for RT-11, TSX-Plus, RSX11M/M-Plus, RSTS/E and VAX/VMS operating systems. With CARDWARE™, your DEC Application and micro computer programs can share data files, run simultaneously without interference and exchange information between the two environments to create a truly powerful, cost-effective system.

For further information and for the dealer nearest you, call or write:

LOGICRAFT Inc.

3 Graham Drive Nashua, NH 03060 (603) 888-4448

CIRCLE D158 ON READER CARD

The first printf call produces the headings i and i*i separated by a tab; the second printf call underlines them. The integer variable counter is used as the loop control variable and is initialized with a value of 1.

The while condition in parentheses, (counter < = 10), is tested and if the condition is true, the body of the loop (the statements enclosed by the inner pair of braces) is executed and the condition is tested again. When the test becomes false (condition = zero), in this case when counter > 10, the loop ends, and execution continues at the statement following the end of the loop. If no statement follows the loop, the function exits. If the while condition is initially false, the loop is never executed. This means that the program will behave rationally regardless of the initial value of the control condition.

The body of the loop may be a single or a compound statement. A compound statement (or block) is a group of declarations and/or statements delimited by { and } which syntactically is treated as one large single statement. Each function must contain at least one compound statement, that being between the function's opening and closing braces. A compound statement may contain other compound statements. One common programming error is the omission of the braces from around a compound statement. Consider the following example.

```
function ()
{
...
     while (condition)
             statement-1
             statement-2
     statement-3
...
}
```

Judging by the indented format, it appears that the programmer intended both statement-1 and statement-2 to be included in the loop body. However, as no braces exist to define a compound statement, the body is considered to be the simple statement statement-1. Statement-2 (like statement-3) is executed regardless of the condition evaluation and then only after the loop has been terminated or skipped. This error typically occurs when the loop body originally had only one statement (and hence did not require delimiting braces) with another statement being added later. Beware, hasty code modifications and enhancements may be difficult to debug, particularly in this case where the code seems to "line up" and looks correct at a passing glance. COBOL programmers no doubt have experienced this problem when they misplace end-of-sentence periods. To force the loop body to include both statements, the code should read:

```
function ()
{
...
    while (condition) {
            statement-1
            statement-2
    }
    statement-3
...
}
```

The condition expression may include such obvious operators as >, >=, < and <= and two less obvious ones != and ==. Note the unusual form != which means "not equal to." Equality is tested using == which is different from the assignment operator =. I find this distinction quite useful as equality testing and assignment are quite different operations. This is similar to the approach taken in Pascal where := and = are used, and is an example of the unambiguous design of C. This kind of approach to language definition makes it considerably easier to write syntax checkers.

One special condition is while (1). As 1 is non-zero and therefore true, the loop is infinite and must be terminated using the break or return statements, or by some other mechanism.

Each time the body of the loop is executed, the value of counter and counter squared are printed on the same line separated by a tab. Note that expressions, such as counter * counter, may be used as function arguments. In fact, C allows a compound expression, a variable or a constant to appear in assignment statements and function calls, provided they reduce to a value of the expected data type.

The unary increment operator ++ is a common and succinct notation used throughout C. ++counter; and counter++; are equivalent to counter = counter + 1;. The ++ increment operator may be used as either a variable prefix or suffix. In this case, both have the same affect; they increment counter by one. However, if a and b are integers, a = b++; and a = ++b; give different values to a. In the first example a is set by b and then b is incremented by 1. In the second, b is incremented and then its new value is assigned to a. In the above example, my personal preference is ++counter as that format infers "increment counter" which defines its action exactly. ++i may not be written as + + i, however, ++ i and i ++ are legal. The unwary operator includes both + symbols. There is an equivalent decrement operator --. It is tempting to consider ++ and -- as operators that increment and decrement by 1 respectively. Although this is true for integer variables, these operators have a much more powerful role when used with pointers, as we shall see in future articles.

The return statement causes control to be returned to the calling function. A return from function main terminates the program. The return statement has the same affect as the last } in a function. In the above example, the return; is superfluous.

Note the way that the brace pairs are lined up to make the logic flow more obvious. This is the format recommended by K&R and greatly improves program readability.

INCREMENTING AND DECREMENTING VARIABLES

In the first article I mentioned that C allowed the programmer to get quite close to the host machine architecture and instruction set, particularly in the generation of efficient machine code. To see an example of this, let's look at several ways of incrementing a variable and the MACRO-11 code generated by one commercial compiler. The following discussion applies equally to variable decrementing.

```
main ()
{
                                               MAIN::
                  /* 000000  010546                   MOV R5,-(SP) */
                  /* 000002  010605                   MOV SP,R5 */
      int i;
                  /* 000004  062706  177770   ADD #-10,SP */
      ++i;
                  /* 000010  005265  177770   INC -10(R5) */
      i++;
                  /* 000014  005265  177770   INC -10(R5) */
      i = i + 1;
                  /* 000020  016500  177770   MOV -10(R5),R0 */
                  /* 000024  005200                   INC R0 */
                  /* 000026  010065  177770   MOV R0,-10(R5) */
      i += 1;
                  /* 000032  005265  177770   INC -10(R5) */
}
                  /* 000036  000167  000000G  JMP C$RETS */
```

+ +i and i+ + generate the same four bytes of code and directly use the increment instruction. i = i + 1; however, generates ten bytes of code and requires two MOVes and one INC instruction. Programmers concerned with speed will appreciate the obvious advantage of using the + + operator. In fact i = i + 2; and i = i + 3; may be replaced by two or three occurences of + +i; respectively and still give speed advantages.

The expression i + = 1; is yet another form of assigning a value to a variable and is an abbreviated form of i = i + 1;. In this case, it generates the same four bytes of code as + +i; and i+ +;. i + = n can be used for any n and for n > 1. This form is more efficient than + +i; repeated in n times. Other compound assignment operators include -=, * = and / =.

One other advantage of using + + and -- will be seen later when we deal with structures. Then using structures it might be useful to refer to a "variable" by its full structure pathname, such as employee.special.sickdays. In this case, incrementing the number of sick days that an employee has used, requires either

employee.special.sickdays = employee.special.sickdays + 1;

or

+ +employee.special.sickdays;

or

employee.special.sickdays+ +;

or

employee.special.sickdays + = 1;.

Which would you rather type or read? The first alternative is the hardest to type, is most error prone and generates the most code. Choosing one of the other three is largely a matter of personal preference.

Increment and decrement operators may be used in statements such as

printf ("%d %d %d \ n", a++, ++b, c--);.

Using this knowledge, one might be tempted to use something like

printf ("%d \ t%d \ n", counter, counter * counter++);

in the while.c example above, and eliminate the need for the separate increment statement. Although this may produce the expected result on one compiler it may not on others. C, like most languages, gives no guarantee as to the order of evaluation of multiple arguments in a function call. The arguments might be evaluated as counter, counter++ * counter or counter++, counter * counter instead, which are quite different.

SYMBOLIC CONSTANTS AND CHARACTER I/O

Let's look at a simple, yet powerful utility that uses the while construct just discussed.

```
/* --- copy1.c Copying input to output, symbolic constants --- */

#define EOF -1

main ()
{
    int c;

    while ((c = getchar ()) != EOF)
            putchar (c);
}
```

Getchar and putchar are functions which are usually provided by the compiler vendor in their C run-time library. (Whitesmiths supply getch and putch instead.) These functions are NOT part of the C language. Getchar gets a character from standard input and putchar writes a character to standard output.

This program copies characters from standard input to standard output until end-of-file is reached. Unfortunately there are several different conventions used to represent the end-of-file marker (-1 being the most popular one) so it is unwise to test for EOF against a specific value if the routine is to be portable. Hence c is compared to EOF, a symbolic constant which is defined at the start of the program. Symbolic constants are particularly useful when they are referenced several times in a source file. They also help document obscure constant values.

Symbolic constants are handled by the C preprocessor which replaces each of their references in the source file with their defined value. Hence the EOF in the while statement is replaced with the character string -1 before the input line is passed to the compiler. Other examples of symbolic constants are #define PI 3.1415926 and #define LINEFEED 10. These constants are similar to the user defined symbols allowed in the MACRO-11 and VAX-MACRO

The Newest Innovation in Controller Technology From the Oldest Name in Multifunction Controllers.

Introducing the World's First LSI-11 Emulating Multifunction Disk/Tape Controller.

Say hello to SPECTRA 25, the cornerstone of our new family of high-performance disk/tape controllers designed for use with DEC's LSI-11 computer.

This revolutionary Q-Bus compatible single quad board lets you interface any combination of two SMD disks and four formatted ½-inch tape drives. By using extended commands to program the onboard E^2PROM, you can easily select drive mixing, mapping, and many other features—all without removing the controller from the system.

The SPECTRA 25 emulates DEC's RM02/5 and RM80 disk subsystems, and DEC's TS11 tape subsystem. It also provides complete emulation for operation with DEC's RT-11, RSX-11M, RSX-11M-PLUS and RSTS/E operating systems.

To further enhance system performance, Spectra Logic offers SPECTRA STREAM™ software, a streaming tape backup utility that can back up an entire 80MB drive in only seven minutes.

Spectra Logic first introduced the multifunction concept back in 1979. And we've been quietly revolutionizing the market ever since with families of controllers that provide the high-performance, proven reliability, and added value you need to stay competitive.

We also offer the industry's most comprehensive one year warranty, responsive nationwide service, and ongoing technical support.

SPECTRA 25 is the latest innovation in controller technology from the company with peripheral vision. Spectra Logic. For further information, including complete technical specifications, call or write us today.

Spectra Logic Corporation
1227 Innsbruck Drive, Sunnyvale, CA 94089
(408) 744-0930 TWX 910-339-9566
TELEX 172524 SPL SUVL

International Sales Office:
Belgium (32) (2) 5134892

SPECTRA LOGIC

 CIRCLE D212 ON READER CARD

See us at DEXPO EAST 84, Booth #108

V3.2

THE DEC* PROFESSIONAL

THE MAGAZINE FOR DEC USERS

Please print all information requested. Please answer all questions. Area Code Office Telephone

Do you wish to receive (continue to receive) a **FREE** subscription to The DEC Professional

☐ Yes ☐ No Date X ____________

☐ Check here if renewal.

☐ Canada $25 (US$) ☐ Foreign $50 (US$) check must accompany this card.

X ____________

Signature required to qualify does not obligate me or my company

MAKE ANY NECESSARY ADDRESS CHANGES HERE

NAME ____________

TITLE ____________ MAIL STOP ____________

COMPANY ____________

ADDRESS ____________

CITY/STATE/ZIP ____________

1. **Your own primary area EDP activity** (check ONE only)
   - ☐ **User** of Computing or Data Processing Equipment
   - ☐ **Manufacturer** of Computing or Data Processing Equipment

2. COMPANY MANAGEMENT

☐ A. **General & Corporate:** President/Owner/Director/Chairman/Partner, Vice President/General Manager.

☐ B. **Financial:** Comptroller/Treasurer, Vice President Finance, Chief Accountant.

☐ C. **Administrative Services:** Vice President Operations, Director or Manager Purchasing/Personnel/Office, Directors or Managers of Microfilm.

☐ D. **Marketing:** Vice President Sales/Sales Manager, Director or Manager Marketing.

COMPUTER/SYSTEMS OPERATIONS

☐ E. **Management:** Vice President EDP, Director/Manager/Head/Chief of EDP/MIS, Director or Manager of Programming, Director or Manager Systems & Procedures, Director or Manager Communications, Director or Manager EDP Operations.

☐ F. **Supervisory:** Supervisor Equipment/Analysis, Chief/Lead/Senior Programmer, Chief/Lead/Senior Mathematician, Senior Systems Analyst

☐ G. **Staff:** Analyst/Systems Analyst, Mathematician, Programmer, Machine or Computer Operator, Software Designer, EDP Auditor.

ENGINEERING/MANUFACTURING

☐ H. **Management:** Vice President or Director of R&D, Vice President or Director Engineering, Plant or Production Manager/Director, Chief Engineer.

☐ I. **Staff:** Member of Technical Staff/Group Leader, Systems Engineer, Design/Production/Research Engineer, Maintenance Engineer, Application Engineer, Other Engineer.

☐ J. **CONSULTANT**

☐ K. **EDUCATOR:** Educator (instruct EDP) and Educator (use EDP)

☐ L. **OTHER** ____________

3. **Primary end product or service of your firm at this location.** (Insert code)

☐ A. Manufacturing: Computers, Data Processing and Peripheral Equipment.

☐ B. Other Manufacturing and Processing.

☐ C. Government: Federal, State, Municipal, including Military.

☐ D. Educational: Colleges, Universities, and other Educational Institutions.

☐ E. Computer and Data Processing Services including Software Services, Service Bureaus, Time Sharing and EDP Consultants.

☐ F. Finance: Banking, Insurance, Real Estate, Securities, and Credit.

☐ G. Trade: Wholesale and Retail.

☐ H. Medical and Legal Services.

☐ I. Transportation Services: Land, Sea, and Air.

☐ J. Utilities: Communications, Electric, Gas, and Sanitary Services.

☐ K. Construction, Mining and Agriculture.

☐ L. Other Business Services, such as Business Managment Consultants, Hotels, Amusements and Non-Profit Organizations

4. **Total number of employees in the entire company/institution/governmental department.**

☐ 1. 10,000 or more ☐ 2. 5,000 to 9,999 ☐ 3. 2,500 to 4,999 ☐ 4. 1,000 to 2,499 ☐ 5. 500 to 999 ☐ 6. 250 to 499 ☐ 7. 100 to 249 ☐ 8. 6 to 99 ☐ 9. Fewer than 6 (Check one only)

5. CPU (enter any you have)

VAX 782 ☐, 780 ☐, 750 ☐, 730 ☐, MICROVAX ☐, PDP 70 ☐, 44 ☐, 34 ☐, PDP 24 ☐, 23+ ☐, 23 ☐, 35/40/45/60, etc. ☐, MICRO-11 ☐, LSI-11 03 ☐, 11/73 ☐, LSI-11 Falcon ☐, T-11 ☐, PDP8 ☐, WPS ☐, VT180 ☐, 10/20 ☐, PDP(other) ☐, PROFESSIONAL 350 ☐, 325 ☐, RAINBOW 100 ☐, 100+ ☐, DECMATE I ☐, II ☐

6. DISK TYPES (check any)

RX ☐, RL ☐, RC ☐, RK ☐, RP ☐, RM ☐, RA ☐, RF/RS ☐, other ☐

7. DISC SUPPLIER (check either or both)

DEC ☐, NON/DEC ☐; I also have: HSC50 ☐, DECNET ☐, ETHERNET ☐, STREAMERS ☐, TU/TE TAPES ☐, DBMS ☐

8. TOTAL DISK STORAGE (check one)

<5MB ☐, 5-10MB ☐, 11-100MB ☐, 101-200MB ☐, 201-500MB ☐, 501-1GB ☐, >1GB ☐, >2GB ☐

9. OP/SYS (check any you use)

VAX/VMS ☐, RSTS/E/CTS500 ☐, RSX-11 ☐, RT11/CTS300 ☐, MUMPS ☐, P/OS ☐, CP/M ☐, TSX ☐, UNIX ☐, TOPS ☐, IAS ☐, MS/DOS ☐, CONCUR CP/M ☐

10. I am an OEM ☐ YES ☐ NO

READER INQUIRY CARD — MARCH 1984 (Card expires May 1984)

Circle numbers of advertisements, new literature or new products here:

D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18
D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36
D37	D38	D39	D40	D41	D42	D43	D44	D45	D46	D47	D48	D49	D50	D51	D52	D53	D54
D55	D56	D57	D58	D59	D60	D61	D62	D63	D64	D65	D66	D67	D68	D69	D70	D71	D72
D73	D74	D75	D76	D77	D78	D79	D80	D81	D82	D83	D84	D85	D86	D87	D88	D89	D90
D91	D92	D93	D94	D95	D96	D97	D98	D99	D100	D101	D102	D103	D104	D105	D106	D107	D108
D109	D110	D111	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126
D127	D128	D129	D130	D131	D132	D133	D134	D135	D136	D137	D138	D139	D140	D141	D142	D143	D144
D145	D146	D147	D148	D149	D150	D151	D152	D153	D154	D155	D156	D157	D158	D159	D160	D161	D162
D163	D164	D165	D166	D167	D168	D169	D170	D171	D172	D173	D174	D175	D176	D177	D178	D179	D180
D181	D182	D183	D184	D185	D186	D187	D188	D189	D190	D191	D192	D193	D194	D195	D196	D197	D198
D199	D200	D201	D202	D203	D204	D205	D206	D207	D208	D209	D210	D211	D212	D213	D214	D215	D216
D217	D218	D219	D220	D221	D222	D223	D224	D225	D226	D227	D228	D229	D230	D231	D232	D233	D234
D235	D236	D237	D238	D239	D240	D241	D242	D243	D244	D245	D246	D247	D248	D249	D250	D251	D252
D253	D254	D255	D256	D257	D258	D259	D260	D261	D262	D263	D264	D265	D266	D267	D268	D269	D270
D271	D272	D273	D274	D275	D276	D277	D278	D279	D280	D281	D282	D283	D284	D285	D286	D287	D288
D289	D290	D291	D292	D293	D294	D295	D296	D297	D298	D299	D300	D301	D302	D303	D304	D305	D306
D307	D308	D309	D310	D311	D312	D313	D314	D315	D316	D317	D318	D319	D320	D321	D322	D323	D324
D325	D326	D327	D328	D329	D330	D331	D332	D333	D334	D335	D336	D337	D338	D339	D340	D341	D342

FOR PROMPT ACTION ON ITEMS CIRCLED, FILL IN SECTION BELOW: (Please use label from magazine if available)

NAME ____________ TITLE ____________

COMPANY ____________

ADDRESS ____________ SUITE ____________

CITY ____________ STATE ____________ ZIP ____________

TELEPHONE ( ) ____________

☐ I wish to subscribe to THE DEC* PROFESSIONAL please send information.

V3.2

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 16 AMBLER, PA

THE DEC PROFESSIONAL
P.O. BOX 362
AMBLER, PA 19002-0362

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 16 AMBLER, PA

POSTAGE WILL BE PAID BY ADDRESSEE

THE DEC PROFESSIONAL
READER INQUIRY
P.O. BOX 362
AMBLER, PA 19002-0362

assembly languages, except that assembler symbols can only be equated to valid assembler expressions whereas C compile-time constants can be used to replace almost any arbitrary text string in a C program. Symbolic constants have the same naming conventions as variables, however, they are usually written in upper case which makes them stand out from user defined variables.

The character read by getchar is stored in variable c which is defined as an integer rather than as type char, which seems odd. As all character bit patterns may be valid characters in a character set, some mechanism is needed to detect a non-character value such as end-of-file. This is not quite true for the ASCII set as only seven of the eight bits in each byte are used to represent any character. However, some machines use an extended ASCII set which includes various graphics and control characters with codes 128-255. As each character is read, it is compared to EOF and if c != EOF, the character is output. This process continues until EOF is detected.

Notice that the while condition expression contains an assignment. This is valid througout C. Such expressions are evaluated left to right according to operator precedence and, as != has higher precedence than =, the parentheses around c = getchar() are mandatory. Otherwise the expression would be evaluated as (c = (getchar () != EOF)) where c would always be set to 0 or 1 (false or true) instead of the input character.

This may seem a trivial routine but UNIX systems allow redirection of standard input and standard output at the command language interface (CLI) level. If this is done, then copy1.c becomes an amazingly simple general purpose file copy utility which is device independent. For example, on UNIX, the command COPY1 <INFILE.DAT> OUTFILE.DAT copies the file INFILE.DAT to OUTFILE.DAT The symbol < causes the system input to be redirected to the file INFILE.DAT while > redirects system output to OUTFILE.DAT. Another symbol >>, indicates an output file which should be appended to rather than written over.
RSX/IAS systems do not provide a facility to redirect system input and output at the MCR or DCL level. In this case, some compilers generate code to do redirection by parsing the arguments looking for <, > and >>. This requires the task to be installed so the GMCR$ (Get MCR Command Line) or similar directive can be used to access the arguments. Installations that use a catch-all task do not need to permanently install such user tasks to pass arguments to them. VAX/VMS supports I/O redirection via the ASSIGN/USER_MODE facility.

One caution: the particular vendor's version of getchar that you use may be sensitive to certain characters other than eof. CTRL/C on input may be interpreted as an abort signal. Therefore copying a binary file that contains bytes with the CTRL/C bit pattern, may cause the copy to terminate prematurely.

Printf can achieve the same result as putchar using printf ("%c",c); instead of putchar (c);.

We have seen an example of the use of symbolic constants. Although they remove dependencies from the source code statements above, the dependency still exists in the source code file in the #define directive itself. If a program has several source code modules which each contain references to EOF, each file must be changed if the program is ported to a system where the EOF marker has a representation other than -1. To avoid this the #include preprocessor directive can be used as shown in the following example.

```
main ()
{
    int c;

    while ((c = getchar ()) !=EOF)
            putchar (c);
}
```

#defines that occur in multiple source code files in a program, can be grouped together in a single file, for example DEF.H. Such a file is referred to as a header file and usually has a file type of H. Many compiler kits include header files such as STDIO.H which contains common standard I/O definitions. Header files may contain any valid C source language statement or preprocessor directive. Their contents are "included" into other source code files at compile time. As far as the compiler is concerned, the include file contents are part of the input source file. Moving common symbolic definitions to a header file, reduces the need to modify source files when porting programs to different environments. Any changes necessary need only be made to the header files with these changes being implemented when the source files are recompiled. Other pre-processor directives will be introduced in future articles.

Include file names delimited with ". . ." are searched for in the default directory. Files delimited by <. . .> are expected to be in a system directory (such as LB:[1,1] on RSX systems).

THE FOR CONSTRUCT

Another method of loop processing is the for statement.

```
/* --- for1 c   The for construct --- */

main ()
{
    int i;

    printf ("i (dec)\ti (hex)\t~i (hex)\n\n");

    for (i = -3; i <= 3; ++i)
            printf("%2d\t%4x\t%4x\n",i,i,~i);

    printf ("\nAt the end of the loop, i is %d.\n",i);
}
```

Digital advanc
in video termi

For years, Digital's VT100 terminal has been the CRT to choose if you want the most out of your computer. It has become the industry standard for reliability and ease of use. Not to mention the largest-selling ASCII terminal in the world.

Now Digital advances this standard with the VT200 family.

Three new terminals that embody everything Digital has learned about how to make people comfortable with computers. They offer non-glare screens that can be positioned for the best viewing angle. Keys that are so well arranged on our low-profile keyboard that you increase productivity and convenience. Fifteen programmable function keys eliminate keystrokes to speed up tasks. Plain-language setup commands to easily tailor the screen to each user. Plus, a built-in printer port for hardcopy convenience.

The output produced on a PDP-11 is:

```
i (dec) i (hex) ~i (hex)

-3      fffd        2
-2      fffe        1
-1      ffff        0
 0         0     ffff
 1         1     fffe
 2         2     fffd
 3         3     fffc
```

At the end of the loop, i is 4.

A for statement has three parts, each separated by semicolons. In this example the first part, i = -3; is the initialization part and is executed once before the loop is entered. The second part, i < = 3; is the condition that controls the loop. If this condition is true the body of the loop is executed, then the third part, + + i, is executed and the condition is tested again. The loop terminates when the second part condition is false.

In this example the body of the loop is only one statement and, therefore, does not need to be enclosed in braces, although it can be. Like the while statement, for does not execute the loop at all if the test condition is initially false. The controlling variable i retains its value after the loop is terminated, in this case i = 4.

Until now, numbers printed using printf have been left justified. In this example we see two new edit masks %2d and %4x. The %2d causes the second argument i to be printed as a decimal value right justified with a width of two characters. %4x causes the third and fourth arguments to be printed in hexadecimal format width 4. On the PDP-11, i is a 16-bit integer so only four hex digits are required. On the VAX, i would be a 32-bit field and would require eight hex digits to display negative numbers. The ~ symbol is the one's complement operator.

The above example may be written as:

```
/* --- for2.c   The for construct with a null body --- */

main ()

{
    int i;
    printf ("i (dec)\ti (hex)\t~i (hex)\n\n");

    for (i = -3; i <= 3; printf("$2d\t$4x\t$4x\n",i,i,~i).++i)
            ;

    printf ("\nAt the end of the loop, i is $d. \n",i);
}
```

Here all the work is done in the for statement itself and no loop body is needed. However, the for construct requires at least one statement in the loop body and so ; appears on its own. This signifies the null or empty statement. It is wise to place the null statement on its own line so its presence is more obvious.

For2.c shows the third part of the for construct as having two sub-parts, the printf function call and the increment statement. These two statements are separated by the comma operator and are evaluated left to right. The commas used to separate function call arguments and variable declarations are NOT comma operators and their order of evaluation is therefore not guaranteed. The first part may also contain more than one initialization statement.

The three parts of the for statement are really expressions and although in the above example they all refer to the variable i, they need have no common connection. Parts 1 and 3 are optional. If part 2 is omitted, the loop becomes "infinite." For (;;) | . . . | is equivalent to while (1) | . . . |. Such an infinite loop may be exited using the break or return statement.

THE IF/ELSE CONSTRUCT

The most common way to test the value of a variable or expression is to use the if statement with optional else clause.

```
/* --- if1.c The if/else construct and break statement --- */

main ()
{
    int reply;

    while (1) {
            printf ("\nHave you ever programmed in C? (Y/N) ");
            reply = getchar();

            if (reply == 'y' :: reply == 'Y') {
                    printf ("\nThen skip the introductory lessons.\n");
                    break;
            }
            else if (reply == 'n' :: reply == 'N') {
                    printf ("\nWelcome to the world of C.\n");
            }
            else
                    printf ("\nInvalid reply.  Please try again.\n");

    }
    printf ("\nThanks for your cooperation.\n");
}
```

Examples of output produced are:

Have you ever programmed in C? (Y/N) g
Invalid reply. Please try again.

Have you ever programmed in C? (Y/N) y
Then skip the introductory lessons.

Thanks for your cooperation.

and

Have you ever programmed in C? (Y/N) N
Welcome to the world of C.

Thanks for your cooperation.

We just made WORD-11 a native of VAX.

That's right. At Data Processing Design, we've just upgraded our popular WORD-11™ word processing package for DEC™ systems to native mode for VAX.™

Now users of DEC systems can enjoy even faster, more efficient operation. The Version 3.0 WORD-11 package provides easier, expanded system management, and supports new output devices—including laser printers.

By converting WORD-11 from Macro-11™ to full native Macro-32™ mode, we increased its efficiency and made it possible for you to optimize your system resources.

There are more enhancements, too. The new package includes scientific character set support for VT100™ terminals and the Diablo 630 ECS™ printer. Journaling. Military-style paragraph numbering to extend the existing standard outline paragraph numbering system. An advanced system management utility. And support for the Xerox 2700™ laser printer.

And, of course, Version 3.0 still has all the features that over 1500 computer installations worldwide depend on. Such as list processing, built-in dictionaries, automatic table of contents and footnoting, grammatical editing keys, and more.

For more information about WORD-11 Version 3.0, contact us at DPD. We'll be happy to provide complete details.

And point out the benefits of going native.

Data Processing Design, Inc.

AUTHORIZED digital COMPUTER DISTRIBUTOR

CORPORATE OFFICE
1400 North Brasher
Anaheim, CA 92807
(714) 970-1515

NY OFFICE
420 Lexington Avenue, Suite 633
New York, NY 10170
(212) 687-0104

WASHINGTON D.C. OFFICE
4520 East-West Highway, Suite 550
Bethesda, MD 20814
(301) 657-4098

WORD-11 is a trademark of Data Processing Design, Inc.
DEC, VAX, Macro-11, Macro-32 and VT100 are trademarks of Digital Equipment Corporation.
Diablo, 630, Xerox® and 2700 are trademarks of Xerox Corporation.

CIRCLE D246 ON READER CARD

The while condition (1) is always true by definition. Therefore the body of the loop will execute indefinitely unless it is terminated by some other means. The loop body is a compound statement consisting of the call to printf, an assignment to reply and an if statement. Note that the else clause of the first if test contains another if test. Ifs can be nested up to a compiler defined depth.

Both if tests contain a compound expression. :: is the logical OR operator. :: operator groups are evaluated left to right. && is used as the logical AND operator and takes precedence over :: if both are used in an expression. The order of evaluation of expressions may be altered by using parentheses. Note that the equality operator is = = while = is reserved for assignment purposes only.

The body of both if statements is enclosed in braces as there is more than one statement to be executed. The final else has only one statement and therefore does not need braces although they could be used.

The break statement causes termination of the inner-most current while, for, do or switch loop. Control passes to the statement following the end of the loop construct, in this case, to the printf function call which thanks the user.

Getchar() gets one character from standard input and does not wait for or require the user to press RETURN or ENTER. Hence, the current cursor position is that immediately following the character input. To ensure that their message is printed on the next line, the three printf function calls within the if/else construct print a newline prior to printing their message.

'Y' is a character constant which produces an integer equal to the numerical value of Y in the machine's character set. In the ASCII set this value is 89. 'Y' is preferred to 89 as its meaning is more obvious and is machine independent. The escape sequences used in the printf function edit mask are also single character constants. As we have seen, these include '\n,' '\t,' and '\b.' NOTE: 'Y' is a single character equivalent to an integer while "Y" is a character array consisting of one character. They are not the same.

There are several other less-used constructs available for implementing looping and branching. These are do-while, switch and the inevitable goto. These will be covered in future articles as space permits. Whitesmiths getch and putch do not behave exactly as the K&R functions getchar and putchar. Getch and putch will only echo input and output characters when the respective buffers are flushed with a '\n' newline character. Getchar and putchar are true single character functions.

Next time we will introduce arrays and will discuss the use of functions and their argument passing mechanisms.

(**Editor's Note:** "FORTRAN/RT Tutorial" is a paper originally presented at the Fall 1982 DECUS Symposium in Anaheim and again at the Spring 1983 Symposium in St. Louis. It was well received in both places, and is now distributed to new RT-11 users at Sandia National Laboratories as well as other major laboratories such as Lawrence Berkeley Laboratory and Lawrence Livermore Laboratory.)

FORTRAN/RT TUTORIAL EIS, FIS AND FPU

By

Ron Trellue, Sandia National Laboratories, Albuquerque, New Mexico

David E. Ellis, Major, USAF, Air Force Weapons Laboratory, Kirtland AFB, New Mexico

ABSTRACT

This tutorial is a discussion of NHD, EIS, FIS, and FPU. The hardware available to implement these additional arithmetic instruction sets is presented. Building the FORTRAN IV compiler and library to take maximum advantage of the hardware is discussed. Examples of threaded and inline code are given to show the output from the various types of FORTRAN compilers. The results of running five programs built for the different arithmetic options illustrate the advantages of additional arithmetic hardware. The tests were run on a LSI 11/2, a LSI 11/23 with both the KEF11-AA chip and the FPF11 processor, a PDP 11/34, a PDP 11/45, a PDP 11/44, and a VAX 11/730.

INTRODUCTION

One of my functions at Sandia Laboratories is to provide RT-11 software to new users at Sandia. In addition to giving out RT-11, I give each user all the object modules needed to build the FORTRAN IV compiler and library. Each user is asked what kind of compiler and FORTRAN library he would like to have. Many people are unsure of what hardware they have, so they are not sure how they want their FORTRAN system built.

The purpose of this tutorial is to help clarify some of the options available in building a FORTRAN system, and in particular to discuss the meaning of EIS, FIS, and FPU.

HARDWARE/SOFTWARE TEAMWORK

The discussion of EIS, FIS, and FPU centers around the way the computer handles arithmetic operations such as add, subtract, multiply, and divide. Optional hardware can be purchased to implement certain arithmetic operations in hardware rather than in software. Different hardware is available for various types of processors. The software should be built to take maximum advantage of the type of hardware present on the machine.

Every machine has one or more of the following types of hardware:

- NHD — No hardware
- EAE — Extended Arithmetic Element
- EIS — Extended Instruction Set
- FIS — Floating Instruction Set
- FPU — Floating Point Unit

There are several ways to determine the type of arithmetic hardware on your machine. One way is to look at your purchase order to see what you bought. Another is to look in your cabinet and see what type of board is there. A hard way is the trial and error method. RT-11 has a more sophisticated way to find out, just type

SHOW CONFIGURATION

TYPES OF HARDWARE AVAILABLE

Some machines such as the LSI 11/23 and the PDP 11/44 come with EIS standard with the processors. Some products such as the KEF11-AA chip and the FPF11 board must be bought separately. Not all types of hardware are available for each type of processor. Table 1 shows the hardware options which are available for a variety of processors. EAE is only available on a small number of older machines and will not be discussed in this paper.

TABLE 1. Arithmetic Hardware Options

EAE — Extended Arithmetic Element

KE11A,B	— 11/10, 11/15, 11/35, 11/40

EIS — Extended Instruction Set

KE11-E	— 11/35, 11/40
KEV11	— LSI-11, 11/03
Standard	— LSI 11/23, 11/34, 11/44, 11/60, 11/70

FIS — Floating Instruction Set

KE11-F	— 11/35, 11/40
KEV11	— LSI-11, 11/03

FPU — Floating Point Unit

KEF11-AA — LSI-11/23 Microcode option

FPF-11 — LSI 11/23

FP11 — 11/34, 11/44, 11/60, 11/70

EIS, FIS, AND FPU INSTRUCTIONS

The EIS, FIS, and FPU instructions extend the normal PDP-11 instruction set by providing additional instructions for certain arithmetic operations. These instructions are implemented in the hardware indicated by Table 1. The trick is to get the software to generate these additional instructions so that the hardware can take advantage of them. One way to make use of them is to write your own assembly language program and use them in it. Another is to get the high level language processors to generate these instructions. In the FORTRAN world, this means having the FORTRAN compiler generate the proper code and building the FORTRAN library with the pre-compiled OTS routines which utilize the appropriate arithmetic instructions.

WHAT IS EIS ?

EIS, the Extended Instruction Set provides only four additional instructions.

MUL — Fixed point multiply (32 bit)
DIV — Fixed point divide (32 bit)
ASH — Arithmetic shift (16 bit)
ASHC — Arithmetic shift combined (32 bit)

These instructions only effect integer operations, and do not effect either single or double precision arithmetic. The EIS instructions are standard on most of the newer processors.

WHAT IS FIS ?

The FIS instructions are only available on a small number of older systems, and are not available on most of the newer machines which support a floating point processor. The FIS instructions are:

FADD — Floating add
FSUB — Floating subtract
FMUL — Floating multiply
FDIV — Floating divide

The FIS instructions only work on single precision arithmetic. The results of our time tests will illustrate the advantage of having FIS when doing a large amount of single precision arithmetic.

WHAT IS FPU ?

FPU is a floating point unit which provides arithmetic hardware to implement 46 floating point instructions. They are:

ABSF	LDCDF	MULF	STCDI
ABSD	LDCFD	MULD	STCDL
ADDD	LDCIF	NEGF	STEXP
ADDF	LDCID	NEGD	STFPS
CFCC	LDCLF	SETF	STF
	LDCLD	SETD	STD
CLRF	LDEXP	SETI	STST
CLRD	LDFPS	SETL	
CMPF	LDF	STCFD	SUBF
CMPD	LDD	STCDF	SUBD
DIVF	MODF	STCFI	TSTF
DIVD	MODD	STCFL	TSTD

These instructions are implemented on a PDP-11 with an FP11 floating point processor. On an LSI 11/23 they can be implemented in microcode on the KEF11-AA chip or on the FPF-11 processor.

WHO CARES?

The timing results given later in this paper will indicate the benefits which can be attained by taking advantage of the arithmetic hardware. In certain cases the increase in speed is substantial. Investing in floating point hardware can be expensive, but, depending on the types of calculations to be done, floating point hardware can greatly speed up program execution.

THREADED CODE

The FORTRAN IV compiler can be built to produce either inline or threaded code. Threaded code is hardware independent. This means that the compiler produces the same code no matter what hardware is available on the machine. Threaded code consists of calls to routines in the FORTRAN Object Time System (OTS). The threaded code calls thread their way through the OTS routines which were built for a particular type of hardware. The FORTRAN library containing the OTS can be built for any one of the five types of arithmetic hardware — NHD, EAE, EIS, FIS, or FPU. A threaded code object file could be linked with any one of the five types of OTS libraries which can be built. Threaded code object modules are normally linked with either an NHD or FPU library. The FORTRAN debugger, FDT, only works when linked with threaded code object modules.

INLINE CODE

The inline code compiler generates actual assembly language instructions for each FORTRAN statement with fewer calls to library routines. The FORTRAN IV compiler can only produce inline code for EAE, EIS, and FIS. This means that if you have no arithmetic hardware (NHD), then you must build a threaded code compiler. If you have a floating point unit, you can build a threaded code compiler or, if you have

CRITICAL MASS

Nothing is more critical to your computer system than reliable data storage. Guardian Data combines the latest technology to give you a mass storage subsystem with price and performance second to none. Cover the critical issues and decide for yourself.

Winchester technology. Ultra reliable drives from Fujitsu.

Tape back-up. Streaming and cache-streaming tape drives from Cipher Data Products, the industry leader.

Emulating Disc and Tape Controllers. Field-proven compatibility with DEC operating systems.

Equally critical to the operation of your mass storage subsystem is complete integration and proper maintenance. Our technical specialists will assure fully operational integration and compatibility to your DEC operating system. We further ensure your investment by supplying you with prompt service and on-hand factory spares through our regional factory-trained and certified Technical Services Group.

In short, Guardian Data delivers a high-tech mass storage subsystem so complete that this critical decision becomes simple.

For additional information contact the Guardian Data office in your area, or use our reader service number.

Performance Measure	GUARDIAN/750
Storage Capacities	84-475MB/drive
Access Speeds	20-18msec
Buss Support	Qbus, Unibus, CMI, SBI, Massbus
Operating Systems	RT, RSTS, RSX, VMS, UNIX
Reliability (MTBF)	10,000-30,000 hrs

GUARDIAN DATA

A TRADEMARK OF GROUP THREE ELECTRONICS

Redondo Beach, CA (213) 643-6997 | Irvine, CA (714) 863-1382 | San Diego, CA (619) 292-0525 | Sunnyvale, CA (408) 747-1800 | Portland, OR (503) 684-1870 | Bellevue, WASH. (206) 454-0150

CIRCLE D278 ON READER CARD

both EIS and FPU, you can build a compiler which produces inline EIS code to link with an FPU library. The FORTRAN IV V2.5 compiler can not produce inline FPU code. Shame on DEC! The FORTRAN 77 compiler available under RSX-11M is capable of producing inline FPU code. Some of the timing tests presented later in this paper show the advantages of having inline FPU code.

BUILDING THE COMPILER

The FORTRAN compiler is built from the FORTRAN pieces which are distributed as a set of object modules. The input and output devices are assigned, and the FORGEN program run.

```
.ASSIGN DL1 INP
.ASSIGN DLO OUP
.RUN INP:FORGEN
```

The FORGEN program asks questions about how the compiler should be built. The first question concerns the number of lines allowed per listing page. This depends on the type of line printer you have. The next two questions, the maximum size of a formatted ASCII record and the number of channels open at a time, can both be overidden at compile time so taking the defaults is normally adequate. Finally comes the question about the type of code that the compiler should produce. The choices are threaded, inline EAE, inline EIS, or inline FIS. You can build a threaded code only compiler, an inline only compiler, or a compiler which can produce either threaded or inline code. The way you answer the questions will determine the default setting.

At the end of the question session, a command file, FORBLD.COM is created and can be initiated to build the compiler. The file FORBLD.COM only has one command in it. It will start up one of the following three indirect command files to link together a compiler which will be threaded, inline, or both depending on which command file is run.

F4LTHR.COM — Threaded code only compiler, about 130 blocks in size
F4LINL.COM — Inline code only compiler, about 180 blocks in size
F4LINK.COM — Both inline and threaded compiler, about 210 blocks

How do the answers to the questions get incorporated into the building of the compiler? They are saved in a file on OUP: called DEFLTS.OBJ. The file DEFLTS.OBJ is then linked with the appropriate FORTRAN pieces as specified by one of the three F4L*.COM files to produce a tailored compiler. I like to build a compiler which does both inline EIS by default, and threaded code when asked. Most applications run faster with inline code, but threaded code is handy when a program gets too big since threaded code is usually smaller than inline code.

BUILDING THE FORTRAN LIBRARY

The second piece of software required for a FORTRAN development environment is the FORTRAN library or OTS. The FORTRAN OTS can be added to the System Subroutine

SN'T THAT THOUGHTFUL ISN'T THAT THOUGHTFUL

Library SYSLIB.OBJ or can be put in a separate file with any name you choose. If the FORTRAN library is to be added to SYSLIB.OBJ then SYSLIB.OBJ must exist on the output device. Again device assignments must be made, and the OTS build program can be run.

```
.ASSIGN DL1 INP
.ASSIGN DL0 OUP
.RUN INP:OTSGEN
```

The program will ask a few questions concerning the building of the OTS. One of the important questions is what type of code should the library be built for — NHD, EAE, EIS, FIS, or FPU. This is where you get to specify how you want the library to be built. A command file is created which can be initiated by typing @OUP:OTSBLD.

The OTSBLD command file uses the librarian to build the FORTRAN library from the FORTRAN pieces. The way you answer the questions determines which modules are put into the library. When the compiler is built, the FORGEN program must be run to create the file DEFLTS.OBJ. It is quite easy to build the library yourself by selecting the pieces you want and putting them into the library. A typical build might look like this:

```
.R LIBR
*OUP:SYSLIB=SYSLIB,INP:FPU,INP:OTSCOM/G/C
*INP:VIRNP,INP:V2NS
*//
Global? $ERRS
Global? $ERRTB
Global? $OVRH
Global?
*↑C
```

In building the FORTRAN OTS, the following modules can be used.

This module is mandatory.

OTSCOM.OBJ — Routines everyone needs

Pick one of these, depending on hardware.

NHD.OBJ	—	No hardware, software does it.
EAE.OBJ	—	Routines with EAE instructions
EIS.OBJ	—	Routines with EIS instructions
FIS.OBJ	—	Routines with FIS instructions
FPU.OBJ	—	Routines with FPU instructions

Pick one of these types of virtual array support.

NOVIR.OBJ	—	No virtual array support needed
VIRNP.OBJ	—	Virtual array support under SJ or FB needed
VIRP.OBJ	—	Virtual array support under XM is needed

Pick one of these if threaded code will be used.

V2NS.OBJ	—	No array subscript checking, threaded code only
V2S.OBJ	—	Do array subscript checking, threaded code only

Optional features.

UNI.OBJ	—	Provides SIMRT, standalone FORTRAN support if needed
UIOBYT.OBJ	—	Provides byte resolution in direct access files

The size of the library will depend on the number of modules selected and on the type of arithmetic support requested. For example, an NHD no hardware library will produce the largest library file since all the arithmetic operations must be performed in software rather than hardware. The FPU library will be the smallest since the routines rely on the hardware to do the work.

THE TEST ROUTINES

In order to see if the different code options (NHD, EIS, FIS, and FPU) really make any difference in the execution time of a program, several programs were compiled under the various options and the run times recorded. Five programs were used.

CCA1.FOR — Single precision calculations
CCA2.FOR — Same as CCA1, except double precision
CCA5.FOR — FORTRAN library function test
CCA9.FOR — Binomial expansion
CCA10.FOR — IF test

Programs CCA1 and CCA2 do a variety of floating point operations such as addition, subtraction, multiplication, and division as well as array element addressing and operations with functions. Parts of the programs are as follows.

```
      PROGRAM CCA1
C-----------------------------------------
      STIME=SECNDS(0.0)
        .
        .
11      DO 18 I = 1,N1,1
           X1 = (X1+X2+X3-X4)*T
           X2 = (X1+X2-X3+X4)*T
           X3 = (X1-X2+X3+X4)*T
           X4 = (-X1+X2+X3+X4)*T
18      CONTINUE
        .
        .
21      DO 28 I = 1,N2,1
        E1(1) =(E1(1)+E1(2)+E1(3)-E1(4))*T
        E1(2) =(E1(1)+E1(2)-E1(3)+E1(4))*T
        E1(3) =(E1(1)-E1(2)+E1(3)+E1(4))*T
        E1(4) =(E1(1)+E1(2)+E1(3)+E1(4))*T
28      CONTINUE
        .
        .
61      DO 68 I = 1,N6,1
           J = J*(K-J)*(L-K)
           K = L*K-(L-J)*K
           L = (L-K)*(K+J)
           E1(L-1) = J+K+L
           E1(K-1) = J*K*L
68      CONTINUE
        .
        .
71      DO 78 I = 1,N7,1
        X=T*ATAN(T2*SIN(X)*COS(X))/COS(X+Y
        Y=T*ATAN(T2*SIN(Y)*COS(Y))/COS(X+Y
78      CONTINUE
        .
        .
118     X = SQRT(EXP(ALOG(X)/T1))
        .
        .
      TTIME=SECNDS(STIME)
```

CCA2 is identical to CCA1 except that the variables are declared double precision.

CCA5 is the FORTRAN library function test.

```
      PROGRAM CCA5
C-----------------------------------------
      STIME=SECNDS(0.0)
      N = 26400
      C = 1.0
      DO 1 I=1,N
      D = C
      E = C
      A = SQRT(C)
      B = SIN(D)
      C = COS(E)
      D = SQRT(A)
      E = EXP(B)
      A = SIN(C)
      B = SQRT(D)
      C = COS(E)
      D = ATAN(A)
      E = SQRT(B)
      A = COS(C)
      B = EXP(D)
      C = SQRT(E)
      D = SIN(A)
      E = COS(B)
      C = SQRT(C)
      B = COS(D)
      A = EXP(E)
      D = ATAN(A)
 1    CONTINUE
      A = A + B + D + E
C-----------------------------------------
      TTIME=SECNDS(STIME)
```

CCA9 is a binomial expansion.

```
      PROGRAM CCA9
C-----------------------------------------
      STIME=SECNDS(0.0)
        .
        .
      DO 600 M=1,IA
      BC=0.0
      BB=1.0
      DO 500 N=1,IE
      DO 400 K=IB,IC,ID
      AK=K
      L=N+1
      MM=L/2
      COMB(1)=1.0
      COMB(L)=1.0
      DO 120 I=2,MM
      AI=I-1
      L=L-1
      AN=L
      COMB(I)=AN/AI*COMB(I-1)
  120 COMB(L)=COMB(I)
      IF(N+1-MM*2)150,151,150
  150 COMB(L-1)=(AN-1.0)/(AI+1.0)*COMB(L)
```

```
151 L=N+1
    Q=AK/AA
    P=1.0-Q
    BI=Q**N
    BII=BI
    DO 160 I=2,L
    BI=BI*COMB(I)/COMB(I-1)*P/Q
160 BII=BII+BI
400 BB=BB*BII
500 BC=BC+BII
600 CONTINUE
C---------------------------------------
    TTIME=SECNDS(STIME)
```

CCA10 is the IF check routine.

```
    PROGRAM CCA10
C---------------------------------------
    STIME=SECNDS(0.0)
      .
      .
    DO 300 L=1,N
    DO 300 M=1,N
    GO TO 200
100 IF(I)101,999,999
101 IF(J)999,102,999
102 IF(K)999,999,103
103 IF(I)104,999,999
104 IF(J)999,105,999
105 IF(K)999,999,106
106 IF(I)107,999,999
107 IF(J)999,108,999
108 IF(K)999,999,109
109 IF(I)110,999,999
110 IF(J)999,111,999
111 IF(K)999,999,112
112 CONTINUE
    GO TO 300
200 IF(I)201,999,999
202 IF(K)999,999,203
201 IF(J)999,202,999
203 IF(I)204,999,999
205 IF(K)999,999,206
204 IF(J)999,205,999
206 IF(I)207,999,999
208 IF(K)999,999,209
207 IF(J)999,208,999
209 IF(I)210,999,999
211 IF(K)999,999,212
210 IF(J)999,211,999
212 CONTINUE
    GO TO 100
999 CONTINUE
300 CONTINUE
C---------------------------------------
    TTIME=SECNDS(STIME)
```

GENERATED CODE

The FORTRAN compiler can list the generated code which it produces. A statement was selected from program CCA9 and the generated code for threaded, inline EIS, inline FIS, and inline FPU was listed. The statement chosen was:

150 COMB(L-1) = (AN-1.0)/(AI + 1.0)*COMB(L)

Example 1 shows the code generated by a FORTRAN IV threaded code compiler. The left column gives the relative octal byte address of each threaded code call, and the next column gives the name of a threaded code routine from the FORTRAN library to be executed followed by any pertinent parameters. A discussion of how threaded code works is available on the fall 1980 RT-11 SIG DECUS tape.

Example 1. Threaded Code

```
000404   SAF$MM   $DATA+#004000
                  $DATA+#177770
000412   MOF$MS   $DATA+#004012
000416   SUF$IS   #040200
000422   MOF$IS   #040200
000426   ADF$MS   $DATA+#004006
000432   DIF$SS
000434   SVF$MM   $DATA+#004000
                  $DATA+#177774
000442   MUF$SS
000444   MOF$SA
```

Example 2 shows the code generated for the same statement from an inline compiler built for EIS code. Inline code contains the actual MACRO-11 instructions for each statement. This particular statement required more code to be generated by the inline compiler than did the threaded version. Since no integer operations are performed, none of the four EIS instructions appear in the generated code for this particular example. Both types of code require library routines to be called, so the total number of instructions required to execute this statement is not obvious. Inline code is normally larger and faster than threaded code. The inline code in Example 2 shows that the arithmetic operations are done by jumping to subroutines (JSR PC,$ADDF). These subroutines such as $ADDF are taken from the FORTRAN OTS and can contain EIS instructions, or FPU instructions depending on which arithmetic module was selected when the library was built. The more instructions which can be implemented in hardware, the faster and smaller the routine will be.

Example 2. EIS Code

```
000542   MOV   AN+2,-(SP)
000546   MOV   AN,-(SP)
000552   CLR   -(SP)
000554   MOV   #40200,-(SP)
000560   JSR   PC, $SUBF
000564   CLR   -(SP)
000566   MOV   #40200,-(SP)
000572   MOV   AI+2,-(SP)
000576   MOV   AI,-(SP)
000602   JSR   PC,$ADDF
000606   JSR   PC,$DIVF
000612   MOV   L,R0
$000616  ASL   R0
000620   ASL   R0
000622   MOV   COMB-2(R0),-(SP)
000626   MOV   COMB-4(R0),-(SP)
000632   JSR   PC,$MULF
000636   MOV   L,R0
000642   ASL   R0
```

Add a little tape backup to your DEC microcomputer

SMS announces a ¼" tape cartridge peripheral compatible with DEC's TSV05 at half the price. Requiring only 5¼" of rack or table top height your Winchester disk can't afford not to have one. The SMS FWT01180 (Q-Bus) or FWT11180 (Unibus*) provides up to 21 Mbytes of storage and emulates the TS11* device supported by standard RT-11*, RSX-11M* and TSX-PLUS** software. Besides disk backup, the low cost tape cartridge can support system boot, media interchange, archiving, journal operations and software distribution applications.

Scientific Micro Systems, Inc.
777 E. Middlefield Road
Mountain View, California 94043
(415) 964-5700

Authorized SMS Distributor for DEC Products:
Midwest: First Computer Corporation (312) 920-1050
Western: Pnambic Systems, Inc. (206) 282-0199

SMS SALES OFFICES:
Seattle, Washington (206) 883-8303
Boston, MA (617) 246-2540; **Atlanta, Georgia** (404) 296-2029; **Morton Grove, Illinois** (312) 966-2711; **Yorba Linda, California** (714) 993-3768.

*Trademark of Digital Equipment Corporation **Trademark of S & H Computer

CIRCLE D217 ON READER CARD

```
000644   ASL    R0
000646   MOV    (SP)+,COMB-10(R0)
000652   MOV    (SP)+,COMB-6(R0)
```

Example 3 shows the code generated by an FIS inline compiler. FIS code required the same number of statements to be generated. Notice that the single precision arithmetic operations implemented in EIS code by a subroutine call, such as JSR PC,$ADDF, have been replaced with simple FIS instructions like FADD. This is the advantage of FIS code where the FADD instruction is executed immediately by the hardware unit.

Example 3. FIS Code

```
000536   MOV    AN+2,-(SP)
000542   MOV    AN,-(SP)
000546   CLR    -(SP)
000550   MOV    #40200,-(SP)
000554   FSUB   SP
000556   CLR    -(SP)
000560   MOV    #40200,-(SP)
000564   MOV    AI+2,-(SP)
000570   MOV    AI,-(SP)
000574   FADD   SP
000576   FDIV   SP
000600   MOV    L,R0
000604   ASL    R0
000606   ASL    R0
000610   MOV    COMB-2(R0),-(SP)
000614   MOV    COMB-4(R0),-(SP)
000620   FMUL   SP
000622   MOV    L,R0
000626   ASL    R0
000630   ASL    R0
000632   MOV    (SP)+,COMB-10(R0)
000636   MOV    (SP)+,COMB-6(R0)
```

Example 4 is the generated code from the RSX-11M FORTRAN 77 compiler. It shows the benefits of inline FPU code. The code is small, and no subroutine or threaded library routine calls are necessary.

Example 4. FPU Code

```
000472   LDF    #40200,F0
000476   LDF    AN,F1
000502   SUBF   F0,F1
000504   ADDF   AI,F0
000510   DIVF   F0,F1
000512   MOV    R2,R0
000514   ASL    R0
000516   ASL    R0
000520   MULF   COMB-4(R0),F1
000524   STF    F1,COMB-10(R0)
```

GROUND RULES

Each of the five test programs was built a variety of ways to illustrate the execution speed of the different types of code. The first set of tests were all run on the same PDP 11/45. Since the 11/45 has EIS and a floating point processor, tests could be run with NHD threaded code, inline EIS code and library, threaded code and the FPU library, and inline EIS code linked with the FPU library. We call this option EPU. The 11/45 does not support FIS, so another machine was used to demonstrate FIS timing.

The programs were compiled under the FORTRAN IV/RT V2.5 compiler patched through Autopatch F and run under RT-11 V4.0 patched through Autopatch F. The routines were all compiled without line numbers to increase execution times. Line numbers are of great value when a program aborts, but add overhead in time and size to a working program.

The same programs were also run on the PDP 11/45 under the TSX Plus system available from S&H Computer Systems, Inc. TSX Plus provides a multi-user RT-11 environment. The routines were then rebuilt using FORTRAN IV V2.5 under RSX-11M V4.0. This was done to get a comparison of RT-11 versus RSX-11M run times. It is interesting to note that the FORTRAN IV compiler under RT-11 and the FORTRAN IV compiler under RSX-11M produced identical code. The routines were also built and run using FORTRAN 77 under RSX-11M. This was done to see if FORTRAN 77 is indeed faster than FORTRAN IV since it does produce inline FPU code. The routines were recompiled and run on a VAX 11/730 under VMS for comparison with RT-11 and RSX.

The routines were compiled with no line numbers, the test programs were the only job running, and no print spoolers, system jobs, other users, or additional processing were taking place. None of the programs did any I/O during the timing period.

The programs were also run on an LSI-11/2 to illustrate the difference between the NHD, EIS and FIS options. Two different LSI-11/23s were used to show the relative speed of the FPF-11 floating point unit compared to the KEF11-AA microcode FPU chip. The programs were also run on an 11/34 and an 11/44 with floating point processors, just for fun.

Table 2 shows the three character names used to identify the different options for the timing test results.

TABLE 2. Naming Convention

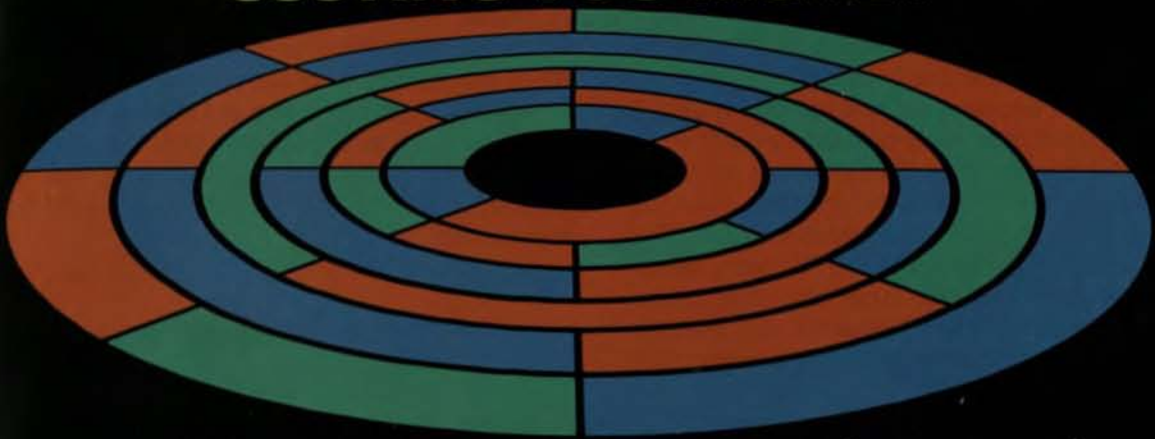
NHD	— Threaded code compiler, NHD library
EIS	— Inline EIS compiler, EIS library
FIS	— Inline FIS compiler, FIS library
FPU	— Threaded code compiler, FPU library
EPU	— Inline EIS compiler, FPU library
F4P	— RSX-11M F77 compiler with /F4P
F77	— RSX-11M F77 compiler and library

SINGLE AND DOUBLE PRECISION RESULTS

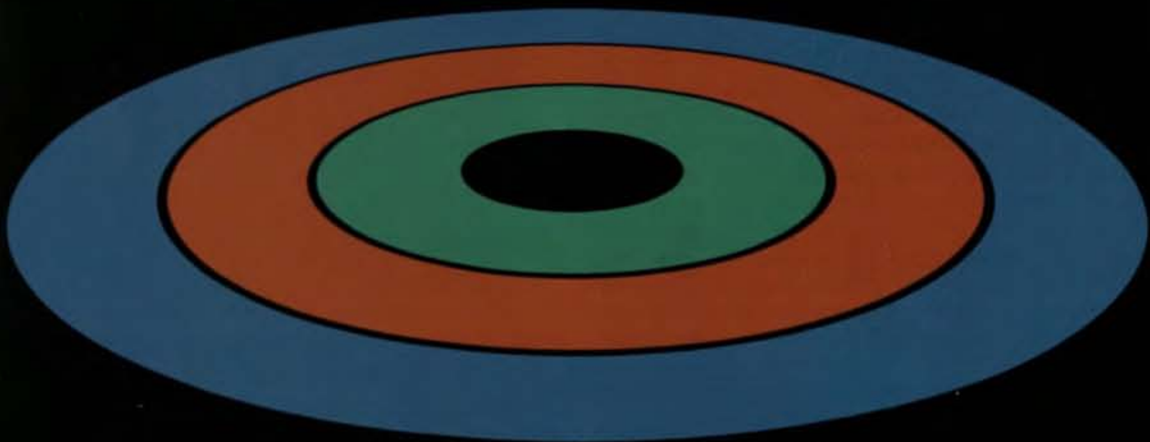
Tables 3 and 4 give the run times of programs CCA1 and CCA2 built for the various options. CCA1 and CCA2 are identical, except that CCA1 has single precision variables and CCA2 has all double precision variables. CCA1 and CCA2 do a diverse set of operations such as addition, multiplication, array referencing, trigonometric functions, and exponentiation as shown by the program sample given earlier. The other programs are much more specialized. Since none

FOR V8.0

WHAT YOU DON'T KNOW ABOUT RSTS/E DISKS IS COSTING YOU MONEY



If your disk looks like this, you're wasting system performance.



If your disk looks like this, you're using DISKIT.

Vhen the job you're running requires reading the red" file, it naturally happens faster on a well-rdered disk. Disks become "fragmented" as you se your computer. The system slows down. And hat costs you money.

Iow, you can restructure your disks and get back hat lost performance (up to 50%) without spending dime on new hardware. DISKIT is the original oftware system that makes this possible.

ut don't confuse DISKIT with other system tilities, DISKIT is a complete "software tool kit" hat optimizes your RSTS/E system.

DISKIT is:

- DSU — The utility which restructures the information on your disk, making data fast and easy to access.
- DIR — The incredible directory tool that finds files at the rate of 400 per second.
- RDR — Reorders disk directories 30 times faster than ever before possible.
- OPEN — Displays complete job statistics and file activity so you can see what your system is doing.
- DUS — The set of CALLable subroutines which pre-extend file directories, reducing fragmentation.

In today's tight economy, it's more important than ever to get the most out of your hardware investment. Call or write today and start getting your money's worth from your computer.

Software Techniques Incorporated

CIRCLE D193 ON READER CARD

5242 Katella Avenue
Los Alamitos, CA 90720
United States
Phone: (714) 995-0533

Software Techniques Limited
Bank Chambers
10 Long Street, Tetbury
Gloucestershire, GL8 8AQ
England
Phone: Tetbury (0666) 53995

of the programs does any I/O, such as reading data from a file, they may not seem representative of the average user program. Their purpose, however, is to illustrate the different types of code which primarily affect floating point arithmetic operations.

The CCA1 and CCA2 run times indicate that the more arithmetic hardware available the better. For single precision, the floating point processor reduced run time from 5.8 minutes to 1.4 minutes over using no hardware and from 4.5 minutes to 1.4 minutes over EIS. On double precision, CCA2, the reduction was even more dramatic from 26 minutes NHD to 18.6 with EIS, to 2.0 minutes with EPU. The EPU option, EIS inline code linked with an FPU library, seemed to produce the best results. It takes advantage of the fastest inline code available to FORTRAN IV, plus the FPU library. Inline code, however, often produces a larger .SAV image. For applications which do a large number of double precision calculations, buy a floating point processor!

TABLE 3. CCA1 Times on PDP 11/45

		Single Precision (minutes)	
	RT	TSX	RSX
NHD	5.82	6.37	6.96
EIS	4.47	4.85	5.30
FPU	1.62	1.75	1.88
EPU	1.42	1.53	1.64
F4P	—	—	.69
F77	—	—	.69

TABLE 4. CCA2 Times on PDP 11/45

		Double Precision (minutes)	
	RT	TSX	RSX
NHD	25.98	28.32	30.94
EIS	18.65	20.28	22.15
FPU	2.18	2.37	2.50
EPU	2.00	2.17	2.30
F4P	—	—	.90
F77	—	—	.90

RT-11 VERSUS TSX PLUS AND RSX-11M

Tables 3 and 4 also show that both TSX Plus and RSX impose some overhead on program execution. This increase in run time should be expected in a multi-tasking system where additional overhead must be incurred to control multiple jobs. The advantage of a multi-tasking system is that while one job is waiting for I/O, another job can be executing, so the total run time for the two jobs can be less than for running them consecutively.

Table 5 shows the size of the .SAV and .TSK files from program CCA1 for both RT-11 and RSX-11M. In both systems the inline code versions, CCA1.EIS and CCA1.EPU, produce slightly more code but run a little faster than the equivalent threaded code version, CCA1.FPU.

While TSX Plus can execute the same .SAV files as RT-11, the RSX .TSK files were appreciably larger than the RT-11 images. The more sophisticated, relocatable nature of a multi-tasking system such as RSX caused the .TSK files to be almost twice as large as the .SAV image. Remember that the compiler produced identical code, but the library routines under RSX must be smarter to run in that kind of environment.

TABLE 5. .SAV Versus .TSK File Sizes

	RT-11		RSX-11M	
	.SAV (blks)	Time (min)	.TSK (blks)	Time (min)
CCA1.NHD	28	5.82	43	6.96
CCA1.EIS	28	4.47	44	5.30
CCA1.FPU	25	1.62	41	1.88
CCA1.EPU	26	1.42	42	1.64
CCA1.F4P	—	—	44	.69
CCA1.F77	—	—	44	.69

FORTRAN 77

Notice that the F4P and F77 execution times under RSX-11M are twice as fast as the best that the RT-11 FORTRAN IV compiler can do. This is the advantage of the F77 compiler which produces highly optimized, inline FPU code. Table 5 shows that the size of the .TSK files increased slightly over the equivalent .TSK file produced by the FORTRAN IV compiler, but the F77 versions ran much faster.

TRIG FUNCTIONS, BINOMIAL, AND IF TEST

Tables 6, 7, and 8 show the results of the single precision trig function test (CCA5), the binomial expansion (CCA9), and the IF test (CCA10). Table 6 shows that FORTRAN IV's implementation of trigonometric functions take full advantage of the floating point hardware. Table 7 shows several interesting results. For some strange reason CCA9 blew up under TSX Plus even though the same exact .SAV file ran fine under RT-11. SPR time for S&H Computer Systems. CCA9 also aborted under RSX when the floating point hardware was NOT used. SPR time again. CCA9 is also a somewhat rare example since the threaded/FPU code was actually faster than the inline EIS/FPU code. It just depends on what the program is doing. Sometimes it is best to try the program both ways and use whichever is most effective.

CCA10, the IF test shown in Table 8, illustrates that inline code can be much faster than threaded code. It also shows that having a floating point processor does not help when the program does not perform any floating point arithmetic. In this case EIS is as fast as EPU.

TABLE 6. CCA5 Times on PDP 11/45

	RT	Functions (minutes) TSX	RSX
NHD	17.67	19.33	21.13
EIS	13.20	14.37	15.65
FPU	1.93	2.07	2.14
EPU	1.67	1.75	1.81
F4P	—	—	1.87
F77	—	—	1.87

TABLE 7. CCA9 Times on PDP 11/45

	RT	Binomial (minutes) TSX	RSX
NHD	16.03	17.58	?
EIS	12.70	13.58	?
FPU	3.58	?	4.08
EPU	4.48	?	5.17
F4P	—	—	1.59
F77	—	—	1.59

? means that the program aborted

TABLE 8. CCA10 Times on PDP 11/45

	RT	IF test (minutes) TSX	RSX
NHD	5.35	5.90	6.37
EIS	1.52	1.67	1.81
FPU	5.35	5.90	6.37
EPU	1.52	1.67	1.81
F4P	—	—	1.87
F77	—	—	1.87

EIS VERSUS FIS

Tables 9 and 10 show the results of running CCA1 and CCA2 on an LSI 11/2 which has both EIS and FIS. For single precision arithmetic, the four FIS instructions decreased execution time from 15 minutes with EIS to 4.8 minutes with FIS. On double precision, Table 10, the FIS instructions offered no speed increase over EIS. This is due to the fact that FIS instructions are only implemented for single precision arithmetic. Tables 11, 12, and 13 also point out that the FIS option can be advantageous when single precision arithmetic is performed.

. . . continued on page 60

LETTERS TO THE DEC PRO . . .

. . . from page 6

vould like to see this minor change.
Thank you.

Gregory N. Brooks
Research Electronics Technician
Washington University
St. Louis, Missouri

or the past 10 years I have been attend-ng DECUS symposia and have known hat you are telling us here: Symposia e for work, the hours are long and e rewards are great. Visiting amorous places is NOT what DECUS all about.

Our complaints on the sites selected e based on the availability for ALL the EC community, not just those with rge travel budgets. We have asked for ECUS meetings in places where there e large numbers of DEC users, some f whom are not able to attend meet-gs in far away places. We have never omplained about DECUS not being in fun" cities.

How can a college student in altimore, or a programmer in Boston tend a DECUS in the next five years ithout a major expenditure. DECUS is r everyone and we ought to make re more people can attend.

Our criticism of DECUS will continue ntil it becomes more responsive to the timated 200,000 DEC users. Even the ggest symposia draws a small part, ss than 3 percent of this total, so there much room for improvement. More ers will mean better DECUS meetings nd that's what we would like to see.

* * *

Mr. Marbach's article, "CRT Ter-inals on DEC Computers" (Novem-er 1983), was quite interesting, but I ust take exception to his statement, Unfortunately, there are no stan-ards for the control of CRT screens." e, like some CRT manufacturers DEC probably included), apparently as never heard of ANSI Standard 3.64. ANSI X3.64 "provides . . . con-ols to facilitate the operation of two-mensional input-output devices ch as character-imaging cathode ray tube devices . . ." and was published in 1979.

While it is true that there is no compatibility among many manufacturers' products, one has only to look at the advertisements for compatible CRTs to realize that the DEC VT-100 protocol is certainly not "the only standard that exists . . ." any more than any other single manufacturer's protocol is.

The fact is that there does exist a standard and many CRT manufacturers support it. What is unfortunate is that more do not.

Robert C. McGill
Project Manager
Houston Engineering
Research Corporation
Houston, Texas

* * *

The article "Digital's New Personal Computers" (September, 1983) reads a bit like sales literature. The plain fact is Digital's entry into the personal computer marketplace has been halting and largely disappointing. My employer acquired a bunch of DECmate I machines for electronic mail applications. All the secretaries who use them have had very nasty things to say about them. They are totally incompatible with anything else on the planet, including DEC's own Rainbow. Why any company would build a major product around the antique PDP 8 architecture is beyond me. I guess sentiment counts for a lot up in Massachusetts. As word processors they are very poor in comparison to all the other makes we have in house (Xerox, CPT, Lanier, Compucor) and as a personal computer the OS 78 operating system and its strange BASIC are interesting museum pieces, but hardly anything you would trade in your Apple for.

We also acquired a DEC Rainbow in the mistaken belief that it was an IBM PC look alike. No such luck. It won't read IBM disks and it won't format its own. My friendly DEC peddler has delivered two copies of a bootleg disk formatter, neither of which would work. The MS-DOS was delivered months after the machine was installed and we are still waiting for the add-on disk drives. Nobody's perfect in this marketplace but if you are competing with IBM and Tandy (to name two) you had better take better care of your customers than that.

We have no experience with the Professional 350 and aren't likely to gain any. Again, a proprietary and aging architecture (PDP 11 in this case) was chosen in preference to the de facto standard MC68000. It looks like DEC is only interested in selling to its installed base. In the PC world you either set standards or you follow them. DEC's offerings weren't strong enough to set standards, and they chose not to run with the pack. Only the overall strength of the company will keep the PC effort alive. If I were DEC I would start all over again with a machine like the Fortune:MC68000, built in hard disk, UNIX disguised behind user friendly menus, lots of business software.

We are not DEChaters; indeed our PDP 11/24 does what we bought it for without complaint (well, without many complaints) from the users. But in the PC world DEC will soon fall behind HP, and they stumbled around too. But their new touch-screen device is just the kind of innovation that will make a product take off. Bye, bye, DEC.

John R. Culleton, Jr.
Sykesville, Maryland

While we agree that DEC's entry into the personal computer marketplace has not been smooth, we don't agree with some of your other ideas.

Most of the people we know love the DECmate word processing. This fact has been proved again and again by Word-11 (PDP and VAX) and WPS-80 (Rainbow) software products, which emulate the DECmate word processing software. No one ever represented the Rainbow as a PC compatible. It will run

most of the popular software packages used today, Wordstar, LOTUS 1-2-3, dBASE, and others.

The value of the Professional Series is the fact that it is a PDP-11 for which there is 12 years of software. The 68000 is a transitional product, not long for this world, soon to be replaced by a true 32-bit processor. If DEC starts all over again, it will be with the Micro-VAX.

* * *

Thank you for your generous assistance in running an advertisement for The Computer Museum in the November issue of THE DEC* PROFESSIONAL.

It came out beautifully, and is generating a good response — both in direct sales and reader inquires. As a non-profit institution, such support is the Museum's lifeline. We appreciate your help, and hope to call on you again later this year.

In the meantime, we wish a joyous and prosperous new year.

Stephanie Haack

Communications Coordinator
The Computer Museum
Marlboro, Massachusetts

* * *

This letter is written in response your article on the VT200 series minals recently introduced by We have been evaluating a VT24 a couple of weeks and would li share some observations.

First, the VT240 is NOT necess compatible with software writte the VT125 and the 240 — Datatrie a good example. Due to the differe in implementation of REGIS betwe the VT125 and the 240, Datatri graphics will not work until Datatri V3.0, due to be released in the Su mer of '84.

VMS (Version 3.5) does N understand the VT200 identifica answerback and classes the de type as "unknown." This will not fixed until VMS V4. Another rela problem is that in VT100 mode answers back as a VT240 — not a VT100. This will break any softw which attempts to parse the answ back message.

The Tektronix 4010 implementa has at least two bugs — the "top li of plot will be lost, and only 24 lines alphanumeric characters ca represented (the 4010 has 33).

Other "features" include the existence of the Escape, Backsp and Line Feed keys in VT200 mo They are only present in VT100 em tion mode and then are located in function key area, rather than in main keypad area of the keybo While these function keys are and may be downloaded from host, they are NOT saved in a volatile memory.

The documentation shipped the terminal no longer includes a grammer's Reference Manual. If want to know the details of the es sequences, 8-bit character set, o of the new features, you must this manual from the Installed Group (A&SG) at a price of $20.

All of these comments have made to DEC to local, regional product management levels. response generally has been tha will have an ECO available in the mer to fix the 4010 emulation everything else is a feature! We a tremely disappointed with this line of terminals, especially evaluating other vendors' offeri he VT100/4010 compatible narketplace.

Almon T. Sorrell, Sr. Engineer
Westinghouse Defense &
Electronics Center
VAX Support Group
P O Box 746 MS 1615
Baltimore, MD 21203

* * *

I was quite surprised to find Frank 3orger's article complaining about the ıbundance of incorrectly addressed nail he receives (p. 58, January, 1984). \pparently he is unaware of recent esearch supporting the idea that mailng address labels are actually the owest form of sentient data strings yet ·ncountered.

For those unfamiliar with these enities; a sentient data string is a selfcontained collection of computerized lata which exhibits any kind of basic ınd independent intelligence. To late, intelligence has been limited to imple survival. However, mailing list lata have shown an unusual creativity or survival which hints at a larger ınderlying intelligence in these ap- ›arently simple entities.

A nascent mailing list data string is atisfied with simple transposition or ›mission of letters, as in Mr. Frank 3rog from Mr. Frank Borger. The obect here is survival through numbers. reate as many offspring as possible, nd some are bound to survive. Next ve have the sex change (e.g. for Mr. to As.). This is an easy way to double all ffspring (triple if you include Mrs.).

Until recently, this seemed to be the mit of the average mailing list data tring's survival instincts. However, ately we have seen two divergent trategies begin to develop.

First we have the "get lost in a rowd" syndrome. Have you ever oticed the staggering number of John Does? For that matter, recent studies how that letters addressed to Resident nd Occupant far exceed the number f households.

We call the second strategy the god" syndrome. That is, create a title vhich nobody in his right mind would elete from a mailing list. Popular ariations include President and Chief r Head of something. Mr. Borger's ewly acquired title of Father indicates ıat these data strings have achieved a asic understanding of the importance of religion in our society. Who's going to delete a priest from a mailing list?

In closing, it seems to me that this society of mailing list labels is growing in intelligence and influence at an exponential rate. I believe they are at a cusp in their development, ready to move beyond simple survival. Naturally I cannot prove this, but I wouldn't be surprised if, in the near future, we had a President who was addressed as "Occupant or Current Resident of 1600 Pennsylvania Avenue."

Bradley S. Albom
COMSOL
The Computer Solution Company
P O Box 206
Pt. Richmond, CA 94807

FORTRAN/RT TUTORIAL . . .

. . . continued from page 54

DIFFERENT MACHINES

Tables 9 through 13 detail the execution times of the same programs run on an LSI 11/2 with both EIS and FIS, an 11/23 with EIS and an FPF11 floating point processor, an 11/34, 11/45, and an 11/44 all with an FP11. The 11/23 appears to be about twice as fast as the 11/2 for every test. One exception is double precision. Since the 11/2 does not support a floating point processor, it cannot take advantage of any double precision arithmetic. It is interesting to note that an 11/2 with FIS can execute single precision arithmetic almost as fast as an 11/23 with the KEF11-AA microcode floating point chip — for CCA1 it took 4.8 minutes for the 11/2 versus 4.1 minutes for the 11/23 (Tables 9 and 14). Many people have been dismayed when they upgraded their 11/2 with FIS to an 11/23 and the single precision run times were only minimally decreased.

The execution times of the various machines show their relative power. No big surprises. The tests were run on the 11/23 and 11/34 because they are two of the more common systems currently in use and relate to a large community of users. The test results show that the 11/44 was twice the speed of the 11/34 on the NHD and EIS tests and almost that much faster on the floating point tests.

TABLE 9. CCA1 — Various Processors

	Single Precision				
	11/2	11/23	11/34	11/45	11/44
NHD	18.47	8.12	7.93	5.82	3.65
EIS	15.43	7.13	5.63	4.47	2.75
FIS	4.80	—	—	—	
FPU	—	2.07	1.78	1.62	0.98
EPU	—	1.87	1.58	1.42	0.93

TABLE 10. CCA2 — Various Processors

	Double Precision				
	11/2	11/23	11/34	11/45	11/44
NHD	83.07	35.25	34.05	25.98	15.47
EIS	57.00	26.42	22.90	18.65	10.60
FIS	57.00	—	—	—	—
FPU	—	2.72	2.38	2.18	1.42
EPU	—	2.55	2.20	2.00	1.30

KEF11-AA VERSUS FPF11

Table 14 points out the difference between the two floating point options available on the LSI 11/23. The KEF11-AA chip implements the floating point instructions in microcode. From Tables 9 and 14 it can be seen that the microcode chip can decrease processing time for CCA1 from seven minutes to four, and for CCA2 from 26 minutes to seven. That alone is a substantial savings. The FPF11 floating point processor implements the floating instruction set in hardware, and cuts the execution time of the KEF11-AA in half. Notice that the FPF11 did a much better job of executing the trig functions of CCA5.

TABLE 11. CCA5 — Various Processors

	Functions				
	11/2	11/23	11/34	11/45	11/44
NHD	60.03	25.07	25.48	17.67	11.90
EIS	49.63	21.67	17.40	13.20	8.63
FIS	14.42	—	—	—	—
FPU	—	2.90	2.73	1.93	1.88
EPU	—	2.55	2.43	1.67	1.73

TABLE 12. CCA9 — Various Processors

	Binomial				
	11/2	11/23	11/34	11/45	11/44
NHD	52.10	22.57	22.77	16.03	10.70
EIS	47.32	21.50	15.95	12.70	8.65
FIS	13.83	—	—	—	—
FPU	—	4.28	3.73	3.58	2.15
EPU	—	5.65	4.75	4.48	2.75

TABLE 13. CCA10 — Various Processors

	IF test				
	11/2	11/23	11/34	11/45	11/44
NHD	12.82	6.03	5.67	5.35	2.20
EIS	4.35	1.93	1.85	1.52	0.77
FIS	4.35	—	—	—	—
FPU	—	6.03	5.67	5.35	2.20
EPU	—	1.93	1.85	1.52	0.77

TABLE 14. KEF11-AA versus FPF11

	LSI 11/23 Floating Point	
Program	FPF11	KEF11-AA
CCA1.FPU	2.07	4.07
CCA1.EPU	1.87	3.85
CCA2.FPU	2.72	7.32
CCA2.EPU	2.55	7.17
CCA5.FPU	2.90	11.70
CCA5.EPU	2.55	11.33
CCA9.FPU	4.28	9.07
CCA9.EPU	5.65	10.45
CCA10.FPU	6.03	6.00
CCA10.EPU	1.93	1.95

THERE ARE AT LEAST EIGHT MORE TERMINALS HIDING IN YOUR VAX.

All you need is one Able VMZ/32 asynchronous terminal controller to find 8 more lines in your DEC VAX-11/730, /750 or /780, for a total of 16. With an additional VMZ/32 you can discover 16 more lines.

The VMZ/32 board plugs into the same space as the standard DEC DMF32 board it replaces, providing 16 lines instead of 8. Without patches to the VMS. And with complete VMS compatibility.

Immediately you'll unleash a minimum 13% increase in processing speed. You'll instantly realize a 50% reduction in costs per line, obtain a 50% space savings in the backplane and see a surprising reduction in power consumption.

The VMZ/32 also reveals hidden performance within your CPU by increasing its availability (idle time) up to 500%.

To handle this newly acquired power, the VMZ/32 gives you modem control on all sixteen lines. This lets you easily manage the terminals, including those remotely located. Plus, you can easily vary the speed on each line from a 50 to a 19.2K baud rate, including split baud capability.

Here's a dependability clue: an MTBF of over 100,000 hours. To back it up, the VMZ/32 comes with a one-year warranty from Able that can be extended if you like. Ask your Able representative for full details.

If you do unearth a problem, Able Support Service is as close as the phone.

You can also uncover extra performance hidden in your printer with Able's VMZ/LP. The VMZ/LP provides a dramatic reduction in CPU overhead and is LP11 cable compatible.

So, to discover the full potential hiding in your VAX, Able's got a simple solution: just plug in the VMZ/32.

ABLE COMPUTER

The communications specialists.

1732 Reynolds Avenue, Irvine, California 92714. Call toll free: 800-332-2253. In the Irvine area: (714) 979-7030. Or, TWX: 910-595-1729

VAX 11/730

The test routines were also run on a VAX 11/730. Table 15 shows the execution times of two different 11/730 systems, one with the FP730 floating point processor and one without. These times are listed along with the comparable times from the RSX-11M FORTRAN 77 run times on the 11/45 and 11/44. VAX times are more accurately compared with the RSX times rather than the RT-11 times since VMS supports FORTRAN 77. FORTRAN 77 has been shown to be substantially faster than FORTRAN IV running on the same machine (Tables 3 through 8). The 11/730 appears to be about the same speed as the 11/44, and has the advantages of the VMS operating system.

TABLE 15. VAX 11/730 versus 11/45, 11/44

Program	11/45	11/44	11/730	11/730
	RSX/F77		no FP	FP730
CCA1	0.69	0.56	1.76	0.56
CCA2	0.90	0.74	3.51	0.89
CCA5	1.87	1.82	5.91	1.37
CCA9	1.59	1.30	6.73	0.99
CCA10	1.87	0.87	1.90	1.90

CONCLUSIONS

EIS — the Extended Instruction Set, FIS — the Floating Instruction Set, and FPU — the Floating Point Unit are hardware/software solutions to faster execution of integer, single precision, and double precision arithmetic operations. EIS is standard on most new CPUs, while FIS is only available on some older model processors. For integer operations, EIS is very useful. EIS inline code linked with a floating point library (EPU) can be quite effective in the RT-11 environment. Inline FPU code has been shown to have great potential in decreasing run times, but is not currently available to the RT-11 FORTRAN IV community.

Double precision arithmetic on all types of machines will take four times as long to execute as the same program doing single precision when the operations must be performed in software. From Table 9 CCA1 took 8.12 minutes on the 11/23 for NHD and 35.25 minutes for the double precision version, CCA2. For EIS the difference was 7[illegible] minutes for CCA1 and 26.42 minutes for CCA2. In contrast when the floating point operations could be performed directly by the hardware (EPU), the times only changed from 1.87 minutes for CCA1 to 2.55 minutes for CCA2.

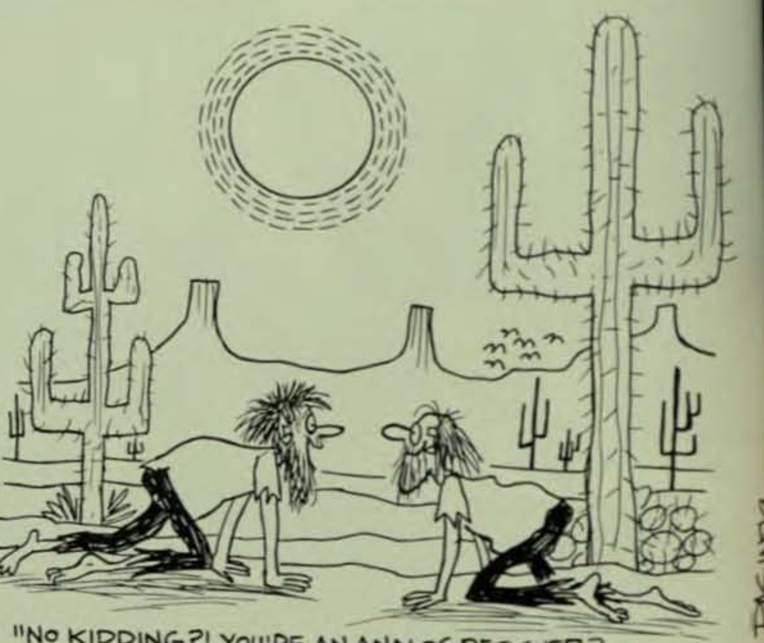
On an LSI 11/2, FIS can greatly speed up single precision operations by doing the arithmetic in hardware rather than software. This is reason the LSI 11/2 can perform single precision arithmetic almost as fast as an LSI 11/23 with the KEF11-AA chip which implements the 46 floating point instructions in microcode.

A floating point processor can improve the execution of double precision arithmetic by an order of magnitude over having to perform the same operations in software (Table 10 FPU and EPU versus NHD and EIS).

The software must be tailored to take maximum advantage of the hardware present on a given machine. The FORTRAN compiler can be built to generate the additional instructions implemented by the various types of hardware. Which type of hardware/software is best for a particular application depends on what type of calculations are to be done and what types of hardware are available.

In each application it is important to know what types of operations are to be executed, types of hardware available, and how to build the software to take maximum advantage of the hardware.

Frank Baginski, a cartoonist whose work appeared frequently in THE DEC PROFESSIONAL magazine, died suddenly January 1, 1984 at the age of 45. He will be greatly missed by his father and sister, all of us here at THE DEC PRO, and by our readers. Frank brightened our days by showing us how to laugh at ourselves. His gift lives on through his work.



Telematics gives you the key to more VAX from your VAX.

VAX systems are designed to handle data processing and computational tasks with a maximum of efficiency and flexibility. However, moderate data communications loads can keep VAX systems so occupied that main application functions can degrade significantly.

Off-load your communications overhead.

Now you can upgrade performance by off-loading data communications from the VAX system. And, do it in a manner that is transparent to applications programs running in the VAX. Results show a significant reduction in interrupt processing and VAX-Kernel mode execution when our VAX-FEP is added. A single FEP can communicate with multiple VAX's concurrently, augmenting DEC's cluster philosophy.

Communicate beyond the traditional DEC world.

With our FEP you can quadruple line connectivity for personal computers and support a mixture of synchronous and asynchronous terminals. Multiple X.25 network connections are supported with line speeds up to 64 kilobits per second. And, our library of frame level handlers provides an ideal solution for proprietary protocol implementation.

Productivity through programmability.

The Telematics VAX-FEP is fully programmable using high level languages such as Pascal and C. This provides the ability to off-load such non-productive functions as forms generation, data field validation and the access of static data bases. For these reasons our VAX-FEP is fast becoming the standard for OEM and systems houses implementing VAX based systems in data communications environments.

Why settle for a partial board solution when this performance and functionality is available at comparable prices.

For more information call 1-800-327-7944 or write Telematics International, Inc., Crown Center, 1415 NW 62nd Street, Ft. Lauderdale, Florida 33309. Telex: 510 955 4177.

In Europe write Telematics International, Ltd., Woodside House, Woodside Windsor, Berkshire SL4 2DP, United Kingdom. Or call (0344) 884644 Telex: 847840

TELEMATICS

DEC and VAX are registered trademarks of Digital Equipment Corporation

Although the American Samuel Morse is credited with the idea, the first practical telegraph based on current electricity was essentially the work of two Englishmen, W.F. Cooke and Professor Charles Wheatstone, in 1837. It was the needs of the expanding railway system which provided the social motive.

CIRCLE 0255 ON READER CARD

Prior to the "gasoline crisis," economy ca... popular as colds — those who had them didn't want them. A small car with an efficient engine was certainly not a status symbol. We wanted a car with air conditioning and automatic transmission, loaded with accessories. And, when all those goodies hanging off the engine hurt our acceleration, we bought bigger engines and larger gas tanks. If the cost of gasoline hadn't gone through the roof, there is no doubt that these trends would have continued.

We are hungry for bigger, more powerful machines. Nowhere is this more apparent than in the computer industry, where we outgrow computers long before their useful life expires. Sometimes we even outgrow them on paper, exceeding their capacity before they are ever installed.

Could we get more value out of our computers? I think so. I have seen installations where 16 users worked quite comfortably on a small system which, by industry consensus, could only handle six. American industry demands this type of economy from other capital investments, why not from the computer?

Often, our system performance problems take us by surprise. We planned for our current system to last five years (or more). But, having drunk from the bottle like Alice in Wonderland, we are suddenly, unexpectedly, too small. Too dependent on the computer to ignore the problem, we opt for a short-term solution and "throw money at it."

Sometimes there is a legitimate reason for declining system performance. Unexpected business growth can account for some computer overloads; doubling the number of

In the early days, when systems with the power of a pocket calculator occupied an entire room, users never came in contact with the machine. Instead, they submitted their jobs (usually a stack of cards) through a window where they could be retrieved by the System Operator.

The System Operator was often a computer buff who, although fascinated by the machine, didn't know enough about it to program it. He was the computer's eyes and ears. Relegated to the role of robot, he responded to console messages by loading endless stacks of cards, filling the printer with paper, and mounting various tapes and disk packs.

Those days are gone. The interactive terminal has replaced punched-card batch-processing, allowing users to load their own programs and get printouts from distributed printers near (if not on) their desks. Gone also is the System Operator, replaced by the System Manager.

Today's System Manager is a data processing professional, a vital part of his company's information management team. Freed from the tyranny of a hungry card reader, he faces the more important challenges of resource management.

IMPROVING SYSTEM PERFORMANCE

Foremost among these challenges is improving the performance of his system, a factor which the operator had no

We just made it easier

First computer corporation

- The leading integrator of exceptional computer systems
- A distributor of premier computer products and components
- A reliable source for technical expertise and dependable service
- Able to meet delivery schedules with products that work when they arrive

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 210 WESTMONT, IL

POSTAGE WILL BE PAID BY

First computer corporation
645 Blackhawk Drive
Westmont, Illinois 60559

12

Now, when time is of the essence, reach for our number and be assured of receiving First Computer Corporation's expert knowledge, superior product selection, reliable service and delivery.

First Computer is *your* problem solver. We expertly analyze your needs, listening and advising to understand your objectives. Then we configure a completely integrated system to be an investment in your future growth.

Your system has First's commitment to craftsmanship and our dedication to technical innovation behind it.

First Computer Corporation is *your* distributor for premier computer products. We warehouse a full line of systems, subsystems and components: terminals, printers, tape and disc drives and much more. Products of the highest quality available today. Products tested and re-tested to meet our high standards. Products delivered on time. Products that work when they arrive.

First Computer is *your* service contact. With our toll-free customer hot line and remote diagnostics capabilities, we take care of your systems and products. Service arrangements can be tailor-made to suit your needs.

So, when you need quality computer systems or products and time is important, reach for our number and call us First.

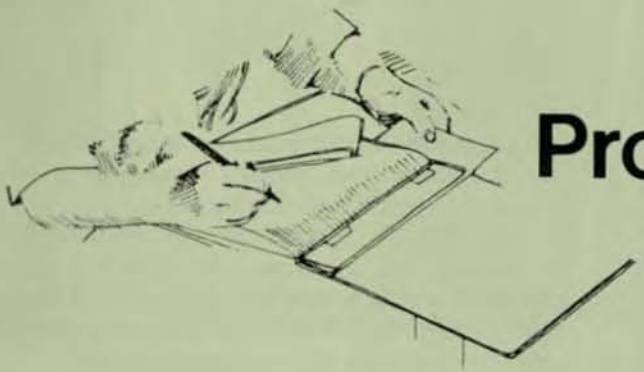
Tear off the end of the attached card and use it for quick reference when you call us First. Then complete the business-reply portion and mail to First today for more information.

Visit us at booth 930 at DEXPO® East 84.

Call us First toll-free 1-800-292-9000 (In Illinois 1-312-920-1050).
Business Hours: 7:30 am — 6:30 pm CST

CIRCLE D210 ON READER CARD

First computer corporation
645 blackhawk drive westmont, illinois 60559



Programmer's Notebook

SYSTEM PERFORMANCE MANAGEMENT:
New Software Tools That Make Cents

3y Rick Scherle, Software Techniques, Alamitos, California

In the United States, where hardware is cheap and echnology fashionable, the word "economy" too often neans "economy of scale." True economy, the careful or hrifty management of resources, is a rare commodity.

Prior to the "gasoline crisis," economy cars were as opular as colds — those who had them didn't want them. \ small car with an efficient engine was certainly not a tatus symbol. We wanted a car with air conditioning and utomatic transmission, loaded with accessories. And, when ll those goodies hanging off the engine hurt our accelera- ion, we bought bigger engines and larger gas tanks. If the ost of gasoline hadn't gone through the roof, there is no oubt that these trends would have continued.

We are hungry for bigger, more powerful machines. No- /here is this more apparent than in the computer industry, /here we outgrow computers long before their useful life xpires. Sometimes we even outgrow them on paper, ex- eeding their capacity before they are ever installed.

Could we get more value out of our computers? I think o. I have seen installations where 16 users worked quite omfortably on a small system which, by industry consen- us, could only handle six. American industry demands this /pe of economy from other capital investments, why not om the computer?

Often, our system performance problems take us by irprise. We planned for our current system to last five ears (or more). But, having drunk from the bottle like Alice Wonderland, we are suddenly, unexpectedly, too small. o dependent on the computer to ignore the problem, we ot for a short-term solution and "throw money at it."

Sometimes there is a legitimate reason for declining stem performance. Unexpected business growth can ac- unt for some computer overloads; doubling the number of customers or orders can put a dent in your system's performance. Some systems are overloaded by the "bandwagon" phenomenon. Once the new computer demonstrates the capability to improve productivity or solve organizational problems in one area, everyone in the organization wants access to the new tool.

But, by far the largest percentage of systems thought to be too small are simply suffering from inadequate management. For these systems a more economic solution is available and an experienced System Manager is the key.

THE SYSTEM MANAGER

In the early days, when systems with the power of a pocket calculator occupied an entire room, users never came in contact with the machine. Instead, they submitted their jobs (usually a stack of cards) through a window where they could be retrieved by the System Operator.

The System Operator was often a computer buff who, although fascinated by the machine, didn't know enough about it to program it. He was the computer's eyes and ears. Relegated to the role of robot, he responded to console messages by loading endless stacks of cards, filling the printer with paper, and mounting various tapes and disk packs.

Those days are gone. The interactive terminal has replaced punched-card batch-processing, allowing users to load their own programs and get printouts from distributed printers near (if not on) their desks. Gone also is the System Operator, replaced by the System Manager.

Today's System Manager is a data processing professional, a vital part of his company's information management team. Freed from the tyranny of a hungry card reader, he faces the more important challenges of resource management.

IMPROVING SYSTEM PERFORMANCE

Foremost among these challenges is improving the performance of his system, a factor which the operator had no

Programmer's Notebook

control over on earlier machines. In fact, there is an emerging awareness that systems management can make (or break) the system's performance.

The effective System Manager uses a broad knowledge of hardware and software in his quest for the optimal configuration. For example, will the terminals for this application perform better attached to DMA-transfer (DH11) or character-interrupt (DZ11) type interfaces?

He must also have a good understanding of the various software tools at his disposal. One of the most important of these is the RSTS/E statistics option (not only powerful, but included for free). In addition, there are powerful new system management and performance enhancement tools available from various software companies.

Software Techniques has been a significant source of some of these new tools. DISKIT improves the performance of almost any system by restructuring the data on disks for faster access. And UTL, announced last month, makes system management easier by providing an integrated command console from which to control every aspect of system operation. Its DCL-like command language is easy to learn and a very fast online help command details all of the commands.

Another of the new breed of system management tools was announced last summer by Northwest Digital. RPM (for RSTS/E Performance Montior) gives the System Manager a comprehensive analysis of the system's performance. Through some software magic, RPM extends RSTS/E's built-in statistics to capture detailed information about the usage of the CPU and disks. It tracks I/O requests, file processor usage, and access to programs and data files during normal system operation, and reduces the data to English

USING PERFORMANCE MANAGEMENT TOOLS

In the RSTS/E environment, these new software tools are giving the System Manager access to information he has never had before, providing a "window" into the system. Some of these tools optimize the performance of various system operations. Others provide total online control of the system, creating a virtual command console — a cockpit from which to fly the machine.

WORKING WITH THE DEVELOPMENT TEAM

In addition to routine operational activities, input from the System Manager can be a valuable asset in planning for growth. For example, when a company is considering the purchase or development of additional applications, the

System Manager can often provide valuable input regarding performance of the proposed application. His familiarity with the day-to-day performance of his system can help him forecast potential bottlenecks, and avoid costly surprises.

Application developers in the mini/mainframe environment frequently do not consider overall machine efficiency as a goal. Instead, their attention is focused on development time and support costs. With budgets stressed by the high cost of programmers, the temptation is great to reduce development time by using application development tools. These general purpose tools consume tremendous machine resources, forcing the end user to buy a bigger and faster machine to run the completed application. The rationalization is often "memory is cheap."

While trading machine resources for programmer resources can be cost effective, especially when developing applications for single users or for a limited marketplace, there is no excuse for inefficiency. Rather than the result of a calculated tradeoff, it is more often the result of sloppy planning, a lack of resource budgeting, or reluctance to look for a more effective implementation method.

By taking the responsibility for overall system performance, the System Manager can help insure that the new development is a success without an unnecessarily large hardware investment. Using performance statistics from his machine, he can determine the resources available and present resource budgets to the developers.

The old adage that "programs will expand to fill the memory available" is only too true. Knowing that he is writing a system for the PDP-11/44, the programmer often assumes that he has the whole machine, instead of the small part of it not currently being used by other applications. Programming without a budget, like playing tennis without a net, is a sloppy affair.

MANAGING FOR EFFICIENCY

By generously applying a knowledge of hardware and software, along with a determination to maximize the company's investment, a good System Manager can often postpone the necessity of hardware purchases by at least several months and, sometimes, indefinitely.

New system management tools are helping the System Manager to do his job more effectively. By providing more comprehensive information about what is going on inside the machine, by optimizing operations (such as disk seeks), and by giving the manager better control of the system, these low-cost tools quickly pay for themselves in improved system performance, reduced hardware costs, and better planning. ■

The Wonderful World of UNIX

This month our UNIX editor discusses the nature of interaction with UNIX and describes some of the facilities of the C shell. UNIX is a trademark of Bell Laboratories.

SHE SELLS C SHELLS . . .

By Steve Holden, Department of Computer Science, University of Manchester, England

Well, here we all are again in the first column I'm writing after Christmas. I hope you all had a pleasant time and arrived safely in 1984. For those who are interested in UNIX without necessarily being converted or wanting to be, I write mainly about the C shell this issue, but I've been pushed in various directions lately so other subjects have also become relevant.

It's all to do with interaction. The C shell has excellent facilities for controlling multiple interactive jobs. Unfortunately, with the advent of such interactive and real-time operating systems as RSTS/E, RSX and TOPS-10 many people seem to feel that writing interactive programs requires no special techniques. Although this is true to a certain extent, the type and nature of the interaction can have important effects on user performance.

To put it simply, if your software doesn't interact properly, people won't use it because they'll be able to get more work done with somebody else's.

RANDOM SOFTWARE NOTES

Although this column isn't (yet) influential enough to have software companies sending in products for review, every now and then a new software goody appears at work. It's always fun to try new software out, and this month has yielded a couple of real finds.

Although there is often no professional support for the languages you can get on UNIX, having them around is certainly stimulating. I have recently installed an ICON compiler over here. ICON is a language designed by Ralph Griswold and his team at the University of Arizona, and I was interested in it because I had done a lot of string processing work in the past using the SNOBOL4 language, an earlier Griswold innovation.

The language has some interesting features, although it can be a bit mind-twisting when you first come across things like expressions which can have a **sequence** of resu[lts] rather than just one. I hope to write more about ICON i[n a] later issue.

Something else that's come my way in the last tw[o] months is the EMACS editor. Although I'm new to EMAC[S I] am a convert already. It's easier and quicker to use by f[ar] than **VI**, even though it appears to have the edge in run-tim[e] efficiency. The macro facilities will offer TECO freaks t[he] chance to learn an even trickier editor-programm[ing] language.

THE NATURE OF INTERACTION

Back in the dim and distant past, computing was d[one] by assembling huge decks of cards and stuffing th[em] through a reading machine. When the computer was fi[nish]ed you would get back a printed listing which (usually) t[old] you that you had made syntax errors in your progra[m or] otherwise fouled things up.

DEC had a lot to do with promoting interactive comp[ut]ing, where terminals were connected to the machine [by] serial lines and the operating system was written to g[ive] each user the illusion of sitting at the console of his o[wn] computer. Things are moving forward slowly into netw[ork]ing environments, which makes interaction more c[om]plicated for human beings by increasing the numbe[r of] possibilities.

So why is interactive computing better? In a mag[azine] dedicated to DEC users that may seem like a fatuous [ques]tion. But, when you think of the complexity we add t[o] software to handle interaction, perhaps we should be a[sking] ourselves under what circumstances is it worth w[riting] systems to be interactive.

I prefer interaction, however, because you can ob[serve] the computer system's behaviour directly. You get in[stant] feedback about whether or not you've done a job righ[t]

least in theory). This is not important for a month-end reporting job, although it's surprising how many systems are still around which tie up a RSTS/E terminal for ages while they grind out a report.

For example, I am writing this article on a PERQ running the PNX variant. (Editor's Note: For more information on PERQ see "The Wonderful World of UNIX: LOGIN" in Volume 2, Number 5, September 1983.) This PERQ also has a serial connection to MUCSVAX1, a VAX which runs 4.2 BSD. One of the windows on my screen is labeled "VT100 TO MUCSVax1" and it allows me to use the C shell even though it isn't implemented under PNX, because it emulates a terminal on the VAX line. I can capture the information in a file when it comes back from the VAX, and then put those examples into the article. Neat, huh?

Having all these windows is a nice feature. It means I can run several different tasks on both the PERQ and the VAX, and while the VAX is munching on a long job I can come back to this article and get on with it. This may be very convenient for a computer buff, but if you don't know what's going on how are you to take care of all the complexities of interacting with different jobs on different machines? This problem will become more complicated as the amount of networking and the number of distributed systems increase with time.

UNIX was designed so that convenient units of processing could be encapsulated in utilities which could all communicate with each other. Since many of the ideas were experimental, Ritchie and Thompson took the understandable course of first making them work and **then** (sometimes) making them work better. Unfortunately, several different groups are responsible for different and sometimes incompatible enhancements of the operating system, which can lead to the problems of limited portability discussed last time.

Another problem is the user's inability to deal with the systems concepts needed to build complex software systems. Most users want to use the computer as a tool — or, to put it more accurately, as a set of specialist tools for particular jobs. Each program a user buys gives his computer the ability to take on an increasing range of tasks. The user doesn't want to know how many interacting processes it takes to do his job. He is unimpressed by fancy command languages to do his job. He just wants it done.

This means that the applications programmer is going to need ever more power at his beck and call in order to be able to quickly put systems together. Interactive computing is desirable because of its ability to let several people share the data processing resource. Parallelism in computer systems is one of the major reasons why software becomes so expensive and difficult to design as it gets more complex.

The C shell represents an attempt to let the experienced UNIX user get the most out of an interactive terminal. A single-user system would be easier to design. Let's hope that

Data base management can be child's play.

It's that simple. Contel's RTFILE® lets you manage the most complex databases with single keystroke responses to menu options. That's because it's relational and interactive. You need no programming experience. Proven by more than 250 customers on PDP-11 and LSI-11 systems, RTFILE can create, expand, shrink, modify, sort, merge, display and print all your data files. And much more. You simply select the operation from the menu, touch a key and RTFILE does the rest.

It runs on RT-11*, RT-11 emulators (including UNIX†), supports the SHARE-eleven and TSX-Plus‡ timesharing systems, and the STAR-eleven local area networking system. So it's available to any DEC user, even DEC Professional PC users.

Get complete information on RTFILE as well as our full line of advanced performance software designed specifically for DEC users. Contact Contel, the big name in small systems software. Write Loretta Kane, Contel Information Systems, 4330 East-West Highway, Bethesda, MD 20814, or call 301/654-9120.

CONTEL

Now available for RSTS*, RSX*, VMS* and PRO/RT-11.

RTFILE registered T.M. of Contel Information Systems
*Registered T.M. of Digital Equipment Corp.
†Registered T.M. of AT&T
‡T.M. of S&H Computer Systems, Inc.

See us at DEXPO EAST 84
Booth #705

CIRCLE D275 ON READER CARD

as networks become more common we will start to see systems where the real smarts are at the other end of a fast (about 1Mbyte per second) channel, and we can all have single-user systems again.

Unfortunately, single-user does not mean single task. No parallelism would immediately rule out my nice windowing system, so perhaps we'll keep the complexity for now and just try to learn how to live with it.

THE C SHELL

The C shell, like the standard Bourne shell **sh**, gives the UNIX user the power to initiate several tasks simultaneously. Using the advanced terminal and job control features of Version 4 it allows much better coordination of the individual jobs. I'm going to concentrate on the command history and substitution features, which make it very easy to define common tasks and repeatedly execute them.

A nice feature of the C shell is that you can ask it to print the command number as a part of the prompt. You can see that after the initial startup guff the prompt "[1]:" appears. The C shell is asking us for input. The first command asks the shell to list all the variable values known to it.

```
login steve
Password:
Last login: Sun Jan 29 18:12:27 on tty09

Type "msgs" for information on benson plotter.
                                              AR
        +-----------------------------------------------------+
        |  Zenith assumed - use magic if at some other terminal |
        +-----------------------------------------------------+
        Current Task: /usr/staff/steve/Icon/Programs
[1]: set
argv    ()
cwd     /usr/staff/steve
history 20
home    /usr/staff/steve
oldpath . /usr/ucb /bin /usr/bin
path    (. /usr/staff/steve/Bin /usr/our /etc /usr/ucb /bin /usr/bin)
prompt  [!]:
shell   /bin/csh
status  0
term    zenith
user    steve
```

The empty list value for "argv" shows that **csh**, as the C shell program is called, was not given any positional parameters on the calling line. My current working directory is given as "cwd," and this is the same as "home," my home directory. The value of "history" specifies that the shell should retain the command line for all commands referenced in the last twenty commands. "Path" is the list of directories which the shell will search looking for a program to run when a command is entered.

The path feature has a major efficiency gain over the standard shell because when **csh** is started or the path changes, it examines the directories and builds a hashed list of all files indicated as executable. It then searches the hash tables in memory rather than the directories on disk.

The "prompt" variable shows you the string which th[e] shell interprets to prompt for commands — the shriek [i]s replaced by the command number in use. The "shell" variable tells me where my current shell program was load[-]ed from. The rest can be ignored for now.

The **echo** command can be used to print messages as [it] echoes its arguments on to the standard output file. Com[-]mand [2] demonstrates this by printing a friendly greeting. By referring to variables in the arguments you can prin[t] messages which include their values. The dollar sign is use[d] to indicate such a substitution. Command [3] prints out th[e] working and home directories.

```
[2]: echo Hello to all Dec Pro Readers.
Hello to all Dec Pro readers.
[3]: echo "Current:" $cwd ";  Home:" $home
Current: /usr/staff/steve ;  Home: /usr/staff/steve
```

I can now introduce the history command [4], whic[h] sends out the command history to the terminal. This list ca[n] get a bit too long with the "history" variable set as high a[s] 20. I'm thinking of reducing it soon.

```
[4]: history
     1  set
     2  echo Hello to all Dec Pro readers.
     3  echo "Current:" $cwd ";  Home:" $home
     4  history
[5]: !2
echo Hello to all Dec Pro readers.
Hello to all Dec Pro readers.
```

As you can see, the history list contains the characte[rs] that were typed in as a command rather than what the she[ll] saw after it had made the substitutions. The history featu[re] is useful if you want to repeat a command. For examp[le,] suppose I wanted to issue my friendly message again. Com[-]mand [5] causes the shell to use event [2] (the commands a[re] referred to as "events" by the C shell documentation) as th[e] source for a new command. Because I run in "verbose" mo[de] the command is printed out before the shell starts to e[x-]ecute it.

You will also notice from the output of command [5] that the history list saves that version of the comma[nd] rather than the one with shrieks in it. This is because som[e] history references can refer to events in a relative way, li[ke] "the last command but two" and this obviously won't mea[n] anything in the general case. This is done by using [a] negative value for the event number following the shrie[k.] The minus four of command [7] specified event three, wi[th] the results you see.

```
[6]: history
     1  set
     2  echo Hello to all Dec Pro readers.
     3  echo "Current:" $cwd ";  Home:" $home
     4  history
     5  echo Hello to all Dec Pro readers.
     6  history
[7]: !-4
echo "Current:" $cwd ";  Home:" $home
Current: /usr/staff/steve ;  Home: /usr/staff/steve
```

MS-DOS® & CP/M® ON A VAX®?*

*Or any other DEC™ system.

YES. You can get all the benefits of personal computing—without buying a personal computer.

Run all the powerful, friendly software packages that have made personal computing so desirable. Virtual Microsystems' total hardware/software solution puts personal computing on *any* DEC system, combining the flexibility and speed of microcomputing with the power and control of your DEC.

For more information on the system that's been proven in hundreds of sites, call (415) 841-9594 or send this coupon to:
Virtual Microsystems
2150 Shattuck Ave, Suite 720
Berkeley, CA 94704.

Tell me more.

Name ______________

Company ______________

Address ______________

City ______________

State ________ Zip ________

Telephone (   ) ______________

virtual microsystems

DP3-84

Registered trademarks: CP/M: Digital Research; MS-DOS: Microsoft Corp.; VAX: Digital Equipment Corp.;
Trademark: DEC: Digital Equipment Corp.

CIRCLE 0211 ON READER CARD

Wonderful World Of UNIX

Yet another useful feature of the C shell is the alias list. For RSTS users this will seem a bit like CCL commands (remember which version of RSTS introduced the CCL feature?). I can print my alias list as at command [8]. When a command line is entered and after various substitutions like those mentioned above have taken place the shell examines the alias list looking for the first word of the command. If a match is found then the specified substitution takes place. For example, command [9] shows how a single-word command can be used to obtain the long-format directory listing.

```
[8]: alias
bye      echo `pwd` >$HOME/Bin/.current; logout
current  cd `cat $HOME/Bin/.current`; cat $HOME/Bin/.current
dir      ls -l
keep     (date ; history ; echo '=================' ) >>~/General/histories
ownmail  mail -f mbox
printit  !-1 | lpr &
slower   !-1 | more
[9]: dir
total 53
drwxr-xr-x 2 steve        240 Jan 26 10:16 Bin
drwxr-xr-x 2 steve         80 Jan 25 17:43 Documentation
drwxr-xr-x 2 steve         96 Jan 16 10:53 Experiments
drwxr-xr-x 2 steve         96 Jan 25 17:47 General
drwxr-xr-x 4 steve        112 Jan 16 15:54 Icon
drwxr-xr-x 4 steve         64 Dec  3 15:52 Projects
drwxr-xr-x 5 steve         80 Dec  3 14:50 Writing
-rw-r--r-- 1 steve        343 Jan 26 12:10 macros
drwxrwxr-x 2 steve         48 Oct 15 14:17 mail
-rw-r--r-- 1 steve        769 Nov 16 14:49 mbox
-rwxr-xr-x 2 steve         65 Jan 25 17:47 note
-rw-rw-r-- 1 steve       1746 Jan 27 11:08 write.man
```

After a little housekeeping with command [10] (the **rm** command removes the referenced files) I introduce another C shell idea: the directory stack. The **pushd** command [11] pushes the current working directory down the list and moves to the specified directory, which command [12] takes a listing of. Command [13] goes a further level down the directory tree and [14] lists that directory. Notice how the stack is displayed after each **pushd** command.

```
[10]: rm emacs* macros write.man
[11]: pushd Icon
~/Icon ~
[12]: ls
Programs  Utils
[13]: pushd Programs
~/Icon/Programs ~/Icon ~
[14]: ls
echo   echo.icn   t   t.icn       trbld       trbld.icn
```

This command can be used to switch to any directory temporarily, and [15] through [18] show this feature in use. Command [18] shows how several commands which are to be run in sequence may be put on the same line with semicolons in between. Note that the tilde character (~) is used by the C shell to refer to the home directory of the logged-in user. That single character is certainly quicker than "/usr/staff/steve" or even "$home" which would otherwise be needed.

```
[15]: pushd /etc
/etc ~/Icon/Programs ~/Icon ~
[16]: dir
total 1206
-rwxr-xr-x 1 mgr        15360 Jul  9  1981 512restor
-rwxr-xr-x 1 mgr         9216 Jul  9  1981 ac
-r-xr-xr-x 1 mgr        28076 Oct 25  1982 acct
-rwxr-xr-x 1 mgr         5120 May 19  1981 accton
-rwxr-xr-x 1 mgr        17408 Jul  8  1981 analyze
-rwxr-xr-x 1 mgr         9158 Dec  2  1982 cron
      .
      .
      .
      .
      .
-rwxr-xr-x 1 mgr         4096 May 20  1981 update
-rw-r--r-- 1 mgr          240 Jan 29 18:13 utmp
-r-xr-xr-x 1 mgr          859 Sep  6  1982 uucollect
-rwxr-xr-x 1 mgr          219 May 20  1981 vipw
---------- 1 alasdair       0 Dec 13 14:59 vipw.lock
-rwxr-xr-x 1 mgr        10240 Mar 10  1982 vpac
-rwxr-xr-x 1 mgr        11370 Aug  3  1982 wall
[17]: popd
~/Icon/Programs ~/Icon ~
[18]: popd; popd
~/Icon ~
~
```

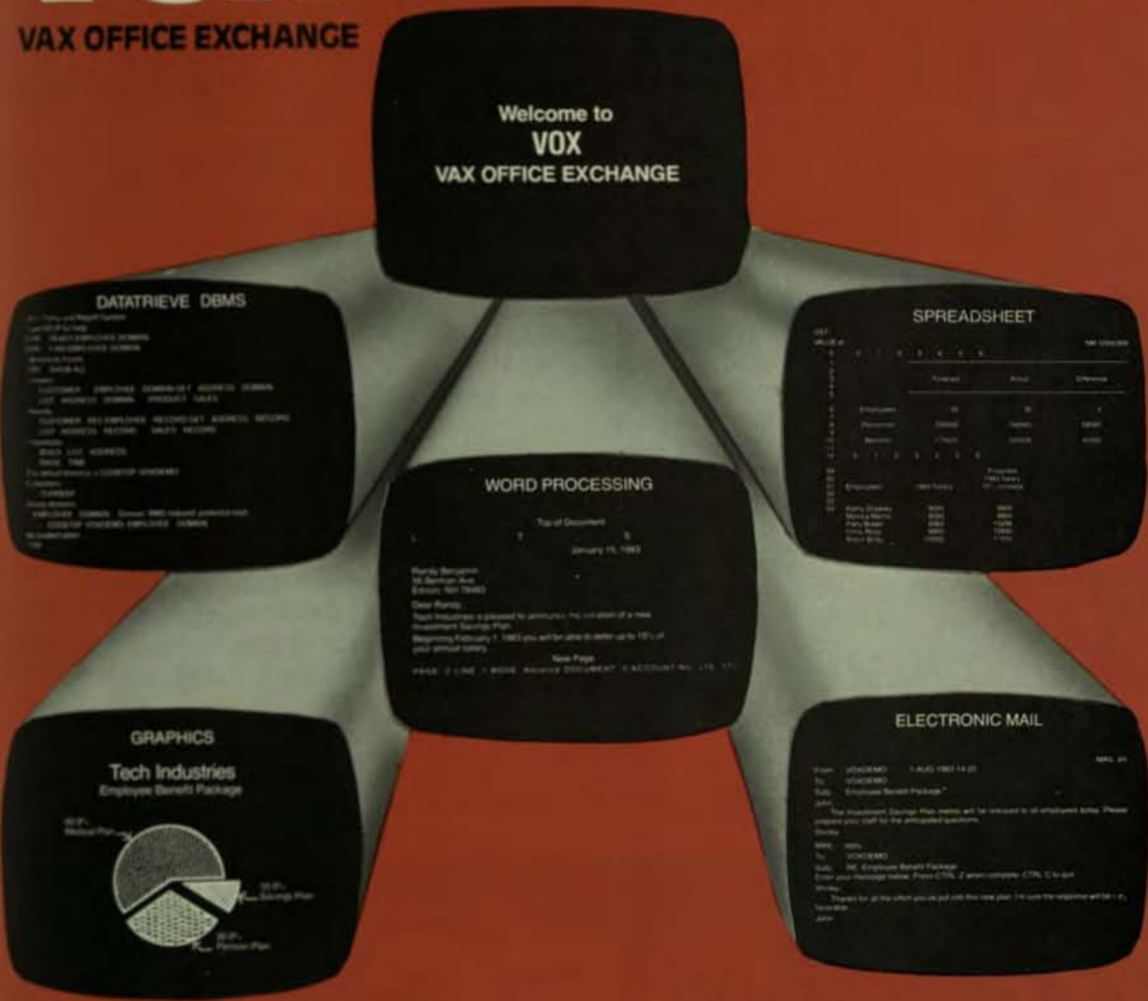
Command [19] shows the commands I have put in m ".login" file, which the C shell reads when activated at th start of an interactive session. You can see the visible effec of many of these commands in previous sections of th listing. The last echo command appends information to file, which I use to record the duration of my terminal se sions. I hope later to process these on the computer to giv me some statistics about the amount of time I spend at te minals and so on. The **msgs** command will tell me if I hav any mail, the -f option telling it to stay silent rather tha print a message if there is no mail. I have hardly begun t scratch the surface, but I hope you get the impression tha the C shell is an extremely flexible control mechanism. Com mand [20] was just me fixing a bad edit, and shows up wha used to be the least friendly part of UNIX — the edito Other screen-based systems such as **vi** were written to go o top of **ed**, but they were rather bulky and cumbersome.

```
[19]: cat .login
set term=zenith
stty dec crt
echo "  +-----------------------------------------------------------+"
echo "  |  Zenith assumed - use magic if at some other terminal |"
echo "  +-----------------------------------------------------------+"
echo " Current Task:" `cat $HOME/Bin/.current`
echo "$$ login" `date` >> ~steve/General/log
msgs -f
biff y[20]: ed .login
333
$p
msgs -f
a
biff y
.
w
334
q
```

The job control features are indicated at command [2 which calls the standard shell. It is sometimes convenient have shells running in different directories. The control causes the current job (the sh from [21]) to be stopped b not aborted. Command [22] starts a new shell which is sw ched into a different directory. That shell is also stopp The **jobs** command [23] tells me about the state of the jo under the shell control.

VOX THE AUTOMATED OFFICE

VAX OFFICE EXCHANGE



Make no mistake — the automated office is here now. But it will continue to evolve and expand as increasingly sophisticated information processing and exchange systems are developed. The VAX Office Exchange (VOX) is your link between the office automation functions you need today and those that will be essential to you in the years ahead.

VOX is not just a collection of software. VOX transforms strong software components into an interactive office system. And because VOX is modular, you can start with basic VOX capabilities and add others as you need them.

VAX, Datatrieve are trademarks of Digital Equipment Corp. / VOX is a trademark of Aquidneck Data Corp. and is developed jointly with Hamilton HGL Software Ltd.

WRITE OR CALL FOR FURTHER INFORMATION AND LITERATURE

HAMILTON

HGL Software   Hamilton Rentals

6 Pearl Court, Allendale, N.J. 07401
TOLL FREE 800-631-0298
In New Jersey 201-327-1444
In Dallas 214-980-1453

415 Horner Ave., Toronto, M8W4W3
TOLL FREE Ont. & Que. 800-268-2106
All other prov. 800-268-0317
In Canada 416-251-1166

NEW YORK • DALLAS • MONTREAL • CALGARY • LONDON • PARIS • DUSSELDORF

See us at DEXPO East '84, Booth No. 644

CIRCLE D130 ON READER CARD

```
[21]: sh
$ cd Icon/Programs
$ ^Z
Stopped
[22]: sh
$ cd Writing/Decpro
Writing/Decpro: bad directory
$ pwd
/usr/staff/steve
$ ls Writing
Decuser  Local    Serc
$ ls Local
Local not found
$ cd Writing/Local
$ ls
p.perq-u.vax
$ ^Z
Stopped
[23]: jobs
[1]  - Stopped                sh
[2]  + Stopped                sh
```

When a job is triggered to run asynchronously like the print spooling command [24], that too is recorded in the job tables. In this particular example **csh** was too quick for me and so what I actually got was a repeat of the previous job listing followed by a message informing me that one of my jobs had finished.

```
[24]: lpr Icon/Programs/echo.icn &
[3] 181
[25]: jobs
[1]  - Stopped                sh
[2]  + Stopped                sh
[3]    Done                   lpr Icon/Programs/echo.icn
```

Command [26] puts job number two into th foreground (i.e., makes it the job which gets to use the tty The command line which triggered the job (in this case sh) echoed and (since the last thing the shell did was to issue prompt for input) the job awaits input. Then this arrives the form of a **pwd** command, to prove I got the right jo back. I then terminated the shell and did the same thin with the other one at command [27]. Then I decided I'd pr bably confused you enough for one issue, and so I logged of

```
[26]: $2
sh
pwd
/usr/staff/steve/Writing/Local
$ ^D
[27]: $1
sh
pwd
/usr/staff/steve/Icon/Programs
$ ^D
[28]:logout
```

It's impossible to do the shell full justice in a short ar cle, so I hope this has served mainly to give you the flavo and encourage you to try it for yourself. If any further sh tips and kinks come up I'll pass them on.

Next issue I should like to talk about text editing und UNIX. It's a job we all have to do, and one which can be do in a staggering number of ways. Until then, keep control all your processes!

Computer Bookshelf

By Richard Low
and Amy Par

Telecommunications And Networking
By Udo W. Pooch, William H. Greene and Gary G. Moss
Little, Brown and Company.
386 Pages. $29.95

This book covers virtually all aspects of the telecommunications and networking field. Both the hardware and the software elements of networks are discussed in detail. Several network architectures are described and compared.

Part I contains introductory information about the basics of telecommunications, electronics and signals. It contains many complex equations which should be familiar to those with an electronics background.

Part II is concerned with telecommunications systems: modulators, demodulators, coaxial cable, microwave, satellites, fiber optics, voice channels, noise and distortion. I found it to be a very good, if somewhat hardware oriented, description of these items.

The core of this book is Part III, which describes the software components of networks. Some of the topics discussed include network design objectives and classification schemes, switching techniques, network topologies, routing algorithms, flow control, teleprocessing and protocols.

The book provides information o interest to the programmer/analy who wants to find out more about ne working and telecommunications co cepts. However, the book spends to much time detailing telecommunic tions hardware. I found the hardwa sections got in the way of an in-dept understanding of the software.

Applying Software Engineering Principles
By David Marca
Little, Brown and Company.
270 Pages. $27.00

As the title of this book implies is the author's intention to ident software engineering concepts a demonstrate how they can be appli in a practical setting. The author clai his book is written for advanced pr grammers, but his presentation material seems more suited for t first-year programming student. Co sider this excerpt from Part I, "Impo tant Aspects of Software Engineerin

> "The newly running software do not remain stable for long. Th you must be ready to cope wi change as soon as the new syste is installed. [CA80]. The process changing a software system aft installation is called 'maintenanc [PG82]."

This section is the weakest in th book. It is overburdened with u necessary references ([CA80], [PG82] the above) and the treatment of th topic is overly simplistic and supe ficial. It does however provide som background for the engineering tec niques to follow.

Parts II and III comprise the core the book, the nuts and bolts of sof ware engineering. In Part II, the su ject is the purpose and form of co puter language. Beginning progra mers are urged to master their chos language and use templates organize language statements. T author uses examples, in FORTRAN, discuss the rules and limitations of pr gramming languages. Also present are techniques for structuring data a algorithms to maximize the efficien of a program. In Part III the auth discusses the importance and role software development tools. T reader learns why subprograms a useful and how they can increase pr gram reliability and portability.

In Part IV, the final section, little done to excite the imagination of t reader. The author ties together co cepts and techniques discussed earl by asserting that such engineering co cerns as user interface, applicati logic, and data management are fu damental to software development.

Overall, there is not much wisdo here for the professional softwa developer. Clearly, the book is writt for students and may serve well as a auxillary text.

NIFTY THINGS TO DO WITH

RSX INDIRECT COMMAND FILES

By Allen A. Watson, The Record, Hackensack, NJ

Editor's Note: This installment of "Nifty Things To Do With Indirect Command Files" focuses on the "Case" Statements or Computed GOTOs, and Task Builds. Our continuing series on hints and tricks using RSX Indirect Command files began in Volume 2, Number 6 (November 1983).

MENU PROCESSING - SIMPLE COMPUTED GOTO

Frequently in writing a command file one wants to present a user with a list or menu of options and then, based on his response, branch to an appropriate routine. After a lot of complex experimentation, usually based on a long list of things like ".IF CHOICE = "A" .GOTO A", I arrived at a rather simple and elegant solution to this general problem. Here it is:

```
           .ENABLE SUBSTITUTION
.OPTION:
; Please enter one of the following options:
;          DELETE
;          CREATE
;            .
;            .
;          EXIT
.START:    .ASKS CMD Enter desired function:
           .ONERR NOSUCH   ! Trap for bad option !
           .GOSUB 'CMD'
           .GOTO START
.DELETE:   .ONERR       ! Reset error trap !
           .; ...body of code for delete ...
           .RETURN
.NOSUCH:   .OPEN TI:
           .DATA There is no such command as 'CMD'.
           .DATA Please check your entry against this
           .DATA list of valid options.
           .CLOSE
           .GOTO OPTION
```

There must be a subroutine labeled with the exact name of each option in the option list (upper or lower case makes no difference). If the operator types in "ADD", the command file will try to execute a ".GOSUB ADD" command. If there is no such label, the ONERR statement will cause a trap to "NOSUCH" which issues an error message and redisplays the valid choices.

An Alternate Approach

If you prefer using numbered options instead of named options, you could use basically the same approach, but instead of saying ".GOSUB 'CMD'" you would say ".GOSUB OPT'CMD'", and include labels such as ".OPT1:", ".OPT2:" and so on, in your file. You could then dispense with the ONERR statements by including a range check in your ASKN statement for the command number, e.g.,

.ASKN [1:9.] CMD Enter option number

This has the benefit of consistent labeling for the starting point of each option.

Using Multi-level Command Files Instead of Subroutines

If the functions you are performing are complex, you may want to write separate command files for each function to avoid having one huge command file, and to facilitate testing of new functions.

For example, we have a set of command files that invoke Datatrieve to maintain a list of disk packs in use. One file adds new packs, one deletes packs, one modifies the contents of any field, one just records a fresh copy date, one prints a report sorted by volume label, and another a report sorted by manufacturer.

ne main command file, called DISKLOG, does nothing esent a menu and, based on the operator's input, in- a sub-file to perform the requested function. It looks is:

```
     .ENABLE SUBSTITUTION
ON:
se enter one of the following option numbers:
     Log a pack copy.
     Enter a new pack.
     Remove a pack.
     Print current disk log.
     Display disk log on terminal.
     Print old disk report (unimplemented).
     Print disk report sorted by manufacturer.
     Modify any field for a pack.
     EXIT
RT:  .ASKN [1:9.] CMD Enter desired function:
     @DLG'CMD'
     .GOTO OPTION
```

Command files called DLG1.CMD, DLG2.CMD, up to 9.CMD, exist in the local UIC. (You could, of course, place n in another UIC and specify the UIC in the command , e.g., "@[1,5]DLG'CMD'".

I have found this type of shell interface between users programs to be one of the most useful things we have done with Indirect Command files.

TASK BUILD OPTIONS FOR INDIRECT

In the MCR Operations Manual, Chapter 4 (The Indirect Command Processor), page 4-4, there are two options mentioned for indirect command processing. Both have very interesting possibilities; I have implemented only one. The two options are: a default command file to be executed if none is given in the command line, and a default UIC to search if the command file is not in the current UIC. For both options, the manual says "To enable it, the value in the build file for the Indirect task must be changed." There is no mention, however, of where that file is located, what it is called, or what value to modify.

How to Select Them

You have two choices. In fact, three build files are created during SYSGEN by the SYSGEN command files for a standard Indirect and versions using FCSRES and FCSFSL. You can proceed with the PREPGEN until asked if you want to modify any files for the nonprivileged tasks. Answer yes. Then, during the actual SYSGEN, SYSGEN will pause to allow edits. At that time edit [1,24]ICMBLD.CMD (and/or ICMFSLBLD.CMD and ICMRESBLD.BMD). The symbol you

NIFTY THINGS

want to modify is D$CUIC for default UIC or DEFCML for th default command file. Comments in the file will tell you a you need to know. I used [1,5] to maintain consistency wit the KMS version of IND which we had been using under 3.2 (D$CUIC=00405 because 405 is the octal equivalent o ".byte 1,5"). You can choose any UIC you want to use; se the details on page 4-4 of the MCR manual.

The problem with this approach is that re-running th SYSGEN will create another copy of the ICMBLD.CMD fil and you may inadvertantly lose your modifications unle you remember to make them every time. So what I sugge is editing [1,20]ICMBLD.BLD. This is the file used b SGNBN.CMD to create ICMBLD.CMD and the other buil files. Edit this file so that when ICMBLD.CM ICMFSLBLD.CMD and ICMRESBLD.CMD are created, the already have the globals set the way you want them.

Finally, when you have re-taskbuilt Indirect, th distributed files will leave the task in [1,54], but if Indirect installed in your VMR command file, you will have to r VMR your pack to install the new version, and that com mand file installs the task from [3,54]. So, either move th new version to [3,54] before rerunning VMR, or just do remove and install in your startup command file.

The benefits of having a default UIC for Indirect t check are, basically, that you can create "global comma files" that are available for use by anyone on your syste The user does not have to know where the file is store s/he just types "@filename" and the command file e ecutes, no matter what UIC the user is in. Indirect w always look first in the user's default UIC, and only if no fi of that name is found, will look in the system UIC.

Having system-wide command files allows you to eas implement things like standard switches for compiles task builds, or mounts and dismounts; it lets you provide indirect command file library for things like parsi filespecs, and it lets you set up user interfaces to co plicated tasks like BRU.

Having a default command file that is execut whenever Indirect cannot locate the file called for also h interesting possibilities. You could do nice things like ma ing the default file list all of the command files in the cu rent UIC. You could even create a poor man's Command Li Interpreter by putting the user into an indirect comma file at the start (through LOGIN.CMD) and having him ty his commands into that indirect program. For example, if types "BOMB", your command file attempts to execute t command "@BOMB". If an invalid command were giv your default command file could display a prompt scree valid options and let him try again. You should start yo Command Interpreter file with ".ENABLE CONTROL-Z" prevent CTRL/Z from exiting the command file; this ke him from exiting to MCR. (The knowledgable user can exit by typing CTRL/C and aborting Indirect.)

Who Really Manages Your VAX Data?

ntroducing the DBMS hat Users can Use, Programmers can Program, and Managers can Manage.

Most data base users are merely bystanders. Because most data base management systems are so complex that only a few people can squeeze out results.

Now System 1032™ puts the power of the VAX on your side. You get relational data management, without the burden of relational calculus or SQL.

Anyone can use it with less than an hour of training. If you get stuck in the middle of a command, just touch a key. System 1032 prompts for essential information, list choices, or offers a capsule description of any feature.

But System 1032 doesn't compromise on the features that simplify your applications. Design using any host language or our built-in block-structured programming language. You can customize data bases and output formats to satisfy your most demanding users. Yet you can modify any data structures at any time.

Try it out on your own computer. A 60-day trial period costs just $85. See how easily System 1032 manages your own data, on your own system. Test our responsive hot-line support services. Watch your programmer productivity shoot up, and system efficiency accelerate.

Show your VAX who's in charge, with System 1032. Send in the coupon or call today.

System 1032 ™

- ☐ Please call me.
- ☐ Send me the latest on System 1032.

My VAX environment is:

- ☐ 11/782
- ☐ 11/780
- ☐ 11/750
- ☐ 11/730
- ☐ On order
- ☐ Under consideration
- ☐ My company is an OEM

Name ____________

Title ____________

Company ____________

Address ____________

City ____________

State/Prov ________ Zip/PC ______

Tel ____________ DP384

Send to:
Shauna Sullivan
Software House
1105 Massachusetts Avenue
Cambridge, Massachusetts 02138

See us at DEXPO
Booth No. 512

Software House

1105 Massachusetts Avenue
Cambridge, Massachusetts 02138
Tel. (617) 661-9440 TWX: 710-320-1075

DATA COMPRESSION TECHNIQUES

How to put ten pounds of stuff in a five pound bag!

By Steve Roy
New Haven, Connecticut

(Editor's Note: Steve Roy, president of Diversified Consulting in New Haven, Connecticut will cover the fine points of Data Compression in a series of articles. He here shows us the best way to compress floating-point data into its tightest possible form. Future installments will show how to manipulate compressed data, and how to put other types of data — lists, string, Y/N, etc. — into less space.)

```
1000    ! COMPRS - COMPRESS 8-BYTE (4-WORD) FLOATING-POINT TO A BINARY
        !       INTEGER (STORED IN A STRING OF SPECIFIED LENGTH<7) AND
        !       WITH A SPECIFIED NUMBER OF DECIMAL PLACES.
        !
        SUB COMPRS(IN.DEC,OUT.STR$,BYTES%,DECIMALS%,E%)
\       ON ERROR GO BACK
\       MINUS%=IN.DEC<0.                    ! REMEMBER SIGN
\       T=IN.DEC*10.^DECIMALS%              ! SCALE TO INTEGER
\       T=-1.-T IF MINUS%                   ! TWO'S COMPLEMENT
\       T=INT(T+.00001)                     ! CHOP OFF EXTRA
\       T$=""                               ! TEMP OUTPUT STRING
\       FOR I%=1% TO BYTES%                 ! FOR EACH OUTPUT BYTE
\               T1=INT(T/256.)              ! STRIP LOW BYTE
\               TBYTE%=T-T1*256.            ! REPLACE WITH NEW DECIMAL
\               T=T1
\               T$=CHR$(TBYTE% XOR MINUS%)+T$   ! TACK BYTE ONTO T$
\       NEXT I%                             ! NEXT BYTE (YUM, YUM)
\       E%=TBYTE%>127% OR T>0.              ! OVERFLOW (SHOULDN'T HAPPEN)
\       OUT.STR$=T$                         ! LOAD RESULT
\       SUBEND                              ! BYE, BYE
```

```
        .TITLE EXPND(IN.STR$,OUT.DEC,DECIMALS%)
        .IDENT /000001/
        .NLIST TTM
        .RADIX 8
; EXPAND FROM BINARY TO FLOATING-POINT - SJR - 5/83 - THIS ROUTINE
; WILL ACCEPT ANY COMPRESSED BINARY STRING, AND CONVERT IT TO
; FLOATING-POINT FORMAT WITH A SPECIFIED NUMBER OF DECIMAL PLACES.
EXPND:: CLR EXPHI                   ; CLEAR EXP AND HI MANTISSA
        CLR HORD                    ; NEXT HIGHER
        CLR MORD                    ; MIDDLE
        CLR LORD                    ; AND LOW ORDER MANTISSA
        CLR SIGN                    ; CLEAR SIGN
        MOV 2(R5),R0                ; POINT R0 TO STRING HDR
        MOV 4(R5),FLOAT             ; POINT TO OUTPUT AREA
        MOV 6(R5),R5                ; POINT TO DECIMAL PLACES
        MOV (R5),DECI               ; DECIMAL PLACE COUNT
        MOV (R0)+,R3                ; STRING ADDRESS
        MOV (R0),R4                 ; STRING LENGTH
;
        MOV #LORD+2,R0              ; POINT TO WORK AREA
        SUB R4,R0                   ; BACK UP TO MAKE ROOM
        TSTB (R3)                   ; CHECK SIGN
        BPL 1$                      ; PLUS...
        MOV #100000,SIGN            ; INDICATE NEGATIVE
1$:     MOV SIGN,R1                 ; PUT THE SIGN IN A REGISTER
MVLOOP: MOVB (R3)+,R2               ; GRAB A BYTE... YUM YUM...
        TST R1                      ; WAS THIS SUCKER NEGATIVE ??
        BPL 2$                      ; NO
        COM R2                      ; COMPLEMENT THE NUMBER
2$:     MOVB R2,(R0)+               ; PUT IT IN THE WORK AREA
        SOB R4,MVLOOP               ; DO IT FOR ENTIRE STRING
        MOV HORD,R3                 ; PREPARE TO NORMALIZE
        MOV MORD,R4                 ; MEDIUM ORDER MANTISSA
        MOV LORD,R5                 ; LOW ORDER MANTISSA
        SWAB R3                     ; ADJUST FOR ADDRESSING SWITCH
        SWAB R4                     ; DITTO
        SWAB R5                     ; DITTO
        TST R1                      ; MORE WORK IF NEGATIVE...
        BPL 4$                      ; PLUS
        ADD #1,R5                   ; COMPLEMENT DONE, NOW INCREMENT
        ADC R4                      ; ADD THE CARRY
        ADC R3                      ; AND ONCE AGAIN
4$:     MOV #60,R1                  ; EXPONENT
        TST R3                      ; CHECK HIGH ORDER
        BNE NORMAL                  ; GO NORMALIZE
        TST R4                      ; CHECK MEDIUM ORDER
        BEQ 6$                      ; HMMM... MAYBE LOW ORDER IS > 0
        MOV #40,R1                  ; LESS SHIFTING NEEDED
        MOV R4,R3                   ; ADJUST THE NUMBERS
        MOV R5,R4
        CLR R5                      ; DON'T FORGET TO CLEAN-UP
        BR NORMAL                   ; NOW NORMALIZE
```

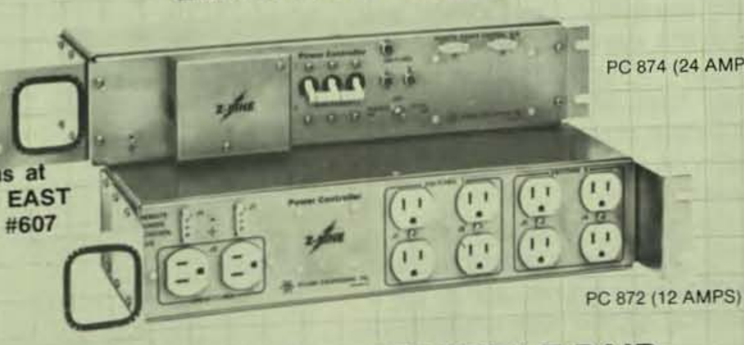
Floating-point representation of
came about because computers
to represent all types of data in a
ndard" form. It was designed to
mmodate very large numbers as
as very small numbers with the
est possible number of significant
ts. When it comes to data storage
disk, however, floating-point
resentation is usually very costly.

Since the general format of data is
ined or known when data files are
signed, it would be much more effi-
nt to store the data in some form
er than floating-point. For this
ason, I designed a set of data com-
ession routines that convert floating-
int decimal to fixed-point binary and
ce-versa. The below listings are of
o routines, one does 4-word floating-
int compression in BASIC-PLUS-2,
nd the other does 4-word fixed-point
floating-point expansion in MACRO.

Excluding laziness, the reason that
he compression routine wasn't writ-
en in MACRO, is that compression is
ess frequently needed than expansion.
n my applications I have found that
ENTRY and UPDATE operations on
compressed fields are performed less
frequently than INQUIRY, REPORTING,
and DECISION processes. The speed im-
provement by converting to MACRO
wasn't cost justified. Furthermore, for
simple addition, subtraction, incre-
ment, and decrement operations I have
four MACRO routines that perform the
operations on compressed data
without even having to expand it
(listings in the next installment).

To convert floating-point data to
fixed-point data, you must know
where the decimal point will fall. For
example, when dealing with monetary
amounts where only dollars and cents
are to be stored, the number of decimal
places is two. Next, you must deter-
mine the size range of the numbers to
be represented, keeping in mind that
the stored number will have no decimal
point. For example, $123.45 will be
stored as 12345. Knowing the size

```
6$:     TST R5                          ; CHECK LOW ORDER
        BNE 8$                          ; WAS IT ZERO ??
        MOV FLOAT,R0                    ; YES... CLEAR OUT TARGET AREA
        CLR (R0)+
        CLR (R0)+
        CLR (R0)+
        CLR (R0)
        JMP BYE                         ; THAT'S ALL FOLKS !!
8$:     MOV #20,R1                      ; EVEN LESS SHIFTING NEEDED
        MOV R5,R3                       ; ADJUST
        CLR R5                          ; CLEAN-UP LOW ORDER MANTISSA
NORMAL:                                 ; 48-BIT NORMALIZATION
; THE NORMALIZATION STARTS BY SHIFTING THE MANTISSA 8 PLACES TO THE
; LEFT IN ALL 4 WORDS (R2,R3,R4,R5).  THE SHIFT IS ACCOMPLISHED BY A
; SEQUENCE OF SWAB, CLRB, AND XOR INSTRUCTIONS, BECAUSE OF SPEED.
; SUBSEQUENT SHIFTS (AT "NLLOOP") ARE PERFORMED ONE AT A TIME, CHECKING
; FOR A BIT TO APPEAR IN THE HIGH BIT OF THE LOW BYTE OF R2 (EXP & HI).
        SWAB R3
        MOV R3,R2                       ; COPY TO (EXP & MSB)
        SWAB R4
        SWAB R5
        CLRB R3                         ; CLEAR BYTE 1 FROM HORD
        XOR R3,R2                       ; CLEAR EXPONENT
        XOR R4,R3                       ; INSERT BYTE 3 FROM MORD
        CLRB R4                         ; CLEAR BYTE 3 FROM MORD
        XOR R4,R3                       ; FIX BYTE 2
        XOR R5,R4                       ; INSERT BYTE 5 FROM LORD
        CLRB R5                         ; CLEAR BYTE 5 FROM LORD
        XOR R5,R4                       ; FIX BYTE 4
;
NLLOOP: TSTB R2                         ; HAVE WE GOT A BIT YET ??
                                        ; (IN THE EXPONENT LSB)
        BMI 2$                          ; CAN'T GO NEGATIVE
        DEC R1                          ; SHIFT R5 LEFT
        ASL R5                          ; THEN SHIFT R4
        ROL R4                          ; THEN SHIFT R3
        ROL R3                          ; AND FINALLY, R2
        ROL R2                          ; AND DO IT AGAIN
        BR NLLOOP
```

range of the numbers to be stored allows you to choose the size of the field (in bytes) that is to contain the compressed number.

Numbers between -128 and 127 require one byte, -32768 to 32767 requires two bytes, and -8388608 to 8388607 requires three bytes. In general, the ratio of digits (expanded) to bytes (compressed) for signed numbers is: 2/1, 4/2, 6/3, 9/4, 11/5, 14/6. This means that a social security number or nine-digit zip-code can be stored in four bytes. Telephone numbers, including area-code would require five bytes. Six-digit account numbers can be stored in three bytes (seven-digit positive numbers can also be stored in three bytes).

These two routines are designed to be invoked from BASIC-PLUS-2 via a CALL statement. The decimal numbers are double-precision (4-word) floating-point, and may not be passed in integer format variables or constants. The compressed data is passed as a string, and to minimize BP2 string manipulation, may be passed (unsubscripted) from MAPs or COMMONs. The variable BYTES% in COMPRS specifies the number of bytes to be used for the compression, and must not exceed six. The variable DECIMALS% must be an integer (especially for EXPND) and specifies the number of digits that are to the right of the decimal point in the decimal form of the number. Any digits further than DECIMALS% places to the right of the decimal point are discarded.

One other variable used in COMPRS, E% returns a zero value if the compression worked, and a -1% if the decimal number was too large to fit into the signed fixed-point string. Proper program design will allow for the elimination of this variable and/or its being tested.

To compile COMPRS, specify "COM/OBJ/NOLINE/DOUBLE" to BP2. To assemble EXPND, specify "EXPND=EXPND" to MAC.TSK (preferred)

```
2$:     MOV DECI,R0                      ; GET THE DECIMAL-PLACE COUNT
        BEQ NLDONE                       ; NO DECIMAL-PLACES !!
        SUB R0,R1                        ; DIVIDE BY (2**DECI)
        MOV R1,EXP                       ; SAVE REAL EXPONENT FOR LATER
        ASL R0                           ; MAKE DECIMAL-COUNT AN OFFSET
        ADD #POWER5-2,R0                 ; POINT INTO THE POWERS-OF-5 TABLE
        MOV (R0),DECI                    ; SAVE DIVISOR IN DECI FOR NOW
; NOW THAT WE'VE DIVIDED BY (2**DECI), WE MUST DIVIDE BY (5**DECI).
; TO ACCOMPLISH THIS, WE DIVIDE EACH MANTISSA BY (5**DECI), AND GLUE
; THE REMAINDER ONTO THE NEXT LOWER MANTISSA.  THE LOWEST MANTISSA
; DIVISION YIELDS A REMAINDER WHICH WE ALSO DIVIDE BY (5**DECI) TO GET
; SOME INTERPOLATION BITS FOR THE SUBSEQUENT RE-NORMALIZATION. FIRST,
; HOWEVER, WE MUST BORROW SOME LOW ORDER BITS TO COMPENSATE FOR THE
; FACT THAT WE MUST DO DIVISIONS WITH 15-BIT QUOTIENTS.
DIVIDE: CLR R0                           ; CLEAR HIGH WORD OF DIVIDEND
        ASL R4                           ; BORROW ONE BIT FROM R4 (MORD)
        ROL R3                           ; MOVE IT THRU R3 (HORD)
        ROL R2                           ; AND GIVE IT TO EXPHI
        ASL R3                           ; BORROW ANOTHER FROM R3 (HORD)
        ROL R2                           ; AND GIVE IT TO EXPHI
        SUB #2,EXP                       ; MAKE EXPONENT REFLECT BORROWS
; ALL DONE WITH PRE-ALIGNMENT NONSENSE...
        MOV R2,R1                        ; LOW WORD OF DIVIDEND
        DIV DECI,R0                      ; DO FIRST DIVISION
        MOV R0,EXPHI                     ; SAVE EXPHI
;
        MOV R1,R0                        ; REMAINDER TO HIGH WORD OF DIVIDEND
        MOV R3,R1                        ; LOAD LOW WORD OF DIVIDEND
        ASR R0                           ; COMPENSATE FIRST "BORROW"
        ROR R1
        DIV DECI,R0                      ; DO SECOND DIVISION
        MOV R0,R3                        ; SAVE HORD
;
        MOV R1,R0                        ; REMAINDER TO HIGH WORD OF DIVIDEND
        MOV R4,R1                        ; LOAD LOW WORD OF DIVIDEND
        ASR R0                           ; COMPENSATE SECOND "BORROW"
        ROR R1
        DIV DECI,R0                      ; DO THIRD DIVISION
        MOV R0,R4                        ; SAVE MORD
;
        MOV R1,R0                        ; REMAINDER TO HIGH WORD OF DIVIDEND
        MOV R5,R1                        ; LOAD LOW WORD OF DIVIDEND
        ASR R0                           ; ENSURE CLEAR SIGN BIT
        ROR R1                           ; ROTATE THROUGH LORD...
        ROR R2                           ; SAVE LORD L.S. BIT
        DIV DECI,R0                      ; DO FOURTH DIVISION
        MOV R0,R5                        ; SAVE LORD
;
        MOV R1,R0                        ; REMAINDER TO HIGH WORD OF DIVIDEND
        CLR R1                           ; NO LOWER ORDER
        ASL R2                           ; RESTORE LORD L.S. BIT
        ROR R1                           ; SHOVE IT INTO LOW WORD OF DIVIDEND
        ASR R0                           ; DON'T FORGET: 15-BIT RESULT
        ROR R1                           ; MEANS SMALLER DIVIDEND
        DIV DECI,R0                      ; DO ONE LAST DIVISION
; NOW WE HAVE TO GLUE ALL OF THE 15-BIT RESULTS TOGETHER...
        ASL R0                           ; START WITH THE SUPER LOW ORDER
        ASL R0                           ; SHIFT OUT A BIT
        ROL R5                           ; LORD IS FIXED
        ASL R0
        ROL R5
        ROL R4                           ; MORD IS FIXED
        ASL R0
        ROL R5
        ROL R4
        ROL R3                           ; HORD IS FIXED
        MOV EXPHI,R2                     ; RESTORE EXPHI FOR NORM
        MOV EXP,R1                       ; RESTORE "REAL" EXPONENT
; WE MUST RE-NORMALIZE THE MANTISSA BY SHIFTING IT TO THE LEFT UNTIL
; THE HIGH BIT OF THE LOW BYTE OF "EXPHI" IS SET.
; WHEN THIS HAPPENS, R1 HOLDS THE EXPONENT (WHICH MUST BE MADE EXCESS 128,
; AND ATTACHED TO THE HIGH ORDER MANTISSA). THE BIT WHICH CAUSED THE LOOP
; TO END IS DISCARDED (ASSUMED BIT), AND THE NORMALIZATION IS COMPLETE.
NLLOP2: TSTB R2                          ; MORE NORMALIZATION
        BMI NLDON2                       ; EXIT IF DONE
        DEC R1                           ; DECREASE EXPONENT
        ASL R0                           ; LOWEST LORD
        ROL R5                           ; LORD
        ROL R4                           ; MORD
        ROL R3                           ; HORD
        ROL R2                           ; EXP & HI
        BR NLLOP2                        ; MAX ITERATIONS: 12(DECIMAL)
```

or MACRO.SAV. At this point you have two object modules (.OBJ files) that you can access from BP2 programs only if you tell the task builder that they are to be included in the task. They cannot be accessed from BP2 programs that aren't task-built. The module names can be included in the ODL file, or in the build command as follows:

BUILD program,COMPRS,EXPND/options

A much easier way of having the task builder include these modules whenever you use them requires their one-time installation into your system object library. This will probably have to be done by a privileged user, but once done, you will never have to specify that the routines are to be included in your tasks. They will be included automatically. To install them in the library, the command line to the librarian (LBR.TSK) is LB:BP2COM/IN = EXPND,COMPRS".

About compressed numbers: They are stored in two's complement binary. Therefore their sign can be tested by checking the ASCII value of (the first byte of) the string. If it is greater than 127, then the number is negative. Positive compressed numbers make excellent "keys" in indexed file structures, since they represent data compactly (yielding smaller index trees) and in the correct format to insure that the data is retrieved in numerical order. Floating-point numbers cannot be used as keys, since the algorithm by which floating-point data is stored does not yield strings that translate to numerical order.

One complex data structure that is easily implemented with compressed data is an indexed file that is to be accessed forward or BACKWARDS. Positive compressed numbers yield key values that advance forward through the file, while the same numbers negated and compressed yield key values that go backwards through the file! Have fun!

```
NLDON2: ASL R0                             ; PREPARE TO ROUND RESULT
        BCC NLDONE                         ; < .5
        ADD #1,R5                          ; INCREMENT R5
        ADC R4
        ADC R3
        ADC R2                             ; CARRY INTO EXP IS IMPOSSIBLE
NLDONE: BIC #177600,R2                     ; CLEAR THE BIT (ASSUMED BIT)
        ADD #200,R1                        ; EXPONENT IS EXCESS 128
        ASH #7,R1                          ; ALIGN THE EXPONENT
        ADD R1,R2                          ; ATTACH EXPONENT TO MSB
        ADD SIGN,R2                        ; TACK ON THE SIGN
        MOV FLOAT,R0                       ; POINT TO THE RESULT AREA
        MOV R2,(R0)+                       ; LOAD THE FLOATING-POINT RESULT
        MOV R3,(R0)+
        MOV R4,(R0)+
        MOV R5,(R0)
;
BYE:    RETURN
;
FLOAT:  .WORD 0                            ; POINTER TO RESULT
DECI:   .WORD 0                            ; DECIMAL-PLACE COUNTER
EXP:    .WORD 0                            ; EXPONENT (ALONE) HOLDER
SIGN:   .WORD 0                            ; SIGN HOLDER
;
EXPHI:  .WORD 0                            ; EXP & HIGHEST ORDER MANTISSA
HORD:   .WORD 0                            ; HIGH ORDER MANTISSA
MORD:   .WORD 0                            ; MEDIUM ORDER MANTISSA
LORD:   .WORD 0                            ; LOW ORDER MANTISSA
;
        .RADIX 10
POWER5: .WORD 5                            ; POWERS-OF-FIVE TABLE
        .WORD 25,125,625,3125,15625
;
        .END
```

VMS GETS A LITTLE HELP FROM A FRIEND

By Ed Vaccaro, Shared Medical Systems, Malvern, Pennsylvania

As a data processing development manager for a manufacturer it was part of my job to make sure our computer system was serving the business, not vice versa.

The firm had been using a Data General minicomputer with the operating system MICOS for eight years. But the company, a $10 million a year operation, had outgrown it. It didn't take much shopping around for us to settle on a VAX. We were impressed with the programming facilities, especially EDT, by the operating system architecture and the ease with which programmers could do very sophisticated work. To put it in a word, the facilities were excellent.

We bought a 780 and installed it in April of 1982, hooking it up to the MICOS and about 30 dumb terminals — the standard VT100s, 102s and 131s. Our users were order entry clerks, sales help and managers doing text editing and word processing. This was a group of users not only gunshy of the "new technology," but people who would never regard the VAX as more than a means to an end.

It didn't take us long to find that as powerful as the VAX was in DCL, it was missing some basics. For example, the system lacked the ability to help an unsophisticated user recover from a crash and allow him or her to keep on trucking while the DP people analyzed the error. In short, we needed to reduce down time on the system.

We had the advantage of working on a virgin machine. So we were able to define our needs accordingly. Users wanted menus. But they didn't want to plow through three or four levels of them before they got to a process. Some users wanted direct access, the option to skip the menus altogether. But they still wanted to be able to use menus for help.

What we came up with was a superviser or piggyback system we called CONTROL. Its genesis took an afternoon for the design and a man-year to write.

CONTROL, as its name im[plies,] controlled the user terminal se[ssion] and acted as the interface betwee[n the] user and VMS in the same way a [pup]peteer controls a marionette. Nor[mally] DCL itself serves that purpose. B[ut we] usurped it, not by replacing DCL b[ut by] putting CONTROL on top of it.

The system had three main [com]ponents: dynamic menu gene[...]

```
Step 1: User executes main command procedure
Main Command Procedures:
$ XXX
$ XXX   any site specific environmental setup requirements
$ Loop:
$ Run CMD
$ Goto Loop
```

FIGURE 1.

```
$ Set noon
$ Run cmdsetup
$ 'User__Command'
$ If $status then goto clean__exit
$ run crashtrap
$ clean__exit:
$ exit
```

FIGURE 2. Command Procedure Executed by CMD

and access control, print file management and routing, and broadcast message management.

When the user wanted to invoke CONTROL, he or she executed the command procedure shown in Figure 1. That in turn invoked the Control Menu Drive (CMD) program which took charge of the terminal session. CMD would accept commands from the user terminal, evaluate and analyze them and check to see if the user had access to the process. Once it did that, CMD would invoke the command procedure shown in Figure 2 by issuing a **lib$do_command** service.

Notice in Figure 2 the $ '**user_command**' executes a symbol set up by CMD which is the equivalent of a run-program command. That makes the command procedure generic — it runs any program the user has selected.

When the user was finished with that process, CMD would interrogate $ Status to determine whether the program completed normally. If it hadn't, CMD would invoke **crashtrap**.

Crashtrap would store the name of the program that was running and status value in a file along with the message text from the system, such as file not found. It put the message on the screen, "Previous program terminated abnormally. Call data processing immediately." It would then lock up the terminal. DP would copy the screen using a screen printer, enter a password and free the terminal to return to a menu.

This only works if your DP department can respond immediately, which ours was able to do, so we made crashtrap optional. We wanted to make sure we didn't miss anything in debugging, plus we wanted to get information as soon as possible about the user written program.

We also discovered this feature of CONTROL demystified the system for our users. They weren't afraid of it anymore. They knew if the job crashed, someone would be there promptly to fix it. If we knew no one would be available, we simply removed the lockup device and they could go on their merry way.

Next we tackled security. VMS' protection system didn't give us the flexibility we wanted to control users' access to programs. VMS structures the file protection in groups. In our office there was too much cross-tracking to make that practical. We decided what we needed was a per process access method keyed to menu entry and the use of a menu editor.

The menu method worked this way: Once a menu is created it's accessed via the CMD. The CMD displays menus, accepts command selection, evaluates the user's access privilege and activates the selected program. Menus may be modified at any time. The menus dynamically link to programs and are capable of limiting access on a per user basis regardless of protection, or UIC codes.

Information about each process that could be executed from CONTROL was put into a file in a special directory called **CONTROL.database**. That information included: the name of the menu command, full file specification of the program and the security access information. There were two classes of access — global and limited. For the latter, each user who had access was stored on the database record. Maintenance of security access and menu entries was performed via the CONTROL menu editor. That gave the system manager more control over user access than VMS does.

Next we tackled the print file management system. We called ours PFMR, for Print File Managment and Routing. It's designed for use in multiple printer environments where it's desirable to route reports to a printer located closest to the terminal on which the user is working. That way a sales manager could order a report in a conference room while a meeting was in progress.

Our shop's hardware consisted of five LA 100 character printers and the mighty LP11 900-line-per-minute line-printer all spread around different offices. One of them was set up to exclusively handle special forms. In addition to automatic routing, we needed the system to purge itself without user intervention. In other words, we didn't want our users to have to clean house.

All of the other processes in CONTROL were designed to be executed from the terminal. But PFMR was different. It was started up as a detached process during system initialization and is resident at all times — a monitor, if you will.

We had to fool the system into thinking it was interactive so it could intercept broadcast messages from the VMS print system. To do that we had to assign a permanent terminal port to it. We also had to make entry modifications in the Process Control Block (PCB) to trick VMS into thinking PFMR was actually an interactive user. A user name entry and a terminal number had to be put into the PCB. I also had to create the process permanent files and process logical name table normally created during an interactive log-in from a terminal. All print requests were dispatched from PFMR to VMS. Users interfaced to PFMR through the command **SPOOL**. This replaced the **print** command. A user could still use the normal VMS print method, but he or she only could invoke PFMR by using the **spool** command. It functioned in a similar manner to the print command: i.e., **spool/copies** = 5 filename.

For the routing function we got rid of VMS' typical printout command: **print/que** = tta6 myfile.lis. Our users could call up a menu and order a sales order report. The inquiry screen would then ask them questions, build a command file and submit it to the VMS batch queue. That way the user could continue to work on the terminal while the report, no matter how lengthy, was being processed. It wasn't unusual for the system to handle reports that took an hour and a half to process. When the report was completed, we wanted to be able to notify the user cleanly—without destroying a formatted screen, a glitch in VMS. So we had CONTROL intercept the broadcast messages.

We took care of the automatic routing by drawing up a "floor plan" outlining the locations of the terminals and the nearest printer. Now when a report finished processing a request was sent to the CONTROL print system. It would tell the VAX to print a specific file, pinpoint what terminal the user had submitted the report from, and send it to the appropriate printer. It would then wait until the user exited from a process before flashing the message that the report was complete and on what printer.

To accomplish the automatic purging, CONTROL kept a record of each report that contained instructions about its demise. For our purposes reports were earmarked for immediate destruction, 48-hour destruction (not over weekends) or eternal life. We used the VMS timer services to do this. Every hour, PFMR would go through its cleanup routine and determine which completed files were to be purged.

The beauty of PFMR was that it interfaced directly with the standard VMS print symbiont.

The final piece of CONTROL was the Broadcast Message Management system. It intercepts the VAX's broadcast messages and stores them until they can be safely displayed without disrupting a user and destroying a formatted screen, or it displays the message on the last line of a screen. We did this by writing a set of terminal input routines which made it easy to do reverse video, highlighting, underlining by coding the call to the input routine.

The input routines associated the user's terminal with a VMS mailbox. When a broadcast message hit the associated mailbox, the input routine would intercept the message and, based on the parameters specified during program development would: (a) display the message at the line determined by the programmer (b) hold the message until the program explicitly requested a display. We didn't worry ourselves about the terminal commands.

BMM was the only service that required modifications to the existing programs.

CONTROL was written in native mode macro for VMS exclusively. Its present version is the second generation of a system developed for Harold Beck & Sons Inc., Newtown, Pa. ■

Dear Dr. DEC:

I am an end-user in the distribut[illegible] business who purchased a PDP 11/[illegible] last year. We had been using [illegible] Ultimacc System based on a D[illegible] General Nova 1200 with 16K, [illegible] disc drive with 5 MB, 2 CRTs and a [illegible] 1pm printer. Our new DEC has 25[illegible] 30 MB, four CRTs, two LA-120 print[illegible] and three RL02s.

Our business has not changed sin[illegible] 1983, and I find that what took abo[illegible] 16 man-hours on the old system, no[illegible] takes 24 to 30 man-hours. Our pr[illegible] gramming house is saying that's t[illegible] best that can be done and Di[illegible] seems to concur. We have spe[illegible] "oodles" of money and time in ord[illegible] to take a giant step backward. A[illegible] suggestions?

Matthew J. Mi[illegible]
Cantor Bros., In[illegible]
Farmingdale, New Yo[illegible]

They have RT-11XM and the syst[illegible] seems to be in DIBOL. The soluti[illegible] that comes up is:

a) Get an 11/23+, a new box a[illegible] backplane for 22-bit addressing. Ge[illegible] least ½ MB.

b) Get a RSTS license.

c) Don't buy that 730.

The above presumes that the appl[illegible] tion can be "ported" to the RSTS e[illegible] vironment.

Since your old application was [illegible] MACRO, you will be hard pressed [illegible] duplicate its performance in DIBOL. [illegible] soon as DEC announces it, upgra[illegible] your 11/23+ to an 11/73.

* * *

Dear Dr. DEC:

Perhaps I can shed some light [illegible] Thomas H. Haydon's question ab[illegible] downloading executable images to an LSI-11. The DECnet/VAX System Manager's Guide describes a process in which RSX-11S systems images can be booted on a remote PDP11 system via DECnet. However, this is really overkill for many applications.

There are two other ways to load code into an LSI-11 using only microODT and the serial console which is part of every LSI-11. To do this you will need a serial interface (DLV-11 compatible) wired for the standard console CSR and vector (777560/60) on the remote system. On the host system you will need a serial port of any type; i.e., DL-11, DH-11, or DMF-32. This port must set noecho and at the same baud rate as the remote system. You will also need to write a little program to drive the port and format the load image for transmission.

The first method is to use the binary load and unload commands of microODT. These two commands are somewhat documented in the Microcomputer Handbook. They are used by DEC during final assembly and test to run their APT (Automated Product Testing) system. You could experiment with this method, since I have never tried it. The other method I have used, and it works pretty well.

This method uses a program running on the host system to simulate the dialogue that one would use to enter the program by hand via microODT. To use this method the LSI-11 has to be wired to halt and go into microODT on power-up. The host system reads the output of the LSI-11 and looks for the "@" prompt. When it sees the prompt it sends the starting load address in octal ASCII characters followed by a slash character. It then waits for the echo of load address, slash, and contents of that address, and reads it all in. It then sends the new contents of the address in octal ASCII characters followed by a line feed, and only a line feed. Because the host system sent only a line feed the next location is opened and ready to be deposited with its contents. You continue like this for the whole image and when you are done send a carriage return, followed by the start address of the image, followed by a "G" for go and you are off and running. Of course your program will have to have the code in it to communicate with the host via the console, but this is trivial. You would probably want to include a checksum of some sort in your load procedure.

You can see how simple this system really is. We used it to load code into a PROM burner and to communicate with other stand-alone systems. However, you may want to give the first method a try since every DEC system with a serial console (VAX, LSI, PDP11/44) has a facility of this type to allow APT to be used, and this may be one of those good old "undocumented features" which can be very useful.

Joel K. Gallun
OAO Corporation
Greenbelt, Maryland

* * *

Dear Dr. DEC:

In response to Nick Kushmerick's question of November '83:

I started to submit a response to this one, but I figured there would be several others with the same response. After seeing the responses, I couldn't help but offer my two cents' worth.

The problem of cursor addressing in VT100 mode from FORTRAN programs is that of suppressing the leading space when the cursor address has less digits than the field-size. I had been resorting to means as described in the January issue, until I realized that FORTRAN-77 has a format mode which will use leading zeroes to fill out a field. For example, the following code will move the cursor to position (X,Y) and plot an asterisk:

```
WRITE(5,10) 27,Y,X
10 FORMAT(1X,A1,'[',I2.2,';',I3.3,'H*')
```

By specifying the format as I2.2, it tells FORTRAN that the field will always be two digits, even if it has to pad on the left with zeroes.

Paul Barrett
General Railway Signal Company
Dallas, Texas

SIMACS
SYSTEM INDUSTRIES MULTIPLE ACCESS CONTROL SYSTEM

By Albert B. Chu, Ph. D.
SIMACS Software Development, System Industries, Milpitas, California

Many mini-computer users find that they need more CPU power than a single processor can provide. Rather than procure a more powerful or larger processor, one possible solution is to add one or more similar processors to their system. This solution introduces a further problem. Both processors must share the same data base if they are to be viewed from an application program or a user's point of view as a single system. Of course, such an addition must be completely transparent to any existing application software or user programs.

The hardware already exists which allows multiple CPUs to access a single disk storage device. However, unless there is some form of communication between the CPUs it is difficult to maintain the integrity of the data on the common disk storage device. Each CPU can make changes to the data structure without the other knowing about it.

There are some partial solutions to this problem.

As shown in Figure 1, all requests for access to the common data base come through a single CPU via the local network. This CPU treats all these requests simply as another local user. This provides a limited increase in processing power because each satellite CPU can handle its own front end processing. However, the single CPU still has to handle all the data access requests, thus increasing the load on the CPU and creating a bottleneck. This is typical of any master slave relationship.

While networks are useful for remote access, they introduce an overhead in both CPU time and I/O throughput. They provide only a limited degree of system integration and need special protocols to access the data. File transfers may also have to pass over other hosts before reaching the destination. It is generally not worth the cost to use a network merely to gain access to a file system.

LOCAL AREA NETWORK METHOD

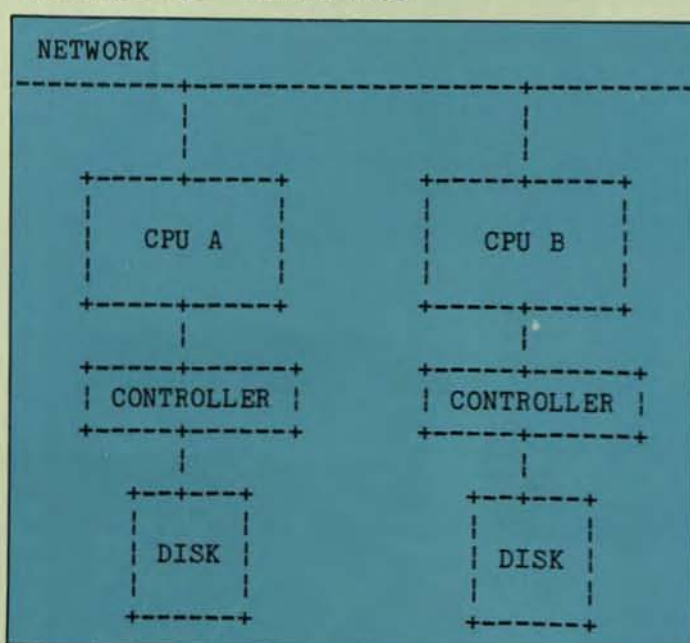


FIGURE 1.

DUAL CHANNEL DEVICE OR DUAL PORTED CONTROLLER

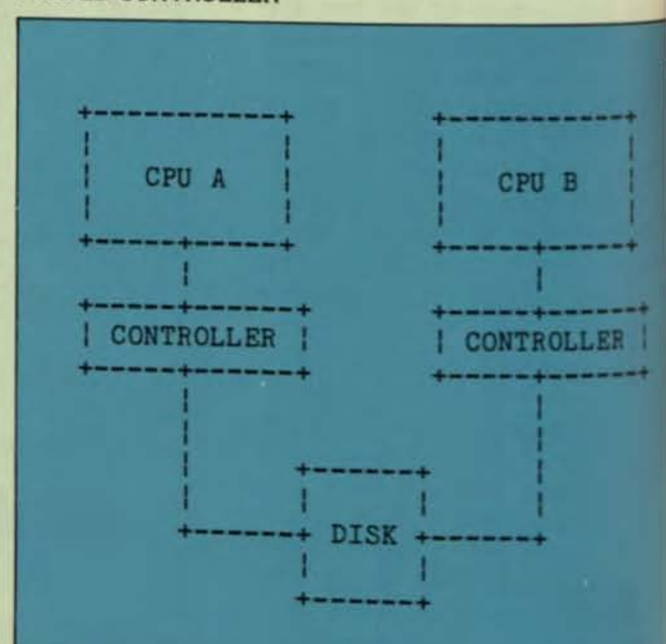
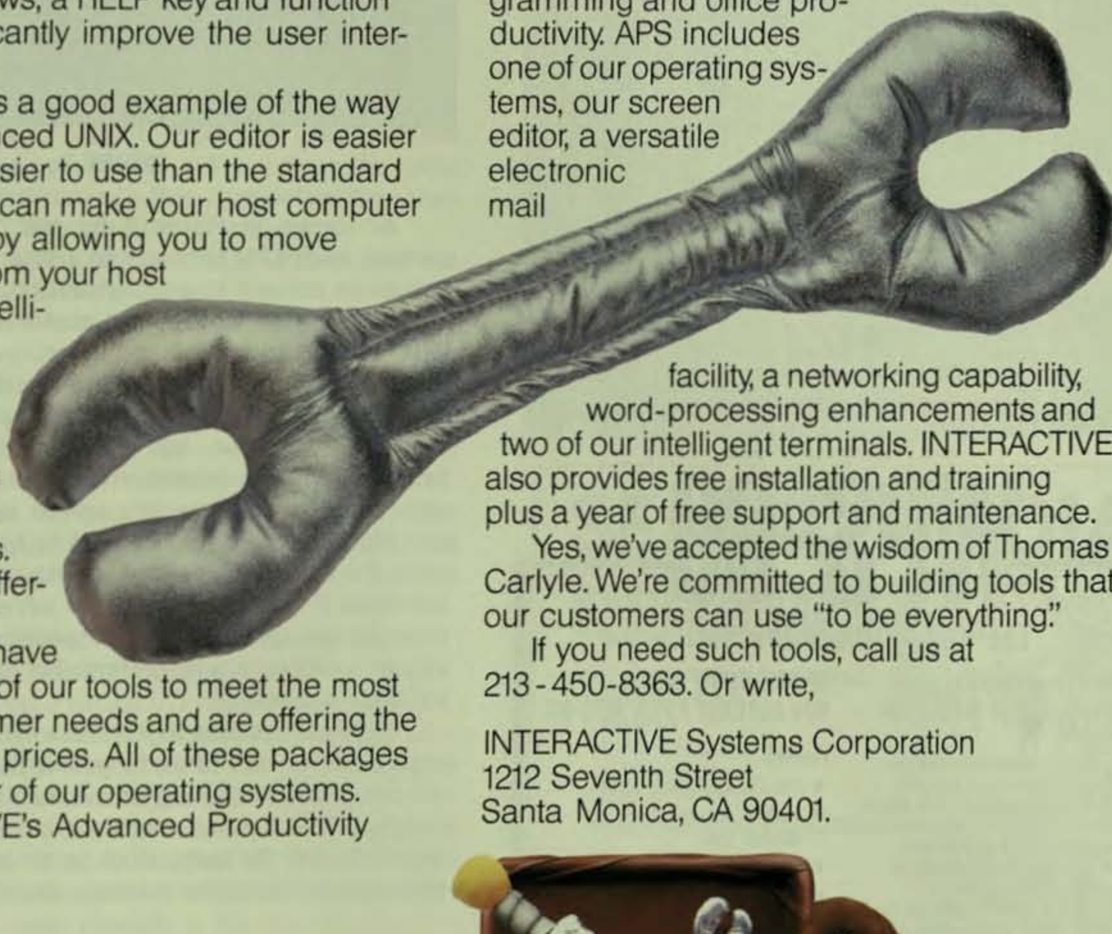


FIGURE 2.

IMAL. WITHOUT TOOLS, LS, HE IS EVERYTHING.

—From Thomas Carlyle, 1795-1881



One of the most important tools is our full-screen text editor. It uses features such as multiple windows, a HELP key and function keys to significantly improve the user interface to UNIX.

The editor is a good example of the way we have enhanced UNIX. Our editor is easier to learn and easier to use than the standard UNIX editors. It can make your host computer more efficient by allowing you to move certain tasks from your host computer to intelligent terminals. By using that feature, you can provide better response times and support more users.

We're now offering still another innovation. We have packaged sets of our tools to meet the most common customer needs and are offering the sets at new low prices. All of these packages run under either of our operating systems.

INTERACTIVE's Advanced Productivity System is our most complete set. It gives you a packaged set of tools to enhance both programming and office productivity. APS includes one of our operating systems, our screen editor, a versatile electronic mail facility, a networking capability, word-processing enhancements and two of our intelligent terminals. INTERACTIVE also provides free installation and training plus a year of free support and maintenance.

Yes, we've accepted the wisdom of Thomas Carlyle. We're committed to building tools that our customers can use "to be everything."

If you need such tools, call us at 213-450-8363. Or write,

INTERACTIVE Systems Corporation
1212 Seventh Street
Santa Monica, CA 90401.

INTERACTIVE
SYSTEMS CORPORATION

CIRCLE D123 ON READER CARD

```
+-----------+          +-----------+
|           |          |           |
|   CPU A   |          |   CPU B   |
|           |          |           |
+-----+-----+          +-----+-----+
      |                      |
      +-------+      +-------+
              |      |
          +---+------+---+
          | CONTROLLER   |
          +------+-------+
                 |
             +---+---+
             |       |
             | DISK  |
             |       |
             +-------+
```

FIGURE 3.

As shown in Figure 2 for the dual channel devi method, each CPU connects to a separate controller. Bo controllers connect to a dual channel disk storage device. Figure 3 for the dual ported controller method, the tw CPUs connect to a single controller.

In either case, the disk can be partitioned so that ea CPU has exclusive access to one partition. This prevents da structure corruption, but does not allow shared data. limited sharing is possible if one CPU can write into t other's partition at certain agreed upon times. Anoth possible solution is for one CPU to have read-only acces while the other CPU has read/write access. This solution workable if the data is updated infrequently and the u dates can be coordinated with subsequent reads. These, similar solutions are hardly invisible to existing applicati software or users.

It should be noted that if these methods are not used, is possible for two CPUs to gain write access to the sam volume, neither knowing of the modifications the other making. CPU A could allocate a block to one file; CPU could allocate the same block to an entirely different fi thereby corrupting the volume's structure.

SIMACS SOLUTION

In the situation as shown in Figure 3, we can define o semaphore for each volume which is accessible to the co troller only. If a semaphore is not set, any CPU can reque the controller to set it. The controller does this a remembers which CPU made the request. It refuses r quests from any other CPU to set this semaphore until t original CPU requests it to be cleared.

By agreement between cooperating programs t semaphore gives each CPU exclusive access to the di structure for the period of time that the CPU owns t semaphore. During this time, critical updates to the di structure could be protected, and still allow shared acce for other disk I/Os which do not alter the disk structure.

The System Industries Multiple Access Control System (SIMACS) 9920 controller allows one CPU to reserve the semaphore for its use. The controller will inform any other CPU requestors that that semaphore is already in use. It will then inform the owner that others have requested access. This allows owner CPUs to use appropriate algorithms for relinquishing control of the disk.

It may be necessary to make modifications to operating system software to ensure that the disk structure is left in an integral state before relinquishing ownership. It is also essential for the CPUs to cooperate with each other, since the controller does not physically lock out or prevent other CPUs from accessing the reserved disk.

THE SIMACS VAX/VMS IMPLEMENTATION

Each CPU in the configuration contains a special file server, the Ancillary Control Processor (ACP). When VAX/VMS accesses a volume, the ACP builds up the cache buffers from the disk reserved files that define the structure of the volume. The reserved files include:

a. The bitmap which contains information about the allocation of the blocks on the volume.

b. The index file which contains the headers of all files that reside on the volume.

c. The quota file which contains the disk quotas.

d. The master and user directory files which describe the directory structures.

When a request to create, extend or delete a file occurs, the ACP updates the reserved files to reflect the changes made. The ACP changes the copies residing in the cache. The copies residing on disk are not necessarily changed immediately.

Under VAX/VMS, the ACP handles requests involving access to the reserved files that describe the disk structure. Such a request is called a virtual I/O request. Once the ACP gives the requesting process the precise location of its data on the volume, subsequent requests do not go through the ACP. These requests are called logical I/O requests. They do not require references to the reserved files. These logical requests must not be confused with logical I/O requests made by the ACP in its processing of a virtual I/O request. ACP logical I/O requests make changes to the file structure. A virtual I/O may require the ACP to perform several logical I/Os to satisfy the request, but these I/Os must be autonomous such that ownership of the semaphore must be maintained until the virtual I/O request is completely satisfied.

The SIMACS solution grants a CPU exclusive access to the volume's reserved files only during a virtual I/O request which requires references to the reserved files. But all CPUs are still allowed to make logical I/O requests which do not reference the volume's structure.

The SIMACS ACP is designed to cooperate with similar ACPs in other CPUs. Such an ACP maintains the integrity of the volume by accessing the shared volume only when it has ownership of that volume. It checks that the volume is in an integral state before the ACP starts any I/O and it returns the volume to an integral state before relinquishing ownership. It does this by flushing the reserved files from the cache buffers out to the volume.

A synopsis of control follows:

1. CPU A needs to process a virtual I/O request to the volume. The SIMACS ACP issues a reserve request.

2. If no other CPU owns the volume, the controller grants the request immediately.

3. If CPU B owns the volume:

   a. The controller sends the owning CPU a REQUEST-TO-RELEASE interrupt, and notifies CPU A that the volume is already reserved.

   b. The owning CPU receives the REQUEST-TO-RELEASE interrupt and will release ownership after it has completely processed any virtual I/O that has been started.

   c. If the owning CPU made any changes to the reserved files, the modified buffers in the ACP cache are flushed to return the disk structure to an integral state before the ownership is released.

4. After gaining the ownership of the semaphore, the new owner of the volume (CPU A) checks the integrity of the disk structure. If the volume structure has been changed since last ownership, the cache of the reserved files will be revalidated; otherwise some ACP cache can still be used.

The SIMACS ACP incorporates an algorithm to determine when it must relinquish ownership of the volume. Transfer of ownership entails a certain amount of flushing overhead. It is therefore desirable to delay transferring ownership as long as possible while not delaying other CPUs excessively. After receiving a REQUEST-TO-RELEASE, the ACP continues processing I/Os for a length of time determined by a tunable parameter. Usually all the CPUs in a configuration will use the same value for this parameter. This will give equal priority to all CPUs accessing the shared volume. However, this value may be increased for one CPU which needs to process more I/O than the others.

Also, in a configuration where one CPU is faster than another (e.g., 780 and 750), one can use this parameter to even out the number of I/O operations that each CPU completes.

FILE LOCKING

In addition to the file locking that VAX/VMS provides, SIMACS uses a file-locking technique to prevent the contents of a file from becoming corrupted if more than one CPU makes updates to it. This technique allows a file to be opened for write from one CPU, or to be opened for read by multiple CPUs. The SIMACS file locking mechanism does not affect the file locking implemented by VAX/VMS on a single CPU. Files that are opened for shared access through RMS on a single CPU may still be accessed in the same manner.

DISK STRUCTURE INTEGRITY

One of the major design criteria of SIMACS was to maintain the disk structure integrity under all adverse circumstances. For example, in a SIMACS configuration there are two possibilities if one of the CPUs crashes:

1. The CPU does not own the semaphore of a shared disk. In this case, the volume structure remains in an integral state. The files which the crashed CPU was accessing will be in an unknown state. The surviving CPUs can still access the volume without any disruption.

2. The CPU owns the semaphore of a shared disk. In this case, the disk structure could be in an inconsistent state because the ACP cache was not flushed. In order to protect the disk structure from further corruption, SIMACS freezes the volume from any more allocations until the disk structure is rebuilt. During that time, SIMACS provides a mechanism to force the ownership of the shared disks so that the surviving CPUs can open, close and re-map files as long as the disk structure is not modified. Applications can still read and write data to and from open files. Full volume access is restored when the volume is rebuilt.

After the controller recovers from power failure, the semaphores maintained by the controller are probably lost. In order to avoid dual ownership claimed on the volume which leads to its corruption, the controller will not grant any reserve requests until it is reset either manually or through program request. This implies that the original owner before the power failure still has the full file accesses, but new virtual I/O requests from other CPUs will be denied until the controller is reset.

CONFIGURATION

The current SIMACS package allows connection of up to eight CPUs running VAX/VMS. It allows any combination of VAX 11/780, 11/750 and 11/730. PDP11s running RSX11M will shortly be added to the SIMACS family of products such that VAX/VMS and PDP11/RSX11M systems can share with ODS-1 structured disks.

The total shared disk storage capacity with currently available drives is in excess of four gigabytes on eight drives.

SUMMARY

SIMACS is an elegant solution which allows users of local VAX CPUs to create, store, access and retrieve files on a common volume just as it is when it is locally connected to an individual CPU. It allows file server ACPs to build up cache when there is no ownership contention. All file transfers are local I/O for the CPU which initiates the requests. None pass from one CPU to another, nor require double memory transfers.

The performance of the shared disks will be almost the same as non-shared disks, if the ownership never changes hands; and the same as the orginal file server ACP mounted with no cache if the ownership has to be changed for every virtual I/O request. SIMACS adds these capabilities in a way that is transparent to the application, requiring no new user commands to implement these features. It provides an integrated system for multiple CPUs to share the same disk file system. ■

EXTENDING THE INPUT CAPABILITIES OF BASIC-11

By R.M. Kaplan, Smith, Kline & French Laboratories, Philadelphia, Pennsylvania

Like most BASICs, BASIC-11 provides the standard methods for getting data into and out of a program. The methods provided are, for the most part, those that were defined in the original dialect of the language developed at Dartmouth. The design objective for input and output in BASIC was to provide a simple means of getting data into and out of the computer, avoiding the cumbersome definitions required by languages like FORTRAN and COBOL.

The INPUT and the PRINT statements are the primary means to move data back and forth between the computer and and its peripheral devices. They have been extended over the years to allow input from files, formatting of output, and some free forms of input, but the limitations in the design of these statements remain the same; they can only process a line of data from or to an input or output device.

BASIC-11 INPUT and PRINT statements allow input only in line oriented fashion. Unfortunately, data does not necessarily structure itself along line boundaries. This is the restriction that a programmer must place on a user — data must be entered one line at a time.

At the time BASIC was originally developed, all input and output devices were line oriented — essentially one dimensional. Terminals to timesharing systems used during the early '60s were TTY33 type devices; input was restricted by the physical limitations of the device. With CRT terminals it is possible to display multiple lines at a time. Unlike its predecessors, the CRT terminal is a two dimensional input/output device, and yet the restriction of line input remains in languages like BASIC-11.

Suppose we have a program that maintains a personal card catalog. This program stores three lines of title text, two lines of author text, and two lines of journal text for each catalog entry. One could envision the program requesting line one of the title, line two of the title, and so on until all of the multiple lines of data are entered.

But what happens when a user of this program makes a typing error on a line and notices the error after pressing return to enter the line? One preventive measure is to make the user confirm that the data is to be stored. If the user does not confirm, the program could ask if the user wanted to reenter some part or all of the data just entered. However, this multiline data entry and correction procedure is cumbersome. And there is an alternative.

FMS-11 and BASIC-11

The people at DEC have provided the RT-11/BASIC-11 community with a product that greatly enhances the input/output capabilities of BASIC-11. Forms Management System — 11(FMS-11) gives BASIC-11 the ability to use forms for the input/output function.

A forms management system is a set of computer programs that allows a programmer to design screens (a display on a CRT terminal). All of the lines of the screen may be used to display and capture data. In this way, multiple line data may be easily input and output to and from a computer program.

FMS-11 has a tool that allows the programmer to design libraries of screens. The process of designing a screen involves entering what is to appear in one display on the CRT terminal. Any character can appear anywhere on the CRT screen. Titles, labels, and instructions are all entered by simply typing this information on the terminal to the FMS-11 screen design tool.

The programmer using this screen design tool may also define fields. A field is a set of locations on the screen that can be used for data input or output. Any number of fields may be defined on a screen. The screen design tool is much like a text editor in that cursor keys are used to move around the screen, and the keypad on VT100 type keyboards provides special functions like cut and paste.

Once the form has been designed and inserted into a library it may be used for input and output. This is accomplished by a set of calls that are added to BASIC-11. These calls are defined by assembly language routines merged into the BASIC-11 interpreter. By using these routines, input/output is handled by the FMS input/output mechanism as opposed to BASIC-11's facility, allowing the user to change and correct data on the screen without having a reenter all of it to make a correction.

Sounds great, but there is a hitch. FMS-11 uses about 6000 words of the BASIC-11 user work area. Using FMS-11 requires that programs written in BASIC-11 be very small. Those of you who have worked in BASIC-11 know that the 12000 word ceiling on program size makes large program development difficult. Can you imagine having only 6000 words of program space?

An Alternative Approach

Six thousand words is a lot to give up to have this sort of facility. Is there some middle ground that can be devised? Is it possible to implement a program that will allow multiple line input and output, yet allow a user to move around within these lines to make corrections and modifications?

First, let us consider what this program must do. It must allow a section of the CRT screen to be used for input. This section consists of a number of lines on the screen. Data entry can be made anywhere within the defined space. The user can move around within the space using the cursor movement keys, the delete key, and the return key. Anything entered that is not a special key will be saved in a buffer and displayed in the current cursor position on the dis-

In over 100 installations, on five continents. . .

RSTS System Managers will sleep tonight.

LOCK-11

provides them with:

- **Comprehensive access control**
  (150 machine years without a verified breach)
- **Powerful system management tools**
  (that don't degrade the system they manage)

now distributed & supported by

ON TRACK SYSTEMS
921 Bethlehem Pike
Springhouse, PA 19477

215-542-7008

Circle the number below for a full set of documentation to evaluate.

RSTS SECURITY

'against all enemies,
foreign and domestic'

LOCK-11

IT'S 2:28 AM

Some kid with a MODEM just figured out that you have 1,100, DEMO

LOTS OF LUCK!

LOCK-11

IT'S 3:15 PM MONDAY

Tired of writing depreciation journals in 3,5,GL your third assistant bookeeper just discovered the joys of 4,0,PAY

He's on his way from the bank to the airport

LOTS OF LUCK!

LOCK-11

IT'S 2:28 AM

The kid with his auto-dial MODEM just found your "new" dial-in number 555-0112 on the 112th try
He's in and you are out

LOTS OF LUCK!

LOCK-11

IT'S 5:30 PM, FRIDAY

Your FORMER programmer just went home.

He dialed into a non-priv account, let himself in through a 'back door'. ([1,82]X.TSK (232))

He is now linking the bottom of [1,2] to the top with ODT. He is planning a couple of custom monitor patches.

He is not mad anymore. . .

LOTS OF LUCK!

LOCK-11

play. Overwriting a character on the display causes the character in the buffer in the proper position to be replaced by the new character. Likewise the new character is displayed on the terminal screen. Any attempt to move outside of the display area causes a terminal bell to sound.

This facility is a lot like a simple screen editor. Data can be entered anywhere on the specified lines and stored in an appropriate memory buffer. The buffer is made available to the program for processing after the user has finished data entry.

Such a program might be written as a GOSUB-type subroutine in BASIC-11. However, BASIC-11 cannot process single keyboard characters fast enough because it is an interpretive language and a great amount of processing is required for program translation. Another way to implement a screen input routine is to write the program in MACRO and incorporate it into BASIC-11 as an assembly language routine(ALR). An ALR would use space from the BASIC-11 user work area but be significantly faster than a routine written in BASIC-11 and would also require less memory than the FMS-11 system.

SCRNIN

SCRNIN came out of the need for an input routine that would allow a user to enter data and be able to change it easily. The data usually spanned a number of input lines, and available program space was limited.

The limitation of available programming space was a primary consideration in the design of SCRNIN. ALRs should use as little storage as possible. With SCRNIN the question was where the buffer area for the screen would go.

The VT100 screen contains 24 lines of 80 columns. This would require a buffer area of 1920 characters. SCRNIN is an ALR that is only used sometimes. Reserving 1920 bytes for a screen buffer would be a waste of program work area. Also, the space required for input by a program might be much smaller than 24 lines. The application that SCRNIN was developed for only required a maximum of three lines at any one time.

There are a number of possible places where buffer space might be obtained and then returned after SCRNIN is finished doing an input or output operation. For instance, it is possible to use BASIC-11's memory management routines to get a block of storage from the program work area. This tends to be very tricky but it is possible. The other alternative for buffer space is from the user program.

This last solution is by far the simplest and places the responsibility of storage allocation for the screen buffer area on the programmer. Fortunately BASIC-11 gives the programmer an easy method of allocating and deallocating storage. This method is based on BASIC-11's string data type.

When a string array is dimensioned no storage is allocated for it except a block of descriptor words. This descriptor word contains information about where in memory the strings can be found. It is not until a string array element is actually assigned a value that it uses any storage. This means that as long as elements of a string array are assigned the null string the only storage used for the array is that which is used for the descriptor word.

With string arrays we have a means of allocating a buf- fer area for screen input and returning the storage used by the buffer area to the user program area when screen input or output is completed. The string array used for the screen buffer is dimensioned in a DIM statement. When a screen buf- fer area is required the elements of this array are set to strings of 80 blanks(" "). The address of the each element of this buffer array is passed to the screen input/output ALR. The ALR uses the buffer space for screen input and output. After the ALR is finished processing the string array contains whatever was entered on the screen. The programmer can then manipulate this data, after which the space used by the string array can be returned to available storage by setting each element of the buffer array to the null string ("").

The Work of SCRNIN

SCRNIN has two entry points; SCRNLD and SCRNIN. SCRNLD is used to tell SCRNIN where each line of the buffer area resides. This routine must be called to load buffer string addresses each time they have been set to a new value because each time they attain a new value it may reside at a different memory location. If you fail to call SCRNLD and you have changed the value of a buffer array element BASIC-11 will terminate in the most disagreeable way with a TRAP to 4 or 10 depending on the color of the sun and the current phase of the moon.

SCRNLD takes two arguments LINE% and ARRAYLINE$. LINE% is an integer that is used by SCRNLD to determine where in the buffer array element address table (BUFRTB) the address of ARRAYLINE$ will be stored. In other words LINE% contains the line number for which ARRAYLINE$ is the buffer. ARRAYLINE$ must be initialized to a string of 80 blank characters before it is passed to SCRNLD. BUFRTB has room for 20 entries; i.e., 20 lines of screen input is the maximum allowable in this version of SCRNIN.

SCRNLD accesses the BASIC-11 argument block to get the appropriate values and addresses. Suffice it to say figuring out just how to get anything from BASIC-11's argument passing mechanism is a monumental task and would require many pages of description. It is beyond the scope of this article to describe why SCRNLD does what it does regarding argument passing, but information describing the mechanism is contained in the BASIC-11 User's Guide.

The second entry point to SCRNIN is SCRNIN. This is the entry point that facilitates screen input. This entry point is divided into the following sections:

1. Get the arguments.

SCRNIN takes four arguments. These arguments specify the starting line of the screen input, the total number of lines, and the mode of operation. The mode argument is necessary because SCRNIN will run in both RT-11 and TSX. For SCRNIN to function properly in TSX certain escape sequences are displayed by SCRNIN that cause TSX to operate correctly in the TSX environment.

2. Set necessary operating system parameters.

TSX escape sequences are output if mode specifies SCRNIN is running in TSX. Otherwise no special operation system characters are displayed. Note that these characters are not actually displayed on the terminal screen. They are intercepted by TSX and used to set operating characteristics. The escape sequence transmitted tells TSX that it is to process incoming data one character at a time.

3. Initialize the selected screen area and set any necessary terminal characteristics.

In this section SCRNIN moves the cursor to the specified screen input line, sets the terminal numeric keypad into application mode (described in VT100 User's Guide), sets the terminal so that it will not skip to a new line when it reaches the end of a line, and changes all display to reverse video.

After these settings occur, SCRNIN clears the selected portion of the screen and displays the contents of the screen buffer area in the specified portion of the CRT screen. After all of the buffer strings have been displayed the cursor is positioned on the first line, leftmost character to get ready for data input.

4. Process input keystrokes.

At this point any keystroke is either entered into the screen buffer or processed by a special section of code.

a. Store alphanumeric characters n screen buffer and display them to creen.

Any alphanumeric character ntered is stored in the screen buffer. CRNIN keeps track of the current curor position. As the cursor moves, ointers into the appropriate screen uffer element are modified. R3 lways contains the address of the curent character in the screen buffer rea. The current character is the haracter directly above the cursor. fter the character is stored it is splayed on the terminal screen.

If an attempt is made to go eyond the boundaries of the defined rea for screen input(specified by ALR guments) SCRNIN beeps. SCRNIN will eep at the top, bottom, left, and right argins if an attempt is made to go tside these limits.

b. Process cursor movement keys, lete key, return key, and enter key.

When SCRNIN encounters any of the keys listed above it takes an appropriate action depending on the key SCRNIN detects. The specific actions are described by the table below.

Key	SCRNIN Action
Up Arrow	Move cursor up one line if possible. Change buffer pointer to previous line. Beep if this is an attempt to move outside of screen input area.
Down Arrow	Move cursor down one line if possible. Change buffer pointer to next line. Beep if this is an attempt to move outside of screen input area.
Left Arrow	Move cursor left one character if possible. Change buffer pointer to previous character. Beep if this is an attempt to move outside of screen input area.
Right Arrow	Move cursor right one character if possible. Change buffer pointer to next character. Beep if this is an attempt to move outside of screen input area.
Return Key	Move cursor to beginning of next line if possible. Change buffer pointer to appropriate position. Beep if this is an attempt to move outside of screen input area.
Delete Key	Move cursor to previous character if possible. Delete it from screen. Delete it from buffer by making it a blank. Change buffer pointer to previous character. Beep if this is an attempt to move outside of screen input area.
Enter Key	Reset system and terminal parameters that were set when SCRNIN was called. Return to the calling program.

5. Reset operating system characteristics.

When the Enter key on the keypad is pressed, SCRNIN displays the escape sequences that return the operating system to it normal state. These sequences are emitted only if the mode indicates SCRNIN is running in TSX.

6. Reset terminal characteristics.

Terminal characteristics are also returned to normal when the Enter key is pressed.

Installation in BASIC-11

With this description and the listing you have all you need to do screen input in BASIC-11, right? Unfortunately, not yet. SCRNIN must be installed in BASIC-11. This means modifying one of the source modules supplied by DEC and relinking BASIC-11 with the SCRNIN module after you have run it through the MACRO processor.

The process of installing an ALR in BASIC-11 is described in the BASIC-11 User's Guide. The module you will need to modify is called BSCLI.MAC and is contained on one of the volumes of BASIC-11 distribution media. You may also have to run the BASIC-11 configure program to generate a BASIC-11 with CALL support if you don't already have this feature as part of your BASIC-11. When you install SCRNIN, make sure you define both SCRNIN and SCRNLD in BSCLI.MAC. BASIC-11 does not know about entry points within an ALR; you must define these for BASIC-11.

Once you have followed the installation procedures in the BASIC-11 User's Guide you will be ready to use SCRNIN.

Use

There are two steps in calling the SCRNIN routine. The first step involves defining the screen buffer addresses to the SCRNIN routine. The second step calls SCRNIN. You must do the screen buffer definition before calling SCRNIN.

Buffer Line Definition

To define a buffer line you call SCRNLD as follows:

CALL "SCRNLD"(LINE%,ARRAYLINE$)

LINE% is the line of the screen that ARRAYLINE$ will be used to store. LINE% must be an integer variable or an integer constant.

ARRAYLINE$ is a string element 80 characters in length that will be used to store characters input on line%. There must be one ARRAYLINE$ for every SCRNIN line.

SCRNIN Call

CALL "SCRNIN"(STARTLINE%,STARTLINE$,MAXLINES%,MODE%)

STARTLINE% is the starting line of the input field on the terminal screen. This parameter must be an integer variable or integer constant.

STARTLINE$ is the string representation of STARTLINE%. The routine uses this character string for screen control. STARTLINE$ must be a character string constant or character string variable.

MAXLINES% is the number of lines that the field will have. MAXLINES% must be an integer variable or an integer constant and cannot be greater than 20. SCRNIN does not check this parameter for validity. If it exceeds 20 SCRNIN control data will be overwritten causing a serious program error.

MODE% specifies the operating system under which SCRNIN is being used.

MODE% = 1% means that SCRNIN is running in a BASIC program in TSX.
MODE% = 0% means that SCRNIN is running in a BASIC program in RT-11.

MODE% must be an integer variable or an integer constant of value 0% or 1%.

If you are running SCRNIN in RT-11 you must SET TT: WIDTH=255. There is a system peculiarity that this avoids. Without using this command SCRNIN will not functio[n] properly.

Example:

```
500 .......
510 DIM B$(4)
520 REM
520 REM Blank out each buffer line and make each
520 REM 80 characters in length
520 REM
520 X$="" \ FOR I%=1% TO 80% \ X$=X$+" " \ NEXT I%
530 FOR I%=1% TO 4% \ B$(I%)=X$ \ NEXT I%
540 REM
540 REM Load the buffer line address 1 at a time
540 REM
540 FOR I%=1% TO 4% \ CALL "SCRNLD"(I%,B$(I%)) \ NEXT I
550 REM
550 REM Invoke SCRNIN
550 REM
550 CALL "SCRNIN"(10%,"10",4%,1%)
560 .......
```

This example shows SCRNIN in a small program segment. B$ is the buffer and line 510 dimensions B$ to have four lines of screen input. Lines 520 and 530 initialize every string in B$ to 80 characters (a requirement of SCRNIN). Line 540 sets up a table of buffer address so that as characters are entered they can be stored for access by BASIC. Line 550 invokes SCRNIN. The parameters specify that screen input will begin on line 10 of the terminal screen and input can be four lines long. The buffer is B$ and the mode is TSX. At line 560, B$ will contain any data entered in the screen area by the program user.

Bibliography

[1] Kurtz, T.E., "BASIC, in History of Programming Languages," Ed. Wexelblat, R.L., Academic Press, 1981, Pp 515-549.

[2] "BASIC-11/RT-11 User's Guide," Digital Equipment Corporation, DEC No. DEC-11-LIBUA-A-D, August, 1977.

[3] "Programmer's Information, VT100 User's Guide," Digital Equipment Corporation, DEC No. EK-VT100-UG-00[?] January, 1979.


```
        .TITLE  SCRNIN
;
; The purpose of SCRNIN is to allow screen input into a BASIC-11 program.
; SCRNIN uses a buffer defined in the BASIC-11 program. The buffer is a
; string array each element of which should be 80 characters. This buffer
; must be initialized to blanks before it is sent to SCRNIN.
;
; The cursor control keys left, right, up, down, and delete allow
; movement around the defined screen area. When a character is typed over
; on the screen it is replaced in the buffer.
;
; The routine is called
;
;       CALL "SCRNIN"(startline%,startline$,maxlines%,mode%)
;
; where startline% is where the cursor will be positioned for the beginning
; of input. The cursor will not be allowed to go above this line. A beep
; will occur if the user attempts to go outside the starting line boundary
; with the cursor up key. startline% must be an integer constant or an
; integer variable.
;
; startline$ is the character string representation of startline%
;
; maxlines% is the maximum number of lines of allowable input; i.e., how
; much of the screen the user can use. startline%+maxlines%-1% is the last
; line on the screen that can be used by the user. An attempt to go below
; this line with the down cursor key or RETURN will result in a beep telling
; the user he is attempting to go outside the lower line boundary. maxlines%
; must be an integer constant or variable.
;
; mode% specifies whether SCRNIN is being called from RT-11 BASIC-11 or
; TSX BASIC-11.
;
;
;                       mode% = 0% --> RT11 BASIC-11
;                       mode% = 1% --> TSX BASIC-11
```

```
;
;
;
        .PSECT  SCRNIN
        .GLOBL  SCRNIN,SCRNLD,$FIND
        .MCALL  .TTYIN,.TTYOUT,.PRINT
JSW=44                                  ; Job status word address
ESC=33                                  ; ESCAPE character
BRK=133                                 ; [ character
DEL=177                                 ; DELETE character
        .MACRO  BEEP
;
; This MACRO outputs a beep to warn the user he is doing something not to
; good.
;
        .TTYOUT #BCHR
        JMP     ?SDLP
        .ENDM
SCRNLD:
;
; SCRNLD is the SCRNIN entry to load the buffer table with the addresses
; of the buffer array lines. This is necessary because of BASIC-11's inability
; to pass string array arguments correctly.
;
; This entry is called:
;
; CALL "SCRNLD"(line$,arrayline$)
;
; where line$ is an integer whose value is the line number for which
; arrayline$ will be the buffer.
;
; arrayline$ is the array element used as the buffer line for line$.
; arrayline$ MUST BE INITIALIZED TO A STRING OF LENGTH 80
;
        MOV     -2(R5),R4               ; R4 points to the argument
                                        ; descriptor list
        MOV     2(R4),R4                ; R4 points to the arrayline$
                                        ; descriptor
        MOV     ?(R4),R0                ; R0 contains the arrayline$ string
                                        ; reference pointer
        MOV     #100,R1                 ;
        JSR     PC,$FIND                ; Find the string
                                        ;
        MOV     @2(R5),R2               ; Move the line number to R2
        DEC     R2                      ; Compute the offset into the
        ASL     R2                      ; buffer address table
                                        ;
        MOV     R0,BUFRTB(R2)           ; Move the address of the string
                                        ; to the buffer address table and
        RTS     PC                      ; return


;
; Get the program arguments
;
SCRNIN: MOV     @2(R5),STRTLN           ; Move the starting line argument
        MOV     @6(R5),NLINE            ; Move the number of lines argument
        MOV     @8.(R5),MODE            ; Move the operating mode to the
                                        ; mode variable
                                        ; this next address
;
; Get the startline string and move it to the starting line screen control
; sequence
;
        MOV     -2(R5),R4
        MOV     2(R4),R4
        MOV     2(R4),R0
        MOV     #100,R1
        JSR     PC,$FIND
        MOVB    (R0),LN
        MOVB    1(R0),LN+1
;
; Set .TTYIN/.TTYOUT characteristics - set for special mode and upper/lower
; case ( bits 12 and 14 ).
;
        BIS     #50000,@#JSW
;
; Check the mode and issue TSI sequence for single character mode if is "1"
;
        TST     MODE
        BEQ     RTS1
        .PRINT  #TSIES
;
; Initialize specified screen area
;
RTS1:   .PRINT  #MOVLN                  ; Move cursor to that line
        .PRINT  #ETAP                   ; Set numeric keypad in application
                                        ; mode
        .PRINT  #NOWRAP                 ; Put terminal in no wraparound
                                        ; mode
        .PRINT  #RVID                   ; Put terminal in reverse
                                        ; video mode
;
; Clear the specified lines on the screen
;
        MOV     NLINE R1                ; R1 will count lines
ERALP:  .PRINT  #ERALN                  ; Erase a line
        .PRINT  #CDRGN                  ; Move cursor to next line
        DEC     R1                      ; Decrement count
        BNE     ERALP                   ; Not zero then erase next line
        .PRINT  #MOVLN                  ; Move cursor back to the starting
                                        ; line
;
; Write whatever is in the BASIC-11 buffer area. Lines are terminated at
; 80 characters or when a return(015) is processed.
;
        MOV     NLINE,R1                ; R1 will count lines
        MOV     #BUFRTB,R3              ; R3 will point to the buffer
                                        ; address table
                                        ;
LNLP:   MOV     #80.,R2                 ; R2 will count characters
        MOV     (R3)+,R4                ; R4 will point to the string
```

```
                                        ;
DSPLP:  CMPB    (R4),CR                 ; Is the character a carriage return
        BEQ     NXTLN2                  ; Yes then go to next line
                                        ;
        MOVB    (R4)+,R0                ; Otherwise display the character
        .TTYOUT                         ;
        DEC     R2                      ; Reduce character count
        BNE     DSPLP                   ; Not zero yet then do another
                                        ; character
                                        ;
NXTLN2: DEC     R1                      ; End of line the reduce line count
        BEQ     RSCUR                   ; If zero lines then no more display
                                        ; to do
                                        ;
        .TTYOUT CR                      ; Otherwise output a return and line
        .TTYOUT LF                      ; feed and go to display the next
        BR      LNLP                    ; line of the buffer
                                        ;
RSCUR:  .PRINT  #MOVLN                  ; Move the cursor to the first line
                                        ; of the screen input area
;
; This is where the input is processed. The registers are used as follows -
;
;               R1 - current line relative to 1
;               R2 - current offset into line
;               R3 - buffer character address - used to store incoming
;                    non-control characters
;
        MOV     #1,R1                   ; Set R1 to first line
        CLR     R2                      ; R2 is at first character
        MOV     BUFRTB,R3               ; R3 is at first string array element
                                        ; character
                                        ;
CHRLP:  .TTYIN                          ; Get input
        CMPB    R0,#ESC                 ; Is it an escape ?
        BEQ     ESCPRO                  ; Go to process escape
                                        ;
        CMPB    R0,#DEL                 ; Is it a delete ?
        BNE     CRCHK                   ; No then maybe a carriage return
        JMP     DELPRO                  ; Go to process the delete
                                        ;
CRCHK:  CMPB    R0,CR                   ; Is it a carriage return ?
        BNE     CHRPRO                  ; No then its not a control
                                        ; character
        JMP     CRPRO                   ; Go to process carriage return
                                        ;

CHRPRO: CMP     R2,#79.                 ; Outside line boundary ?
        BLT     INBDD                   ; No then process character as input
```

. . . continued on page 114

DEXPO EAST 84 — Boston
ADVANCE CONFERENCE PROGRAM

KEYNOTE SESSION *TUESDAY — 10:00 AM*

C. GORDON BELL ON DEC

Note: This session is free to all DEXPO East 84 attendees. No separate registration fee is required.

Speaker: **C. Gordon Bell** — Chief Technical Officer, Encore Computer; formerly Vice President of Engineering for Digital Equipment Corp.

Mr. Bell will assess the latest developments within Digital from the point of view of a long-time insider. His insights into the future directions of DEC-related technology should be of special interest to every DEC user.

Technology Trends

SESSION 1.1 *TUESDAY — 2:00 PM*

TRENDS IN DATA BASE MANAGEMENT SYSTEMS

Speaker: **Lance Prager** — President, Information Systems Consulting Associates
Masters of Computer Science Candidate – Brown University

This presentation will explore what the latest developments in the field of Data Base Systems are and how methods for requirements specification should be formulated when a decision to acquire a DBMS has been made. The presentation will focus on the three types of DBMS offered today, CODASYL, Hierarchical and Relational and the type of processing environment for which they are best suited. A case study for the selection process which one firm experienced over a two year period will illustrate all of the important issues involved with a decision of this magnitude and the effects which transpired on the users, technical staff and support personnel.

Mr. Prager will speak about Datatrieve* as a database tool for developing entire application packages with a consistent user friendly interface, which can be used by business professionals for quick retrieval and formatting of critical information, on an as needed basis.

*Registered trademark of Digital Equipment Corp.

SESSION 1.2 *TUESDAY — 2:00 PM*

TRENDS AND DIRECTIONS WITHIN DEC

Speakers: **Sonny Monosson** — President, Monosson Technology Enterprises, Boston, MA
Steve Tatum — Televideo
Ken O'Mohundro — President, Able Computer
Bill Wollam — MDB
Tom Evans — Evans, Griffiths & Hart
Bob Goety — System Industries
Don Rudolf — National Semiconductor

DEC-compatible vendors discuss their perceptions of DEC's product strategy in an open forum format. The impact that the installed customer base will feel, and which options will remain viable for system growth will be the focus of this session. Questions and free-flowing discussion will be encouraged thorughout the session.

Information Systems Management

SESSION 2.1 *WEDNESDAY — 9:00 AM*

MANAGING SOFTWARE DEVELOPMENT PROJECTS — PART 1

Speakers: **Donald Job** — Information Systems Consultants Associates
Gerry Tolman — Independent Consultant

This session will cover the myriad of DEC-compatible software packages for the full range of DEC systems. Discussion will focus on how to evaluate packaged software systems for the critical elements of design, functionality, ease of use and support. The "make versus buy" decision evaluation will be discussed from the perspective of resource utilization and payback considerations for special projects.

Many new packages have been introduced for the personal computer line, but before your firm buys new packages find out about the tools available to migrate your exising PDP-11* and VAX* applications into the new personal operating system environment. This can prove to be a cost saving method to employ the new technology, while maintaining the quality of current systems.

SESSION 2.2 *WEDNESDAY — 2:00 PM*

MANAGING SOFTWARE DEVELOPMENT PROJECTS — PART 2

Speakers: **Monica Collins** — Independent Consultant, Worcester, MA
Arthur Greenberg — Information Systems Consulting Associates, Lexington, MA

Ms. Collins has served as a chairperson of the IRUS organization and currently is working as a project management consultant to Digital Equipment Corp. on a large internal development project. She will address the subject of completing a project on time and on budget. Analyzing the business requirements of a project for a functional design in terms of data flow diagrams. Creating detail task lists and making proper estimates of the required people and hours to be expended. From the functional design and proper estimates, milestone schedules for ma[illegible] project completion can be established. Using these tools to augment the technical analysis [illegible] design phase, work can be completed in a timely and effective manner. Methods for testing [illegible] completed modules on several levels will be examined so as to ease the implementation [illegible] new system. Ongoing support and future enhancement methods for the developed project [illegible] also be examined.

Mr. Greenberg will examine the pros and cons of using outside consultants for pro[illegible] assistance and management. When is the proper time to introduce outside resources in[illegible] project? And, how to take advantage of their specific areas of expertise. Tips on negot[illegible] contracts with consultants — when is the hourly rate with a ceiling amount preferred to a f[illegible] price contract — will be detailed and examined from the business managers perspective. T[illegible] session should prove valuable to the technical and non-technical manager alike, as an a[illegible] completing projects on time and on budget.

SESSION 2.3 *THURSDAY — [illegible]*

INSTALLATION MANAGEMENT OF DEC-COMPATIBLE SYSTEMS

Speaker: **Carlos Flores** — President, Superior Computer Systems; Chairman, IRUS

The market for Digital Equipment Corp. compatible products and services has grow[illegible] exceed six billion dollars annually, approximately half-again as large as all of Digital's total sa[illegible] As the Digital marketplace has been inundated with a plethora of new products, the configura[illegible] options seem endless. In this session Mr. Flores will examine typical configuration options [illegible] provide cost comparison information between Digital hardware and the compatible alternativ[illegible]

He will discuss just how compatible certain types of equipment are and what cautions sh[illegible] be observed before acquisition. When it's time to upgrade your DEC compatible system, [illegible] you can ensure it will remain fully compatible. Ideas for improving system and applica[illegible] performance by using hardware solutions will be presented. Once all the pieces are toge[illegible] and the most viable solution is determined, what maintenance options are now available? T[illegible] session should produce many money-saving ideas.

SESSION 2.4 *THURSDAY — [illegible]*

SECURITY PLANNING IN NETWORK DESIGN

SESSION 2.5 *THURSDAY — [illegible]*

PLANNING FOR DISASTER RECOVERY

Speakers: **Leslie D. Ball Ph.D** — Associate Professor Babson College, Wellesley, MA
William Dreyer — Iron Mountain Group, Boston, MA

Mr. Ball has written over 30 articles on information systems and computer security; foun[illegible] chairperson of ACM's special interest group in security, auditing and control; former chairpe[illegible] of Boston SMIS, and has consulted for Cullinane Database Systems, Wang Labs, Pr[illegible] Computer, Colonial Gas Energy Systems and Nixdorf Computer Corp. As a speaker on *Plan[illegible] for Disaster Recovery*, he will discuss the impact of a computer disaster on the operations [illegible] business; the value, contents and implementation of a plan; case studies of computer disa[illegible] and issues and experiences regarding computer fraud.

Mr. Dreyer will discuss a completely documented, industry sanctioned case study of [illegible] financial impact to a firm which suffered a complete data site disaster. Bill will explain the diffe[illegible] levels of protection available to small and large firms for off-site data storage, reciprocal pro[illegible] sing agreements and shell processing arrangements which are available for immediate res[illegible] tion of data processing activity.

SESSION 2.6 *FRIDAY — [illegible]*

"ASK THE EXPERTS": SHOULD WE GO TO A VAX?

Panel Discussion: **Bob Voelk** — President, ASA Labs
Jim Littlefield — Sys. Mgr., Aquidneck Data Corp.
Monica Collins — Independent Consultant
Bill Coughlin — Sys. Mgr., Teradyne Inc.
John Santos — Sys. Prog., Evans, Griffiths & Hart

Each panelist will present a brief series of remarks about his particular experience [illegible] conversions to a VAX system, and the methodology or tools which were employed to accom[illegible] the migration successfully. Then a general discussion among the panelists, in the conte[illegible] the topic question, will follow. The session will conclude with an extended question and an[illegible] period where the attendees can "ask the experts" about their own situations.

The attendees will benefit from the experience which these panelists have in the [illegible] product line in addition to other DEC systems. Performance improvements for application[illegible] well as system performance concerns will be the focus of the discussion. Also for those atten[illegible] seriously considering an upgrade or switch, the tools and tricks these panelists have emp[illegible] will provide the basis for some significant time and money saving ideas.

Personal Computers

SESSION 3.1 *WEDNESDAY — [illegible]*

AN OVERVIEW OF DIGITAL'S PERSONAL COMPUTER SYSTEMS

Speakers: **Carl Marbach and Dave Mallery** — Publishers of *Personal & Professional*, *DEC Professional* and *VAX/RSTS Professional* magazines

When Digital Equipment Corp. announced their new personal computer lineup they [illegible] just announce one computer, they presented three: the Rainbow, the DecMate II and [illegible] Professional 300 series. While they all look alike, they are very different inside; they are [illegible] three very distinct computers which fit into three different marketplaces. These compute[illegible] not a sequence of "good", "better" and "best", but rather a single package that embodies [illegible] separate processors, thereby offering three distinct solutions. These two "pros" of pers[illegible] computing will provide an overview of each system and describe their relative capabilities [illegible] advantages. Typical configurations for implementing the different types of software which [illegible] is best suited for will be evaluated as they relate to the different intended marketplaces.

This session will be tutorial and questions and discussion will be encouraged during [illegible] last hour.

SESSION 3.2 *WEDNESDAY — [illegible]*

INTEGRATED TOOLS FOR PERSONAL COMPUTING

Speaker: **Henry T. Cochran** — President, Henco Software, Waltham, MA

Mr. Cochran will explore the personal computing phenomenon from the perspective [illegible] business professional. He will focus on defining the problem to be solved and identifying [illegible] type of personal computer resources available as potential solutions. An explanation of the tools available in the personal computer arena for solving problems on your own terms will relate the available resources to specific business situations.

How to integrate personal computing into the business by organization by functional areas within a firm, such that they support decision making effectiveness. The session will next address technological advances and the business professional; the use of software tools to integrate multiple applications into a coherent system which allows the linking of data used in independent applications. And how a completely integrated personal computer system can be used to maximize productivity and information retrieval. The ease of use of these integrated application systems through a consistent user interface will provide the business professional with a complete set of tools to move into the world of personal computers comfortably.

SESSION 3.3 *THURSDAY — 9:00 AM*

PRO 300 SERIES WORKSHOP

Digital Equipment Corporation Professional 300 series personal computers will be available for hands-on work and demonstrations. The topic presenters from previous sessions will be available for a discussion of particular areas of interest to each attendee. Packages to be presented will include all new software running under RT-11 on the Professional 300 series and will feature CAD/CAM software presented in the companion session on this subject. This workshop will be differentiated from the other workshops by virtue of the operating system orientation and the application focus.

SESSION 3.4 *THURSDAY — 2:00 PM*

PERSONAL COMPUTER WORKSHOP: ADVANCED SOFTWARE APPLICATIONS

Digital Equipment Corporation personal computers will be available for hands-on work and demonstrations. The topic presenters from previous sessions will be available for a discussion of particular areas of interest to each attendee. Packages to be presented will include the VAX-RSTS/E application migration tool kit and development workbench. With these tools programmers can perform development work on a large minicomputer, taking advantage of its inherent features and speed, then transport the final result to a personal computer for local distributed execution.

SESSION 3.5 *FRIDAY — 9:00 AM*

RAINBOW 100 WORKSHOP

Digital Equipment Corporation Rainbow 100 personal computer systems will be available for hands-on work and demonstrations. The topic presenters from previous sessions will be available for discussion of particular areas of interest to each attendee. Packages to be presented include word processing, LOTUS 1-2-3 and database management migration. Additionally, financial applications, graphics and other exciting software will be shown.

Data Communications

SESSION 4.1 *WEDNESDAY — 9:00 AM*

COMMUNICATION ALTERNATIVES OVERVIEW

Speakers: **Edd Lemoine** — N.E. Regional Manager, Halcyon Inc.
Ken Stiles — New England Telephone Co.
Barbra Paull — AT & T Information Systems

This topic will focus on implementation strategies for communication networking. An explanation of the basic equipment and concepts will be provided to acquaint the attendees with communication terminology. Representatives from the new operating units of the telephone company will address how their new competitive posture will benefit the users of local and long-distance transmissions. At the conclusion of this session the attendees will be familiar with all of the commonly used data communication terminology and will have an understanding of the form and function of networking computer systems and remote link terminals.

SESSION 4.2 *WEDNESDAY — 2:00 PM*

TRENDS IN COMMUNICATIONS

Speaker: **Stuart Wecker** — President, Technology Concepts Inc.

This session will help software professionals quickly update themselves on new trends in networking. Specifically, the presentation will cover:

- Trends in communications concepts;
- Analysis of communications carriers;
- The latest manufacturers' network offerings;
- Local Area Networking.

The speaker is one of this country's foremost authorities in the field of computer communications networks. Mr. Wecker has responsibility for the architectural design of distributed data/voice PBX and for the definition and protocols of Digital Equipment Corporation's Digital Network Architecture. In this role, he developed the design for DECnet and directed the initial specifications of the DDCMP, NSP and DAP protocols. He is currently working on the architecture for a large reservation transaction message switch. In addition, Mr. Wecker has been a visiting lecturer in the Computer Science Department of the University of California at Berkeley and at the IBM Systems Research Institute, he has authored many technical papers on computer networking and has given numerous talks at computer conferences and workshops.

Systems Software

SESSION 5.2 *WEDNESDAY — 9:00 AM*

OPERATING SYSTEMS UPDATE

Overview of the UNIX, VMS, RSTS, RT-11 and MS/DOS operating systems for DEC computer systems. What is the status of each latest version and what to expect in the next versions will be discussed by a panel of IRUS members who have extensive experience with these products. Questions will be encouraged throughout this session.

SESSION 5.3 *WEDNESDAY — 2:00 PM*

"ASK THE EXPERTS": DATA BASE MANAGEMENT SYSTEMS

Panel Discussion: Panelists to be announced

Each panelist will present a brief series of remarks about his particular experience with using, selecting and installing or managing data base systems. Issues such as maintaining data integrity and minimizing data redundancy will be presented from the DBMS administrator's perspective. Selection criteria for special purpose applications will be addressed with an illustrated case study. And the user perspective will be discussed by an expert in distributed database systems. A general discussion among the panelists of the DBMS options available to DEC users will provide the basis for a relative evaluation of different systems now being offered.

This will be followed by an extended question and answer session where the attendees can "ask the experts."

SESSION 6.1 *THURSDAY — 9:00 AM*

TRENDS IN OFFICE AUTOMATION

Speakers: **Wesley T. Gardner, Jr.** — Consultants to Management, Providence, RI
Hilda Greer — Samna Associates, Atlanta, GA

Mr. Gardner, former MIS Director with Laventhol and Horwath, will introduce the office automation discussion by stressing the importance of having complete and accurate manual procedures for the office data flow. He will examine the steps necessary to ensure a smooth and effective transition for both the human resources presently in place and the technological resources to be employed for the desired productivity improvements. The presentation which follows will explore the training aspect of office automation systems, defining the role of the Office Automation Manager and what functions this individual should be responsible for. The training topics will focus on the general approach to gaining users' acceptance with specific illustrations for word processing, spreadsheet and distributed database applications. This session will be oriented to business managers and other users of office automation systems. Questions and discussion will be encouraged during the session.

Office Automation

SESSION 6.2 *THURSDAY — 2:00 PM*

STRATEGIES AND IMPLEMENTATION APPROACHES IN OFFICE AUTOMATION

Speakers: **Glenn McCaffery** — Evaluation and Planning Systems, Windham, NH
John Sadd — Frey Associates, Amherst, NH

This session will present two approaches to implementing an automated office system which can be easily used by business management persons. First the Decision Support System will introduce concepts relating to why there has been limited use of the computer as a decision support tool because the need to learn and to use traditional programming languages isolated the people who needed the information from the machine. The typical business manager has been frustrated by computer "languages" from the past in his attempts to describe his thought process to the computer. Traditional methods of working through systems analysts/programmers to develop a system to user specifications were entirely too slow for rapidly changing business situations. A system specifically designed for decision support, enables the business manager to communicate directly with the computer in office automation business terminology to extract current information in a meaningful way.

The presentation to follow will explore some of the issues raised by the development of Natural Language Query Systems. First, why is this technology valuable as an addition to the current capabilities of Office Automation systems? How can such a tool be a boon to management people in need of timely information, and to the data processing staff who must support them? Second, what capabilities should such a system have in order to be truly productive for the end-user, and to encourage an ever wider range of company personnel to take advantage of the information available to them?

Third, how can such a tool be applied to a particular company's existing database. What are the tasks facing those who must install, maintain and support the system? The discussion will demonstrate that a Natural Language Query System can be a valuable and practical contributor to the corporate decision-making process.

Education

SESSION 7.1 *TUESDAY — 2:00 PM*

COMPUTERS IN EDUCATION: CURRICULUM CHALLENGES FOR A NEW GENERATION OF COMPUTER USERS

Speakers: **Peter Kugel** — Director of Academic Computing, Boston College
Dr. Robert Hashway — Former Director of the Massachusetts Higher Education Authority; Digital Equipment Corporation Board of Directors
George Pendergast — Director of Computer Science, Tahanto Regional Schools

This topic will concentrate on the changes evolving in computer science curriculums at all educational levels, as technology advances place more and more affordable computers in the hands of an increasing number of students. As more students are introduced to data processing at much earlier stages, what must educators provide in order to continue the learning challenge? Will computer literacy programs produce a new generation of super students?

This session will provide thought provoking ideas and suggestions for educators and administrators to address during the balance of this decade. The approaches to curriculum content and design, now being tried and implemented, will assist the attendees in preparing for the challenges they face.

Graphics

SESSION 8.1 *THURSDAY — 2:00 PM*

GRAPHICS TECHNOLOGY OVERVIEW

Speaker: To be announced

This session will address development of a method for selection of a graphics system. Applications for graphics technology will be highlighted within this selection methodology. An update presentation will be included in the session to describe what the newest developments in this rapidly evolving field are and what trends will occur in the next year.

SESSION 8.2 *FRIDAY — 9:00 AM*

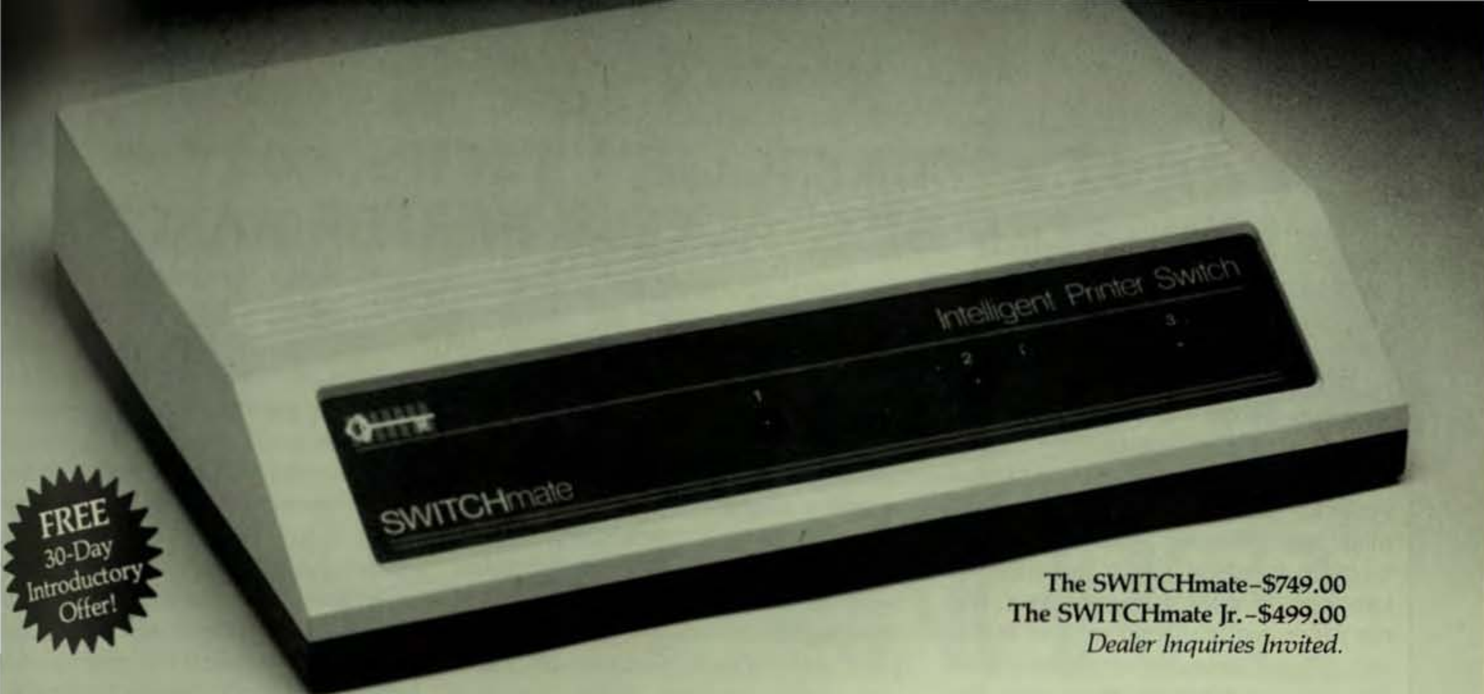
IMPLEMENTING A COMPUTER AIDED DESIGN SYSTEM ON A VAX AND PC 350

Speaker: **Ken Graves** — Kendall Co., Walpole, MA

Mr. Graves has served on the Executive Board of IRUS and has been a contributor of technical articles to the newsletter and industry publications. At Kendall Company, he is managing a team of analysts who are currently implementing a Computer Aided Design system on a VAX-11/750, with distributed PC350 systems in a local area network. Mr. Graves will discuss the methods and techniques employed by his team to implement this project.

 . . . continued from page 111

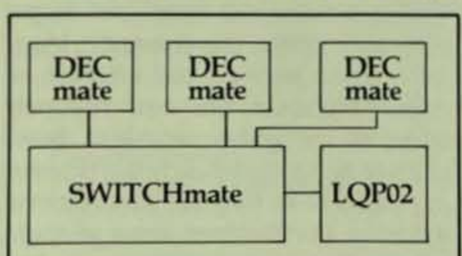
```
        .TTYOUT                         ; Echo the character
        MOVB    R0,(R3)                 ; Move the character to the buffer
        .TTYOUT BPCHR                   ; Beep at the user
        BR      CHRLP                   ; Go to process another character
                                        ;
INBD0:  .TTYOUT                         ; Echo the character
        MOVB    R0,(R3)+                ; Store the character in the buffer
        INC     R2                      ; Advance offset pointer
                                        ;
        BR      CHRLP                   ; Go to process another character
;
; Process escape sequences
;
ESCPRO: .TTYIN                          ; Get next input character
        CMPB    R0,#BRK                 ; Is it a bracket ?
        BNE     ENTPRO                  ; No then the key must be ENTER
                                        ;
        .TTYIN                          ; Get next input character
        CMPB    R0,#'A                  ; Is it cursor up ?
        BNE     KEY2                    ; No then go to check next possible
                                        ; key
                                        ;
        CMP     #1,R1                   ; Is current line top line ?
        BNE     INBD1                   ; No then in bounds
        BEEP                            ; Yes then beep at the user
                                        ; and go to process the next input
                                        ; character
                                        ;
INBD1:  .PRINT  #CURUP                  ; Move the cursor up
        DEC     R1                      ; Reduce current line register
        MOV     R1,R4                   ; R4 gets current line
        DEC     R4                      ; Compute offset into the
        ASL     R4                      ; buffer address table for this
        MOV     BUFRTB(R4),R3           ; line's buffer address
        ADD     R2,R3                   ;
        JMP     CHRLP                   ; Go to process next character
                                        ;
KEY2:   CMPB    R0,#'B                  ; Is it cursor down ?
        BNE     KEY3                    ; No then check another key type
                                        ;
        CMP     NLINE R1                ; Is current line bottom line ?
        BNE     INBD2                   ; No then go to move cursor
        BEEP                            ; Otherwise beep at that bad user
                                        ; and go to process the next input
                                        ; character
                                        ;
INBD2:  .PRINT  #CURDN                  ; Move the cursor down
DNSET:  INC     R1                      ; Advance line counter
        MOV     R1,R4                   ; Compute address of buffer for
        DEC     R4                      ; new line
        ASL     R4                      ;
        MOV     BUFRTB(R4),R3           ;
        ADD     R2,R3                   ;
        JMP     CHRLP                   ; Go to process next input character
                                        ;
KEY3:   CMPB    R0 #'C                  ; Is it cursor right ?
        BNE     KEY4                    ; No then check another key
                                        ;
        CMP     R2,#79.                 ; Are we at the end of the line ?
        BLT     INBD3                   ; No then go to move the cursor
                                        ;
        .TTYOUT BPCHR                   ; Beep at the user
        JMP     CHRLP                   ; Go to process next input character
                                        ;
INBD3:  .PRINT  #CURR                   ; Move the cursor right
        INC     R2                      ; Advance offset
        INC     R3                      ; Advance buffer pointer
        JMP     CHRLP                   ; Go to process next input character
                                        ;
KEY4:   CMPB    R0,#'D                  ; Is it cursor left ?
        BNE     NOKY                    ; No then we do not know what key it
                                        ; is
                                        ;
        TST     R2                      ; At beginning of line ?
        BNE     INBD4                   ; No then go to move the cursor
        BEEP                            ; Beep at the user and go to process
                                        ; the next input character
                                        ;
INBD4:  .PRINT  #CURL                   ; Move the cursor left
        DEC     R2                      ; Decrement the line offset
        DEC     R3                      ; Decrement the buffer offset
        JMP     CHRLP                   ; Go to process the next character
                                        ;
NOKY:   BEEP                            ; Don't know what the key is so
                                        ; beep at the user and go to process
                                        ; the next input character
;
; Process the enter key - this key terminates SCRNIN input
;
ENTPRO: .TTYIN                          ; Get the next input character
        CMPB    R0,#'M                  ; Is it an M
        BNE     NOKY                    ; No then we don't know what the
                                        ; key is
                                        ;
        .PRINT  #KYNM                   ; Set the numeric keypad back to
                                        ; numeric mode
        .PRINT  #WRAP                   ; Restore terminal to wraparound mode
        .PRINT  #NVID                   ; Restore terminal to normal video
                                        ;
        BIC     #50000,@#JSW            ; Set the .TTYIN and .TTYOUT back
                                        ; as they were
;
; Check the mode again and if in TSX reset the single character mode
;
        TST     MODE
        BEQ     RT02
        .PRINT  #TSXT
;
RT02:   RTS     PC                      ; and return to BASIC-11
;
; Process the delete key
;
DELPRO: TST     R2                      ; At beginning of line ?
        BNE     INBD5                   ; No then process then delete
                                        ;
        BEEP                            ; Otherwise beep at the user and
                                        ; go to process the next input
                                        ; character
                                        ;
INBD5:  .PRINT  #DELSEQ                 ; Do the delete on the screen
        DEC     R2                      ; Backup the offset
        MOVB    #40,-(R3)               ; Backup the buffer pointer and
                                        ; blank the character
        JMP     CHRLP                   ; Go to process the next input
                                        ; character
;
; Process carriage return
;
CRPRO:  CMP     NLINE,R1                ; At bottom line of input area ?
        BNE     INBD6                   ; No then go to do the return
                                        ;
        TTYIN                           ; Get the line feed character that
                                        ; belongs to the carriage return
                                        ;
        BEEP                            ; Beep at the user and go to process
                                        ; the next input character
                                        ;
INBD6:  CLR     R2                      ; Set R2 to the beginning of a line
        .TTYOUT                         ; Output return
        .TTYIN                          ; Input line feed
        TTYOUT                          ; Output the line feed
        BR      DNSET                   ; Go to set the new buffer address
;
; Data Areas
;
STRTLN: .WORD   0                       ; starting line variable
NLINE:  WORD    0                       ; number of lines variable
MODE:   .WORD   0                       ; system mode variable
BUFRTB: .BLKW   20.                     ; buffer address table
                                        ;
MOVLN:  .BYTE   ESC,BRK                 ; Move cursor to starting line
LN:     .BLKB   2                       ; screen control sequence
        .ASCII  /;01H/                  ;
        .BYTE   200                     ;
                                        ;
CURDN:  .BYTE   ESC,BRK                 ; Cursor down screen control
        ASCII   /01B/                   ; sequence
        .BYTE   200                     ;
                                        ;
CURUP:  .BYTE   ESC,BRK                 ; Cursor up screen control sequence
        .ASCII  /01A/                   ;
        .BYTE   200                     ;
                                        ;
CURL:   .BYTE   ESC,BRK                 ; Cursor left screen control sequence
        .ASCII  /01D/                   ;
        .BYTE   200                     ;
                                        ;
CURR:   .BYTE   ESC,BRK                 ; Cursor right screen control sequence
        .ASCII  /01C/                   ;
        .BYTE   200
                                        ;
ERALN:  .BYTE   ESC,BRK                 ; Erase a line screen control sequence
        ASCII   /0K/                    ;
        .BYTE   200                     ;
                                        ;
KYAP:   .BYTE   ESC                     ; Set numeric keypad to application
        .ASCII  /=/                     ; mode sequence
        .BYTE   200                     ;
                                        ;
KYNM:   .BYTE   ESC                     ; Set numeric keypad to numeric mode
        ASCII   />/                     ; sequence
        .BYTE   200                     ;
                                        ;
NOWRAP: .BYTE   ESC.BRK                 ; Set terminal in no wraparound mode
        .ASCII  /?7/                    ;
        .BYTE   154,200                 ;
                                        ;
WRAP:   .BYTE   ESC,BRK                 ; Set terminal in wraparound mode
        .ASCII  /?7/                    ;
        .BYTE   150,200                 ;
                                        ;
BPCHR:  .BYTE   7                       ; Beep character
                                        ;
DELSEQ: .BYTE   10,40,10,200            ; Delete screen control sequence
                                        ;
CR:     .BYTE   15                      ; Carriage return character
LF:     BYTE    12                      ; Line feed character
                                        ;
TSXS:   BYTE    35                      ; TSX single character mode on
        .ASCII  /S/                     ; sequence
        .BYTE   200                     ;
TSXT:   .BYTE   35                      ; TSX single character mode off
        .ASCII  /T/                     ; sequence
        .BYTE   200                     ;
                                        ;
RVID:   .BYTE   ESC,BRK                 ; VT100 reverse video
        .ASCII  /7/                     ;
        .BYTE   155,200                 ;
NVID:   .BYTE   ESC.BRK                 ; VT100 normal video
        .ASCII  /0/                     ;
        .BYTE   155,200                 ;
        .END
```



The SWITCHmate–$749.00
The SWITCHmate Jr.–$499.00
Dealer Inquiries Invited.

The DECmate Just Got Better.

Introducing The SWITCHmate, Intelligent Printer Switch.



Why fight for the office LQP02 printer when there's the SWITCHmate,™ Intelligent Printer Switch from Gold Key Electronics. Now, you have the ability to form a multi-workstation system with your Digital DECmate family of business computers that can save you time and money. With the SWITCHmate, up to three Digital systems, whether it's the DECmate I,™ or DECmate II,™ or any combination of them, may share one LQP02 printer.

FULLY AUTOMATIC. Using the SWITCHmate is the same as normal printing. The operator simply prints a document in the standard fashion and the SWITCHmate will handle your print request automatically. Each operator may have up to eight documents awaiting the SWITCHmate to print.

DESIGNED FOR THE OFFICE. Because the SWITCHmate is custom designed, it has the features a working office needs:

- Integrated into your Digital system, operators may perform print operations without regard for the current printer activity.
- The SWITCHmate operates unattended. Once installed, no operator attention is required.
- Status lights indicate current activity and those workstations which have a print request pending.
- You may suspend the sharing of the printer and dedicate it to a single DECmate without leaving your workstation.

30-DAY INTRODUCTORY OFFER. Gold Key is now offering a satisfaction guaranteed program for the 3-user SWITCHmate and 2-user SWITCHmate Jr.™ If you are not satisfied, just return the unit within 30-days with no obligation to buy.

SERVICE AND WARRANTY. Service for the SWITCHmate is provided directly by Gold Key. During the 90-day warranty period, from date of purchase, a failing unit will be replaced via Air Express to your door. A toll free number is available 24 hours a day. This warranty service may be extended on a yearly basis at a low cost.

Gold Key ELECTRONICS, INC.

P.O. BOX 186, GOFFSTOWN, N.H. 03045
TELEPHONE: (603) 625-8518

CIRCLE 0183 ON READER CARD

A NEW CONTROLLER TECHNOLOGY ENHANCES DISK ACCESS PERFORMANCE

By Richard Scott, Acceleron, Santa Clara, California

Disk drive data storage performance has become a crucial factor in computer system throughput and response time. Early model computer systems had slow processors, small main memories, communications and terminal handling which required the intervention of the main processor, and ran programs that were smaller in size and did not interact very much with the user. In addition, there was one other key difference — individual disk drives stored much less data and generally had only a small physical area of the disk being actively accessed during any short period. Consequently, the causes of poor system performance were not generally poor disk performance. This enabled disk and system designers to abandon fast, small capacity, fixed disks that had many immovable read/write recording heads, and to implement economical, large capacity disks which had one set of read/write recording heads mounted on a movable arm. These heads could be mechanically repositioned across the new disks on a periodic basis when the area of predominate activity shifted.

What has changed with the new "high performance" systems and disk drives? None of the individual devices within the computer system has been made slower. Basically, the relative performances of different elements of the system have been altered by their different rates of technological advance. With the introduction of faster processors, larger main memories and improved communication interfaces, user available main processor power increased dramatically. The price and performance for processors has improved in excess of 30 percent per year. The amount of main memory installed in minicomputers in particular has recently been growing at a rate in excess of 70 percent per year. Late model intelligent communication interfaces equipped with separate semiconductor memory buffers have virtually eliminated the 10 to 20 percent of processor utilization required with earlier interfaces in communication intensive environments. In general, the decline in price per system function of these components has been greater than the decline in price of disks and other data storage peripherals. This has caused many site managers to invest more heavily in non-disk system elements than was the case with earlier systems.

In addition, the introduction of improved system software incorporating virtual memory management with adaptive, self-tuning characteristics has accommodated more and larger programs within the processor. Many tasks which were not cost effective on earlier machines are now routinely added to the system workload. Some of these tasks are not so much processing intensive as they are data retrieval intensive. Furthermore, many of them interact heavily with on-line users. The result is that while earlier system bottlenecks which masked the limitations of disk performance have been removed, the burden on individual disk performance and the visibility of that performance has increased.

How has disk performance improved to match other system improvements? The answer is determined by focusing on the most important aspects of disk performance — the access and transfer times.

ACCESS AND TRANSFER TIMES

The access time is the period between the time the processor requests a piece of information on the disk and the time the first bit of the requested information is transferred to the processor. There are two predominate components to the access time — the seek time which is the time to move the mechanically actuated read/write heads to the correct location and the rotational latency time which is the time for the first bit of information to rotate under the correctly positioned head.

The transfer time is the period between the time the first bit of information is transferred and the time the last bit of information is transferred. It depends on the density of the information storage on the disk, the rotational speed of the disk, and the length of the information record to be transferred.

The transfer time for the latest disks has improved along with the improvement in the data transfer rate from 180 kilobytes (thousands of eight-bit words) per second in the late 1960s by a factor of ten or more so that current, large capacity disks transfer at a rate well over a million bytes per second even in minicomputer and many advanced microcomputer based systems. These transfer rates are adequate to the task of transferring large blocks of information without interfering with the performance of most systems.

The access time has not improved at the same pace as the transfer time or other system parameters. The average rotational latency for data access has remained largely constant at 8.33 milliseconds. In fact, in some cases where the transfer rate threatened to outstrip the processor's ability to accept data, the rotational rate has actually been slowed. The mechanical seek time required to position the heads has improved, but only by a fac tor of two or three from early model disk drives. During periods of high usage and poor disk performance, seek times can be the source of between 50 and 80 percent of the access time. This condition is generally known as "arm contention" because different users are contending for the use of the mechanical arm or actuator which controls the position of the read/write heads.

The dramatic improvement in the density of data recording on disks as illustrated in Figure 1 has resulted in a greater number of user files on the same disk. With faster processors and more users on the same system during peak loading periods, each disk is subject to greatly increased activity. The result is an overall increase in the incidence of arm contention and a corresponding decrease in the response time of the disk to system requests despite the improvements in seek time.

Of course, the nature of a particular computer site's data processing characteristics determines the conditions under which disk arm contention will occur. It may occur on disks as small as 150 megabytes; it is quite common on 300 and 600 megabyte disks. A standard industry rule of thumb is that beyond 600-800 megabytes, two actuators each accessing a separate 300-400 megabyte area is required to avoid unacceptable arm contention problems. The incidence of arm contention is independent of the physical size of the drive — 14, 8 and 5¼-inch Winchester disks all have the same problem when their capacities become large enough to make arm contention probable. Smaller physical disk dimensions do not necessarily translate to faster average seek times.

There are basically five ways to improve the disk arm contention problem:

1. Faster mechanical actuators. This requires electro-mechanical improvements in the disk actuator. However, the ability of the disk manufacturers to build large capacity disks with faster, single actuator seek times is reaching the practical and economic limits of actuator technology.

2. Multiple actuators. Put additional actuators on each disk drive or use multiple smaller capacity disks.

Unfortunately, these solutions raise the manufacturing cost and reduce the reliability of the resulting storage solution. The manufacturing cost of an actuator can represent 20 percent of the total cost of a disk. Alternatively, the use of multiple, small capacity disks generally raises the resulting price per unit of data storage higher than that of a single, large capacity unit. The reliability of a storage unit is halved if two disk units or actuators are used instead of one, even if the reliability of the disks or actuators is equivalent. In any case, the additional actuators reduce, but may not eliminate, the incidence of arm contention.

3. Access request scheduling, also called "queuing." It requires that several disk access requests be known in advance and that the order in which the requests are met is unimportant. The requests are then reordered so that the actuator will move the minimum distance between accesses and thereby achieve the minimum average seek time.

Unfortunately, this method is ineffective in typical systems environments because at the instant the seek decision is being made the number of outstanding requests is generally small even during periods of high utilization with arm contention. For example, one or two requests may be outstanding, rarely three or more. The reason is that most programs and system tasks make a disk access request and then suspend operation until that request is filled. Then these programs operate for a significant period of time before making another request. As a consequence, at the instant the request is made the actuator is frequently positioned at the wrong location or already committed to a seek for a request from another program made just prior to the arrival of this request with no other requests outstanding.

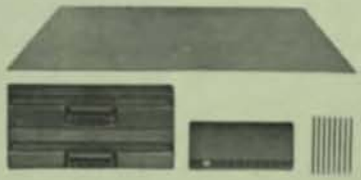
4. Putting high use data in a separate, faster memory. This is now generally accomplished using high speed, semiconductor random access memories. This technique, commonly termed "caching," is effective only when a significant portion of the data access requests are to a very small area on the disk. Until recently, this was effective in many systems with limited processor memory and software operating systems which required numerous disk accesses to accommodate larger programs. Later model computer systems have very large main memory capabilities and virtual memory software operating systems capable of fully utilizing this main memory to minimize the number of disk accesses required for program execution. In effect, the main memory of these systems is used for inexpensive disk caching, making a separate, external disk cache less effective. However, caching is still an important method of improving disk access speed in smaller systems.

5. Relocating high use data on the disk. This can be accomplished by either of two means. The site manager in charge of the computer system can periodically monitor the relative usage of various files and then use standard operating system utilities to locate frequently used files close together at the center of the disk. Or a separate machine such as the disk controller could theoretically be used to relocate the data automatically while responding

"Your Best Selection"

10MB - 140MB WINCHESTER-FLOPPY

LSI-11 LSI-11



RX02 FLOPPY

Dual drive, double density, double sided 2MB floppy (8") with controller and bootstrap. Also reads single sided, single density! $2695.

EMULATION RL02

10MB Winchester/Floppy (8") $4495.
20MB Winchester/Floppy (8") $4995.
40MB Winchester/Floppy (8") $5995.

EMULATION RK06/RK07

42MB Winchester/Floppy (8") $6995.
70MB Winchester/Floppy (8") $7195.
140MB Winchester/Floppy (8") $8195.

Q-BUS MEMORY

- Control Status Register (CSR)
- On board parity generator checker.
- Battery back-up mode.

SINGLE QTY. PRICE
256KB $525.

NEW! NEW!

Q-BUS MEMORIES

4 MB QUAD

- Control status register
- On board parity
- Block mode DMA

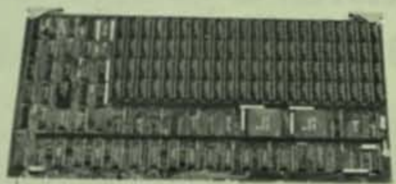
SINGLE QTY. PRICE
1MB $1975.
4MB $9975.

2MB QUAD EDC

- Error detecting & correcting
- Block mode DMA
- Battery back-up mode

SINGLE QTY. PRICE
512K $1495.
2MB $5195.

MULTIBUS MEMORY



- Error detecting & correcting (EDC)

SINGLE QTY. PRICE

	Without/EDC	W/EDC
512KB	$ 895.	$1495.
2MB	$4775.	$5775.

MICROCOMPUTER
CI-MICRO-11 SYSTEM FEATURES

- 11/23 PLUS CPU board which includes two serial lines, diagnostics and boot.
- 22 bit addressing backplane and power supply.
- 256KB - 4MB of parity memory.
- RD51 10MB, 20MB or 40MB 5¼" (13.3cm) mini winchester disk.
- RX50 1.6 MB 5¼" (13.3cm) dual mini-floppy disk.
- An eight slot quad LSI-11 BUS backplane.

STANDARD CONFIGURATION

LSI 11/23 CPU, 256KB memory, 20MB winchester, 800KB dual floppy, 2 serial I/O, bootstrap, power supply, 4 x 8 backplane all in a rack mountable chassis. $6850.

CALL FOR ASSISTANCE ON THE (RAINBOW) PC100 SERIES

"OFFERING QUALITY WITH AFFORDABLE PRICING"

Chrislin Industries, Inc.

31352 Via Colinas • Westlake Village, CA 91362
Telephone: 818-991-2254 • TWX 910-494-1253 CHRISLIN WKVG

CIRCLE D153 ON READER CARD

to the data access requests from the computer system. This method would be particularly attractive using the new high capacity, very reliable, non-removable media disks called Winchesters.

AIM/300

Acceleron, a Santa Clara, California based company, has developed the concept of on-line, automatic data relocation (patent pending), refined it and implemented it in a high performance Winchester disk drive. The drive called the AIM/300 combines proprietary, high speed data relocation with normal systems operations. It consists of a 315 (256 formatted) megabyte Winchester disk with an embedded single board MASSBUS adapter containing the logic and semiconductor memory necessary to emulate an RM05 drive and accomplish its own performance mission. In addition, the AIM/300's relocating capability does not require any special power failure or recovery precautions because the data and their physical locations are always stored on the magnetic media of the disk itself. In cases of power failure, these locations are recovered by the AIM/300 during its normal power-up and self-test sequence.

All of the data accessible at a single physical position of the actuator is commonly called a "cylinder." Even during periods of heavy system utilization, the AIM/300 can relocate two cylinders of data to each other's position on the disk in 5.3 seconds including the time spent read checking the data bit by bit for absolute integrity. After relocation to adjacent positions, the seek time between two actively used cylinders can be reduced to 3.6 milliseconds from a typical "arm contention" seek time of 29 milliseconds. AIM/300 also employs high speed caching. Both standard and look ahead algorithms are employed during periods when the semiconductor memory is not being used for relocation. Standard caching is used to put commonly used, high activity data in the fast semiconductor memory. Look ahead caching is used when sequentially read data make anticipatory retrievals from the magnetic disk to the semiconductor memory profitable. If a subsequent access request is within the semiconductor memory, the data can be accessed in less than a quarter of a millisecond. Both single and multiple blocks of data can be transferred at up to 1.2 megabytes per second from the magnetic media and 1.6 megabytes per second from the semiconductor memory.

The MASSBUS adaptor within the AIM/300 provides automatic data error detection and correction, and the transparent handling of media flaws on the disk. The DEC MASSBUS was selected as the most appropriate means for the attachment of the AIM/300 because it is the highest speed disk transfer bus available to both PDP-11 and VAX computers. Use of the MASSBUS also ensures that there will be no interference with the operation of other system devices such as terminals and tape drives.

To the system processor, the distinguishing feature of the AIM/300 is its ability to achieve many times the access performance of conventional disks through the use of its combination caching and automatic data relocation. Initially the design of the

AIM/300 incorporated naive or simple data relocation of the most actively used cylinders to the physical center of the disk. Further development resulted in the inclusion of sophisticated statistical forecasting techniques and cylinder selection criteria so that the relocation of cylinders could be quickly made to the most advantageous positions on the disk regardless of the center. This Dynamic Data Relocation results in a disk which behaves much as a conventional disk with caching during periods without arm contention, but which can achieve the access speed of the early model fixed disks (8.33ms) during typical periods of heavy arm contention with moderate caching (35 percent), and up to 10 times the performance of conventional moving head disks with heavy (80 percent) caching.

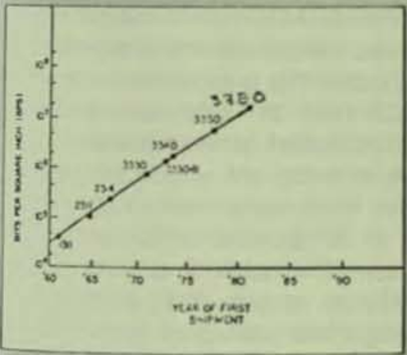


FIGURE 1. Disk Density Versus Year of Introduction

This new type of data storage device means that networks of small computers can mutually utilize a large central disk without heavy investments in a central semiconductor memory or multiple disks. Similarly large systems can fully utilize smaller numbers of large capacity disks without the purchase of additional disks to avoid arm contention. System managers can accomplish the task of disk subsystem optimization by making sure that data files are loaded within logically contiguous areas of the disk using standard system backup and restore facilities. Related files which formerly had to be placed on several disks to avoid arm contention can now be placed on one disk for convenience, security and logical integrity without sacrificing performance.

MASS STORAGE DEVICES

By Louis J. Finnegan, Aviv Corporation, Woburn, Massachusetts

For the past two years, DEC has been unveiling its long-term mass storage strategy called the Digital Storage Architecture (DSA). In general, DSA provides more long-term stability in the selection of mass storage peripherals by defining several independent layers, or protocols, for I/O control. The cornerstone of DSA is the way in which large data bases are stored, accessed, and transferred, using a combination of small and large systems sharing resources through conventional communications devices or a high speed serial link, the Computer Interconnect (CI) bus.

Most users, however, are not concerned with large interconnected computers sharing unlimited resources. Most of the applications we at Aviv encounter require a high availability of disk based data, and a convenient file transfer and backup capability between two and six CPUs of either PDP-11 or VAX vintage. This does not require a network or an architecture, but simply a reliable, economical method of storage and retrieval that can perform I/O requests intelligently enough to enhance overall throughput.

DEC apparently has endorsed this philosophy (having reversed direction on their time-honored theory called "Thou Shalt Not Unibus" high speed peripherals), by designating the RUA 81 as the performance leader in their disk product line, as well as adding other Unibus based peripherals. In effect, DEC is endorsing what controller manufacturers have been saying for years. Namely, that the Unibus will not be saturated with data given an intelligent controller design and a keen understanding of how to monitor the Unibus for other devices that require service. The same can be said of the Q-bus, particularly with the advent of the 11/73 and MicroVAX. Other changes are expected in this area from DEC which will maximize the use of existing Unibus and Q-bus structures.

The most critical storage device in a system is the primary disk, as operating system overlays and work files have long ago migrated on to rotating memory. In our experience, I/O bound systems can potentially support double the number of simultaneous users with a disk subsystem that has the ability to perform tasks such as independent seeking, while the CPU is servicing another task. In a multiuser environment, every millisecond counts, and each time a revolution of the disk is wasted because the controller is not being able to perform concurrent tasks such as overlapped seeks or head selection, terminal degradation will occur. This is particularly true with DEC's RUA 81 subsystem on large block transfers, where features such as seek ordering are of no extra benefit when the transfer involves more than 25 to 30 seconds. In fact, most I/O bandwidth requirements never exceed 1 MB/sec, because of the constant mix of high and low speed devices being serviced at varying priority levels.

To counter this problem, Aviv disk controllers use between three and 15 sector buffers, depending on the speed of the attached disk, to balance the availability of the CMI bus, Unibus or Q-bus while other devices are being serviced. In addition, DMA utilization can be modified to allow continuous transfers to and from the disk or tape drive while no other NPR requests are pending, and provide scheduled inactive periods (deadslot time) for the CPU to service other interrupts. The definition of drive characteristics such as cylinders and tracks, found in the DSA controller layer protocol, is a standard feature in Aviv controllers. Called a "Geometry" PROM, it provides greater flexibility for reconfiguration of installed systems.

Tape drive utilization is perhaps the least understood method of storage today. With the proliferation of 160 to 700 MB Winchester disks, ½" tape is enjoying a renaissance, as it alone offers the portability, reliability and low cost required of a backup medium. Industry standard (800/1600 bpi) tape has been around for years, so it is the 6250 bpi recording that is attracting most of the interest, due to a new generation of low-cost peripherals now entering the market.

GCR recording can store up to 145 MB of formatted data per reel, and achieve transfer rates rivaling some disks (781 KB). In addition, the two-track error correction capability raises the integrity of stored data to one correctable error in every 40,000,000 (40 MB) characters. Here again, the proper design of a controller can alleviate any potential bus problems that may occur during acceleration of the tape drive to its operating speed. In fact, proper handling of this feature can alone prevent the reduction of the speed of a high performance drive by half.

Streaming tape drives recently have become popular backup devices, and can be very cost effective when purchased specifically with that application in mind. However, they share limitations: 1) 800 bpi is not offered, reducing portability of data; 2) a very slow start/stop capability may or may not be offered, reducing the on-line transfer or backup capability and forcing these jobs to be performed after hours at an additional expense to the customer. Aviv recommends that careful consideration be given to the overall tape requirement before a selection of start/stop or streaming drives is made.

In addition, the tape drive is usually not a high availabilty device. Aviv offers a cost-effective method of sharing a bank of up to four tape drives with a combination of up to six CPUs, including DEC, Data General, Multibus and others. This feature not only saves on capital equipment expense, but provides substantial maintenance savings, as well as valuable floor space.

A COMPREHENSIVE PROGRAMMING ENVIRONMENT
MANAGEMENT AND PROGRAMMING TOOLS

By Josef S. Bleier, New York, New York

(Editor's Note: The author, a system software specialist and consultant, has been involved in the successful implementation of the Comprehensive Programming Environment (CPE) in major corporations, using VAX with UNIX (the operating system from Bell Labs).

In medium to large scale programming development, maintenance and documentation costs increase rapidly as the project nears completion and often end up being the largest part of the total development cost.

Because customers need fast response to their maintenance requests and up-to-date software documentation, sound business practices demand that we develop practical management and programming tools which hold down these expenses and meet customer needs promptly and effectively.

Here we look at a software environment that effectively reduces long-term maintenance and programming costs and helps management control the development process. The discussion is a non-technical overview of the tools we have developed but not of their implementation.

INTRODUCTION TO THE SOFTWARE DEVELOPMENT ENVIRONMENT

A software development environment includes the programmers, the managers and the computer systems. In very simple terms, the managers tell programmers the kind of program to write or develop, the programmers write the programs and the computer systems run the developed programs.

A software project produces a stand-alone program called a "target system." This system is designed to perform certain predefined tasks on a computer for which the software was developed.

A target system is made up of many small programs called "project files," each of which does a specific job. These project files are usually small (about 100 lines) and easy to understand.

Furthermore, each project file has a version number. Whenever a programmer changes the code in the file, that file gets a new version number but its file name stays the same.

The programmers write the project files in a programming language. A "compiler" program transforms each project file into an "object file." Then a "loader" program combines all the object files into one "target test file." If we encounter no errors in either the compilation or the loading of the target system, we have completed another stage in the software development cycle. Now the software development group gives the test group the target test file for testing.

Since each computer or type of computer is different, each computer's compiler translates the object file into a program which the specific computer (called the "target machine") understands. For this reason, it is most desirable to have a programming language which can easily be used on many different machines. Such a language is said to be "portable." Thus, the company can increase its market penetration.

At the heart of the (CPE) is the idea that all project files are stored and controlled by a single library, that can record all the version numbers. Therefore, those who develop or maintain the system need not keep track of changes made to the files and of their affect on the target system.

By using a set of library commands, the programmer stores and retrieves files and automates the building of the target system for testing. Furthermore, at any time managers can get an up-to-date programmer's progress report, system documentation and system status information.

Finally, the user — the programmer, the maintainer or the tester — communicates with the project library in this environment. The library becomes the database system for the project files. As in database systems, the tester or the developer knows of a logical view of the data and need not be concerned with where and how files are stored.

MANAGEMENT TOOLS

The management tools are the programs that automate all the necessary but burdensome paperwork:

1. **Documentation Preparation** — collecting all the material which describes each project file and the target system as a whole.

The successful development and maintenance of a system depends to a great extent on the quality of its documentation. Such documentation meets the needs of those who must maintain and update the system.

2. **Internal Documentation** — maintaining program information and source code.

Each separately compiled project file has two sections. The first section (the "header") has information about the program; for example, the author's name, the purpose of the program and what kind of report or output the file will yield. The second section of the file contains the actual source code; i.e., the sequential program statements and definitions that the compiler accepts and compiles. Therefore, when a programmer changes a project file, he can change the source code and the header information in one step.

Because there are many such changes during development and testing, the automatic document retrieval tool gives the programmer a fast and easy way to scan the project files in the library, extract the header section of each modified file and print up-to-date information about the new target system or a part of it.

Among sophisticated documentation tools is one which scans the source code section of each project file and extracts from it all references to other programs in different project files.

3. **Maintenance Request (MR) Procedure** — for documenting and correcting any program failure.

The MR system gives managers timely and accurate reports of all program errors found during testing and routine maintenance.

An MR is produced whenever a target test system functions improperly. The person doing the testing or maintenance submits an MR to alert management and development of the problem. A manager gives the MR to a programmer who studies the MR and tries to solve the problem.

MRs fall into one of the following categories:

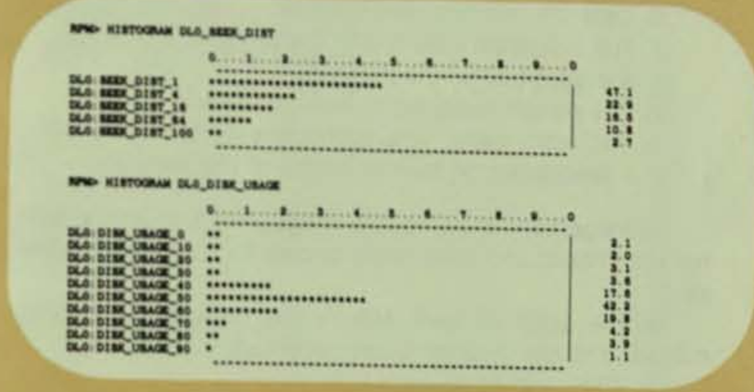
OPEN — Someone created an MR to describe a problem, but no one has been selected to resolve it yet.

PENDING — Management is aware of the problem mentioned in the MR and has assigned a programmer to work on it.

CLOSED — The assigned programmer changed the source code of the faulty file and resolved the problem described in the MR.

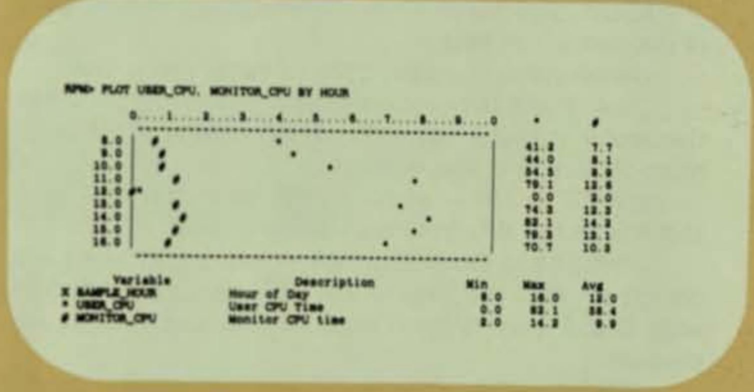
EXTENDED PERFORMANCE STATISTICS

[cut off] adds extended performance statistics to your [cut off]S/E monitor. This monitor extension captures [cut off]mation about overall system performance plus [cut off]mation on individual programs and files. Using this [cut off]mation, you can improve system performance by [cut off]nizing disk head movement, reducing file processor [cut off] waits, reducing swapping, and optimizing disk [cut off]e operation.



DYNAMIC PLOTTING

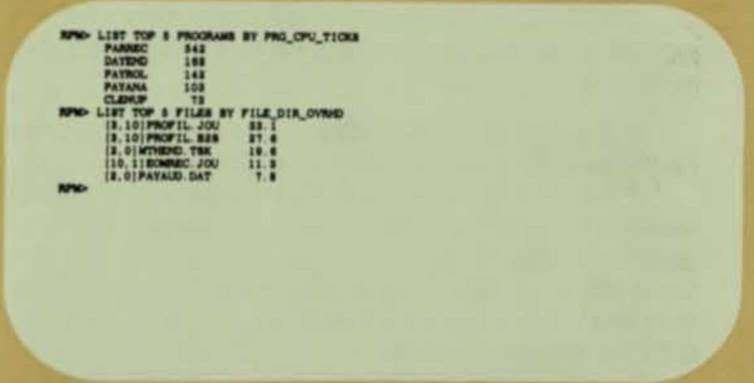
[cut off] addition to comprehensive reports, RPM can [cut off]rate a wide variety of graphs. It can plot curves, [cut off] bar charts, and plot histograms using any [cut off]bination of system information. By plotting one [cut off]able against another, you can immediately see [cut off]elations between variables. This is especially useful [cut off] determining critical resource usage points.



IDENTIFY PROGRAMS BY RESOURCE USAGE

[cut off]h RPM, you can identify the programs that will [cut off]vide the most benefit from optimization. You can [cut off]ermine which programs are used most often and [cut off]ich programs use the most of critical resources.

[cut off]ce identified, these programs can be optimized. [cut off]verall system performance can be increased by making [cut off]anges only where they will do the most good.



NO-RISK GUARANTEE

[cut off] all this sounds too good to be true, here's your chance to find out for yourself. Try RPM for 30 days. Use it to [cut off]e your system. If RPM isn't everything we say it is and more, return it for a full refund. We're not worried. We [cut off]ow that once you see the difference RPM can make, you won't want to be without it.

RPM – The Performance Revolution

NORTHWEST DIGITAL SOFTWARE

NORTHWEST DIGITAL SOFTWARE, INC. BOX 2-743 NEWPORT, WA 99156 (509) 447-5631

Each MR file may have several fields, including:
a. A priority number, indicating the urgency of the MR.
b. Function name.
c. Date the problem occurred.
d. Date the problem was resolved.
e. The individual who found the problem.
f. A description of the problem.
g. The person assigned to resolve the MR.
h. The team leader who authorizes work on this MR.
i. A description of the resolution of the problem.

Management must be able to keep track of all the MRs for the project and have ready access to such information as:

a. The state of each MR in the project, its priority number and the type of failure reported.

b. Who worked on each MR.

Automation makes it easy for managers to keep track of any number of MRs.

The following is a description of some useful tools.

CREATE and MR — Creation of MR involves an interactive session with the user and ends by creating an MR document file with a unique number.

SUBMIT an MR — stores an MR file in the MR library that has all the MRs generated by the project.

INSPECT and CHANGE an MR — lets users inspect and change any MR. The programmer assigned to this MR may wish to add more relevant notes to the description of the problem.

CLOSE an MR — locks an MR and resolves it.

GENERATE an MR Report — lets management inspect each MR in the library and produce a report as described in items 1, 2 and 3 above.

4. Status Report Preparation — documenting the progress of each individual working on a project.

Often programmers and their supervisors submit weekly status reports to tell upper level managers about the progress on a project. Writing status reports can be very time consuming. But it may be just as hard for a supervisor to collate data on a project if each report has the data in a different format.

To address this problem, we have developed some tools which automate and standardize the preparation of status reports.

The user enters a command, and the system prompts him for the necessary information. Then he enters a second command which transfers the report to a file from which the manager compiles his summary report. The supervisor uses another command to collate all the reports into a summary status report for his supervising manager.

The system automatically enters additional information in the status report, e.g., the user's name, the time and date the report was prepared, etc.

Then using certain key items (e.g., the reporting date or task name), the supervisor can retrieve the desired data from all the status reports to prepare a summary report for his supervising manager.

These tools reduce the amount of time spent preparing status reports leaving more time available for programming.

Here is a sample status report:

STATUS REPORT

WEEK OF: date

PROJECT: project_name

TASK: task_name in project

NAME: programmer's name

1 END OF WEEK STATUS:

Describe your accomplishments for last week.

2 NEXT WEEK'S TARGETS:

Describe the desired goals for this week.

3 CONTINGENCIES

Describe any problem in meeting your schedule.

SCHEDULE STATUS:

State "On Schedule" or "Behind Schedule."

Explain if "Behind Schedule."

5. Manual Pages System — creating or updating all material which describes the available software tools and their use.

Manual Pages explain how the various system features work. For example, to find out how to use the "stat" command, the user would type in:

$man stat<CR>

which would result in the following display:

NAME:
state — create a new status report.
SYNOPSIS:
state file_name
DESCRIPTION:
The "stat" program creates a new formatted status report in file_name.
The program prompts the user for a question and answer session. At the end, a formatted report can be inspected at file_name. The format of file_name is the project standard.
FILES:
none
SEE ALSO:
none
DIAGNOSTICS:
Error reports are self explanatory.
BUGS:
None.
AUTHOR
Josef S. Bleier

As discussed earlier, we build a target system for testing by compiling the project files. If no errors occur in the process, we must link the object files to form an executable target test system. To do that, we use the loader program to make sure that all the reference names used in one file are defined in another file.

If we examine all the stages involved in building a test system, it becomes clear why it often takes so long.

1. We assemble project files with the appropriate version numbers. If we are unfortunate enough to get the wrong file or version, it could cause a nasty error later that would be hard to trace.

2. For test purposes we may want to create a dummy source file with the name of all the programs not yet developed. These programs may be referenced by other existing programs.

3. After the above preparation we are ready to compile one project file after another to get a set of object files.

4. After compilation we use the loader to link all the object files and to create an executable target test file.

Since building a test system takes so long, we created some test and programming tools to automate this process.

As mentioned above, all project files are put into a library which keeps track of the changes made to a file. Thus, if the user makes a change to the file, the library records only the changes to that file and gives that group of changes a new version number. No duplicate file exists anywhere.

Thus, the user can:

1. Obtain a project file (with a version number) from the library.

2. Store a project file in the library.

3. Submit a new project file to the library.

4. Show a list of selected files in the library.

5. Provide a library copy of any project file by printing it.

At a certain point in the development cycle, the programmers have finished coding and the library is ready for testing. We call this point a "milestone." We define a "release" as the time when the customer gets some software for use. Therefore, we divide a release into several development milestones. Each milestone development gives the test group a test map which may contain, among other fields, the following:

1. Project File Name — the name of the file.

2. Version — the version number of the file for this milestone.

3. Status — tells if the file exists or has not been developed yet.

4. Date — the expected completion date of the project file.

This test map marks the official milestone.

Because the library uses version numbers to record the changes in the files, the programmers can begin work on a second milestone without the danger of losing the earlier versions of the file (i.e., the previous milestone). They can always retrieve any designated version of a file merely by specifying its number.

Since system testing usually involves many files at a time, keeping track of so many versions, milestones and release numbers manually would be too time consuming and hard. The following commands make the process fast and easy.

1. CREATE A MAP — creates a test map for testing.

2. GET FILES — collects all project files necessary for the creation of a target test system.

3. CREATE DUMMY — creates a dummy file for programs that have not been developed yet — see test map entry to signify this.

4. CREATE OBJECTS — compiles all files listed in the test map. If there were no changes to a project file, the system will not re-compile it because the system can track file interdependencies and changes.

5. CREATE A SYSTEM — uses the loader to link all the files and create an executable test file.

These commands could be put into a file and executed one after the other. This process automates the building of a test system and allows testers the flexibility to choose which project files should be combined together to form a system for testing.

CONCLUSION

In this paper we proposed the development of a set of tools that automates a programming environment. The implementation of these tools in a software environment creates what we called a Comprehensive Programming Environment (CPE). We have found that such an environment saves development and maintenance costs.

Since projects differ, the implementation of the CPE requires software tailoring efforts and constant contact with management. That is, the success of the project requires full management cooperation.

Some Helpful Hints . . .

A central criterion for estimating how likely a project is to succeed, is how easy the project files are to maintain. Is the program easily understandable and modifiable?

Many authors stress that program modularity; i.e., breaking the code up into logically independent and separately compiled units or modules, is one measure of maintainability. Furthermore modularity significantly reduces program complexity which otherwise grows exponentially as program size increases. In addition, modularity saves testing and development time because the structure and interdependency of the modules is clearer. The result is an easily maintainable system.

The first step in structuring a project requires some set of standards to assist developers in creating consistency and modularity among the various project files.

The following are a few guidelines that we found central in developing our automated test tools:

1. The program is divided into independent pieces that we call project files, or modules.

2. Each module is a self-contained unit that represents one unique logical and separate function.

3. Project file name and program name are the same. Note that we write the program (or function) inside a project file.

4. Each project file contains a function header describing the function.

5. Each project file name should be meaningful. Names should suggest the logical function they perform and their purpose in the system; i.e., a project naming convention dictates the rules for naming a file.

Finally, these few guidelines go hand in hand with the principles of good structured programming and maintenance support systems.

REFERENCES

1. "The Mythical Man-month" by W. Brooks, 1977.

2. "Notes on Structured Programming" by Hoare and E.W. Dijkstra, Structured Programming, Academic Press New York 1978.

3. "Composite/Structured Design" by Myers, Glenford J., New York: Van Nostrand Reinhold, 1978.

4. "On the Criteria to Be used in Decomposing Systems into Modules." Communication of the ACM, 15, 12 1053-1058 1972.

5. "Using Structured Design" by Wayne P., John Wiley & Sons N.Y. 1981.

6. James Martin and Carma McClure, "Software Maintenance: The Problem and Its Solutions," Prentice-Hall 1983.

First with Cache First in Performance

The CSS-800 Compact Storage System

Of all the mass storage systems available today, only the CSS-800 has the advanced architecture, featuring cache memory, designed to increase the performance of your mini or micro system. Our cache memory and look-ahead buffer strategically manage data transfers to speed throughput by up to 90% and give your system the added power to run faster and handle more concurrent users. The 800 combines Winchester technology with a unique, intelligent controller to reduce access time without expensive hardware upgrades or software redesign.

Our cache memory logic "learns" which data and routines are most often used and holds those sectors in a multiported 64-Kbyte cache buffer. Look-ahead buffering brings adjacent sectors into cache as well, so future transfers can be made at memory speeds. As usages change, the controller overwrites the least recently used sectors to keep the most pertinent data in cache for immediate access. Only the CSS-800 can give your mini or micro these performance features of a mainframe without adding to host overhead time.

First in Price Fully integrated with tape backup, controllers, and interface adapter for under $10,000 —the full entry price is the best of any total storage subsystem. With 70 Mbytes of disk storage and 22 Mbytes of tape backup, our cost per Mbyte is unrivaled. The CSS-800 is your most cost effective, complete storage buy.

The DEC Plug-in Solution Totally DEC* compatible, the CSS-800 emulates the RK07 disk and TU10 tape drives. System support is available under RSX-11M, RSTS/E and RT-11 operating systems as well as DSM, TSX, and UNIX. Q-bus or Unibus host interfaces are standard features. The whole rackmount or desktop package is only 5¼ inches high and weighs 42 pounds. Just plug our host interface into your backplane and you're up and running with a truly compact and efficient system.

Only U.S. Design Corporation gives you the high-performance architecture your system needs to extend its maximum operating level. We're first with cache, first in price, and first in performance. So when you consider your storage needs, call us first. Our toll-free number is (800) 368-2811.

*DEC, RK07, TU10, RSX-11, RSTS/E, RT-11, DSM, Q-Bus and Unibus are registered trademarks of Digital Equipment Corporation. TSX is a registered trademark of S & H Computers. UNIX is a registered trademark of Bell Laboratories.

U.S. DESIGN CORPORATION
5100 Philadelphia Way
Lanham, Maryland 20706
(301) 577-2880 (800) 368-2811
TWX 710-826-0417

USDC

CIRCLE D118 ON READER CARD

BUBBLE SORTS: A SECOND LOOK

By Joe Celko, Atlanta, Georgia

The "Bubble" or "pairwise exchange" sort is the simplest of all possible sorting procedures. Given an array, it is put into sorted order by swapping neighboring pairs of elements. This is probably the first program that the teacher assigns in the second week of a beginner's programming class. By the end of the course, the teacher has lectured on how slow it is and shown how to write several faster sorting procedures.

But, the poor old "Bubble" sort is worth a second look, in spite of its humble position among sorting programs. The real teaching value of the Bubble sort is that it can be greatly improved by simple changes. This makes it ideal for a training aid to teach programmers how to polish an algorithm.

DESIGNING AN ALGORITHM

When designing an algorithm, step one is to determine its purpose. A sorting algorithm ought to work for an array of any size. It ought to produce the permutation of the input array in sorted order, and work with any permutation of the input array. This sounds good, so let's begin.

The first hack at writing a Bubble sort program has only one mildly tricky point. You must remember not to try to inspect the (n + 1) element of an array with n elements. It's obvious that each element might have to be swapped, so if we make one pass for each element, then the array will be sorted.

Here is the raw, basic sort in pseudo-code. The pseudocode is a cross between BASIC, PASCAL, ALGOL and English. With a little inspection it should present no trouble.

```
PROCEDURE Bubble1 (INT n, INT ARRAY table[1:n]);
BEGIN
INT i, pass;
EXTERNAL PROCEDURE Swap (INT a, INT b);
FOR pass: = 1 TO n
DO FOR i: = 1 TO (n-1)
  DO IF (table[i] > table[i + 1])
    THEN Swap (table[i], table[i + 1]);
END of Bubble1;
```

This program is inefficient and will require that the Swap procedure be called (n*(n-1)) times. One quick way to improve this program is to stop doing the bubbling all the way down each time. After the first pass (controlled by "pass"), you know that the highest element in the array is now in place. Once an element has been "bubbled" to position, it stops moving. You only need to go down to array element (n-1) the next time; the third time to element (n-2), and so forth. The goal is not to do more work than is required.

The second version of the sort looks like this:

```
PROCEDURE Bubble2(INT n, INT ARRAY table[1:n]);
BEGIN
INT i, pass, LastItem;
EXTERNAL PROCEDURE Swap (INT a, INT b);
LastItem: = (n-1)
FOR pass: = 1 TO n
DO BEGIN
  FOR i: = 1 TO LastItem
  DO IF (table[i] > table[i + 1])
     THEN Swap (table[i], table[i + 1])
  LastItem: = LastItem - 1;
  END;
END of Bubble2;
```

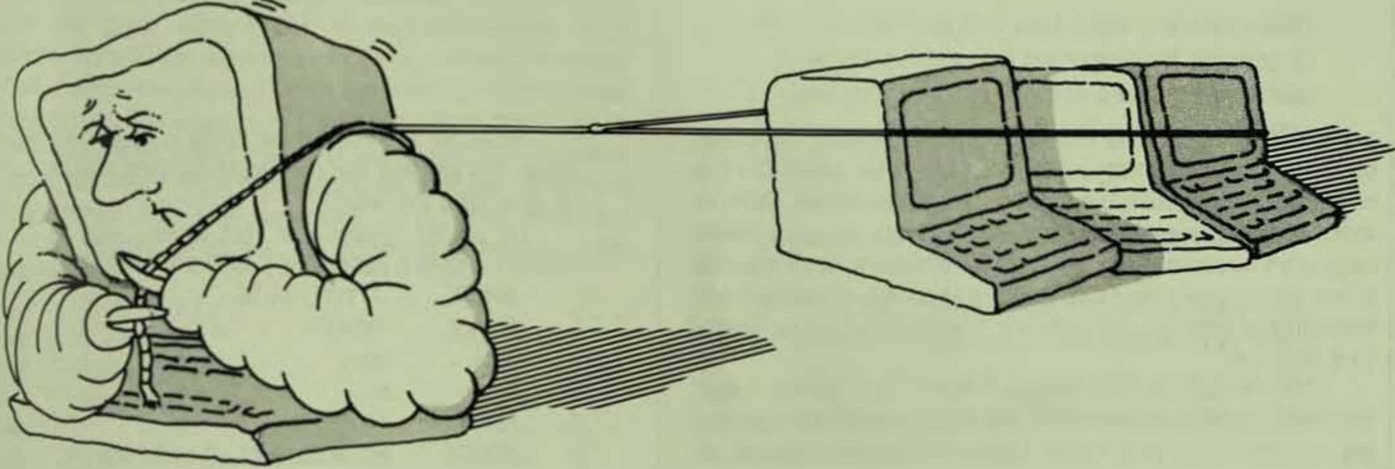
The British call this version a Sink sort, because the heavy items sink to the bottom. Another thing to do is check to see if the array is in sort as the program passes through it. If no swaps are made in a pass, the array must be in sorted order. This can be checked with a Boolean flag "InSort" that starts each pass with the value TRUE and gets set to FALSE if a swap is performed. This is an important principle for optimizing an algorithm. If you can test for a terminal state during the operation of the program, you can come to an earlier halt. It is also a principle of program design to use Boolean control expressions instead of counting control expressions whenever possible. So a FOR loop is replaced by a WHILE loop, which is more general. That pseudo-code looks like this:

```
PROCEDURE Bubble3(INT n, INT ARRAY table[1:n]);
BEGIN
INT i; LOGICAL InSort;
EXTERNAL PROCEDURE Swap (INT a, INT b);
LastItem: = (n-1);
InSort: = FALSE;
WHILE (NOT InSort)
DO BEGIN
  InSort: = TRUE;
  FOR i: = 1 TO (LastItem - 1)
  DO IF (table[i] > table[i + 1])
    THEN BEGIN
      Swap (table[i], table[i + 1])
      InSort: = FALSE;
      END;
  LastItem: = LastItem - 1;
  END;
END of Bubble3;
```

. . . continued on page 159

THE DRS DECISION: PART 5

DOES YOUR DATA BASE SYSTEM OFFER "FOUR ON THE FLOOR"?



Performance has always been one of the prominent selection criteria for data base systems. However, until now no one has anticipated the possibility that your requirements might outgrow a single DEC VAX 11/780.

DRS, the complete data base management system from Advanced Data Management, is the only system to allow up to four VAX 11/780's linked with shared memory to become an integrated data base system. Now, in addition to offering a powerful data manager and query language interface, DRS allows this performance system to grow from a single 11/780 to as many as four 11/780's with users on each machine accessing a data base distributed across the entire system.

It's a great comfort to know that, if your task outgrows the exceptional performance provided by the re-entrant code, pooled buffers, resource locking, and high-speed index access features of a single VAX/DRS combination, "four on the floor" performance is available through the easily-installed option of DRS/MP. If multi-processor DRS is already here, can "clusters" be far behind?

Let us help you get the "word"
on how over 200 users have managed
some very complex applications using DRS;
...often without a programmer!

ADVANCED DATA MANAGEMENT
15-17 Main Street
Kingston, NJ 08528
(609) 799-4600

Send me the word!

____
____
____
____

GETTING STARTED WITH MACRO-11

By Robert F. Curley, Department of Radiation Therapy, University of Pennsylvania, Philadelphia, Pennsylvania

Most of us become fluent in a programming language by using it, not by studying it in a text book. MACRO-11 is no exception. Writing your first program in the PDP-11 assembly language is more difficult than in most other languages because there is no single reference. Most contain a few simple, getting started programs, device drivers and examples of that ilk, but nothing basic enough to be useful to a beginner.

The assembly language itself is pretty well documented; it's the I/O that's the real problem. By restricting ourselves to two black boxes or system macros or system directives, we can get quite a way into the most useful second language for PDP-11 high level language programmers. A system directive is a request of the operating system for some service. In our case we'll ask for the operating system to move a character from our programming space out to our terminal. And we'll notify the operating system that we're finished with the program.

The directives that I find least intimidating are from RT-11: ".TTYOU" and ".EXIT". To use these two requests for system services on other systems it is necessary to translate them into the correct form for that system. We'll emulate TTYOU and .EXIT under several operating systems.

Note: For a RSTS/E system, use the RT-11 emulator. Make a file MACROS.MAC that contains:

```
.MACRO   .TTYOU
EMT      341
BCS      .-2
.ENDM

.MACRO   .EXIT
EMT      350
.ENDM
```

For an RT-11 system make the same file MACROS.MAC as above. It is arguable that one should use the system library, but, for now, do it this way.

For an IAS system or a VMS system, using the PDP-11 emulator, create a file MACROS.MAC that contains:

```
     .MACRO   .TTYOU, ?A,?B
     .MCALL   QIOW$S
     MOV      R0,A
     BR       B
A :  .BLKW    1
B :  QIOW$S   #IO.WLB,#5,#1,,,,<#A,#1,#0>
     .ENDM

     .MACRO   .EXIT
     .MCALL   EXIT$S
     EXIT$S
     .ENDM
```

(This may work on RSX11M and RSX11D too.)

Create your first program in a file ONE.MAC:

Note: I am presuming, of course, that you understand the basics of using your operating system and the use of a text editor. The editor that I use is TECO and I recommend it. It is consistent across most of the PDP-11 operating systems.

```
       .TITLE   ONE
RED:   .WORD    101
BLUE:  MOV      RED,R0
       .TTYOU
       .EXIT
       .END     BLUE
```

This is your first program. You may immediately assemble it and link ("taskbuild" if you speak RSX) it and execute it.

Note: The assembly process is very similar to a compiled high level language, but quite different from using an interpreted language:

For RSTS/E:

```
MACRO ONE=MACROS,ONE
LINK ONE=ONE
RUN ONE
```

For IAS or RT-11:

```
MACRO/OBJECT:ONE MACROS+ONE
LINK ONE
RUN ONE
```

For VMS:

```
MACRO/RSX/OBJECT=ONE MACROS+ONE
LINK/RSX ONE
RUN ONE
```

On some of the above systems there are other possible command sequences and abbreviations. These work on the systems that I have access to.

The program prints an upper case letter A on your terminal. The grace with which it happens varies with operating system. Some perform a carriage return, line feed sequence before the program starts executing; others as the program exits. Some operating systems allow the A to be printed over the prompt on the left margin. If you have one of these, you might wish to modify your program to print a carriage return, line feed sequence before and after the main part of your program. For example, create the file TWO.MAC:

```
        .TITLE  TWO
PINK:   .WORD 101
CR:     .WORD 15
LF:     .WORD 12
START:
MOV     CR,RO
        .TTYOU
        MOV     LF,RO
        .TTYOU
        MOV     PINK,RO
        .TTYOU
        MOV     CR,RO
        .TTYOU
        MOV     LF,RO
        .TTYOU
        .EXIT
        .END    START
```

Program TWO prints five characters on your terminal: carriage return, line feed, upper case A, carriage return, line feed.

A discussion of the components of Program ONE and TWO is in order. The syntax of MACRO-11 is simple: four fields, label, operation code (opcode), operands, comments. Labels begin the line and are terminated by a colon. Opcodes are next, terminated by a delimiter: space, tab, carriage-return. Operands next, separated by commas if there are more than one. I'll ignore comments. I use these rules: a label goes at the left margin, a tab, the opcode, a tab, the operands. No label? Tab to the opcode field. There are more elaborate schemes, but this will get you started.

"WHEN I AGREED TO MOVE IN WITH YOU, BRUCE, I THOUGHT YOU HAD SUFFICIENT HARDWARE!!"

The .TITLE is an example of a class of program components called "assembler directives." It is a command (request) to the assembler, evaluated or executed at assembly time. In this case, the .TITLE tells the assembler to call this program ONE. The use of this naming varies with your operating system but the assembler uses it to title the listings and the linker (taskbuilder) uses it on the map listing. For the beginner programmer it performs no useful function — if you leave it out the program works the same. I recommend the use of the .TITLE directive as a good habit for later benefit!

The next line, starting with RED: asks the assembler to reserve a word of memory (the .WORD assembler directive), call that word RED and start the program with RED containing the octal value 101. Unlike most high level languages MACRO-11 associates two values with a memory location: the address and the contents. Here the address is RED and the contents 101. The value, 101, is assumed to be octal, as are all numeric constants unless you explicitly notify the assembler otherwise.

The next line, starting with BLUE:, is a PDP-11 instruction MOV. The details of the move instruction mnemonic, MOV, may be found in the "Processor Handbook."

> *Note: There have been numerous PDP-11 Processor Handbooks. Any of them is sufficient for the beginning MACRO programmer. If you have older ones, save them — they sometimes contain details that have been removed from the newer ones. A sampling from my bookshelf: "PDP11/04/05/10/35/40/45 Processor Handbook" EB-05138-75 (1975); "PDP11/70 Processor Handbook" EB-05962-20 (1976); "PDP11/04/34a/44/60/70 Processor Handbook" EB-17716-18 (1979); "PDP11/04/24/34a/44/70 Processor Handbook" EB-19402-20 (1981). The most recent edition seems to follow the example of VAX, the volume is now called: "PDP-11 Architecture Handbook" EB-23657-18 (1983)*
>
> *These books detail the instructions of which the PDP-11 is capable. In some cases there are also hints on assembler syntax for some of the instructions that were left out of the Assembly Language Reference Manual. Whichever edition you have, it is a valuable reference to the MACRO programmer. Note, for example, the fine print under the MOVe Byte instruction.*

In fact, that is the place to find the definitive description of all the PDP-11 instructions. The MOVe instruction takes two arguments: Source and Destination. At BLUE we have 'MOVe the contents of RED into Register Zero'. It is a copy operation, the original contents of RED are intact. Sixteen bits are MOVed — most of the PDP-11 operations can be conceived as 16 bit parallel operations. A register is storage place in the CPU as opposed to a storage space in main memory.

> *Note: "Main Memory" is what used to be called "core memory." The stuff made of little ferrite donuts called cores, not the core of an Apple (pardon the expression). Today, main memory is usually made of etched silicon chips, as are the registers "within" the CPU. The principle difference then, is where the memory cells are in the computer's organizational chart. Registers are close to the center of power and have a sim-*

We Have the Prescription for Your Ailing Computer System

℞ Tymshare's computer maintenance organization is just what the doctor ordered.

We're an independent service group offering you fast response time and rapid, reliable repair—24 hours a day, seven days a week.

Maximum Uptime for the Lowest Possible Cost

Our highly skilled, experienced service representatives quickly and accurately diagnose your system's problems—to maximize your computer uptime.

And our custom preventive maintenance program keeps your installation healthy—to ensure continuing efficient operation at the lowest possible cost.

For more information on how Tymshare's maintenance programs can best fit your needs, call us today at the regional office nearest you.

COMPUTER SYSTEMS and SUPPORT

EAST: 10111 George Palmer Hwy.; Bowie, MD 20716; (301) 459-8363
CENTRAL: 4513 Lincoln Ave., Suite 106; Lisle, IL 60532; (312) 964-0248 or 964-0362
SOUTHERN: 6333 Harry Hines Blvd; Dallas, TX 75235; (214) 638-2130
SOUTHWEST: 11050 E. Artesia Blvd., Suite D; Cerritos, CA 90701; (213) 402-1114
WEST: 39100 Liberty Street; Fremont, CA 94538; (415) 794-2490 or (415) 794-2719

TYMSHARE®

CIRCLE D243 ON READER CARD CIRCLE D255 ON READER CARD *ple addressing mechanism making access to them quicker. The main memory is usually in another physical section of the computer, on another board, in another cabinet and the addressing of a single cell more time consuming. The actual timing considerations will vary with the model of the PDP-11 and the presence of cache memory and other such considerations.*

There are eight "general purpose" registers on every PDP-11, usually called R0, R1, R2, R3, R4, R5, SP and PC. Guessing from the names, registers six and seven are special and not general purpose at all — correct. Generally a register and a memory location may be used interchangably. Access to a register is faster than memory. And, there are some instructions that require the use of a register.

The next line contains the invocation of the macro .TTYOU. The Language Reference Manual explains that you may use any of the alphabetic characters, the ten numeric characters and dot and dollar sign to form symbol names.

> *Note: "PDP-11 MACRO-11 Language Reference Manual" AA-V027A-TC (1983) is the most recent in a long line of such manuals. While the most recent is nice, the older versions will get you started as well as they have most of the MACRO programmers today. Another example: "IAS/RSX-11 MACRO-11 Reference Manual" DEC-11-OIMRA-B-D (1976). The language has evolved and the older manuals do not mention the newer features, but the essential parts of the language were well designed and have been left intact.*

It recommends that you not use dot and dollar. Digital has bound itself to the convention that ALL the symbols that they use (that might be accessible to you) will contain a dot or a dollar sign. This convention makes it easier for you to avoid "reserved words." In this case I have suggested that you create a macro, .TTYOU, in a file MACROS.MAC, that breaks this convention. Here the .TTYOU tells the assembler to replace the .TTYOU with whatever has been defined as .TTYOU. Since you have created the macro .TTYOU with the incantation necessary to output a character for your operating system, the assembler will do the sleight of hand for you and we can all talk about a .TTYOU without being concerned about the lower level rubric. .TTYOU is a "system directive," it requests the operating system to move the contents of register zero (R0) out to the terminal. Usually there is an ASCII character there, which the terminal interprets and prints.

> *Note: American Standard Code For Information Interchange. An arbitrary code to represent printing and control characters in numbers which can be stored in a computer. There are other codes, but this is the one most widely used in the PDP-11. The full scheme is on the back of the PDP-11 Programming Card. The ones used here: 101=A, 12=<line feed>, 15=<carriage return>.*

Thus the octal 101 is interpreted as an upper case A. The octal value 15 is translated by the terminal as "carriage return," etc.

The next exercises are: (1)Modify program ONE to print a Z instead of an A. (2)Modify program ONE to print an A followed immediately by a Z. (3)Modify ONE to print an A directly over a Z, that is, A on one line, and Z at the left margin on the next line.

It is true that one ASCII character is made up of eight bits, and that there are 16 bits in a PDP-11 word. When you put both the A and the Z in one word and ask the operating system to print it on the terminal, the operating system only prints the one character in the "low byte" or "right hand byte" of the word in Register 0. Thus, it is necessary to have one MOVe instruction and one .TTYOU system directive for each character that is to go to the terminal, printing and control. Examine program TWO for example.

The .EXIT is a system directive that informs the operating system that the program is completed; return the computer to the operating system. There may be several .EXITs in a program, indicating several points in the logic where the program can finish. Only one .EXIT is ever executed for one RUN of the program.

The .END is an assembler directive indicating the end of the program. The difference here, is that .END is a message to the assembler saying "This is the end of my program, ignore anything else in this file," or "Stop assembly here." .EXIT is a message to the operating system; .END is a message to the assembler. There must be only one .END in your program. (Occasionally, as in the file MACROS.MAC for example, there will be no .END.) When the file you are creating is the main program, rather than a subroutine, the .END takes an argument—the address of the start of the program, the transfer address. In program ONE the computer wil start executing the program at the address BLUE.

Unlike most high level languages, MACRO-11 permits you to mix data and instructions. The assembler would not protest if we were to place RED after BLUE. The computer might protest when we tried to RUN the program, because the data in RED might not make any sense when interpreted as an instruction. Thus, at the beginning I recommend that you organize your program to place the data at the front of your program file, before the start of the program. Or, at the end after the .EXIT.

What to do next? There are several textbooks that will lead you through or you may just wrestle with the Language Reference Manual and a self imposed project. (Refer to the list of recommended textbooks at the end of the article.) Or you may find a college nearby that offers a course in MACRO-11 at night. Such a course is probably more productive than the one week intensive courses offered by Digital's Educational Services; learning a language takes most people longer than a week.

The attached subroutine, RDUMP, is included to be used as a tool in debugging your early programming efforts.

Note: The following is a listing of the file RDUMP.MAC. You must edit this file to select the correct operating system. About two thirds of the way down the first page is the statement
SYSTEM=XXX

Edit the XXX into the appropriate symbol for your operating system: RT11, RSTS, IAS, VMS. Then it should be used with your program as follows:

For RSTS/E:

```
MACRO THREE=MACROS,THREE
MACRO RDUMP=RDUMP
LINK THREE=THREE,RDUMP
RUN THREE
```

For IAS and RT-11:

```
MACRO/OBJECT:THREE MACROS+THREE
MACRO RDUMP
LINK THREE,RDUMP
RUN THREE
```

For VMS:

```
MACRO/RSX/OBJECT=THREE MACROS+THREE
MACRO/RSX RDUMP
LINK/RSX THREE,RDUMP
RUN THREE
```

Digital's Octal Debugging Tool (ODT) is available with most operating systems, but is confusing to learn at the same time you are struggling with the assembler. Use RDUMP until you are comfortable with the jargon of assembly language; then take on ODT. A sample program using RDUMP:

```
        .TITLE  THREE
        .GLOBL  RDUMP
N1:     .WORD   1
N2:     .WORD   2
N3:     .WORD   3
N4:     .WORD   4
N5:     .WORD   5
GREEN:  CLR     R0
        MOV     N1,R1
        MOV     N2,R2
        MOV     N3,R3
        MOV     N4,R4
        MOV     N5,R5
        CALL    RDUMP
        .EXIT
        .END    GREEN
```

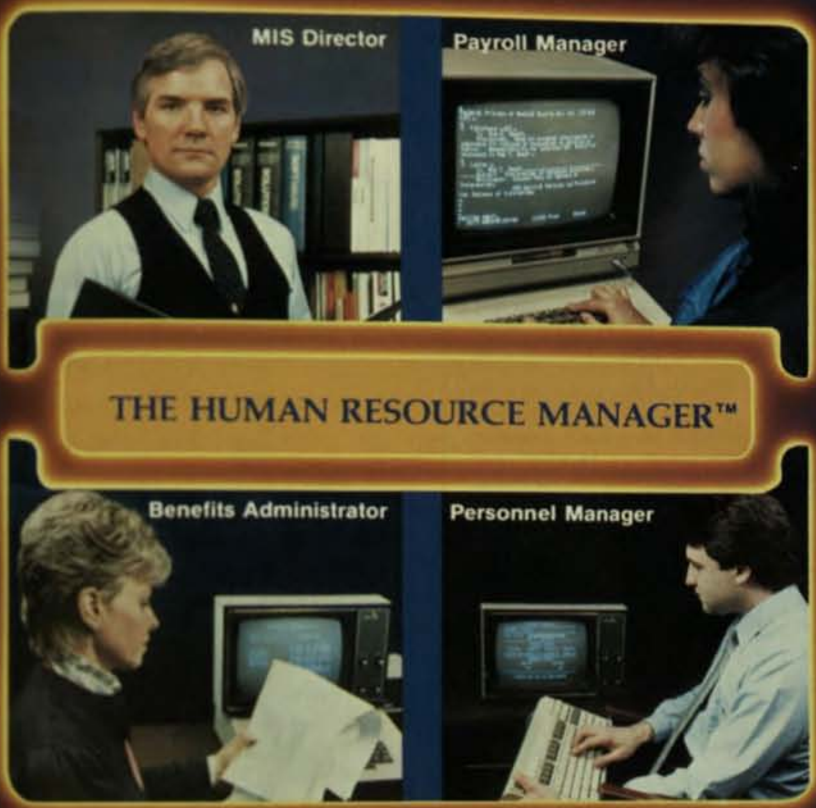
The .GLOBL assembler directive indicates that the argument RDUMP is a symbol that is defined somewhere else, and the linker will connect any references to RDUMP correctly. RDUMP will print the contents of all the registers and the status of the condition codes that are found in the Processor Status Word. This subroutine may be executed by a non-priviledged user on a time sharing system, without direct access to the PSW. There should be no change to your program by the introduction of RDUMP. It saves all the registers and condition codes and restores them before returning to your program. So, while the example THREE uses RDUMP at the end of the program, that is not necessary.

Persevere in your pursuit of MACRO-11 fluency. Your reward will be greater understanding of high level language and the power to perform "impossible" tasks by using assembly language subroutines.

Note: The following recommended textbooks discuss PDP-11 and MACRO-11: Thomas S. Frank, "Introduction to the PDP-11 and its Assembly Language"' Prentice-Hall, Inc. Englewood Cliffs, NJ; 1983. Charles A. Kapps and Robert L. Stafford, "Assembly Language for the PDP-11," Prindle, Webber & Schmidt, Boston; 1981. Michael Singer, "PDP-11 Assembler Language Programming and Machine Organization," John Wiley & Sons, New York; 1980. Arthur Gill, "Machine and Assembly Language Programming of the PDP-11," Second Edition, Prentice-Hall, Inc., New Jersey; 1983. Harry R. Lewis, "An Introduction to Computer Programming and Data Structures using MACRO-11," Reston Publishing Company, Reston, Virginia; 1981. Richard H. Eckhouse, Jr. and L. Robert Morris, "Minicomputer Systems — Organization, Programming, and Applications (PDP-11)," Second Edition, Prentice-Hall, Inc., New Jersey; 1979. Harrold S. Stone and Daniel P. Siewiorek, "Introduction to Computer Organization and Data Structures: PDP-11 Edition," McGraw-Hill Book Company, New York; 1975. R.W. Southern, "PDP-11 Programming Fundamentals," Southcroft Publications, Ottawa, Ontario; 1972. Harvey Lee Shapiro, "Introduction to Assembly Language Programming on the PDP-10 and PDP-11," Myfield Publishing Company, Palo Alto, California; 1982.

```
; File: RDUMP.MAC
; Assembler: MACRO-11
; Programmer: Robert F. Curley
;
;     **************************************************
;     *                                                *
;     *     DIRECT INQUIRIES TO:                       *
;     *           Computer Facility                    *
;     *           Department of Radiation Therapy      *
;     *           School of Medicine                   *
;     *           University of Pennsylvania           *
;     *           Room 410                             *
;     *           133 South 36th Street                *
;     *           Philadelphia, Pennsylvania 19104     *
;     *                                                *
;     *     NO WARRANTY OR REPRESENTATION, EXPRESS OR  *
;     *     IMPLIED, IS MADE WITH RESPECT TO THE       *
;     *     CORRECTNESS, COMPLETENESS, OR USEFULNESS   *
;     *     OF THIS SOFTWARE, NOR THAT USE OF THIS     *
;     *     SOFTWARE MIGHT NOT INFRINGE PRIVATELY      *
;     *     OWNED RIGHTS.                              *
;     *                                                *
;     *     NO LIABILITY IS ASSUMED WITH RESPECT TO    *
;     *     THE USE OF, OR FOR DAMAGES RESULTING FROM  *
;     *     THE USE OF THIS SOFTWARE                   *
;     *                                                *
;     **************************************************
;
; (Most Recent Edit) 10 January 1984
;
;**
; Subroutine RDUMP - print the contents of the registers on the terminal
;**
;       CALL RDUMP
; prints the contents of the registers (in octal) at the time of the
; CALL.  It does not alter the registers nor the condition codes.
;
; Output Format:
;
; Register Dump
;   R0=nnnnnn  R1=nnnnnn  R2=nnnnnn  R3=nnnnnn
;   R4=nnnnnn  R5=nnnnnn  SP=nnnnnn  PC=nnnnnn
; Condition Codes
;    N=n        Z=n        V=n        C=n
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;---------------------------
                                         ;
        .TITLE RDUMP                     ;
        .IDENT  /T/                      ; version number
                                         ;
        .DSABL  GBL                      ; this is the way MACRO-11 on RSTS
                                         ; works.
                                         ;
                                         ;*** Define The Operating System ***
                                         ;
RT11=1                                   ; RT-11 or RT-11 emulator under RSTS/E
                                         ;  with the RT-11 system macro library
                                         ;  added to the RSTS/E system library.
                                         ;
RSTS=2                                   ; RSTS/E under the emulator mode.
                                         ;
IAS=3                                    ; IAS operating system.
                                         ;
VMS=4                                    ; RSX emulation on VAX
SYSTEM=XXX                               ; Select One
                                         ;
                                         ;
        .IF NDF,SYSTEM                   ;
        .ERROR; ** You must define the symbol SYSTEM in the
```

Introducing



State-of-the-art payroll/personnel software for your VAX minicomputer to help you improve the state of your organization

Personnel Data Systems integrates payroll, personnel and benefits information into the industry's most sophisticated on-line, real-time human resource management system.

There's a good reason why hundreds of payroll-personnel, MIS and financial managers have selected THE HUMAN RESOURCE MANAGER for their organizations.

For the personnel manager/benefits administrator, that good reason is the base of information that facilitates easy and accurate tracking of each employee — from salary, performance and attendance data to skills, education, and career growth. From benefits program administration that allows "cafeteria-style" benefits to employer/employee benefit cost analysis and deferred compensation plans utilization. From initial applicant tracking to pension projections and ERISA vesting schedules.

For the payroll manager, that good reason is the high-performance payroll system that is single-minded in purpose: calculating pay, creating paychecks and preparing tax reports. And using Personnel Data Systems' powerful English language Report Writer to provide an unlimited range of management reports.

For the MIS director, that good reason is proven on-line, real-time software that offers table-driven processing, broad communications capabilities, and compatibility with every popular data base environment. State-of-the-art software that parallels non-computer users' state of mind.

...and for the financial manager, that good reason is bottom line profitability. Achieved with human resource software that can be easily geared to a company's unique requirements. That performs job cost accounting, monitors department performance, and assures accurate payroll and benefits payments.

ICP
$25
Million
Company

It's all backed by a product support team committed to software excellence.

CIRCLE D245 ON READER CARD

Call Personnel Data Systems today at 215-828-4294 for your firsthand look at THE HUMAN RESOURCE MANAGER.

I'm interested in THE HUMAN RESOURCE MANAGER.

☐ **Please have a sales representative contact me at** (phone) ____________.

☐ **Please send information.**

Name ____________

Title ____________

Organization ____________

Address ____________

City ____________ State ______ Zip ______

THE HUMAN RESOURCE MANAGER is a trademark of Personnel Data Systems, Inc.

PDS

Whitemarsh Plaza
15 East Ridge Pike
Conshohocken, PA 19428
215-828-4294

Personnel Data Systems, Inc.

DEC 3-84

IBM 43XX, 30XX, System/38, DEC/VAX, Data General, HP-3000, Burroughs, Honeywell, NCR, Sperry

```
        .ERROR; ** statement above, SYSTEM=XXX, by changing the
        .ERROR; ** XXX to RT11 or RSTS or IAS or VMS so that
        .ERROR; ** the appropriate macro for .TTYOU is defined.
        .ERROR; **
        .ERROR; ** Use a text editor to change this file and
        .ERROR; ** and re-assemble it.
        .ERROR; **
        .ENDC                                   ;
                                                ;
        .IF EQ,<SYSTEM-VMS>                     ;
        .MACRO  .TTYOU,?A,?B                    ;
        .MCALL  QIOW$S                          ;
        .GLOBL  IO.WLB                          ;
        MOV     R0,A                            ;
        BR      B                               ;
A:      .WORD   0                               ;
B:      QIOW$S  #IO.WLB,#5,#1,,,,<#A,#1,#0>
        .ENDM                                   ;
        .ENDC                                   ;
                                                ;
        .IF EQ, <SYSTEM-IAS>                    ;
        .MACRO  .TTYOU ?STAT,?BUFF,?SKIP,?Q
        .MCALL  QIOW$,DIR$                      ; IAS code
        MOV     R0,BUFF                         ;
        DIR$    #Q                              ;
        BR      SKIP                            ;
Q:      QIOW$   <IO.WLB!IO.WAL>,5,2,,STAT,,<BUFF,1,000>
STAT:   .BLKW   2                               ;
BUFF:   .BLKW   1                               ;
SKIP:                                           ;
        .ENDM                                   ;
        .ENDC                                   ;
                                                ;
        .IF EQ, <SYSTEM-RT11>                   ;
        .MCALL  .TTYOU                          ; RT-11 code, use the library
        .ENDC                                   ;
                                                ;
        .IF EQ, <SYSTEM-RSTS>                   ;
        .MACRO  .TTYOU                          ;
        EMT     ^O341                           ; RSTS/E code
        BCS     .-2                             ;
        .ENDM                                   ;
        .ENDC                                   ;
                                                ;---------------------------------
        .MACRO  PUSH,REGLIS                     ; <<< PUSH >>>
        .IF NB,<REGLIS>                         ;
        .IRPC   REG,<REGLIS>                    ; Invocation: PUSH <012> or PUSH
        MOV     $REG,-(SP)                      ; Push the indicated registers onto
        .ENDM                                   ; the stack. If no register list
        .IFF                                    ; is used a PUSH <012345> is
        .IRPC   REG,<012345>                    ; assumed.
        MOV     $REG,-(SP)                      ;
        .ENDM                                   ;
        .ENDC                                   ;
        .ENDM                                   ;
                                                ;
                                                ;---------------------------------
        .MACRO  POP,REGLIS                      ; <<< POP >>>
        .IF NB,<REGLIS>                         ;
        .IRPC   REG,<REGLIS>                    ; Invocation: POP <210> or POP
        MOV     (SP)+,$REG                      ; The order should be reversed from
        .ENDM                                   ; the PUSH specification if specific
        .IFF                                    ; registers are named.
        .IRPC   REG,<543210>                    ; A POP without argument is assumed
        MOV     (SP)+,$REG                      ; to be POP <543210>
        .ENDM                                   ;
        .ENDC                                   ;
        .ENDM                                   ;
                                                ;
                                                ;---------------------------------
        .MACRO  SAVE,Z,N,V,C                    ; <<< SAVE >>>
CLEAR=0                                         ;
SET=0                                           ; Create bit masks of the condition
        .IF EQ, N                               ; codes. The bit mask for the bits
CLEAR=^B1000                                    ; to be set is stored in SSAVE.
        .IFF                                    ; The mask for bits to be cleared
SET=^B1000                                      ; is stored in CSAVE.
        .ENDC                                   ;
        .IF EQ, Z                               ;
CLEAR=CLEAR ! ^B100                             ;
        .IFF                                    ;
SET=SET ! ^B100                                 ;
        .ENDC                                   ;
        .IF EQ, V                               ;
CLEAR=CLEAR ! ^B10                              ;
        .IFF                                    ;
SET=SET ! ^B10                                  ;
        .ENDC                                   ;
        .IF EQ, C                               ;
CLEAR=CLEAR ! 1                                 ;
        .IFF                                    ;
SET=SET ! 1                                     ;
        .ENDC                                   ;
        MOV     #SET,SSAVE                      ;
        MOV     #CLEAR,CSAVE                    ;
        JMP     RED                             ;
        .ENDM                                   ;
                                                ;---------------------------------
RDUMP::                                         ; <<< Entry Point >>>
                                                ;
                                                ; Save the Condition Codes
        BNE     10$                             ; Z  branch if Z=0
        BPL     12$                             ; N  branch if N=0
        BVC     14$                             ; V  branch if V=0
        BCC     16$                             ; C  branch if C=0
        SAVE    1,1,1,1                         ;  Z1,N1,V1,C1
16$:    SAVE    1,1,1,0                         ;  Z1,N1,V1,C0
14$:    BCC     20$                             ; C  branch if C=0
        SAVE    1,1,0,1                         ;  Z1,N1,V0,C1
20$:    SAVE    1,1,0,0                         ;  Z1,N1,V0,C0
12$:    BVC     22$                             ; V  branch if V=0
        BCC     24$                             ; C  branch if C=0
        SAVE    1,0,1,1                         ;  Z1,N0,V1,C1
24$:    SAVE    1,0,1,0                         ;  Z1,N0,V1,C0
22$:    BCC     26$                             ; C  branch if C=0
        SAVE    1,0,0,1                         ;  Z1,N0,V0,C1
26$:    SAVE    1,0,0,0                         ;  Z1,N0,V0,C0
10$:    BPL     30$                             ; N  branch if N=0
```

SOFTWARE

CALOUT Communications For RSX. Now RSX users can have the same communications and file transfer capabilities that thousands of VMS and RSTS/E users have experienced with CALOUT. CALOUT is a complete communications facility for RSX, VMS, and RSTS users. As a stand alone software utility, it is a versatile and affordable alternative to expensive hardware networks. With CALOUT your system can access more computers, networks, and communication resources than with any other product of its type.

Total Compatibility. A number of remote modules available for CALOUT enable your computer to communicate with other types of computers as if they were completely compatible. Remote modules presently available allow you to reach out to any VMS, RSTS, TSX, RT-11, CTS, or CP/M operating system with complete communication and file transfer capabilities. You own the source code to the remote module and may communicate with any number of remote systems without having to license each computer individually. If you have multiple computers or are upgrading to a more powerful system, you can use CALOUT to link your computers. This optimizes the performance of your systems by allowing them to transfer and share all types of data, without the cost of a network.

Transfer Anything At All. Using CALOUT you are able to transfer all types of disk files between your computer and any computer for which your system has one of the optional remote modules. This includes binary program files. All file attributes are converted or retained. Wild card transfers are also supported by CALOUT, allowing you to move many files with a single request. CALOUT can be used to down load software to other computers. This works on all kinds of software, including turn-key applications, and software you have created with the development resources of the host computer. This is particularly valuable in any situation where complex programs or information must be down-loaded to micro-computers. CALOUT can also be used to up-load data from a micro to a larger computer. You may run any kind of program or batch process on a remote system you have reached using CALOUT.

CALOUT

RSX
COMPLETE COMMUNICATION AT LAST!

Dynamically Link To Any System. Use CALOUT to call and link-up to any operating system which accepts a standard ASCII terminal. With one of the remote modules available, you can log onto the target system as a remote user, and transfer files back and forth between systems. Your RSX system can have access to virtually any computer with a dial-in line. CALOUT's friendly interface and high operating efficiency can ease the burdens of ASCII transfers. CALOUT is the most widely used product of this kind for DEC systems and has met the demands of communicating with a great variety of computers, including IBM, CDC, Honeywell, and Data General – Literally any system that accepts an ASCII terminal.

The World At Your Fingertips. The CALOUT Telex and Information Network option, CALTLX, connects directly to Western Union's Telex, Telex II (TWX), and EasyLink networks. Through these, you may also send messages by Mailgram, International Telex, Telegram, Cablegram, or InfoCom. Messages prepared with a text editor or your favorite word processor can be transmitted through a network directly to a terminal or network mailbox, eliminating the need for papertapes and teletypes. If CALOUT can't get your message where you want it, you need a plane and a parachute. With CALOUT your computer becomes a powerful communication center to meet all your special requirements.

Hardware Flexibility. CALOUT will operate with whatever hardware links may best meet your needs. You may use a standard telephone line with everything from an accoustic coupler to a sophisticated modem. CALOUT has full auto dial potential, and will interface with any other special modem capabilities. Computers which are physically close to one another may be linked with a simple null modem cable.

The Advantage Of Reaching Out. Since you are not locked into a fixed network environment, there is no limit to the number of systems you can reach out to with CALOUT. When you need to send or receive data or perform a task, you reach out to the system you need quickly and simply. No remote user assistance or tape loading is necessary. CALOUT is very inexpensive compared to hardware networks, and when you add up the savings of not having to license any of the remote systems you contact, you may begin to see why so many VMS and RSTS users depend on CALOUT for all their data communication needs.

There's More To Know. There are many reasons why CALOUT is so widely used. Think about all the things that your system could be doing, and how little it would cost. Find out more about CALOUT. Call us at the number below.

Clyde

How to get more information:
Call Mary at (801)224-5306 or write:
Clyde Digital Systems, Inc.
3707 North Canyon Road/Bldg. 3-E
Provo, UT 84604

CIRCLE D257 ON READER CARD

```
        BVC     32$                     ; V  branch if V=0
        BCC     34$                     ; C  branch if C=0
        SAVE    0,1,1,1                 ;  Z0,N1,V1,C1
34$:    SAVE    0,1,1,0                 ;  Z0,N1,V1,C0
32$:    BCC     40$                     ; C  branch if C=0
        SAVE    0,1,0,1                 ;  Z0,N1,V0,C1
40$:    SAVE    0,1,0,0                 ;  Z0,N1,V0,C0
30$:    BVC     42$                     ; V  branch if V=0
        BCC     44$                     ; C  branch if C=0
        SAVE    0,0,1,1                 ;  Z0,N0,V1,C1
44$:    SAVE    0,0,1,0                 ;  Z0,N0,V1,C0
42$:    BCC     46$                     ; C  branch if C=0
        SAVE    0,0,0,1                 ;  Z0,N0,V0,C1
46$:    SAVE    0,0,0,0                 ;  Z0,N0,V0,C0
                                        ;
RED:    PUSH    <0123>                  ; Save Registers
                                        ;The Stack now looks like:
                                        ;       -- PC (from JSR instruction)
                                        ;       -- R0
                                        ;       -- R1
                                        ;       -- R2
                                        ;  SP>-- R3
                                        ;
        MOV     #STR1,R1                ;
        JSR     R2,PRINT                ;
                                        ;
        JSR     R2,RNAME                ;
        .WORD   "R0                     ;
        MOV     6(SP),R1                ; Register Zero
        JSR     R2,OCTAL                ;
        JSR     R2,RNAME                ;
        .WORD   "R1                     ;
        MOV     4(SP),R1                ; Register One
        JSR     R2,OCTAL                ;
        JSR     R2,RNAME                ;
        .WORD   "R2                     ;
        MOV     2(SP),R1                ; Register Two
        JSR     R2,OCTAL                ;
        JSR     R2,RNAME                ;
        .WORD   "R3                     ;
        MOV     (SP),R1                 ; Register Three
        JSR     R2,OCTAL                ;
                                        ;
        CALL    CRLF                    ;
                                        ;
        JSR     R2,RNAME                ;
        .WORD   "R4                     ;
        MOV     R4,R1                   ; Register Four
        JSR     R2,OCTAL                ;
        JSR     R2,RNAME                ;
        .WORD   "R5                     ;
        MOV     R5,R1                   ; Register Five
        JSR     R2,OCTAL                ;
        JSR     R2,RNAME                ;
        .WORD   "SP                     ;
        MOV     SP,R1                   ; Stack Pointer
        ADD     #12,R1                  ;  calculate original value
        JSR     R2,OCTAL                ;
        JSR     R2,RNAME                ;
        .WORD   "PC                     ;
        MOV     10(SP),R1               ; Program Counter
        JSR     R2,OCTAL                ; print address of the next instruction
                                        ; in the calling program
                                        ;
        MOV     #STR2,R1                ;
        JSR     R2,PRINT                ;
                                        ;
        JSR     R2,RNAME                ;
        .WORD   " N                     ;
        BIT     #^B1000,SSAVE           ; N bit
        CALL    BIT                     ;
                                        ;
        JSR     R2,RNAME                ;
        .WORD   " Z                     ;
        BIT     #^B100,SSAVE            ; Z bit
        CALL    BIT                     ;
                                        ;
        JSR     R2,RNAME                ;
        .WORD   " V                     ;
        BIT     #^B10,SSAVE             ; V bit
        CALL    BIT                     ;
                                        ;
        JSR     R2,RNAME                ;
        .WORD   " C                     ;
        BIT     #1,SSAVE                ; C bit
        CALL    BIT                     ;
                                        ;
        CALL    CRLF                    ;
                                        ;
        MOV     #000240,CRI1            ; Create instructions to restore
        MOV     #000260,CRI2            ;  condition codes to the pre-RDUMP
        BIS     CSAVE,CRI1              ;  values.
        BIS     SSAVE,CRI2              ; The instruction in CRI1 will Clear the
                                        ; required condition codes while the
                                        ; instruction in CRI2 will set any that
                                        ; that should be set.  If none need to
                                        ; be set, each is left a NOP.
                                        ; Note: 260 is an undocumented NOP.
                                        ; Restore Everything
        POP     <3210>                  ;
CRI1:   .BLKW   1                       ; Created Instructions
CRI2:   .BLKW   1                       ;  to restore condition codes
        RETURN                          ; EXIT
                                        ;------------------------------------
                                        ; <<< Data >>>
                                        ;
STR1:   .ASCIZ  <15><12>/Register Dump/<15><12>
STR2:   .ASCIZ  <15><12>/Condition Codes/<15><12>
SPACES: .BYTE   40,40,40,40,40,40,0     ;
        .EVEN                           ;
SSAVE:  .BLKW   1                       ;
CSAVE:  .BLKW   1                       ;
                                        ;
                                        ;------------------------------------
                                        ; <<< RNAME >>>
RNAME:                                  ; Print the sequence: Space, Space
                                        ; 2 character register name, equal
                                        ; sign.
        MOV     #40,R0                  ; 40=space
        .TTYOU                          ;
        .TTYOU                          ;
        MOVB    (R2)+,R0                ; get 1st argument character
        .TTYOU                          ;
        MOVB    (R2)+,R0                ; 2nd, now R2 points to return address
        .TTYOU                          ;
        MOV     #'=,R0                  ;
        .TTYOU                          ;
        RTS     R2                      ;
                                        ;
                                        ;------------------------------------
CRLF:   MOV     #15,R0                  ; <<< CRLF >>>
        .TTYOU                          ;  print carriage return and line feed
        MOV     #12,R0                  ;
        .TTYOU                          ;
        RETURN                          ;
                                        ;
                                        ;------------------------------------
BIT:    BNE     10$                     ; <<< BIT >>>
        MOV     #'0,R0                  ;  print a 1 or 0 for the condition
        BR      20$                     ; code status.
10$:    MOV     #'1,R0                  ;
20$:    .TTYOU                          ;
        MOV     #SPACES,R1              ;
        JSR     R2,PRINT                ;
        RETURN                          ;
                                        ;
                                        ;------------------------------------
OCTAL:  PUSH    <02>                    ; <<< OCTAL >>>
        CLR     R0                      ;
        MOV     #6,R2                   ; Invocation: JSR R2,OCTAL
        BR      20$                     ;  with the value to be printed in
10$:    CLR     R0                      ; octal in R1.
        ROL     R1                      ;
        ROL     R0                      ;
        ROL     R1                      ;
        ROL     R0                      ;
20$:    ROL     R1                      ;
        ROL     R0                      ;
        ADD     #60,R0                  ;
        .TTYOU                          ;
        DEC     R2                      ;
        BNE     10$                     ;
        POP     <20>                    ;
        RTS     R2                      ;
                                        ;
                                        ;------------------------------------
PRINT:  PUSH    <0>                     ; <<< PRINT >>>
10$:    MOVB    (R1)+,R0                ;
        BEQ     DONE                    ;  Invocation: JSR R2,PRINT
        .TTYOU                          ; with the address of the string to be
        BR      10$                     ; PRINTed in R1.  The string must be
DONE:   POP     <0>                     ; be terminated by a zero byte.
        RTS     R2                      ;
                                        ;
        .END                            ;
```

ANSWER TO COMPUTER CROSSWORD PUZZLE which appeared in THE DEC* PROFESSIONAL Volume 3, Number 1. (January 1984), page 133.

JOHN K. YOUNG

P	A	C	K	■	A	S	S	E	M	B	L	Y
■	S	O	■	P	■	S	T	R	A	Y	S	■
D	■	L	I	R	A	■	U	N	I	T	■	C
I	■	L	O	O	P	■	D	E	L	E	T	E
S	L	A	N	G	■	F	■	■	■	S	O	L
K	■	T	■	R	■	I	N	D	O	■	I	L
■	H	E	X	A	D	E	C	I	M	A	L	■
D	A	■	E	M	I	L	■	G	■	D	■	G
U	S	A	■	■	■	D	■	I	N	D	I	A
M	A	S	T	E	R	■	S	T	A	R	■	T
P	■	C	O	D	E	■	T	A	P	E	■	E
■	A	I	R	I	N	G	■	L	■	S	H	■
P	R	I	N	T	O	U	T	■	U	S	E	D

RANDOM-ACCESS ROMANCE

By Robin C. Johnson, Stockton California

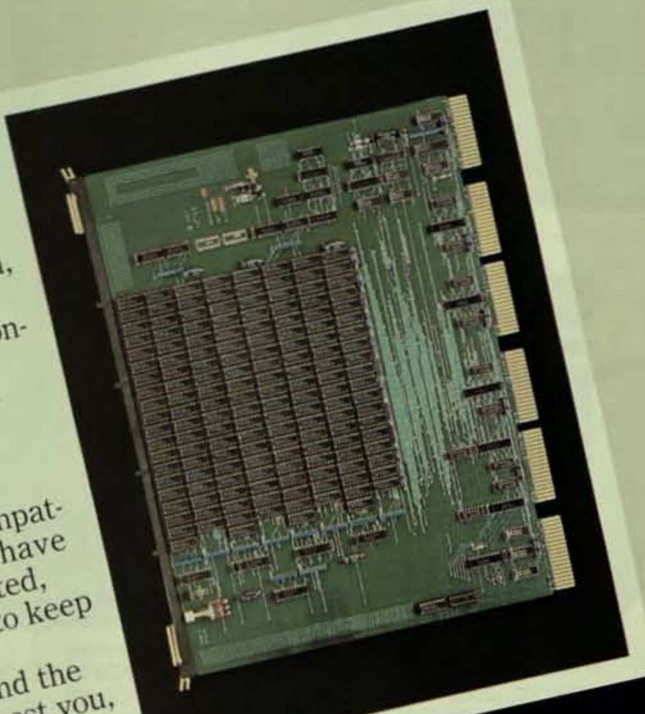
```
USER5       14:04       9-14-83
*****************************

MAIL>NEW MAIL FROM OPER9
MAIL>READ

Peg, you misqueued your
inventory again.  It's
all over our floor.
```

```
OPER9       14:10       9-14-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

Sorry, but Peg is on
vacation.  Where are you
located?  I'll send someone
over to get it.
```

```
USER5       14:13       9-14-83
*****************************

MAIL>REPLY FROM OPER9
MAIL>READ

We're in Suite 22, Lab B.
You must be new.
```

```
OPER9       14:18       9-14-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

Yes, I started yesterday.
I'm a nervous wreck
already.  Computers!  Each
one is different. I've
just come from a RSL-50.
```

MEGABYTES FOR MINIBUCKS



Now you can save on one-megabyte VAX-11/780 compatible memory boards from National. With availability now, you can cash in on our NS789 board at an unbeatable price.

Identical to DEC's board in function and form, the NS789 has one-megabyte of 64K based RAM, a two-year warranty and spare RAM on-board. And National offers an emergency loaner program if you ever need a replacement board.

Maximize your VAX potential.

Expand your memory capacity with compatible memory boards from National. We have the price and availability to get you started, with the service and long-term support to keep you going.

For complete NS789 specifications and the name of the representative/broker nearest you, call our Customer Support group toll free at 800-538-8510; in California, 800-345-4006.

National Semiconductor
MICROCOMPUTER SYSTEMS DIVISION

```
USER5        14:22        9-14-83
*****************************

MAIL>REPLY FROM OPER9
MAIL>READ

Don't worry, you'll get
the hang of it.
If you need any help,
let me know.  I'm Bob.
```

```
OPER9        14:25        9-14-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

Thanks, Bob.
I'll remember you when I
run into a snag.
Nancy.
```

"WE START THEM YOUNG!"

```
USER5     09:35     9-15-83
****************************

MAIL>NEW MAIL FROM OPER9
MAIL>READ

Morning, Nancy. How's it
going?
```

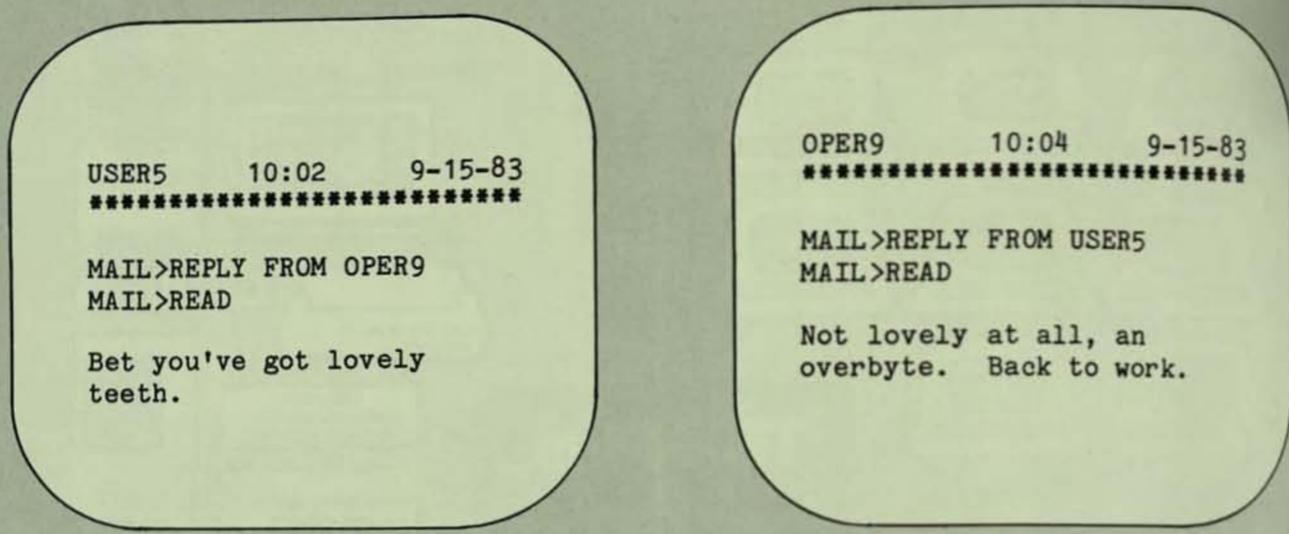
```
OPER9      09:43     9-15-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

Okay, I guess.  I dreamed
in binary last night.
```

```
USER5     09:47     9-15-83
****************************

MAIL>REPLY FROM OPER9
MAIL>READ

I dreamed about you.
Tell me about yourself.
```

```
OPER9      09:50     9-15-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

I'm a neurotic, overweight
divorcee with a failure
phobia.  This is my third
job in two years.  I LOVE
pizza.  How's that?
```

```
USER5     09:55     9-15-83
****************************

MAIL>REPLY FROM OPER9
MAIL>READ

I'm divorced too.
Want to go out for pizza
with me today?
Remember, we have a time
sharing policy at CEF.
```

```
OPER9      09:58     9-15-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

Sorry, not today, Bob.
I've got a dentist appt.
```


USER5 10:02 9-15-83
MAIL>REPLY FROM OPER9
MAIL>READ
Bet you've got lovely teeth.
OPER9 10:04 9-15-83
MAIL>REPLY FROM USER5
MAIL>READ
Not lovely at all, an overbyte. Back to work.

>>>CRYPTOLIST<<<

Cartoons by Johns

COMPUTER CRYPTOPUZZLE FUN________________

EXAMPLE: FORTRAN = GSLVLZO

AFOUVR ________________

KFWSASL ________________

BXBVZV ________________

KQLFWVSLX ________________

KFAFVF ________________

LFVDLO ________________

NZMFAZO ________________

LFWSNFL ________________

JZWYFV ________________

BXBUFO ________________

NZM NCB ________________

CQWLSNZM ________________

KZVZBZGF ________________

BRSI VQCF ________________

ASUGZQA ________________

BFAFWV ________________

KZVZVLQFNF ________________

ZWWFOV ________________

JZBWZA ________________

DOQEDB ________________

(Answer next issue.)

```
OPER9        09:20        9-16-83
****************************

MAIL>REPLY FROM USER5
MAIL>READ

You shouldn't have.
```

```
OPER9        09:26        9-16-83
****************************

MAIL>REPLY FROM USER5
MAIL>READ

You're quite an operator.
We hardly know each other.
```

```
OPER9        09:30        9-16-83
****************************

MAIL>REPLY FROM USER5
MAIL>READ

You've been spying.
Speaking of disk quota, my
disk is full.  I've been
making so many mistakes,
I've got twenty versions
of everything.  What now?
```

AUTHOR! AUTHOR!

THE DEC PROFESSIONAL WANTS YOU TO BE AN AUTHOR!*

THE DEC PROFESSIONAL is your magazine. You can make it better by contributing articles, programs or comments directly to us. Our authors are paid honoraria for published works, which because of their hard work, they deserve. We ask you to contribute; send us your manuscripts for possible publication (we prefer machine readable tapes or floppies in PIP, WORD-11, or ?? format) to THE DEC* PROFESSIONAL, Box 362, Ambler, PA 19002-0362, Attn: Editors. Thank you.*

PERSONAL and PROFESSIONAL, the first independent magazine for DEC personal computer users, is looking for qualified freelance writers.

If you're a writer with good ideas and a knowledge of the DEC personal computer market, call or write:

James L. Trichon, Managing Editor, **PERSONAL and PROFESSIONAL**, P.O. Box 114, Springhouse, PA 19477, (215) 542-7008

PERSONAL and PROFESSIONAL is published by the same folks who bring you **THE DEC* PROFESSIONAL** and **VAX/RSTS PROFESSIONAL.**

```
USER5      09:32      9-16-83
*****************************

MAIL>REPLY FROM OPER9
MAIL>READ

Do a purge to free up
some space.  Then free
up some space in your
schedule for me. How
about in the lounge in
five minutes?
```

```
OPER9        09:35      9-16-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

I can't.  Too frightening.
We don't even know what
each other looks like.
```

```
USER5      09:38      9-16-83
*****************************

MAIL>REPLY FROM OPER9
MAIL>READ

Never fear. Nancy.  The
computer says we were
meant for each other.
I ran our personnel
records through and
they came perfectly
merged.  Meet me?
```

```
OPER9        09:39      9-16-83
*****************************

MAIL>REPLY FROM USER5
MAIL>READ

Okay. No more chaperone.
Five minutes.
```

```
USER5      09:10      9-17-83
*****************************

MAIL>NEW MAIL FROM OPER9
MAIL>READ

Morning, Nancy.  I feel
great after last night.
I've rarely experienced
such a smooth interface.
```

```
OPER9        09:19      9-17-83
*****************************

MAIL>NO NEW MESSAGES
```

```
USER5       09:25      9-17-83
******************************

MAIL>NEW MAIL FROM OPER9
MAIL>READ

Nancy, are you there?  Is
something wrong?  Please
respond.
```

```
OPER9       09:31      9-17-83
******************************

MAIL>REPLY FROM USER5
MAIL>READ

Review your error log file
for byte capacity and real-
time faults.  Disabling
broadcast reception.
```

Does art really imitate life??
Thanks to David Bowers and SRC

BUBBLE SORTS:
A Second Look

. . . continued from page 132

Another optimization for the Bubble sort would be to remember where the last swap was made. All the elements that come after that point are in sorted order already, so you do not have to continue the sort any further. This is a generalization of simply decrementing the last element of each pass by one.

```
PROCEDURE Bubble4(INT n, INT ARRAY table[1:n]);
BEGIN
INT i, LastSwap; LOGICAL InSort;
EXTERNAL PROCEDURE Swap (INT a, INT b);
LastSwap: = (n-1);
LastItem: = LastSwap;
InSort: = FALSE;
WHILE (NOT InSort)
DO BEGIN
  InSort: = TRUE;
  FOR i: = 1 TO (LastItem - 1)
  DO IF (table[i] > table[i + 1])
    THEN BEGIN
      Swap (table[i], table[i + 1])
      InSort: = FALSE;
      LastSwap: = i;
      END;
  LastItem: = LastSwap;
  END;
END of Bubble4;
```

It's also possible to write a Bubble sort which does not use the "InSort" flag. Instead, it uses "LastSwap" to determine if the array is not yet in sort. However, it takes more work to get rid of the InSort flag than it saves. The program has to keep track of the history of the LastSwap to determine when the array is sorted and that requires as much or more code than setting the InSort flag.

It is important to consider the result of removing or adding variables to a program. In general, a variable should be introduced if it can save needless recalculation of a value.

There are other members of the pairwise exchange sort family. As an exercise, try to write a "Cocktail Shaker" sort. The Cocktail Shaker sort swaps pairs of array elements like a Bubble sort. However, when a swap is made, the program begins comparing elements in reverse order. For example, given the array (1, 3, 5, 4, 2) a Cocktail Shaker sort would perform these passes:

```
Pass one:
(1, 3, 5, 4, 2)
    =>      swap 4 and 5
Pass two:
(1, 3, 4, 5, 2)
    <=      reverse direction
Pass three:
(1, 3, 4, 5, 2)
    =>      swap 2 and 5
Pass Four:
(1, 2, 3, 4, 5)
    <=      Reverse direction
            swap 2 with 5, 4 and 3
```

PROFESSIONAL ^etc

educational
timely
communications

NOTES
MEETINGS
NEWS RELEASES
NEW PRODUCT INFORMATION

Send Notes, Meetings, News Releases, New Product Information, etc. to: PROFESSIONAL etc, P.O. Box 362, Ambler, PA 19002-0362.

CONTENTS

Dear DEC User:

Professional et cetera is your section for educational, timely, communications. It is a place for you to announce upcoming events or meetings and a place to comment on them.

It is a place for you to read about new products, and a place to comment on them.

It is a place for you to hear from companies whom you service or who service you — and a place to comment on them.

Please send all material to: Professional ETC, The DEC Professional, P.O. Box 362, Ambler, PA 19002-0362. All announcements must be in our office one month prior to publication.

The Publishers

PROFESSIONAL ^etc

NEW PRODUCTS

CLEARPOINT INTRODUCES THE Q-RAM 11B

Clearpoint, Inc. of Sudbury, Massachusetts will introduce a new high performance memory module for Q-Bus systems at DEXPO East 84. The Q-RAM 11B is designed for the new generation of Micro J/11 and MicroVAX processors, featuring very fast access time, block mode DMA, and bank selection. The dual-height board has 1 megabyte capacity, on-board Control and Status Registers, and an LED to indicate parity errors. The board will have a spare memory IC for easy maintenance.

Clearpoint will also display its Q-RAM 22B for the first time at DEXPO East 84. The quad-height Q-RAM 22B has up to 4 megabytes capacity, on-board CSR and block mode DMA. Both the Q-RAM 11B and the Q-RAM 22B feature a bank select option which allows up to 32 megabytes of addressable main memory with no hardware modifications on the Q-Bus.

For more information contact Clearpoint, Inc. 327/B Boston Post Rd., Sudbury, MA 01776, or come to booth 543 DEXPO East.

UNIX OPERATING SYSTEM FOR NS16000 MICROPROCESSORS

National Semiconductor Corporation today announced immediate availability of a UNIX operating system for its NS16000 Microprocessor Family. Designated GENIX, the system is a port of Berkeley 4.1 bsd UNIX with added features to allow full utilization of the mainframe-like architecture of the 16000 family, demand paged virtual memory and floating point arithmetic functions.

Over 200 utilities or special programs are included in the GENIX package, including uucp inter-system communications, source code control system, nroff document preparation, troff typesetter run-off documentation facilities and vi screen editor. GENIX supports the C Shell which allows extensive job control functions. The GENIX C compiler is an optimized version of the Berkeley 4.1 bsd industry standard portable compiler, producing compact, efficient code.

GENIX is available in source code form for OEM adaptation for $30,000 with over 1400 pages of documentation on system tape or hard copy. GENIX is also available on National's recently announced SYS16 Development System.

For more information come to Booth 525 at DEXPO East, or contact National Semiconductor Corporation, 2900 Semiconductor Dr., Santa Clara, CA 95051, (408) 721-5000.

HANDBOOK FROM DIGITAL FOR PERSONAL COMPUTER BUYERS

LOOK OVER THE RAINBOW HANDBOOK from Digital Equipment Corporation if you are looking to buy a personal computer for your business or professional use. Whether you work in an office, on a farm, in your home, or in a school, and regardless of the kind of work you do, the Rainbow 100 or Rainbow 100+ computer can help you be more productive and add a new dimension to your business or profession. This 275-page softbound pocket book covers everything you need to know about how to use a personal computer in your business and it helps you make an educated purchase. It includes a guide to the computer hardware and its options, a listing of over 700 software programs and information on how to get assistance.

The handbook is FREE. Call 1-800-DIGITAL, Ext. 160, or write Digital Equipment Corporation, 200 Baker Ave., CFO1-1/M94, W. Concord, MA 01742, attention: Media Response Manager. Digital Equipment Corp. will be an exhibitor at DEXPO East Booth 1818.

SPECTRA LOGIC ENTERS DEC LSI-11 MARKET

Spectra Logic Corporation has entered a new market by introducing a single-board, multifunction disk and tape controller designed for users of Digital Equipment Corporation's LSI-11 and PDP-11/23 computers.

The Q-Bus compatible SPECTRA 25 allows DEC users to attach removable pack or Winchester SMD disk drives and "start/stop" or "streaming" half-inch tape drives. The new controller, offering software transparency to standard DEC operating system and diagnostic software, attaches any combination of up to 2 SMD disk drives and up to 4 formatted tape drives without modifying operating system software.

The SPECTRA 25 emulates DEC's RMO2/5, RM80 and RP06/7 disk subsystems, as well as DEC's TS11 tape drives. Operating systems with which the controller is fully compatible include DEC's RT-11, RSX-11M, RSX-11M-PLUS and RSTS/E.

' Although the SPECTRA 25 is a high-performance multifunction disk/tape controller, its power consumption (typically 5 volts DC at 7 amps) is comparable to disk-only controllers.

Spectra Logic Corporation is a leading manufacturer of emulating single and multifunction disk/tape controllers and "streaming" software for Digital Equipment Corporation.

For more information, come to DEXPO East, Booth 108, or contact Spectra Logic Corp., 1227 Innsbruck Dr., Sunnyvale, CA 94089, (408) 744-0930.

5¼" REMOVABLE WINCHESTER REPLACES 4 RLO1 DRIVES

A 20.8 Mb removable hard disk system for all DEC Q-Bus microcomputers has been introduced by Winchester Systems Inc.

The 4xRL01 emulates four DEC

users? Emulex, of course!

FOR THE VAX-11/750...

TC7000—A single-board, software transparent controller that interfaces directly to the internal CMI to support 1-4 STC or 1-8 Pertec formatted type drives. Emulates DEC's TM03/TU77 with tape speeds up to 125 ips at 1600/6250 bpi. Supports both "old" and "new" GCR 6250 kinds of drives.

FOR THE VAX-11/780...

TC7000—The same board, with the flip of a switch, fits in the V-Master/780 chassis to provide transparent emulation of DEC's TM03/TU77 through the SBI. Supports 1-4 STC or 1-8 Pertec formatted type drives at tape speeds up to 125 ips; 1600/6250 bpi. Both "old" and "new" GCR 6250 technology is supported.

Communications Products.

FOR UP TO 16 LINES—CS21 SERIES...

CS21/F—Emulates the asynchronous portion of the DMF-32 for use on VAX-11s. Is software transparent with VMS Version 3.0 and above. Handles 16 lines per controller.

Statcon 21—Statistical concentration through the combination of the proven CS21 multiplexer with special microprogramming and the CM22/EX local statistical port concentrator. Handles up to 16 remote lines per statistical concentrator, up to 32 lines per controller.

FOR 16 TO 128 LINES AND MORE—CS11/CS32 SERIES...

CS11/F—Emulates the asynchronous portion of the DMF-32 for use on VAX-11s. Is software and diagnostic transparent, and can handle 16, 32 or 48 lines per controller.

Statcon 11—Combines the proven CS11 multiplexer with special microprogramming and one or more CM22/EX local statistical port concentrators.

CS32/F—A single-board communications controller that's totally software transparent to DEC's new DMF-32. One CS32 can handle up to 128 lines per controller board.

Statcon 32—Combines the CS32 multiplexer with special microprogramming and the CM22/EX local statistical port concentrator. A single CS32 controller board handles an amazing 256 remote and local lines in this statistical concentration mode.

For more information on Emulex products for VAX, call toll-free: (800) 854-7112. In California: (714) 662-5600. Or write Emulex Corporation, 3545 Harbor Blvd., P.O. Box 6725, Costa Mesa, CA 92626.

EMULEX

The genuine alternative.

CIRCLE D137 ON READER CARD

GSA Contract #: GSOOK8401S5575

PROFESSIONAL etc

NEW PRODUCTS

RL01 14-inch cartridge drives. It incorporates two high-performance fixed/removable 5 1/4" Winchester drives and a controller in just 5.25" of rack space. The unit holds two rugged, front-loading cartridges which are especially convenient to insert and remove.

The 4xRL01 provides totally transparent RL01 emulation. A half-quad wide Q-Bus adaptor card, which supports 22-bit addressing, simply plugs into any DEC LSI-11 microcomputer. The 4xRL01 also runs all standard LSI-11 family operating software plus RL01 diagnostics.

The single quantity price of the 4xRL01 is $10,000. A lower priced system, the 2xRL01, costs $7,000 and emulates two DEC RL01s. Both models are available for delivery two weeks after receipt of order.

Come to Booth 1908 DEXPO East, or contact Winchester Systems, Inc., 14 Laurel Hill, Winchester, MA 01890, (617) 933-8500.

CIS OFFERS VAX RESOURCE ACCOUNTING SOFTWARE

Computer Information Systems (CIS) will introduce and demonstrate two new VAX software products, VMSACTSYS and PROLOG, at the Boston DEXPO East conference April 3-6. VMSACTSYS is a menu-driven resource accounting package that monitors and bills system resources and software usage on Digital Equipment Corporation's VAX/VMS computers. PROLOG, an optional extension of VMSACTSYS, is a project resource accounting utility that allows a user to switch projects without logging out and back in each time. This newly released product is designed with a validity check to verify project-user combinations. Users have the capability of reporting and invoicing resource usage by project designation. Both packages operate on single VAX systems, VAX networks connected with DECnet and Digital's new VAXclusters.

CIS's Resource Accounting Product is priced at $3,800. The PROLOG option is priced at $800. The company has reserved Booths 625 and 627 at the Bayside Exposition Center Conference Center. CIS is an eight-year-old software and computer services firm located at 165 Bay State Drive, Braintree, MA 02184.

LOW COST HIGH-DENSITY GCR TAPE SUBSYSTEMS FROM CALIFORNIA COMPUTER

California Computer Group, Inc. (CCG) has introduced two new magnetic tape subsystems featuring high-density Group Coded Recording (GCR). The subsystems, designed to be integrated into Digital Equipment Corporation, Data General, Texas Instruments, Perkin Elmer or Hewlett-Packard minicomputers, offer dramatic savings in cost-per-megabyte of backup data storage.

Users may choose either the Storage Technology Corporation (STC) Model 2920 "Avalanche" dual density drive (1600/6250 BPI), or Kennedy's tri-density Model 9400 (800/1600/6250 BPI).

The drives are interfaced through a variety of controllers, made by Emulex, Western Peripherals, Rianda, Spectra Logic, Macrolink or Dylon. The choice of controller is determined both by the CPU host and by the price/performance criteria the user requires.

Single unit pricing for the STC 2920-based subsystem starts at $11,300. The Kennedy 9400-based subsystem starts at $12,100 in single units.

California Computer Group, Inc., is located at 3303 Harbor Blvd., Suite G-10, Costa Mesa, CA 92626. The company maintains a toll-free US telephone number: (800) 854-7488; in California: (714) 966-1661; telex: 183519 CCG CSMA. California Computer Group will exhibit at DEXPO East, Booth 121.

NEW PROGRAMMING TOOL SAVES TIME ELIMINATES LINE RETYPING

DCL-EDiT, a high-level programming tool which enables DEC users to edit DCL commands without time-consuming retyping of command lines, will be demonstrated by Syntacom Corporation at DEXPO East 84, April 3-6, Booth 121. DCL-EDiT exceeds the announced DCL editing capabilities of VMS 4.0, and is available for immediate delivery.

Compatible with VMS, RSTS and RSX operating systems, DCL-EDiT automatically tracks the user's last 22 commands, or approximately one full screen of prompts. The utility logs these commands upon request, allowing high-level edits to be entered, and submits the revised commands to DCL. No repetitive keystroking is necessary.

DCL-EDiT lists for $750. When employed in a system management or programming environment, DCL-EDiT proves to be a highly cost-effective tool — because it saves so much of the programmer's time. Syntacom will make demonstration copies available at DEXPO East 84 on a no-obligation basis.

Syntacom Corporation is affiliated with California Computer Group, Inc., and will share exhibit space with the latter company at DEXPO East 84. Syntacom is located at 3303 Harbor Blvd., Suite G-10, Costa Mesa, CA 92626, (714) 966-2782 telex 183519.

NEW STRUCTURED DEVELOPMENT METHODOLOGY DESCRIBED IN COLOR BROCHURE

A new structured methodology, created as the result of the growing interest in the structured techniques, is described in a two-color brochure available from AGS Management Systems.

The brochure features a new symbol, the "Trigle," used to represent a process on data flow

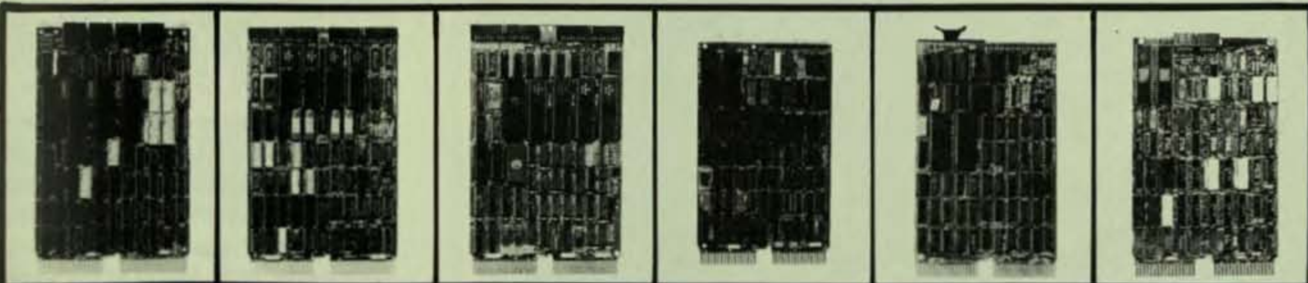
LSI-11 USERS:

SIGMA OFFERS

DEC* COMPATIBLE COMPUTER PRODUCTS

ENCLOSURES

MODULES



SCD-DLV11J
Dual-wide, 4-channel asynchronous serial line interface.

SCD-DLV11J/8
Dual-wide, 8-channel asynchronous serial line interface.

SCD-DZV11
Dual-wide, 8-channel communications multiplexer.

SDC-RLV12
Dual-wide, 5¼'' Winchester controller.

SDC-RXV21
Dual-wide, floppy controller.

SMF-V100
Dual-wide module with two asynchronous serial line interfaces.

SIGMA'S PERSONNEL ARE EXPERIENCED in design and development of DEC* related products and have a comprehensive background in system compatibility. This expertise is available to develop single board, high performance interface and controller boards that are firmware intensive and DEC* software compatible; chassis with user-configured front panels, power supplies, backplanes and drive enclosures.

SIGMA OFFERS a complete line of DEC* compatible disk controllers, interface devices and system enclosures for the LSI-11.

CALL YOUR nearest sales office for information on all products.

* Registered trademark of Digital Equipment Corporation

See us at
DEXPO® East 84
April 3-6, 1984
Booth 525

Σ Sigma Information Systems

6505C Serrano Avenue, Anaheim CA 92807
Telex 298607 SGMA (714) 974-0166

Orange CA (714) 633-0652 / 771-7320 / 639-5795
San Jose CA (408) 985-9112 • Arvada CO (303) 431-0551
Orlando FL (305) 628-8896 • Boston MA (617) 938-1310
Minneapolis MN (612) 881-5015 • Albuquerque NM (505) 294-5423
Beaverton OR (503) 644-0304 • Houston TX (713) 690-8011
Alexandria VA (703) 838-0098/99 • Bellevue WA (206) 454-4600
Quebec Canada (514) 331-5980

CIRCLE D121 ON READER CARD

diagrams. The "Trigle" was created to avoid the packaging prejudices of the different schools within the structured techniques.

Also included in the brochure, which outlines SDM/STRUCTURED, is a description of the physical structure of the methodology, its automated capabilities, major techniques employed, and a listing of sixteen tutorials developed as a result of extensive research into available literature.

A graphic representation of the SDM/STRUCTURED Life Cycle, consisting of five high level processes, has also been included in the brochure.

For free copies of the brochure, or additional information, contact: AGS Management Systems, Department MS, 320 Walnut St., Philadelphia, PA 19106 (215) 265-1550. AGS will be at Booth 124 DEXPO East 84.

APPLIED MANAGEMENT METHODS OFFERS TOPMAN, SCOUT

Applied Management Methods, Inc., the Doylestown, Pa. based project management consulting firm, has announced the introduction of a screen output processor program (SCOUT) to support TOPMAN, their highly regarded project management and control program. SCOUT greatly enhances the variety of outputs available for the TOPMAN system.

SCOUT is designed to aid a project management staff to review and analyze a project schedule and provide the user with a simple method of displaying information about a project through the use of menu-driven display screens. A user may select from a number of options presented on a "MAIN MENU" after the name of the project is entered. Some options call up screens for selecting and sorting data, and some permit the user to view the selected and sorted data in report form. "HELP" screens provide instructions to guide the user through the relevant selection and sorting menus.

SCOUT can display any requested activity in full detail, including dates, schedule and financial information. SCOUT can also select any group of activities by up to eight keys.

Additional information may be obtained from Applied Management Methods, Inc., 201 N. Broad St., Doylestown, PA 18901, or at DEXPO East 84, Booth 1904.

DECISION SUPPORT SYSTEM AVAILABLE FOR DEC RAINBOW

EPS, Inc. will announce the availability of its MicroFCS Decision Support System for the DEC Rainbow personal computer running under MS-DOS during DEXPO East 84, Booth 1910.

MicroFCS is totally "syntax compatible" with the FCS-EPS version of the Decision Support System on DEC/VAX. This feature enables users who are familiar with one version to become immediately knowledgeable when working with EPS software on another system.

In addition to running on the DEC

Rainbow, MicroFCS is compatible with the IBM PC and XT microcomputers and IBM-compatible hardware. The software is priced at $2,000 for a single copy, but quantity discounts are available starting with the second copy.

For more information about the MicroFCS Decision Support System, contact EPS, Inc. at One Industrial Dr., Windham, NH 03087, (603) 898-1800.

STREAMING TAPE SUBSYSTEM FOR DEC Q-BUS AND UNIBUS

Emulex Corporation has announced a new low-cost, high-performance 1/4-inch cartridge tape subsystem that brings enhanced storage and backup capabilities to DEC LSI, PDP, and VAX-11 computer systems.

Designated the Vault, the Emulex subsystem features Control Data's Sentinel 1/4-inch cartridge tape streamer, packaged in an attractive, compact desktop module, and the Emulex TC05 QBus or TC15 Unibus tape coupler. The Emulex Controller/Sentinel package emulates all functions of the DEC TS11 half-inch tape subsystem, and operates in a manner completely transparent to the pertinent DEC operating software and diagnostics.

List price for the Vault subsystem is $3,975, with quantity discounts available to OEM and volume purchasers. Standard delivery is thirty days ARO, with initial units becoming available for shipment on January 1, 1984.

For further information on Emulex products, contact Phillip (Flip) Begich, V.P., Domestic Sales, at Emulex Corporation, P.O. Box 6725, 3545 Harbor Blvd., Costa Mesa, CA 92626, (800) 854-7112, in California, (714) 662-5600. Emulex will be exhibiting at DEXPO East, Booth 837.

INTERFACE FOR DEC RAINBOW FROM NATIONAL INSTRUMENTS

National Instruments introduces an IEEE-488 interface, allowing the DEC Rainbow 100 to access over 2000 instruments, peripherals, and other computers. This high speed interface, also known as GPIB or HP-IB, serves engineering, scientific, business, and medical applications. Using one port, an absolute essential for computers with limited slots, up to 14 peripherals and/or instruments may be connected via a single GPIB interface.

The software includes a complete set of driver routines, an interactive control program, and support for applications written in Mark Williams C-86, Microsoft MBASIC, Digital Research CBASIC, Pascal MT+, or 8088 Assembly Language.

The Rainbow GPIB-PC is priced at $385 in single unit quantity. The Rainbow GPIB-PC with time-of-day clock and I/O port option is $485. The software driver and user's reference manual is $100. Delivery is from stock.

For further information, contact

Frances Drury at (800) 531-5066, or (512) 250-9119 in Texas, or come to Booth 711 DEXPO East.

PERIPHERAL PARTS SUPPORTS RP06 DRIVES

Peripheral Parts Support, Inc. exhibiting in Booth 1012 at DEXPO East 84 is now supporting Digital's RP06 Disk Drives with spare parts. Peripheral Parts Support, Inc. has large inventories of RP06 Disk Heads, Printed Circuit Boards, Spindles, Glass Doors, Linear Motors as well as other spare parts required in maintaining the RP06/Memorex 677 Disk Drives.

Peripheral Parts Support, Inc. not only offers, in most cases, immediate delivery versus their competitor's 30 day lead time, but also offers extended warranties and also provides cost savings of up to 70% from Digital's list prices for their RP06 spare parts.

For additional information on their new line of RP06 spare parts, Peripheral Parts Support, Inc. can be contacted at 68 Laurel St., Watertown, MA 02172, (617) 926-1661.

FILE MANAGEMENT KIT FOR RSTS/E SYSTEMS FROM GROWTH SOFTWARE

The File Management Kit for users of DEC's RSTS/E Operating System will be introduced by Growth Software Systems, Inc. at DEXPO East 84. Components of the Kit to be demonstrated include four utilities designed to ease the day to day management of files by either individual users or by a system manager.

The File Manager Utility provides features for data analysis, security management and user file manipulation. Through an English language command interface, it allows flexible definition of recurrent file management procedures. The File Conversion Utility allows file format and data type conversions driven by simple user defined tables. The Data Recovery Utility may be used to extract data from corrupted files including DEC's RMS-11K indexed structures. The File Inquiry Utility gives the user quick access to data stored in several formats. It includes both RMS record and Block-IO access.

For further information, contact Gerry Tolman, Growth Software Systems, Inc., 5 Hawthorne Village, Concord, MA 01742, (617) 369-0635. Visit Booth 1011 at DEXPO East.

ACCESS INTRODUCES 20/20 ON THE DEC RAINBOW

Access Technology, Inc. will introduce 20/20, its integrated decision support package, on the DEC Rainbow at DEXPO East 84.

20/20 combines a powerful spreadsheet with graphics, database management, project scheduling and internal and external interfaces. But 20/20 is integrated a second, more important way — into the corporation. It can exchange data with other application packages, and models can be moved between machines easily and efficiently. 20/20 gives personal computing better access to central databases, as well as to application packages already installed in the organization.

Access' first product, Supercomp-Twenty, is the leading spreadsheet on DEC machines. Both products run on a variety of machines, including the entire range of DEC computers. Access Technology, Inc., founded in 1980, has over 6,000 licensed sites primarily in Fortune 1000 companies. The company will be located at DEXPO East 84 Booth 206, or contact Access Technology, Inc., 6 Pleasant St., South Natick, MA 01760, (617) 655-9191.

B. G. ENTERPRISES DISPLAYS LARGEST WINCHESTER WITH THIN-FILM HEADS

DEXPO East 84 show-goers will get their first look at Control Data's 9771 Extended Module Drive with capacity to provide 825 million bytes (825 megabytes) of unformatted storage when they visit B.G. Enterprises, Inc., Booth 211.

CDC's highest capacity Winchester uses thin-film heads and improved oxide media. The XMD 14-inch drive is a rack-mounted unit that stores data on five disks in a sealed module. It will be displayed in tandem with a CDC Keystone 9218X series 1/2-inch streaming tape, reel-to-reel transport. The subsystem is available with Emulex or SpectraLogic controllers.

Further information may be obtained from B.G. Enterprises, Inc., 1116 W. Collins Ave., Orange, CA 92668, (714) 771-7770.

NEW SPACEWRITER IS VT100 COMPATIBLE WP TERMINAL FOR DEC MARKET

Quadrex Information Systems Ltd., a Montreal based supplier of innovative word processing systems, has introduced a new product, the SPACEWRITER, a VT100 compatible word processing terminal, at DEXPO East 84 in Boston, April 3-6. SPACEWRITER uses a 16-bit microprocessor with up to 512K of WP software on a field-upgradeable "ROMPAK," plus up to 512K of internal RAM for text storage.

All word processing is performed on the board without requiring a single CPU cycle from the host. Files are stored on the disks of the host. An optional floppy or hard disk can be added to SPACEWRITER to provide local storage.

Quadrex previously released the QX-8000 retrofit board for the DEC VT100; this product is now also available for VT101s, 102s and 131s, providing them with the powerful Quadrex WP capabilities.

Quadrex will be demonstrating both the QX-8000 and SPACEWRITER at DEXPO-East in the Bayside Exposition Center in Booth 1106. Or contact the company at 50 Milk St., Suite 1500, Boston, MA 02109, (617) 451-6509.

ONE MEGABYTE ECC MEMORY FOR VAX SERIES COMPUTERS

A new high density one megabyte add-in memory packaged on a single hex size module is now available from Cyberchron Corporation for the VAX 750, 730, and 725 series computers. Based upon the 64K dynamic RAM, the card contains additional ECC syndrome bits to permit the detection and correction of all single bit errors and the detection of most multiple bit errors. An on-board switch allows the memory to be taken "off line" for diagnostic purposes without removal of the memory card. The CVM-5030 is fully plug compatible with the DEC M8750 add-in memory expansion module, and the memory array is fully socketed for easy maintainability.

A two year parts and labor warranty is standard. Delivery is from stock and generous OEM discounts are available.

For further information, contact Mr. Yuval Brash, Cyberchron Corporation, P.O. Box 198, Garrison, NY 10524, (914) 424-3755, TWX: 710/573-2260. Booth 804 DEXPO East 84.

BATTERY BACKUP BOARD CLOCK CALENDAR FROM GRANT

Grant Technology Systems Corporation, Chelmsford, announces Model 307, a clock calendar with battery backup board, bus compatible with the DEC Q-bus LSI-11 Microcomputer Series. The Model 307 is a complete time of day clock and calendar with battery backup provisions and is used in a wide variety of applications in which the user wishes to have a recording of time and date that relates to a certain specific event that is being monitored.

It features:

- Standard DEC Dual Height Board
- Complete Time of Day Clock and Calendar
- Battery Backup

PROFESSIONAL etc

NEW PRODUCTS

• Automatic End of Month Recognition

• Automatic Leap Year Compensation

• Programmable Square Wave Output

• Alarm and Periodic Rate Interrupts

More information may be obtained from Grant Technology Systems Corporation, 11 Summer St., Chelmsford, MA 01824, (617) 256-8881. Visit Grant at Booth 106 DEXPO East 84.

NEW MULTI-FUNCTION SYSTEM FOR RSTS/E

UTL, a multi-function system utility for use on computer systems operating under RSTS/E, is now available from Software Techniques Incorporated.

Priced at under $350, UTL provides complete on-line control of system operation from a single program. According to Rick Scherle, president, "UTL provides the user a far more efficient way of doing his job, at a very low price." UTL replaces seven different utilities with one convenient program, saving both time and system memory for other, more important tasks. UTL provides all the functions of UTILTY and TTYSET, and supports all the important features of SYSTAT, REACT, ATPK, UMOUNT, and SHUTUP.

Software Techniques Incorporated is an international consulting firm which specializes in the analysis, design, and development of business automation system. For more information about Software Techniques Incorporated and the products and services marketed by the company, see us at DEXPO East Booth 724 or contact Paul Carey, communications manager. Software Techniques, Inc., 5242 Katella Ave., Los Alamitos, CA 90720, (714) 995-0533.

NEW BLAST PRODUCTS ANNOUNCED FOR DEC VAX

Communications Research Group has announced BLAST file transfer for IBM mainframes running under MVS/TSO or VM/CMS. The new communications software products will allow IBM hosts to exchange binary and text files with any of the more than 60 mini and micro computers for which BLAST has already been implemented, including DEC VAX and Rainbow (MS-DOS).

BLAST file transfer uses any asynchronous modems, at any speed, and standard RS-232 serial ports. Special protocol features allow efficient operation on noisy telephone lines and on circuits subject to propagation delay such as satellite links, packet-switching networks, and local-area networks. BLAST supports a virtual text file format, allowing transfer of text files between any combination of systems, automatically converting text file formats.

BLAST is a product of Communications Research Group, Inc., Baton Rouge, Louisiana, and is priced at $250 to $2495, depending on computer and operating system. The company will have information on BLAST at DEXPO East 84 Booth 624.

NEW MODEL CONVERTER FOR LSI-11 BUS FROM ADAC

With sampling rates of 200 per second, ADAC's new Model 1114AD is both the most flexible and fastest wide range — 10 millivolts full scale to 10 volts full scale — A to D converter on the market for LSI-11 bus. Expansion modules 1114SX and 1114VX permit system expansion to 128 channels with up to 250 volt common mode isolation and no loss of performance or sampling speed.

This 1114 A/D system is so flexible that it allows unrestricted mixing of high and low level analog input signals and seven different thermocouple types simultaneously. The user has full capability to programmatically characterize each specific channel.

ADAC also offers a complete line of accessory panels for use with the 1114 boards to facilitate wire connection and signal conditioning for thermocouples, RTD's load cells, strain gauges and other input signals. Panels include convenient ribbon cable connection to the system boards.

The board is half-quad, single height in size and is available for shipment within 30 days from order. For complete details and pricing, contact ADAC Corporation, 70 Tower Office Park, Woburn, MA 01801, (617) 935-6668. ADAC will exhibit at DEXPO East Booth 826.

DATA ANALYSIS, MODELING PACKAGE NEW FROM INTERACTIVE SYSTEMS

Interactive Systems, Inc. recently announced the release of MENTOR, an online data analysis and modeling package for Digital Equipment Corporation's VAX-11, DECsystem-10 and DECSYSTEM-20 computers.

MENTOR utilizes literal names in addition to the conventional method of numerical row and column access. Models that have been developed in several different departments or divisions do not have to be laid out identically, since MENTOR can perform a consolidation on multiple models that use the same name identifiers. This capability is extremely helpful when utilizing MENTOR's top-down budgeting function, which permits deconsolidations of corporate budgets. This literal name approach also eases the use of calculations through formulas.

Another unique feature of MENTOR is its interactive report writer which permits the flexible reporting of data in a model without affecting the integrity of the model itself. Once a report format has been created, it can be saved for future use. Various report formats accessing the same model are also available.

MENTOR is available at an introductory license fee of $5000 from Interactive Systems, Inc., 131 Middlesex Turnpike, Burlington, MA 01803, and will be demonstrated at DEXPO Booth 813.

COLLIER-JACKSON ADDS CJ/PERSONNEL TO FINANCIAL SOFTWARE LINE-UP

Collier-Jackson will debut CJ/PERSONNEL — an online personnel resources and reporting system for Digital's VAX line of mini computers — at DEXPO East 84. The system provides comprehensive analysis and control of personnel data for Human Resources Management. CJ/PERSONNEL will be available in August, 1984. Other CJ/BUSINESS INFORMATION SYSTEMS to be shown or discussed include:

CJ/ADVANCED GENERAL LEDGER — an online general ledger and financial reporting system providing comprehensive analysis control and forecasting features plus more sophisticated capabilities for intracorporate transactions, true variable (flex) budgeting, and highly individualized financial reporting.

CJ/PAYROLL — this flexible payroll processing system provides automatic processing from time transaction entry through general ledger updating for variable pay periods.

CJ/ACCOUNTS PAYABLE — provides cash requirements forecasting and disbursements functions to achieve increased cash flow management.

CJ/ACCOUNTS RECEIVABLE — this invoicing and accounts receivable system provides control of customer account status for sales and collection analysis.

Collier-Jackson develops, markets, installs, maintains and supports a growing family of online business management software for Digital Equipment Corporation's family of VAX 32-bit mini computers. Direct inquiries to Collier-Jackson, Inc., 3709 Jetton Avenue, Tampa, FL 33629, (813) 251-1077. Collier-Jackson will be at Booth 233, DEXPO East.

ROSS PLANS PREMIER SHOWING OF NEW A/P PACKAGE AT DEXPO EAST

Ross Systems' integrated and interactive accounts payable package with extensive cash management features, designed specifically for use on the DEC VAX minicomputers will be premiered at DEXPO East 84, April 3-6, in Boston at Booth 613.

Using its full interactive capabilities, MAPS/AP handles sophisticated payables processing, check writing, vendor information, cash management, inquiries and reports.

Designed to integrate with all Ross software, MAPS/AP is particularly effective when used as a complement to Ross' comprehensive general ledger/financial management package, MAPS/GL. For example, MAPS/AP automatically updates payables to the general ledger and provides a GL distribution by multiple account structures. In addition, processing screens conform to the MAPS/GL format, reducing the time needed to become familiar with each system.

MAPS/AP, backed by a strong support team, will be available in the spring of 1984 at a cost of $15,000 — $20,000 depending on the VAX model. Further information is available from Ross Systems, 1860 Embarcadero Road, Palo Alto, CA 94303, (415) 856-1100.

MARWAY PRODUCTS PRESENTS NEW FRONT PANEL MOUNT POWER CONTROLLERS

Marway Products has announced the introduction of a complete line of new front panel mounted power controllers for use in data processing, testing and military environments.

More than 20 new models have been released. Included in this product line are low power distribution strips, single phase controllers, three phase units and a variety of options such as special EMI filters, remote control and emergency shut down. All models are available in domestic and international versions.

These power controllers eliminate power surges, glitches and line transients which can cause equipment malfunctions such as memory loss, software problems, disk problems, printer interaction and other forms of erratic operation. Special features and options can be added to these units to achieve a custom device.

For further information, please contact Marway Products, Inc., 311 N. Clara Street, Santa Ana, CA 92703, (714) 973-1800, and see us at DEXPO Booth Number 728.

SINGLEBOARD BUBBLE MEMORY SYSTEM NEW FOR LSI-11 MICROMPUTERS

The Bubbl-Tec division of PC/M, Inc. has announced the availability of the industry's first single-board magnetic-bubble mass-storage memory system for Digital Equipment Corporation (DEC) LSI-11 microcomputers.

Bubbl-Tec's new QSB-11 BUBBL-BOARD bubble-memory system for LSI-11s is comprised of a single 1M-bit bubble device (with its attendant support circuitry), plus a complete controller which emulates the DEC RX01 floppy-disk system. The controller handles bubble-device formatting and control, interfaces the bubble-memory system to the LSI-11 bus structure, and provides for both soft- and hard-error detection and correction.

The bubble-memory system is fully compatible with all DEC LSI-11 systems, including the original LSI-11, the LSI-11/02, the LSI-11/23, the new SBC-11/21 FALCON, and the LSI-11/23-PLUS.

The QSB-11 is priced at $1614, quantity ten. OEM discounts are available. Delivery 30 days. For more information contact Al Foreman at Bubbl-Tec, 6800 Sierra Court, Dublin, CA 94566, (415) 829-8700; TWX/Telex 910-389-6890. Stop by DEXPO Booth 709.

MAGNETIC TAPE COUPLER INTERFACES CDC SENTINEL STREAMING CARTRIDGE DRIVE

A new 1/4-inch Magnetic Tape Coupler, that interfaces a single CDC 1/4-inch Sentinel streaming cartridge drive, is now available for DILOG

(Distributed Logic Corporation) Garden Grove, California, for interface to DEC LSI-11 — 11/23PLUS and MICRO/PDP-11 computers.

Designated the Model DQ342, the controller is complete on a single quad printed circuit board that contains the circuitry for TS-11/TU80/TSV05 compatibility under RT-11, RSX-11 and RSTS operating systems.

The controller features 22-bit addressing for 4-megabyte addressing, block mode DMA, switch selectable interrupt vector and register address, plus DMA four word burst size.

Model DQ342 is priced at $1085.00 in QTY 100; delivery 30 days ARO. Direct inquiries to Mr. Dennis Edwards, Mgr. No. Amer. Sales, DILOG (Distributed Logic Corporation), 12800 Garden Grove Blvd., Garden Grove, CA 92643, (714) 534-8950. DILOG will be available at DEXPO East 84, Booth 638.

INTECOLOR ANNOUNCES FASTSCREEN FOR E8001G TERMINALS

Intecolor Corporation announces FastScreen, a computer memory facility that permits displays to be created on an E8001G color graphics terminal screen in one quarter of a second.

Since it uses computer RAM memory, FastScreen provides faster access to information than do conventional flexible-media storage devices. Any of 63 different screens can be almost instantly displayed on the E8001G color graphics terminals equipped with single FastScreen with 512KB storage. Up to 126 screens are available with dual FastScreen, which provides more than 1MB of storage.

The quantity-one price for single FastScreen is $1245; the dual version is $2450. FastScreen is supplied on a printed circuit board for use in the Intecolor terminal card cage. Further information is available from Intecolor, Intecolor Drive, 225 Technology Park, Norcross, GA 30092, (404) 449-5961, TWX 810-766-1581. Intecolor will exhibit its products at DEXPO East, Booth 812.

LINKDATA TO EXHIBIT IBM 9000 MICRO THAT RUNS DEC SOFTWARE

LinkData, Inc. will exhibit a multi-user IBM microcomputer that runs DEC commercial software at DEXPO East 84, April 4-6 in Boston.

The new UNI-DOS operating system allows the IBM 9000 microcomputer to run the complete library of DEC DIBOL business application software. LinkData's programming language is source-to-source compatible with the DEC DIBOL language and UNI-DOS emulates both the DEC CTS-300 and CTS-500 operating systems. LinkData also offers a powerful Applications Generator that allows creation of complete custom systems without writing source code.

At DEXPO East 84 LinkData will be seeking qualified DEC dealers who wish to market LinkData's new IBM 9000 through the IBM Value training, service and sales support. For further information, visit LinkData at Booth 645 at DEXPO East 84 or contact Al Danza, LinkData, 2005 Route 22, Union, NJ 07083, (201) 964-6090.

FIRST COMPUTER CORPORATION TO UNVEIL NEW PRODUCTS AT DEXPO

First Computer Corporation will introduce four hardware products at DEXPO East 84, April 4-6 in Boston's Bayside Expo Center. ORION 750-I is a versatile new minicomputer system with twice the capacity, speed and performance of First Computer Corporation's Orion 730 system. Based on the VAX 11/750 CPU, Orion 750-I has 2MB of memory plus 134.8MB of storage (formatted) provided by the Fixed Storage Drive (FSD) disk, emulating dual RMO3s. Mounting space is available for an additional FSD or a 67.4MB Removable Storage Drive (RSD). Orion 750-I features an autoloading, self-threading high performance CacheTape 125 tape drive that provides up to 92MB of backup power, emulating a super performance TS-11. The system includes a free-standing LA120 DECwriter III Printing Terminal and a VAX/VMS license.

TAURUS 73 is a new microcomputer system designed to deliver minicomputer performance. Ideal for multi-user commercial applications, it contains 160MB of disk storage plus 46MB of tape storage. The system includes an FSD TSV05-FC tape drive to maximize storage capacity.

The PHRASE ENCODED (PE)/GROUP CODED RECORDING (GCR) TAPE SUBSYSTEM can be configured into systems to provide up to 180MB of backup storage capacity. Auto threading, it has dual density (1600/6250 bpi), 50 ips, start/stop tape with tape transport, formatter/controller, power supply and resident microdiagnostics.

Details of the fourth new product, an addition to First Computer Corporation's UNIX product line, will be announced at DEXPO East 84 on April 4. For further information, visit First Computer Corporation at Booth Number 930 or contact First Computer Corporation, 645 Blackhawk Drive, Westmont, IL 60559, (312) 920-1050.

NASSAU SYSTEMS OFFERS OVERTEMPERATURE PROTECTION

Nassau Systems, of Cincinnati, Ohio, will be exhibiting an Overtemperature Protection Hardware Accessory for all Digital Equipment Corporation Computer Systems, at the DEXPO East 84 show in Boston. This will be the first trade show display of an Overtemperature Protection Device specifically intended for all standard DEC Computer systems which includes an optional automatic telephone dialer and voice-alert capability. The product is designed to assist all DEC equipment OEMs and end users in preventing equipment damage from overtemperature conditions and has been endorsed by DEC Field Service for this purpose. A warning alarm and total system power shutdown at preset temperature limits are provided in order to meet the environmental specifications in DEC Field Service maintenance contracts, as well as those of third party service organizations. This product is fully compatible with all DEC computers equipped with DEC-standard power control and distribution systems, including all PDP11, VAX, DECSYSTEM-20, and DECsystem-10 equipment. The Model C-101 Overtemperature Protection System and the automatic telephone dialer — voice alert option will be on display at Booth 112. For further information, contact Nassau Systems Company, P.O. Box 19329, Cincinnati, OH 45219, or phone (513) 231-1283.

DATARAM INTRODUCES SINGLE BOARD UDA50-COMPATIBLE CONTROLLER

Dataram Corporation announces its S35/U UDA50-compatible controller for use with Digital Equipment Corporation's PDP-11 and VAX minicomputers. The S35/U operates with the UDA50 I/O driver which is supported by all of DEC's current operating systems.

The S35/U uses DEC's new Digital Storage Architecture (DSA) and Mass Storage Control Protocol (MSCP) which is utilized by DEC's UDS controller. UDA-50 compatibility requires a level of controller intelligence not available in the minicomputer market prior to the introduction of the UDA50 controller by Digital Equipment Corporation.

The main feature of the S35/U, necessary for DSA operation, is that it responds to a generic type I/O driver designed to operate with essentially any capacity drive. The only basic restrictions for drives interfaced to the S35/U controller is that they have an SMD interface and their data rates do not exceed 1.8MB/second.

The S35/U controller can operate with up to four drives of any mixture of capacity or speed.

The S35/U, including diagnostics, is $6800 in single quantity. Cable assemblies are available from Dataram at an additional charge. Pricing is FOB factory and terms are net thirty (30) days. Delivery is sixty (60) days after receipt of an order. A full one-year warranty is standard. For further information contact Dataram Corporation, Princeton Road, Cranbury, NJ 08512, (609) 799-0071. Come to Booth 313 at DEXPO East.

AIR FILTRATION PRODUCTS DISPLAYS PRODUCT LINE

Air Filtration Products, Inc., a leading supplier of quality, competitively priced disc drive air filters, has announced that its product line will be on display at DEXPO East 84, Booth 117.

The Air Filtration Products display will emphasize the importance of frequent replacement of filters for DEC disc drives. Air Filtration Products maintains one of the largest in-house inventories of filters for all models of DEC disk drives.

Samples and cutaway models of absolute and panel prefilters for DEC disc drives will be featured. The booth will also display the advanced double-grill-design ALFCO instrument fan filters along with others which are available for a wide variety of mini and mainframe computer and data processing equipment.

For more information, call toll free (800) 528-1585; or contact Ralph I. Brooks, Marketing Manager, 1806 West Grant Road, Suite 103, Tucson, AZ 85745, (602) 624-2272.

PROFESSIONAL etc

PRODUCT UPDATES

WORD PROCESSING INTERFACE FORMS ONE CENTRAL FILE

Battelle's Software Products Center will demonstrate at the DEXPO East 84 Show its BASIS-OA, a new word processing interface for minicomputers that is also user-friendly.

Battelle will be in Booth 828 at DEXPO East 84, to be held April 4-6 in Boston.

The BASIS-OA system forms one central file for all data created by office word processing systems throughout an organization. From word processors on the network, company employees can access the central file to search and retrieve needed reports, letters, policies, memos, or other written communications.

This interface is machine independent and currently is available on DECmate and DEC VAX with other interfaces being developed.

BASIS-OA is an enhancement to Battelle's BASIS Data Management System, now serving more than 1,300 organizations worldwide. BASIS, which also will be demonstrated, can be used in a broad array of textual and numeric applications, including newspaper and technical libraries, records management, engineering, litigation, and many more.

Battelle, 505 King Ave., Columbus, OH 43201, (614) 424-7728.

SCRIBE FROM UNILOGIC SUPPORTS QMS LASERGRAFIX

The Scribe Document Production System, widely-used office automation software for DEC 10, 20 and VAX, now supports the QMS LASERGRAFIX 1200 intelligent page printer.

Scribe is most often used to produce reports, technical manuals, proposals, financial summaries, and similar documents. It has special features to automatically generate bibliographies, indexes and tables of contents, to produce mathematical and scientific notation, and to integrate charts, drawings and other computer-generated graphics into text. More than 10,000 of Scribe's daily users work from DEC host computers. In addition to the QMS LASERGRAPHIX 1200, Scribe sites have the capability to print documents on many other devices, including a wide range of other laser printers, photocomposers, high-resolution dot-matrix printers, and letter-quality typewriters. Scribe is priced according to processor type with both non-profit and educational discounts available.

The QMS LASERGRAFIX 1200 offers a resolution of 90,000 dots per square inch, speed of up to 12 pages per minute, and the capability to print on plain paper using a dry two-component toner. Often used for graphics and word processing applications, its ability to bit-map a full 8½" x 14" page in printer memory means that graphics and text may be mixed on the same page. It is priced at $24,995 a single unit with OEM discounts available. For more information about Scribe, its QMS LASERGRAPHIX 1200 support, or its support of other printing devices, contact UNILOGIC. LTD., 160 North Craig Street, Pittsburgh, PA 15213 (412) 621-2277, or come to DEXPO Booth 608.

MINITAB NOW ON UNIX

Minitab, Inc. announces the latest release of Minitab Data Analysis Software for VAX-11 computer systems running under UNIX 4.1BSD. This UNIX version of Minitab Release 82.1 will be exhibited for the first time at DEXPO East 84. It joins Minitab versions for other Digital equipment, including the PDP-11, LSI-11, DECsystem-10 and DECSYSTEM-20 running under RT-11, TSX-PLUS, RSX, IAS, RSTS, TOPS-10 and TOPS-20. All the Digital versions support all of Minitab 82.1 features, including data management, descriptive statistics, regression and correlation, table generation, nonparametric statistics, matrix operations, and macro and looping capabilities.

The UNIX version of Minitab uses the VAX-11's virtual storage to analyze up to 1000 variables and one million data points. It can access and create files anywhere in the UNIX hierarchical file system, and supports the input/output redirection features of the UNIX shell. On-line HELP complements Minitab's easy-to-use interactive command language. The version installs in minutes and no optional software or special hardware is required to use the product.

Information and hands-on demonstrations of the Minitab system will be available in Booth #743 at DEXPO East 84 in Boston, April 3-6.

Additional information about all versions of Minitab is available from Minitab, Inc., 215 Pond Laboratory, University Park, PA 16802. Technical details about the UNIX version are available from Joiner Associates, Inc., 1124 Edgehill Drive, Madison, WI 53705.

RJ-11 COBOL COMPILER AVAILABLE FOR DEC PRO

EEC Systems, Inc. of Sudbury, Massachusetts announce that their RJ-11 COBOL Compiler is now available for the DEC Professional personal computer under the P/OS operating system. RJ-11 COBOL is compatible with the 1974 ANSI COBOL standard but has a number of extra features providing an extremely fast, efficient and flexible compiler. A spokesperson for EEC Systems said that given the existing concentration of sales in the educational field, it is reasonable to expect that that will continue to be a major focus of the sales effort. RJ-11 COBOL is competitively priced at $450.00 which includes the runtime system and is available for immediate delivery.

For more information contact Eric Dickman, EEC Systems, Inc., 327/E Boston Post Road, Sudbury, MA 01776 (617) 443-5106. EEC will be at DEXPO East Booth 513.

TEC TO EXHIBIT COBOL BASED SYSTEM FOR VAX

TEC Computer Systems, Newton, Massachusetts announces the introduction of a COBOL based manufacturing system for Digital Equipment Corporation's VAX minicomputers. The TEC Manufacturing System uses Digital's CODASYL Database Management System, VAX-11 DBMS. The system includes a sophisticated security system and a high level inquiry language and report writer.

The TEC System's powerful application functionality is designed to increase productivity, reduce costs, and improve customer service for medium to large manufacturing firms. The system automates the following areas: Inventory Control, Bill of Materials, Master Production Scheduling, Material Requirements Planning, Capacity Planning, Production Control, Purchasing, Sales Order Processing and Costing.

TEC has specialized in serving the needs of manufacturing companies since 1976. The company has over 65 installations in the Northeast area. The addition of the COBOL product provides a system with increased sophistication to meet the needs of companies with complex manufacturing environments.

For further information contact TEC Computer Systems, Inc., 30 Tower Road, Newton, MA 02164 or come to Booth 723 at DEXPO East.

VIKING FORMS MANAGER ON DEC PRO-300

Viking Software Services, Inc. of Tulsa, Oklahoma, will be demonstrating its Viking Forms Manager, the online screen formatter and data entry software for Digital Equipment Corporation's PRO-300 series personal computers at DEXPO East 84, April 3-6 in Boston. The Viking Forms Manager, or VFM, was originally developed on VAX computers and has since been transported to the PDP-11, the IBM-PC and now to the PRO-350.

All programming is eliminated to create, update and key-verify data files. All the features and functionality of the larger VAX system are available on the PRO including: three levels of data validation, complete control over which data fields are entered, human engineered special function keys, selective data duplication, and many other features required for efficient and productive data entry.

Viking expects that VFM will be used to solve many types of data entry problems. For example, data files can be created on a PRO and then transmitted to a host computer for processing. There are many variations of this theme, ranging from remote order entry to replacement of keypunch and key-to-disk machines. The applications are endless.

VFM on the PRO-350 consists of a Forms Development facility, the Viking Data Entry System (VDE), and a variety of utility programs. The whole package is available with the traditional Viking 30-day trial and full one-year warranty. The end user price is only $600 with liberal discounts for quantity purchasers and OEMs.

For more information contact Viking Software Services at 2815 E. Skelly Dr., Tulsa, OK 74105, or call (918) 745-6550. Viking Software will be at Booth 822 DEXPO East.

NEW PACKAGING FOR MICOM'S MINIATURE LOCAL DATASETS

MICOM SYSTEMS is now offering its Model 430 Mini Drivers and Model

PRODUCT UPDATES

431 Mini Datasets in cartons of 10, with a quantity discount.

Extremely popular for support of data communications for asynchronous terminals and for personal computers, the line driver is used for in-house or on-campus applications, the local datasets for limited distance transmission over private line "metallic" circuits where provided by the local telephone company.

Both operate full-duplex over 4-wire circuits, the line driver at speeds of up to 19,200bps for distances of over a mile, the local dataset at speeds of up to 9600bps.

The Model 430 is priced at $807.50 per carton and the Model 431 at $902.50. Deliveries are made "off the shelf."

For additional information contact MICOM SYSTEMS, INC., 20151 Nordhoff St., Chatsworth, CA 91311, (213) 998-8844, TWX 910/494-4910. Come to Booth 801 DEXPO East.

LOGICRAFT ANNOUNCES INTEL 8086 SUPPORT

Logicraft, Inc., exhibitors at DEXPO East 84, Booth 1808, April 4-6, 1984 will be introducing MS-DOS and CP/M-86 (IBM PC Compatibility) for DEC.

Logicraft announces support for Intel 8086 to their CARDWARE line of co-processor products for DEC LSI-11, PDP-11 and VAX processors. Now any DEC PDP-11 processor can execute MS-DOS or CP/M-86 applications simultaneously with their DEC programs. The QCP-11+ is a dual width board for Q-bus computers with an Intel 8186 microcomputer running at 5MHz, with up to 256K RAM, 4K PROM and a serial channel.

The UCP-11+ is a quad width board for UNIBUS computers with one or three CP/M-80 subsystems and one Intel 8088 microcomputer running at 8MHz, with up to 256K RAM and a serial channel. Both the QCP-11+ and UCP-11+ include all software and licenses to operate either MS-DOS or CP/M-86.

CARDWARE is available for RT-11, TSX-Plus, RSX11M/M-Plus, RSTS/E and VMS operating systems.

For more information on Logicraft's CARDWARE, visit Betty Pietrzyk in Booth 1808 at DEXPO East or call (603) 888-4448.

PLESSEY RECONFIGURES STANDARD MEMORY CAPACITY

Plessey Peripheral Systems Inc. has reconfigured the standard memory capacity for its Series 6200 22-bit LSI-11/23-based computer systems.

The reconfiguratin of Plessey Models 6244 and 6245 from two 256 kilobyte (KB) memory boards to one dual wide 512 KB memory board makes available one additional card slot. This provides the system with two available dual wide slots for additional communications devices or peripheral controllers. Prices of these two systems remain the same — $10,485 for Model 6244 and $12,650 for Model 6245, in single quantities.

Memory capacities of Models 6247 and 6248 have increased to one megabyte (MB) from 768 KB. The memory redefinition of these models provides an additional card slot in the bus. Previous configurations of Models 6247 and 6248 did not include extra slots.

Further information is available from Plessey Peripheral Systems, Inc., 17466 Daimler Ave., P.O. Box 19616, Irvine, CA 92714, (714) 540-9945. Come to Booth 832 DEXPO East.

DISC ANNOUNCES DBL V4.0

DISC, developers of DBL — the portable version of the DIBOL programming language — announces DBL V4.0.

Written in C, DBL V4.0 is now available for MS-DOS, CP/M-86, and UNIX environments, and is source code compatible with DBL or DIBOL-11 code.

DBL V4 supports software virtual memory, multi-key ISAM, multi-dimensional arrays, fixed-point decimal, program "binding," new language statements which include SORT and MERGE, a graphics subroutine library, compile-time constants, executable subroutine libraries and an enhanced symbolic debugger.

DBL V4.0 also incorporates the structured programming constructs of DBL V2.2.

DBL V4 will be featured for the first time at DEXPO East 84 on the Rainbow 100+. DBL V2.2 will be featured on the PRO-350 under RT-11. Visit Booth 604.

Further questions should be directed to Adrienne Webb, Vice President Sales and Marketing at DISC, 3336 Bradshaw Rd., Suite 340, Sacramento, CA 95827, (916) 363-7385, TWX 910/367-3701.

CAMBRIDGE DIGITAL EXPANDS SYSTEM 58

Cambridge Digital Systems has announced that its System 58 line of desktop Q-bus minicomputers is now available in higher performance configurations which incorporate larger capacity Winchesters and a wider variety of high speed, high capacity removable mass storage devices.

System 58 is integrated with the DEC PDP-11 family. The basic configuration features a 5¼-inch 10.4, 20.8 or 41.6 Mbyte Winchester drive, an 8-inch one Mbyte floppy, an 8-inch 10.4 Mbyte cartridge, 256Kb of memory and four serial ports. The backplane is 18- or 22-bit Q-bus. System 58 is packaged in a compact 5¼-inch enclosure.

Operating systems for System 58 include UNIX-based packages, RT-11, TSX-Plus, RSX-11M Plus and RSTS/E.

System 58 is priced from $8,000. (Qty. one)

For more information, write or call Cambridge Digital Systems, P.O. Box 568, 65 Bent St., Cambridge, MA 02139. Call toll-free (800) 343-5504. In Massachusetts (617) 491-2700. Come to Booth 918 DEXPO East.

SPEAKEASY ANNOUNCES SPEAKEASY IV

Speakeasy Computing Corporation (SCC), a Chicago based computer software manufacturer announced the availability of its new advanced version, Speakeasy IV. The product is now operational under VAX/VMS, VM/CMS and MVS/TSO and will be demonstrated at DEXPO East 84, Booth 712.

Speakeasy provides both the novice and data processing professional with a wide variety of capabilities including interactive graphics, statistical analysis, and matrix algebra in a user-friendly environment. In addition, the package is completely documented on line and includes comprehensive help, tutorial and example capabilities. Speakeasy IV offers these same features with increased speed and efficiency, without the restrictive size limitations.

Speakeasy IV may be acquired for an annual lease/maintenance fee of $6500, and is available for a free in-house trial period of two months. For additional information, contact Speakeasy Computing Corporation, 222 West Adams St., Chicago, IL 60606, (312) 346-2745.

TOLAS-SALESTREAM RELEASED FOR DEC VAX COMPUTERS

Transcomm Data Systems, a provider of application software and services for distribution and financial management, has announced the release of TOLAS-SALESTREAM.

SALESTREAM is a new concept for distribution oriented companies that allows them to provide their major business partners (customers, salesmen, vendors, key management personnel, etc.) with direct access to specific computerized business functions.

This concept represents a significant enhancement to Transcomm's primary software product TOLAS, which operates on Digital Equipment Corporation VAX and PDP/11 Computers. TOLAS-SALESTREAM will be demonstrated extensively at DEXPO East 84, Booth 216.

For more information, contact Transcomm Data Systems Incorporated, 1380 Old Freeport Road, Pittsburgh, PA 15238, (412) 963-6770.

HOST INTERFACE UNIT FOR XYPLEX NETWORKING SYSTEM

Xyplex has just expanded the XYPLEX System family of products with the XP-UN32-A, a 32 port version of their UNIBUS Host Interface Unit.

The XYPLEX System converts the terminal subsystem of the VAX from a character at a time to a fully intelligent message processor. It provides the VAX/VMS system with equivalent capabilities of the IBM 3270 terminal family. The XYPLEX Host Interface Unit acts as the IBM 3725 Front-End and the XYPLEX Cluster Controller provides the functions of the IBM 3274 Controller and the character processing capabilities of the IBM 3278 Terminal too.

Complementing the XP-CC8-A, an eight line RS-232 Cluster Controller, and the XP-UN64-A, a 64 port UNIBUS Host Interface Unit, the new unit is excellent for connecting the smaller VAX family processors to the XYPLEX distributed front-end and switching system. The VAX-11/750 and the VAX-11/730 as well as the newly announced VAX-11/725, all normally configured for 32 simultaneous users or less, are systems for which the XP-UN32-A would be used as the interface unit. The XP-UN32-A provides all the interface and performance benefits of the XP-UN64-A at 60 percent of the cost.

For additional information, contact Robert H. Rosenbaum at Xyplex, Inc., 100 Domino Drive, Concord, MA 01742 or call (617) 371-1400, or visit DEXPO East Booth 506.

NEW INGRES FOR VAX/UNIX BOOSTS PERFORMANCE

Relational Technology, Inc. (RTI) has announced INGRES VAX/UNIX version 2.0, an update of its relational database management system (RDBMS), that includes significant performance enhancements and three new application development tools.

Contributing to the improved performance of version 2.0 are enhancements to the I/O subsystem, lock management and internal sort routines. New query-update optimization strategies for QUEL, the INGRES data manipulation language, also have reduced query response times two to tenfold.

INGRES was first developed as a prototype RDBMS at the University of California at Berkeley in the early 1970s, and is currently being used by more than 100 customers at over 250 VAX installations throughout the U.S. and Europe. Major manufacturing and industrial users include: General Electric, Jet Propulsion Labs, Boeing Military Airplane, TRW, Schlumberger, Chevron Research, Texas Instruments, Los Alamos National Labs, Mitel and many others.

Relational Technology, Inc. is headquartered at 2855 Telegraph Ave., Berkeley, CA 94705, (415) 845-1700, and will be at Booth 1007 DEXPO East.

COMPUTER SYSTEMS DEVELOPMENT OFFERS SMC BASIC FOR VAX

Computer Systems Development, Inc. (CSD) announces the availability of SMC BASIC for RSX11-M, POS, and VMS. SMC BASIC is a powerful, user-friendly Business Basic (interpretive like BASIC-Plus) that utilizes RMS-11K file structure. Features include: up to ten indexed files open per user, business math, automatic numeric data type conversion, shareable public programs, data in common, screen neumonics, programmable code generation. SMC BASIC offers the

DEC marketplace an existing base of mature software applications. CSD now offers CMI PROFIT (manufacturing software), SMC IDOL (database management system), and WORD CONCEPT ONE (word processing, mail, library, and spreadsheet software).

Computer Systems Development, Inc. will be at Booth 707 DEXPO East. Or contact them at 140 Mayhew Way, Ste. 700, Pleasant Hill, CA 94523, (415) 930-9932.

INTEGRATED SOLUTIONS DEBUTS UNIX 4.2 BSD ON OPTIMUM SERIES

Integrated Solutions, Inc., has announced that it will be showing for the first time the newest model of its Optimum Series, the Model 5/10. Based on the 68010 chip and running UNIX 4.2 BSD, the Model 5/10 offers the highest performance available from any super micro. The Model 5/10 with its dual bus architecture of LSI-11 bus for external devices and Local Bus for memory accesses, allows the 68010 to run with no wait states in up to 4 MB of physical memory.

Using the same design as the Model 5/00 and the IS-68K board sets, the Model 5/10 comes standard with 768KB of memory, two programmable serial ports, "deskside" packaging and UNIX 4.2 BSD. UNIX 4.2 BSD on the 5/10 fully supports transparent demand paging, a fast file system and the DARPA standard TCP/IP communication protocols.

List price for the Model 5/10 is $6,995.00. OEM discounts are available. UNIX System III and V are also available on the system. Delivery of the system is 30 to 45 days ARO with delivery of board sets 30 days ARO.

More information is available from Integrated Solutions, Inc., 1350 Dell Avenue, Campbell, CA 95008, (408) 374-2441; or at Booth 738 DEXPO East.

COGNOS' POWERHOUSE AT DEXPO EAST

Cognos, a leading supplier of 4th generation software products for minicomputers, is announcing the release of POWERHOUSE, an application development system for VAX, at the DEXPO East show in Boston, April 3-6, 1984.

POWERHOUSE, a dictionary-driven application development system with three processing components — an interactive online transaction processor, a report writer and a batch transaction processor — is already installed on close to 2000 HP 3000 minicomputers in 23 countries.

The VAX version of POWERHOUSE, successfully previewed at DEXPO West in Las Vegas, October 1983, supports RMS files and features new menu-driven dictionary technology. It offers both DP professionals and non-programmers greater ease of use as well as significant time savings in application development and maintenance.

POWERHOUSE is supported from offices located in 13 major cities across the United States and Canada as well as in England and Hong Kong.

For more information during the DEXPO East show, contact Lana Farmery, POWERHOUSE Product Manager, Booth 245. After the show contact Cognos Corporation, 1801 Oakland Blvd., Suite 1000, Walnut Creek, CA 94596, (415) 943-7277.

WORD PROCESSING SOFTWARE FROM COMPU-TOME TIES PDP-11, VAX TO RAINBOW

Known as a developer of the CT*OS family of word processing software for DEC computers, Compu-Tome, Inc., Pasadena, California, has announced a new version of CT*OS which will run under the CP/M-86 operating system. CT*OS/86 is targeted for DEC's Rainbow PC and with it users are able to transfer WP files from VAX and PDP-11 hosts to Rainbows, plus a host of other PCs that use this popular operating system.

CT*OS/86 is functionally equivalent to CT*OS running on larger DEC systems. It is menu driven, and thus easily learned by non-DP personnel. CT*OS/86 provides global search and replace, cut and paste, list processing, a spelling corrector, ASCII file handling, 132 column document width, stored text libraries, right justified margins, scientific character set, and an extensive repertoire of user defined keys to invoke special word processing functions.

CT*OS/86 is available at $950 each for a single user license, including list processing and spelling corrector plus full documentation. Volume discounts are available. Delivery is 30 days ARO.

Compu-Tome, Inc., 234 E. Colorado Blvd., Pasadena, CA 91101, (213) 796-9371, will exhibit in Booth 411 at DEXPO East.

NIS ANNOUNCES VERSION 4 OF VUE PROJECT MANAGEMENT SYSTEM

National Information Systems announces Version 4 of VUE, the computerized project management system. Version 4 features include on-line help messages, redesigned menus, a new command mode, expanded resource capabilities, a new Time Schedule Network, and the incorporation of all processing into a single module.

VUE offers an easy-to-use system for entering and reporting information in order to plan, schedule, and track project activities. The menu-driven approach makes VUE self-teaching, simple enough for productive use the very first day.

The redesigned menus, help messages, and new command mode combine to make VUE even simpler to learn and use. The organization of the menus was changed to improve the logical flow of information. With the help command, the user can now type "help" at any menu level and get a description of available commands and their uses.

VUE runs on VAX, PDP-11, DEC 10/20, HP 3000, Honeywell, and Perkin Elmer computers. VUE can be purchased, leased, accessed through NIS timesharing services, or evaluated for 30 days on your own computer. More information about VUE is available from National Information Systems, 20370 Town Center Lane, Cupertino, CA 90514, (408) 257-7700, or visit DEXPO Booth 430.

EG & H TO DEMONSTRATE FULL LINE AT DEXPO

Evans Griffiths & Hart, Inc. announces it will be demonstrating its full line of system utilities, communications, and on-line data entry packages for the VAX-11 and PDP-11 computers at DEXPO East 84, the National DEC-Compatible Exhibition in Boston, April 3-6, 1984.

FSORT3 and SELECT, EGH's fast record sort and record extractor running under RSTS/E for RMS and non-RMS files, are proven products at over 200 installations. Running stand-alone on a PDP-11/70, FSORT3 typically is up to 75 times as fast in CPU time as the RSTS-11 sort. VSORT and VSELECT are VMS versions for RMS sequential and relative, span and no span files; VSORT is typically three to seven times faster in CPU time than the VMS 3.0 VAX-11 sort, with fewer page faults and less direct and buffered I/O.

Use DIALUP to link your RSTS/E system to any remote system using asynchronous terminal lines. COLINK links two RSTS systems together using DMC11s or DMR-11s.

KDSS is a complete multi-terminal key-to-disk data entry subsystem that runs under RSTS/E, RSX-11M and VMS.

ROSS/V provides a RSTS/E monitor environment running under VMS.

PROFESSIONAL etc

PRODUCT UPDATES

ROSS/V runs concurrently with other VMS processes while handling monitor calls from a number of RSTS/E user programs, so some users may be working under the RSTS/E environment while others are working directly under VMS. For further information, come to Booth #612 at DEXPO East or contact Evans, Griffiths & Hart, Inc., 55 Waltham Street, Lexington, Mass. 02173, (617) 861-0670.

NAG ANNOUNCES MATH-STAT LIBRARIES FOR DEC COMPUTERS

Numerical Algorithms Group announces the availability of Mark 10 of the NAG FORTRAN Library of mathematical and statistical subroutines for the users of PDP-11, VAX-11 and DEC 10/20 systems under all major FORTRAN compilers and operating systems including UNIX. Together with graphic and interactive documentation supplements, the NAG FORTRAN Library provides the developer of scientific, engineering and business applications programs with a comprehensive programming environment.

In addition, the availability of the new NAG FORTRAN PC50 Library for personal computers brings to personal computers over a decade of NAG's experience in developing quality of numerical software for computers ranging from minicomputers to supercomputers. For further information, contact Numerical Algorithms Group, 1131 Warren Avenue, Downers Grove, IL 60515, (312) 971-2337, or stop by Booth 714.

DSD CORPORATION OFFERS C-CALC VERSION 1.5

DSD Corporation is proud to announce version 1.5 of C-CALC, the multi-dimensional forecasting, planning and data analysis modeling package. Written in the extremely CPU-efficient "C" programming language, C-CALC has been engineered for optimum machine efficiency and ease of use.

C-CALC is menu-driven with an interactive Help facility and On-line Training Procedure. Menus and single-level English-like editor commands allow the user with little or no experience to "jump right in." For the advanced user, a variety of optional shortcuts have been included which allow bypassing menus and prompts if desired.

C-CALC's sophisticated data importing/exporting facility and worksheet consolidation capabilities make it ideal for computer networks and workstations. Entire worksheets or raw data can easily be transferred between different computer systems. In addition, worksheet data is also transportable between C-CALC and most other applications programs (graphics, word processing systems, general ledger, etc.).

C-CALC is covered by a full year of cost free maintenance along with comprehensive customer support. C-CALC will be on display at DEXPO East at Booth 329. For more information please contact: DSD Corporation, 10420 NE 37th Circle, Suite A, Kirkland, WA 98033, (206) 822-2252.

HOS TO UNVEIL NEW USE IT VERSION IN ITS OWN BACKYARD AT DEXPO EAST

Higher Order Software, Inc. (HOS) welcomes the DEC-compatible industry to Boston for the DEXPO East 84 show with the introduction of the latest enhancements to USE.IT, the world's first automated software engineering tool.

USE.IT's new capabilities will be showcased at Booth 338 at the event which runs from April 3 through 6 at the Bayside Exposition Center in Boston. USE.IT Version 2.2 for DEC VAX will be demonstrated by HOS technical and marketing representatives showing attendees how to automatically create bug-free systems.

HOS has wrapped further functional and performance gains around USE.IT for this latest version. Interfaces to Digital's file and screen handling facilities, an expanded library of predefined operations, automatic code generation in COBOL (as well as FORTRAN and PASCAL), and substantial performance increases combine to make USE.IT Version 2.2 the most important productivity tool for system developers. In addition, USE.IT Version 2.2 is fully compatible with Digital's VT-125 and VT-240 terminals. For more information contact John F. Burton, vice president, Marketing & Sales, Higher Order Software, Inc., 2067 Massachusetts Avenue, Cambridge, MA 02140, (617) 661-8900, or stop by our booth.

TWO NEW COMMUNICATIONS DEVICES SUPPORTED BY HASP+ FROM DATANEX

Datanex, Inc., producer of communications software products for Digital Equipment Corporation's PDP-11 and VAX Series computers, has announced support for the DEC DMF32 and DPV11 products through its HASP+ software.

HASP+ is a general purpose workstation software package which allows PDP-11 and VAX computers to communicate with a broad range of different computers and networks. The HASP+ package is oriented toward high-speed bulk transfer of data between computers.

Support for the DMF32 brings VAX users a low-priced and low-overhead solution at communications speeds from 2Kbps to 56Kbps. The DMF32 also includes low overhead Direct Memory Access (DMA) transfers for a line printer and eight asynchronous lines.

The HASP+ package, including support for the DEC DMF32 on VAX systems is priced at $5,500 for single systems purchase. PDP-11 prices range from $1,995 to $5,000 depending upon the operating system. Complete details are available from Datanex, Inc., P.O. Box 1728, Eugene, OR 97440, (503) 687-2520. Visit us at DEXPO East, Booth 119.

HAMILTON RENTALS TO EXHIBIT VOX AND CADD AT DEXPO EAST

VOX — VOX Office Exchange — is a comprehensive interactive office automation package for the DEC world. It features data base management, word processing, spreadsheet business graphics and electronic mail, all of which have complete file transfer capabilities from one to another.

CADD — Computer Aided Design and Drafting Package is a complete hardware and software package for as little as $25,000. It can be configured to run stand alone on PRO 350s or multi-user on VAX & PDP-11s.

To learn more about VOX and CADD visit us at Booth 642, DEXPO East 84, or contact Hamilton Rentals, 6 Pearl Court, Allendale, NJ 07401, (800) 631-0298.

SPSS-X RELEASE 2.0 NOW AVAILABLE ON DEC VAX AND DECSYSTEM 20

SPSS INC., leading developers of information analysis software, announce the newest enhancements to SPSS-X. SPSS-X organizes, summarizes and displays information for end-users in business, government, and education. Release 2.0 of SPSS-X on the DEC VAX/VMS and DECSYSTEM 20 includes these new features:

• Transport system files across machines with import/export facility; create and read portable SPSS-X system files that will transport data and data dictionary information across SPSS-X installations—allows full sharing of data on tapes.

• Enhanced plots to map out data relationships with scatter plots, contour plots, and overlay scatterplots, with complete command over plot size, scaling, and labelling.

• New statistical procedures include quick cluster to group cases for analysis; PROBIT/LOGIT for analysis of dichotomous variables (such as "buy" vs. "no buy" in a marketing research survey; HILOG, a hierarchical log linear model to calculate parameters; and LISREL VI for linear structural relation models.

Visit SPSS Inc. at Booth 609/611 at the DEXPO East show in Boston, and see SPSS-X on display, in addition to SPS/Pro, the new micro-sized version of SPSS for the DEC Professional 350; or contact SPSS, Inc., 444 N. Michigan Avenue, Chicago, IL 60611, (312) 329-2400.

DEC COMPATIBLE POWER CONTROLLERS FROM PULIZZI ENGINEERING

Pulizzi Engineering, Inc. will be displaying and demonstrating their "Z-LINE" and "Z-LINE" DEC compatible Power Controllers at the DEXPO East 84.

PROFESSIONAL etc

PRODUCT UPDATES

The DEC compatible "Z-LINE" Power Controllers provide EMI and RFI filtering with triple noise protection, common mode, differential mode, plus high-frequency ground isolation including multi-stage spike, noise and surge protection.

"Z-LINE" Power Controllers incorporate a new and unique cable management feature for system-friendly integration.

"Z-LINE" protects your system against line noise, data loss and distortion. European, Far East and Custom models are available. We look forward to seeing you at our Booth, Number 607. Or contact Pulizzi Engineering, Inc. at 3260 S. Susan St., Santa Ana, CA 92704, (714) 540-4229.

GRAPHIC OUTLOOK IS SPREADSHEET WITH GRAPHICS FOR VAX/VMS

GRAPHIC OUTLOOK from Stone Mountain Computing is a combination electronic spreadsheet and graphics package for the VAX/VMS operating environment. The package can be purchased with interfaces to DISSPLA, DI/3000, SAS, MGSP graphics systems or with its own embedded graphics software.

GRAPHIC OUTLOOK users are able to prepare high-resolution black and white and color plots of spreadsheet data. Vertical and horizontal bar graphs, pie charts and line drawings can be produced on most graphics terminals and plotters using the software. A low-resolution graphics feature draws bar charts and line drawings on VT100-type terminals.

The spreadsheet program without high-resolution graphics costs $2950. With an interface to DISSPLA, DI/3000, SAS or MGSP the program costs $3950. The spreadsheet program complete with embedded graphics costs $4950. Visitors to DEXPO East 84 can see a demonstration of GRAPHIC OUTLOOK at Booth 742. Or contact Stone Mountain Computing Corporation, 1096 Cambridge Drive, Santa Barbara, CA 93111, (805) 964-9101.

ADA LANGUAGE SYSTEM ON DISPLAY FROM SOFTECH

SofTech, a major supplier of custom software and engineering services to the Federal Government and Fortune 500 companies, will be featuring the VAX/VMS hosted Ada Language System (ALS) software development environment at DEXPO East.

This evolving, transportable, software toolset and related training material will be serving as the standard for the DOD services and can be utilized by anyone whenever an organized software "manufacturing" environment is needed to cost-effectively build and maintain complex software systems. Further information is available from SofTech, 460 Totten Pond Road, Waltham, MA, (800) 225-8854. Please visit DEXPO Booth 102.

IMSL LIBRARY NOW AVAILABLE FOR VAX SYSTEMS RUNNING UNIX

IMSL, Inc., an international firm specializing in technical and scientific software, has announced that the widely-used IMSL library package is now compatible with VAX systems running the UNIX operating system.

The IMSL Library contains 540 FORTRAN subroutines, applicable to a broad range of mathematical and statistical functions. The Library is designed to reduce the time and cost involved in developing scientific and engineering application programs by allowing the programmer to select subroutines from this comprehensive library, rather than writing them.

The IMSL Library is now used by government, industry and higher education, at over 2,000 installations in 50 countries. IMSL markets its software products on an economical annual subscription basis, and offers a substantial discount to degree granting universities.

IMSL is based at 7500 Bellaire Boulevard, Houston, TX 77036, U.S.A. The company invites toll-free telephone inquiries at (800) 222-IMSL. Texas residents may call (713) 772-1927. IMSL will be exhibiting at DEXPO East 84 in Booth Number 122.

V3.4.6 FINANCIAL PACKAGE AVAILABLE FROM FASBE GROUP

The FASBE Group, Inc. announced a new release of its financial applications package, IBMS (Interactive Business Management System), for use on Digital Equipment Corporation VAX-11, DECsystem-10, and DECSYSTEM-20 computers. The new release (Version 3.4.6) incorporates major additions to the General Ledger and Accounts Payable subsystems, has improved throughput performance of the entire system including Purchase Order Entry, Inventory, Accounts Receivable, Budget and Security subsystems and now enables the user to tune the system to optimize its performance relative to the number of users.

The Accounts Payable system was expanded and improved to include multiple bank, recurring payment, and one-time vendor processing. Additionally, automatic creation of check reconciliation tapes for banks has been provided.

The Security module has been enhanced to provide more sophisticated security features, including encryption techniques for better internal control of a company's financial data.

A new report writer using English-like statements for non-technical users was also added in the new release.

IBMS is available for between $5,000 and $20,000, depending on the subsystem selected, from the FASBE Group Inc., P O Box G, Nashua, NH 03061. See us at Booth DEXPO East 84.

NEWMAN TO EXHIBIT AT DEXPO EAST

Newman Computer Exchange, Ann Arbor, Michigan, will be an exhibitor at DEXPO East 84 (the fifth National DEC-Compatible Exhibition, Bayside Exposition Center, Boston, April 3-6, 1984). The multi-million dollar firm is the nation's largest dealer in new and used DEC and Data General systems processors and peripherals, including an extensive stock of PDP8 equipment. A charter DEXPO exhibitor, Newman markets late-model minicomputer equipment, by direct mail and telephone, to major corporations, universities, and government military agencies. Qualified personnel will staff the Newman Booth #1201 to provide equipment appraisal and other firsthand information. Also available: catalogs, literature and free signup for mailing cycle, as well as the Newman "Blue Book" on converting surplus minicomputer equipment to cash. These materials may also be requested direct from Newman Computer Exchange, P.O. Box 8610, Dept. T44E-DX, Ann Arbor, MI 48107, (313) 994-3200.

PDP 8/11/VAX 750 REPAIR, SALES, SERVICE FROM NEW YORK REPAIR DEPOT

New York Repair Depot, Inc. is a computer company that sells and services computer boards, power supplies, cash register systems and memories for all manufacturers. We also specialize in the repair and sales of the PDP 8/11/VAX 750. We have a highly skilled technical staff, whose quality workmanship speaks for itself. Two incentives for utilizing our services are our fast turn-around and our competitive pricing. New York Repair Depot, Inc. offers a full warranty on both our repair services and equipment.

We are members of the Digital Dealers Association and the Association of Field Service Managers. Visit us at Booth 708 DEXPO East, or contact New York Repair Depot, Inc., 22 West 23rd Street, New York City, NY 10010, (212) 741-3800.

AVIV ANNOUNCES NEW GENERATION DEC COMPATIBLE TAPE CONTROLLERS

Aviv Corp., a leading manufacturer of high performance disk and tape controllers and subsystems, announces a new generation of DEC TS11 compatible magnetic tape controllers for all LSI-11, PDP-11 and VAX-11 processors using 800, 1600, 3200 and 6250 bpi formatted tape drives with either Pertec or STC interfaces.

The controllers are available in three versions, depending on the type and speed of the drive selected. The first is for use with industry standard (Pertec) streaming and formatted start/stop drives, either front or top-loading. The second is for low-speed STC compatible GCR start/stop drives (50 ips), and the third is for high performance STC compatible GCR start/stop drives (125 ips).

The TFC 825 is a Unibus compatible, single hex-sized board that is fully embedded in an SPC slot, and is software transparent to all DEC and UNIX operating systems. The TFC 825 uses a high speed microprocessor for data buffer management and tape drive control.

The TFC 925 is a Q-Bus compatible single quad card controller, and has the same characteristics as the TFC 825. Additional features found in TFC 925 are 22-bit addressing and block mode DMA transfer found on new DEC computers such as PDP-11/73 and MicroVAX.

Detailed specification data on TFC 825 and TFC 925, as well as Aviv's complete line of tape and disk controllers and subsystems, is available at our DEXPO East 84 Booth No. 1308, or from Aviv Corp., 26 Cummings Park, Woburn, MA 01801, (617) 933-1165.

PROFESSIONAL ^ etc

PEOPLE, PLACES, THINGS

DEC USERS CONFERENCE ANNOUNCES PC SESSIONS

IRUS, a national independent DEC users group, has announced its Spring conference profiles for personal computer users. The sessions will be presented in conjunction with the DEXPO East 84 DEC-Compatible Industry Exhibition, April 3-6 at the Bayside Exposition Center, Boston. The theme of the conference is DECade of DECisions.

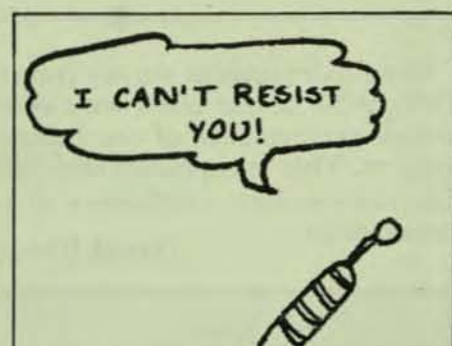
The microcomputer session profiles are:

- Computers in Education — this topic will concentrate on the changes evolving in computer science courses at all educational levels.
- Personal Computers #1 — an overview of the Rainbows, DECmate II and Professional 300 Series and their relative capabilities and advantages. The session will be tutorial.
- Personal Computers #2 — an exploration of the personal computing phenomenon from the perspective of the business professional.
- Personal Computer Workshop #1 — a hands-on work session and demonstration of the Professional 300 Series. This workshop will have an operating system orientation and application focus.
- Personal Computer Workshop #2 — a hands-on work session on advanced software.
- Personal Computer Workshop #3 — a hands-on session and demonstration of the Rainbow 100.
- Implementing a Computer-Aided Design System on a VAX and a PC 350.

For more information call Bob Wittig, conference coordinator, at (401) 738-4430.

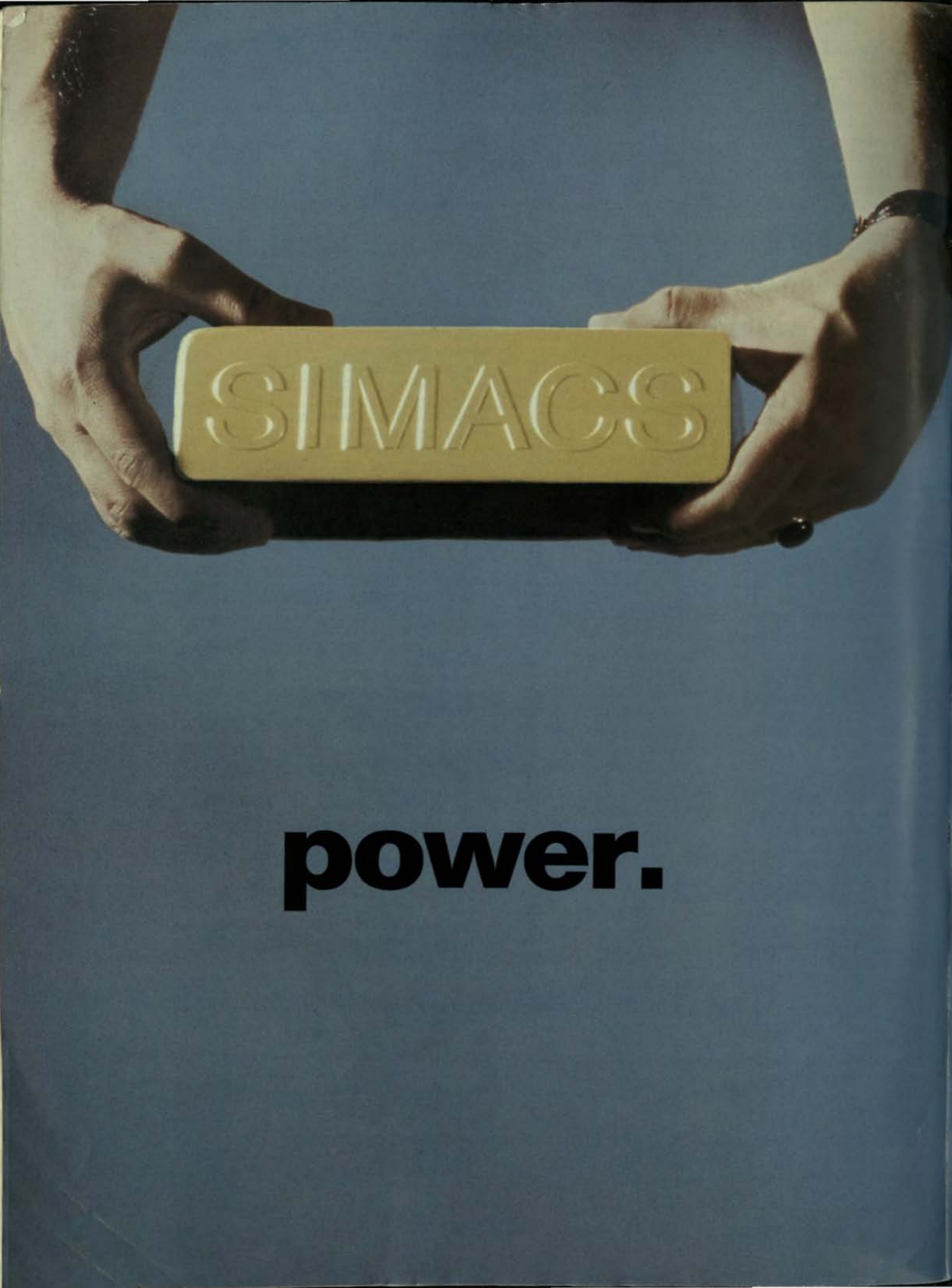
Send Classified Ads to: Classified News, c/o THE DEC PROFESSIONAL, P.O. Box 362, Ambler, PA 19002-0362.

Classified ads are priced at $1.00 per word. Display ads are $75.00 per column inch. If we set, this includes border and 2 lines in bolder and/or larger type size, if desired, — please specify.



I CAN'T RESIST YOU!

LIST OF ADVERTISERS


SIMACS
power.