DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER

November Number 25 1977

Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator
12 Bit SIG
c/o DECUS
146 Main Street
Maynard, MA 01754

..or..

Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

DECUS/Europe contributions are solicited through:

Lars Palmer
DECUS/Europe 12 Bit SIG Newsletter Liaison
Hassle
Fack
S-431 20 MOLNDAL 1
SWEDEN

(Please include reference to Newsletter number and page when inquiring about material published.)

NEWSLETTER DEADLINE

The deadline for ready-to-use material for the next Newsletter is 30-December-1977. Material requiring editing/re-typing must be in earlier. Ready-to-use material should use an area 6 1/2 inches (16.5 cm) wide by no more than 9 inches (23 cm) long on each page. It should be single spaced on white bond paper whenever possible and must be reasonably clean, legible and sufficiently dark for good photographic reproduction.

OS/8 VERSION 3D (ETC.) INFORMATION

I have been receiving many complaints that ordering information for the new updates to the OS/8 software components is still (i.e. November 7, 1977) very hard or impossible to find. Due to the fact that the OS/8 DATE stops working on January 1, 1978, it is very important for everyone to get the updates as soon as possible. I personaly have been unable to extract the needed information from my salesman who says he is very busy but will try to get around to it if he gets a chance!

I have been told that a notice appeared in a recent issue of the PDP-8 Digital Software News but for some reason the last issue I received was back in April! I called Maynard to try to sort that problem out but no one had any idea what was going on or why I was not getting the DSN. The people I talked to promised to fix the problem and send me back issues but it has not happened as yet so that approach to getting the

update information also failed in my case. I was recently told that many people are having trouble with the DSN and cannot get it without some sort of intervention at the local or district level!

After trying several other approaches in an effort to get some information in time for this Newsletter, I finally reached Dave Rogers in the PDP-8 product line who was surprised that anyone was having a problem. He did note that he had the impression the updates were only just now (first week in November) actually becoming available from the Software Distribution Center.

Dave was good enough to give me the following information over the phone. It would be a good idea to double check it before placing an order.

OS/8 Binary Kit		
First time	QF015-Ax	$ 440
Update	QF015-Hx	$ 175
OS/8 Source Kit		
First time	QF015-Ey	$1050
Update	QF015-Ny	$ 350
OS/8 Extensions Kit (Binary) (BASIC, BATCH, TECO)		
First time	QF006-Ax	$ 220
Update	QF006-Hx	$ 200
OS/8 Extensions Kit (Source)		
First time	QF006-Ey	$1050
Update	QF006-Ny	$ 350
OS/8 Fortran IV (Binary)		
First time	QF008-Ax	$ 770
Update	QF008-Hx	$ 200
OS/8 Fortran IV (Source)		
First time	QF008-Ey	$1100
Update	QF008-Ny	$ 350

Where the media codes (x or y) are as follows:

x = A,B,C,N, or Y (i.e. for binary kits)
y = A,C,E or Y (i.e. source kits)

A = LINCtape
B = Paper Tape
C = DECtape
E = DECdisk (i.e. RK05)
N = Cassette
Y = Floppy Disk

Many of us will find January 1 rolling around and we still will not have the updates. As an emergency stop gap measure I plan to do the following:

1) I will subtract 8 from the year (i.e. 1978 becomes 1970) so that the system will accept the date (.DATE 1/1/70).

2) This will result in the low order 3 bits of the year in the system date word being correct but the new extended year bits will not be set in the system head area.

3) As a result, the year will print out as 1970 instead of 1978 in directories and elsewere. More importantly however, the year bits saved in directories will be correct and compatible with the new date algorithm so that when the updated software arrives everything will work as it should.

4) The day-of-the-week from the DATE command will be wrong (usually dates in 1978 fall on a different day of the week than the same dates in 1970) so I will just ignore that printout.

5) If I use some of the user written software (such as the latest version of "DIRECT.05") which has already implemented the new date algorithm, I will find a way (i.e. ODT or a little program) to force the extended year bits to be correct in the system area. These bits were documented in a previous Newsletter.

6) The best thing to do would be to fix up a version of the DATE command to go into CCL that will accept the real date and set both the basic date word and the extended year bits and to use DIRECT.05 in place of DEC's DIRECT.

7) When accessing the system date with functions in the various languages I will get 1970 rather than 1978 so I will have to ignore it or make a temporary compensation in the program until new versions of the software become available.

8) If I had access to a legal staff I would ask them to investigate the propriety of and possible relief from DEC's prices, policies and actions with respect to this "date problem" which they can be shown to have been fully aware of since 1970 (through at least five releases of the system prior to OS/8 V3D).

ADVANCE INFORMATION FROM RON JANSEN

I recently spoke to Ron Jansen about some items of concern to our membership. In response, he wrote the following articles for the PDP-8 Digital Software News and forwarded advanced copies for the Newsletter. Incidently I noticed that Ron's PDP-8 software development group at DEC has been advertising recently for experienced PDP-8 programers. Hopfully this portends increased software development in the 12 bit world.

"OS/8 DATE ALGORITHM

The OS/8 DATE algorithm has been changed so that OS/8 files can be dated with any date up to December 31, 1999.

The modifications are included in OS/8 V3D which is now shipping from SDC. The change has the following effects:

(1) You must enter the date in the following format: DAY-MONTH-YEAR. For example - 11-Oct-77. The old format is now illegal.

(2) The date word on the media directories has not changed. When you list a directory, no dates will be given unless you have entered the DATE. If you have entered a date, the algorithm assumes that the file was created either in the current year or in the past 7 years.

(3) The new algorithm only allows a span of 8 years. Therefore, 1970 dates will cease to be supported in 1978. Similarly, one year of date support will cease each suceeding year."

"MACREL/LINKER HARDWARE RESTRICTIONS

The SPD for MACREL/LINKER excludes support for pre-omnibus machines. This does not mean that it does run on these systems, only that DEC does not guarantee it. The program has been tested on a PDP-12 and on a PDP-8I successfully. Problems with MACREL/LINKER on pre-omnibus machines will be investigated by the supporting organization and will be fixed if they are not hardware related."

"OS/78 FUNCTIONALITY

The most frequently asked question about OS/78 is 'how does it relate to OS/8?'. The answer is that OS/78, V1 is a true subset of OS/8, V3D. It is, essentially, OS/8 configured for a system with floppy disks, a scope, and a line printer. The CUSP's included are those DEC thought appropriate for the DECstation 78 market.

Anyone who wishes to use a DECstation 78, but finds OS/78 too restrictive, can use OS/8 V3D on floppies.

The DECstation 78 (from an OS/8 standpoint) is just a 16k PDP-8A with floppy disks, a VT-52 scope, and an optional line printer. The 78 is slower than an 8A and there are some other minor differences, but OS/8 will run as is off floppy disks."

DECUS LIBRARY NEWS

As most of you know, cards have recently been sent out to everyone on the mailing lists for the catalogs for the DECUS 12 bit libraries. Only those returning cards and requesting a copy of the new catalog will recieve one. This is being done to cut down on the very high cost for the 12 bit library catalogs. In the past very few names were ever removed from the lists so we expect that DECUS is mailing a rather large number of wasted catalogs to people who no longer need them.

Chuck Conley called a few days ago with some statistics he had collected from the first 300 cards to be returned. They give some interesting insights.

31% are interested in ordering programs on floppy disks
37% are interested in packages of programs
29% are interested in packages of programs on RK05 disks.

79% are using OS/8 (and "OS/12")
8% are using DIAL
4% are using EDU50
11.5% are using RTS-8
5% made an unsolicited note that they were using COS 300
16% are using other systems (in particular paper tape, LAP6W, and homegrown systems)

59% are using BASIC
50% are using PAL8
47% are using FOCAL (!!!)
38% are using OS/8 FORTRAN IV
34% are using PAL3 and MACRO
30% are using OS/8 FORTRAN "II"
The same 5% as above are using DIBOL of course

LONGER DECTAPES

As you may have noticed in past Newsletters, there seems to be a never ending quest for ways to get more data on DECtapes. In that connection I noticed an obscure item in a brocure from Computer Operations, Inc. recently. They refer to 400 foot "LINC tape"s (as opposed to the standard 260 foot DECtape or 150 foot LAP6 style LINCtape). In PDP-11 format they claim to get 1024 blocks of 512 16 bit words on the 400 foot tape compared to 656 blocks on a standard 260 foot tape or 400 blocks on a 150 foot tape.

I contacted the company to try to find out more about this breakthrough but found it hard to get really clear, definitive answers. It seems as though they are respooling tape that is not of the special "sandwich" construction in standard DECtape. This probably explains how they get more tape on the reel. The impression I got was that the tape would not be good for hard duty as a result of the way it is made (i.e. it would wear out quickly) but that for data storage with infrequent access it might be OK. I got conflicting stories on whether they are really going to offer the long tapes seriously. One report was that they had tried the tapes and decided not to sell them but after that I got the opposite response plus a new brocure that listed the tape (not cheap by the way!).

The company's address is 9700-B Palmer Highway, Lanham, Maryland 20801 USA.

RAW SPR's

Lars Palmer sent copies of the following SPR's which are for OS/8 V3C.

PIP - 1) If a PIP transfer in ASCII mode involves a file that reaches the end of a device, PIP reports an input error. The file copy can be done correctly with TECO or FOTP.

PIP - 2) A transfer with PIP under the syntax

FILE<FILE(TD)

can (not illogically) go wrong if the file expands. This is really a user error but should be documented.

FRTS - There appears to be an error in the FRTS emulation of the FPP. Apparently when FRTS does an FLDA in STARTD mode the emulator version clears the high order bits of the FAC. The FPP-12 hardware version does not do this.

THE EUROPEAN DECUS SYMPOSIUM

The 12 Bit SIG has been very active this year to try to raise our part of the meeting and have achieved several things. The meeting started as always with training seminaries and this year for the first time two training seminaries, one on TECO and one on advanced FORTRAN 4 were given by DECUS. Teachers were PDP 8 users with a lot of experience. The seminars were given at a much cheaper rate than the DEC-training seminaries and were well attended. All together about 25 persons took part in the two training seminaries.

The actual symposium consisted of the traditional parts: papers by users, presentations and discussions on DEC-material and DEC- and user-demonstrations. Let me say a few words on each of these. The 12 Bit users had a whole day available for their presentations. There were four user-presentations of which the presentation by Szabo from Hungary on "some enhancements of RTS/8" really took the day. He showed how, with minor work, it is possible to run BASIC in the foreground of RTS 8 and demonstrated clearly ideas that would

probably make it possible to run a whole OS 8-system in the foreground. The DEC-presentations were made by Gary Cole, presenting the DEC-station 78 and the KT 8 A memory management and Ron Jensen, presenting the Macrel assembler. All this material has been thoroughly reviewed in previous newsletters and there is no need to go into details. The presentations by Gary Cole were very well acclaimed. However, due to a misunderstanding between the users and the product line, Ron Jensen's presentation came to be aimed at another audiens than what was present. In Europe very few persons have heard anything in detail about Macrel and even less persons had ever seen a copy of it before this symposium. There was a PDP 8 with 16 K and DEC-tape available and DEC-station 78. There was a lively interchange of programs and a concentrated effort of several of the DECUS-members to interface the DEC-station 78 as a terminal to the PDP 8 which, however, failed, probably due to some hardware trouble in the 78. However, the 78 ran by itself and the demonstration of OS 78 was quite good. Gary Cole was presented with his first SPR on the DEC-station 78 during this symposium (in OS 78 the R and RUN commands are normally inhibited. The system is only meant to be run via CCL-commands. This situation is of course catastrophic if the CCL-command is typed in, in which case nothing can be done with the system). Both the PDP 8 timesharing systems, ETOS and MULTI-8 were demonstrated. The MULTI-8 demonstration was also accompanied by a demonstration of the memory management unit produced in Holland and described in an earlier newsletter. With this memory management installed the timesharing users, when there is no competition for core, run at a speed which is virtually the same as the run in a standard stand alone-machine. At the conclusion of the symposium the general feeling was that the symposium, at least for the 12 Bit users, was one of the best so far, very much due to the fact that we have achieved a reasonably good contact with the PDP 8 product line in Maynard.

We have heard Gary Cole and other people from the product line at the symposium and have had a chance to talk to them several times, and Lars Palmer was in Maynard this summer and talked to the product line and the old feeling that nothing can be substituted for personal contacts very much goes through in the data's business. A final conclusion was that a concentrated effort will be made to include commercial and OEM users in the next symposium to be held in Copenhagen in September, 1978.

P.S. A little outside the 12 Bit activities, but very rewarding for the 08 users, was the fact that Richard Larry was at this symposium. He was there to discuss the micro-coding for the PDP 11/60 but he also found time to sit down and discuss many of the intricacies of OS 8 with its users - a very exciting experience.

Lars Palmer
1977-09-22

TO: OS8 Editor

FROM: Pat Caroom, DECUS Standards Coordinator

RE: Spring Symposia Standards Planning and Standards Coordinator Election

A new Standards Coordinator for DECUS will be elected this spring. This is a position on the DECUS U. S. Board of Directors with the specific duty of coordinating standards activities within DECUS and a broader role of policymaking for the entire DECUS U. S.

Anyone interested in this position or as Standards Coordinator for the LCG or Mini/Midi Group should contact me as soon as possible to help in planning the Spring Symposia. DECUS Standards activities include Digital Standards for products across main frames and National Standards which w ry to mpact before they are finalized and Digital is required to comp with them.

This has been a fascinating and educating two years tor me. I hope you will take this opportunity to learn about a very important facit of our industry.

DECUS Standards Report
October 12, 1977

The fall meeting of ANSI X3 was held in Washington, D. C. on October 11. I met with Barbara Ham, LCG Standards laison, on October 10 to discuss X3 positions and finalize plans for the Fall Symposia Standards Sessions. Ham also attended the X3 meeting.

The Fall X3 meeting found DECUS and Digital in conflicting votes on three major standards. I would like to emphasize that this is a point of information and, to the best of my knowledge, not a problem. The standards involved are Fortran, I/O Interface and Power Control Interface. DECUS voted in favor of each; Digital cast a negative vote.

The Fortran Standard is up for a final 30 day "reaction period." There were two negative votes on the final ballot before X3; one belonging to Digital. Digital's objection (attached) is to the zero trip Do Loop. X3J3 - the technical committee charged with the responsibility of the Fortran Standard - has listened to arguments pro and con on this issue for three years. It's been a debate and their decision was to go with zero trip. I announced to our membership that I would vote for the proposed Fortran Standard unless I received a strong argument to do otherwise. I received no such argument nor did Pat White (Digital). However, DEC's argument is based on zero trip being a bad thing for the "user." The 30 day "reaction period" was given to X3 members to review the negative ballots and responses from X3J3. Before the end of the 30 day period, X3 members may change their ballot.

I feel we are badly in need of this standards and can definitely say that X3J3 is not going to change the zero trip to single trip. I anticipate DEC will change their vote. The Fortran Standard will be proposed as an International Standard at a November ISO meeting.

The I/O and Power Control Interface Standard are going out for final X3 ballot. Briefly, these standards are to standardize peripheral device interfaces on "large-medium" systems. During the Public Comment Period I voted against these standards. At both Spring Symposia I announced and discussed our vote. Gordon Bell (Digital Engineering VP) was present for one of the meetings. Bell is strongly opposed to these standards. He feels engineering money that could go to new developments would have to be spent on implanting the new standards which are "old" technology.

The user's perspective, which is shared by the Federal Government, is that the standards will yield cheaper peripherals. We voted at the conclusion of the discussion to change our vote to favor the standards. I expect DEC will oppose these standards on final vote.

DECUS Standards Report
page 2

The Standards and Planning Requirements Committee (X3/SPARC) has finally released their "final" report on Data Base. I anticipate this study group will be renewed and will be open for membership. The Study Group Report will go to Copy n Mail and the RSTS and RSX Newsletter Editors.

Digital and DECUS have voted for the new minimal BASIC. It is being forwarded to ANSI for processing as an American National Standard. The one negative ballot was due to the exclusion of editing capabilities from the Standard. X3 has requested X3J2 to add Editing to its Level 1 announcements of BASIC.

Senate Bill 825 which proposes taking Voluntary Standardization away from ANSI and making it an arm of the Federal Government with mandatory compliance will probably not come up this year. SB325 is being rewritten for resubmittal next year. The result for ANSI has been some healthy reevaluation of how to make the process work in a more timely maner.

I think without exception all of X3 is against a government take-over of the standardization process primarily because it is felt that the technological arguments which are essential, though time consuming, may be cut short.

SPARC Study Groups will be emphasized at the Fall Symposia. SPARC Study Groups are where the goals and objectives of a standard are determined. There are openings on several SPARC groups which we would yield greatly from.

Respectfully submitted,

Patricia M. Caroom
Standards Coordinator
DECUS

PMC/dt

Having just received your newsletter no 24 I would like to add some remarks on various points in it.

Page 9. I for one appreciate Jims DIRECT greatly. It really improves searching on devices. I definitly think it should be in OS8 but I can punch several copies of it if required.

Page 21. The question of user distribution of modified OS8 components must be solved. As far as I can see it seems resonable that OS8 cusps in standard form (save format for all exept CCL and KL8E that are in source) could be distibuted. Who wants these if he doesn't have the monitor any way ?. The monitor and sources to cusps and all parts of the extensions(TECO, BASIC and FIV)are more difficult but it should be possible to solve somehow.

Page 33. Robert Phelps rutines are in the new version of the DECUS program catalouge under distribution now. Several people have asked me about them so this should be pointed out. Regarding the saving of FIV programs under OS8:

1)If the program is small (doesn't destroy loc 17600 upwards) there is no problem at all. The following sequence works beautifully:

```
.EX NAME/L$
*TEMP.BN</P/9$                Punch binary
.LOAD TEMP.BN/S               N.B. the /S
.SAVE DEV TEMP
.RUN DEV TEMP
```

Also by changing a few locations its possible to chain to the program. We have such a program callable by CCL commands. I do not remember the exact locations but look with ODT at loc 200 of the save module. Also the program halts after executi[illegible] simply find the halt and change it to JMP i .+1;7600
We use this procedure often when debugging FIV programs with PDP8 rutines in them. If the programmer takes care to ensure that loc 10004,5,&6 are free its quite feasable to use ODT with this program.

If the program is more than 8K it is nessicary to ensure that the monitor head in 17600 is intact. This can be done by ensuring that this area is covered by a data area not touched by the program. Its a little tricky but can certainly be done.

greetings to all 12bit users in the USA

Lars Palmer

NOTES FROM IAN TEMPLETON

Some problems and ideas raised both here and in recent newsletters may be worthy of comment.

(1) CCL command on system startup (#20, p. 21)

This should be modified for two-page system handlers: the 1077 should be deposited at location 77 of block 66, not block 0. I have changed the .UC to the .DA command on bootstrapping since my version of BOOT.SV retains the 'current' date, which may or may not need to be modified. The 325 and 303 corresponding to U and C simply have to be changed to 304 and 301.

(2) BAT handler (#23, p. 37)

The example given works only in a 32K system in which BATCH has been allowed to access Field 7 (see Software News, Sept. '76 for BATCH V6: however, BATCH V5 should have 711 changed from 1370 to 7200 instead). In any other system the BATCH monitor and input buffer is overwritten by the FRTS loader, and BAT: then refuses to work. However, it works perfectly well with FORTRAN II or with PIP.

e.g.
```
$JOB BATPIP
.R PIP
*DEV:<BAT:
 THIS IS YOUR FRIENDLY BAT SPEAKING
$
$END
```

will write the message on DEV:.

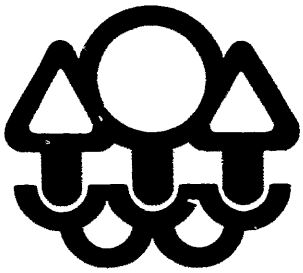
(3) RALF 'ALN' with hardware and software FPP

This problem arose with an early version of Phelps' USR. Dan Smith had patched it to work with the FRTS FPP simulator and it went crazy with our FPP8A. It turns out that the software code to handle ALN for double-word mode shifts retains the last bit shifted right in AC1 and picks this up on a left shift, so that 'shift right 3' followed by 'shift left 3' only strips 2 bits while the FPP8A strips 3. I have discussed this with Dan and have submitted an SPR. (I have enclosed my supporting documentation for Bob Hassinger.) It could be fixed by a DCA AC1 after ALNSHL (FRTS loc 6443) but there's no room on the page.

(4) Disabling the TD8E ROM

A colleague recently upgraded his 8/E to 32K and wanted to disable his TD8E ROM whenever he was using his floppy as SYS: so that he could use all of field 7. I found a simple way to use existing spare hardware plus one resistor and a piece of wire to allow the front panel SW to do the job. Anyone needing this information please contact me at (613) 992-2113.

Ian Templeton
National Research Council of Canada
Ottawa, Canada K1A 0R6



Minnesota Pollution Control Agency

October 28, 1977

Mr. Robert Hassinger, Coordinator
12 - BIT SIG
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

Dear Mr. Hassinger:

I am enclosing a copy of the documentation of a sort-merge package we developed for OS/8. Briefly this package has the following features - it runs in 8K of memory, it will sort ASC II records up to 256 characters long, it handles variable length records, it prints (optionally) sort/merge statistics on either the console device or a user defined ASC II terminal, it handles tab characters corrently, it accepts sort/merge commands from either a formatted input file or in an interactive mode from the console or a user defined ASC II terminal, if commands are entered interactively it will (optionally) produce a formatted file of sort commands acceptable to the program, it contains several options for sorting large files and/or restarting sorts terminated abnoramlly due to lack of disk space, it will sort files in ascending or descending order on up to 8 keys, and the original order of the input file(s) is conserved on equal comparisons so that more keys may be sorted by making multiple sort passes on the file. We use this package extensively for sorting files and have sorted files in excess of 700 disk blocks with it. If people are interested, they can contact me for a copy. If there is sufficient demand I will submit it to DECUS.

I also have a suggestion for a useful "SET" command for OS/8. Having a CRT terminal as well as a teletypwriter we occassionally would like to change console device codes for OS/8. Switching interfaces is not very desirable to me for this purpose, so to accomplish this, I have made some systems with the IOT's changed to the CRT. This process is reasonably slow (especially without OS/8 listings) and one must be careful to not mix systems. I would like to suggest a "SET CONSOLE" command to change CCL, the CD, KBM, ODT and all programs which use device codes 03, 04 in a special, non-standard way to use a different set of device code.

Inclosing, I would like to complement you on the fine 12 - BIT SIG NEWSLETTER. I always find it interesting, informative, and a most valuable source of information and ideas about PDP-8's (e.g. I have included a copy of Ed Steinberger's unsigned comparison notes in my OS/8 Software Support Manual). Keep up the excellent job.

Sincerely,

Bryan Fredrick

BRYAN FREDRICK
Technical Services Section
Division of Air Quality

Saint Francis College

Biddeford, Maine 04005

November 4, 1977

Mr. Robert Hassinger, Coordinator
12 Bit SIG
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, Mass. 07148

HELP.

New academic user of PDP-8A, DS 310 with BASIC compiler needs assistance in developing file structured programs. Anyone in immediate area with this expertise and wishing to assist please contact Michael Denoncour, Assistant Professor, Saint Francis College, 505 Pool Road, Biddeford, Maine 04005. Telephone (207) 282-1515 Ext. 57.

Dear Bob:

Would you include this request for help in your next newsletter. My problem stems from having learned the compiler language on a PDP-8E and now find some difference on the PDP-8A. Thanks,

Michael Denoncour

PURDUE UNIVERSITY DEPARTMENT OF BIOLOGICAL SCIENCES

September 22, 1977

Robert Hassinger, Coordinator
12 Bit SIG
% DECUS
146 Main Street
Maynard, MA 01754

Dear Mr. Hassinger:

Having been involved with a lot of PDP-8 programming the past several years, I have followed your newsletter with interest. The reason I am writing is to inform your readers about some OS/8 programs which have been developed here that increase the power of OS/8 in some novel ways.

In a recent issue, Lars Palmer of Sweden wrote concerning the problem of directory overflow with OS/8. His suggestion, while it would work, seemed to me rather wasteful of disk space and also required changes to PIP. A solution to the directory overflow problem is a natural by-product of our program called PFILES. Program PFILES allows the user to create and maintain "pfiles". "Pfiles" are structured OS/8 files that segregate groups of related files as a single file whenever the separate files are not needed. The commands are natural and varied. PFILES has been both popular and useful at our installation. It requires no changes to any of the CUSPs, excepting the suggested addition of several CCL commands.

Another new program we have is called BATCHM. It allows the user to selectively create batch files from macro structures. Features include conditionals and substitutable parameters formed from the parts of a Command Decoder specification (Special Mode). The batch file formed is automatically submitted to BATCH . BATCHM requires no changes to any OS/8 CUSPs, excepting again the addition of a few CCL commands.

I have enclosed copies of the DECUS abstract and User Documentation for these two programs. The write-ups are rather lengthy, but the abstracts should be interesting to your readers. These two programs are in the process of being submitted to DECUS. In each case, a core image file and a "pfile" containing the source and help file will be submitted.

Separately we are submitting an SRCCOM comparison of our revised CCL vs CCL Version D. It allows 200 plus commands with plenty of room for command subroutine and with no apparent ill effects. It seems to me that SRCCOM comparisons are a fairly satisfactory way of documenting changes to Dec's coprighted sources. Any thoughts from your readers about that?

Sincerely,

Greg A. Gardner

GAG:po

Lilly Hall of Life Sciences
West Lafayette, Indiana 47907

PFILES USER DOCUMENTATION

1. INTRODUCTION

Abstract

Program PFILES is a system CUSP which allows the user to create and maintain pfiles. Pfiles are structured OS/8 files with the extension ".PF". They are used to store groups of related files in a convenient form that is essentially transparent to the user.

Commands available include creating a pfile, storing files in a pfile, loading files from a pfile, deleting or renaming a file inside a pfile, and a number of others. Each command is identified by a distinctive CCL mnemonic. These are described in detail in section 5 of this documentation.

Pfiles may be created on any file-structured device. The internal structure of a pfile is comprised of three parts. The first block of each pfile contains the pfile's directory. In the directory are stored the names, lengths, and creation dates of all files stored in the pfile. Blocks following the directory are used to store the files themselves. At the end of the pfile are a number of blocks (possibly zero) of free space. This is called the IAS, the immediate available storage. Having this free space speeds a number of PFILES functions significantly.

Uses of Pfiles

The most obvious use of pfiles is the storing of related user sources or binaries. In the pfile form, they may be punched as a single tape or transferred from disk to disk as a single file Several pfiles representing different on-going projects may co-exist, without interference, on a disk. Additionally, there are several suggested system pfiles of common interest.

From the user's standpoint, there is pfile space, the storage used by pfiles, and working space, the empty files and non-pfiles. At the start of a work session, the user comes to a "cleared" disk, one containing no non-pfiles excepting CUSPs on the system device. The files needed are loaded from one or more pfiles and the session proceeds normally. At the end of the work session, changed files are stored, if desired, back into pfiles, and the disk is once again cleared of non-pfiles. Simple methods for performing these basic steps are given in section 6.

There are several suggested system pfiles. The first of these is SYS:HELP.PF in which all the help files are stored. CCL command HELP is changed to access this pfile when typing help documentation. Another is SYS:SYSTEM.PF. This is the default pfile accessed. In it are stored infrequently used core images, PAL8 utility package sources, and the like. Another system pfile is SYS:BTMLIB.PF, the default macro library for program BATCHM. BATCHM is described in seperate documentation. Finally, at our installation, the character font definition files for our CRT are stored in pfile SYS:FONTS.PF on each disk.

Restrictions

Since PFILES uses a one block directory in each pfile, a maximum of 41 files may be stored in any single pfile. One way around this is the fact that one or more of the files inside a pfile may be pfiles. Although it is possible that several users may share a disk, each having one or more pfiles for storage, no attempt has been made to add security keys or passwords. Their use would be pointless since a snooper could easily delete a pfile and then gain access to its contents through use of PIP's /I option.

2. REQUIREMENTS

Hardware

PFILES requires a standard OS/8 configuration with at least one mass storage device of medium to large capacity. Program PFILES requires 8K to execute.

Software

PFILES runs under the OS/8 operating system and equivalent operating systems. Core image file PFILES.SV must be present on the system device. Help file PFILES.HL should be in the system help pfile. A number of associated CCL changes, documented in Appendix B, must be made. Finally, since pfiles have the associated file extension ".PF", the user should remove any files that have this extension but aren't pfiles.

BATCHM USER DOCUMENTATION

1. INTRODUCTION

1.1 Vital Statistics

DECUS Files : BATCHM.SV, BATCHM.PF
DECUS Number :
Systems : PDP-8 OS/8 and RTS-8 OS/8
Release Date : September 1, 1977
Version : C1
Author : Greg A. Gardner, Purdue University

1.2 Abstract

This documentation describes OS/8 program BATCHM. This program accepts a user text file containing a BATCHM macro. The marco is expanded forming a batch file which is subsequently submitted to OS/8 CUSP BATCH. The method allows versatile handling of multi-program tasks with such special features as multi-level batch files, non-static batch files, and a proposed new response for CCL to unknown command mnemonics.

BATCHM macros feature substitutable parameters and directive commands. Parameters are set from the input a user gives the Command Decoder using its special mode. These parameters and several scratch pad parameters, also available, may be set and reset during expansion of the macro. Directives provide for conditional expansion, repeat calls to the Command Decoder, resetting of parameters, and many other functions to make BATCHM macros as versatile as possible. Appendix A describes BATCHM parameters and directives in detail.

In several areas, BATCHM is still being developed. For instance, the idea of non-static batch files is still very new, as regards its inclusion in BATCHM's repertoire. Another is the ability of BATCHM macros to directly create data files by preventing the final link to BATCH. Also BATCHM has some close links with program PFILES, which is documented as a separate DECUS entry. Macros and batch files may be submitted to BATCHM from inside pfiles. Also, BATCHM macro can determine the statistics of files residing inside pfiles, as well as those outside pfiles.

BJØRN RUNGE
Gålgebakken Øster 9-14
2620 Albertslund

telf: (02)64 06 75

September 1977.

EXTENDED OVERLAY FOR OS/8 BASIC.

by Bjørn Runge.
MSc in Electrical engeneering.
Ph.D.

OS/8 BASIC has an overlay feature which allows a user to implement up to 16 assembler programmed functions, using a maximum of five pages of core (Ø34ØØ-Ø4577).

Very often these restrictions limit the more advanced use of BASIC. Overlay swapping may be so time consuming (i.e. DEC-TAPE or Floppy disk) or cause such excessive wear (i.e. Floppy disk) as to render BASIC useless. On computers with special devices (i.e. LAB-8) a maximum of 16 user functions is a hopeless limitation. For users needing special functions (i.e. bit manipulation or extended arithmetic precision) five pages of core may be the limiting factor.

Since the BASIC RUN TIME SYSTEM (BRTS) does not allow inclusion of new overlays, a different solution is needed.

The space restriction.

The space restriction may be overcome in two ways:

1. In BASIC.UF is embedded a smaller overlay
2. In BASIC.UF is embedded a core resident part.

Solution 1 is applicable if core is restricted to 8 or 12 k and swapping is not a restricting factor. Two pages (Ø4ØØØ-Ø4377) corresponding to an OS/8 block are used as an internal overlay area. This gives a virtually unlimited overlay size. In the current version the number of overlays is limited to 16.

Solution 2 is applicable when core is no restriction and

swapping is prohibitive. The first time any user function is called, a special program is executed. It is checked whether the core resident part of the overlay (CRO) and the BASIC program are overlapping. If overlap occurs an error message is given, including top of CRO address and bottom of program address and execution is terminated.

If there is no overlapping the CRO is read into core, the block number of BASIC.UF in the BRTS overlay administration is updated to prevent further execution of the intial program, the proper FIELD Ø overlay part is read into core (Ø34ØØ-Ø4577) and the called user function is executed. Further calls of user functions are executed in the normal way. In the current version 12ØØØ-17577 may be used for CRO programming.

To keep swapping at a minimum, the most used functions from the other BASIC overlays (BASIC.AF, BASIC.SF and BASIC.FF) may be copied to the CRO.

A combination of solution 1 and 2 may be implemented as well.

The number of function restriction.

The number of functions may be extended by reserving two of the 16 entries for two generalized functions:

1. USN (O,A,B,C,A$,B$)
2. USS$(O,A,B,C,A$,B$)

USN is a numeric function and USS$ is a string function. Both functions have the maximum number of arguments allowed by BRTS (four numeric and two string).

O is the internal overlay number combined with the internal function number. (In the current version up to 15 functions may be implemented per overlay.
A,B,C are numeric arguments.
A$,B$ are string arguments.

Special routines.

The administration of USN and USS$ including internal overlay swapping is a part of the "core resident" FIELD Ø overlay.

Furthermore a set of often used routines are implemented. They include:

1. ARGNXT, loading the next numeric argument into FAC.
2. GETN , fix the first numeric argument into AC.
3. NXTN , fix the next numeric argument into AC
4. CALPUT, call from any field a BRTS routine ending with JMP I ILOOPL.
5. JMSØ , call from any field any FIELD Ø resident subroutine with one parameter (if applicable). On return data field and AC are as set by the called subroutine.

The virtual numeric file functions PUT and GET (available from DECUS) are implemented in a slightly improved version.

User function entry points.

In order to facilitate overlay programming and debugging, the BRTS entry point links (Ø156Ø-Ø1577) are fixed, moving the actual branching to the overlay itself. This means that changing the overlay program, does not influence the BRTS links. On the other hand each entry point uses three cells in the overlay.

Special functions

A set of user functions has been implemented for various applications:

1. An improved core resident ODT for debugging since OS/8 ODT may fail while debugging programs using the SYS handler.
2. Bit manipulation functions working on FAC.

3. String store and retrieve (similar to PUT and GET).
4. Extended precision integer mathematics (up to 48 digits), storing the numbers in string variables.
5. Printing with full ASCII set (upper and lower case)
6. Cursor manipulation for VDU's.

Applications

The extended overlay technique was utilized in several applications:

1. Administrative system for medical general practitioners. The system includes patient index, medical record keeping, accounting, information retrieval and statistics
2. Production control and accounting system for business and industry
3. Real-time data collection

Developement.

The extended overlay system was developed by the author in Cooperation with

Peter Kjær
General Practitioner
MEDATA I/S
Einarsvej 18
DK - 2800 Lyngby

Birch & Krogboe K/S
Consulting Engineers
Teknikerbyen 34
DK - 2830 Virum

For further information please contact the author

Bjørn Runge

UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON 98195

Department of Biochemistry

October 19, 1977

Bob Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

Dear Bob,

I now have a program (OPTF4) which optimizes the RALF code that is generated by the Fortran IV compiler. It accomplishes the following:

1. Unnecessary FLDA instructions are eliminated,
2. FCLA, FADDM and FMULM instructions are implemented,
3. addition and subtraction of zero are eliminated (this optimizes some IF statements),
4. absolute values are computed in-line,
5. squaring is reduced to a multiply instruction.

Much of the above is also applicable in extended precision mode.

OPTF4 is designed to be included in the set of chained programs that comprise the Fortran IV system. It executes after PASS20 (or PASS3) and before the RALF assembler. Simple patches must be made to PASS20 and PASS3, then optimization is activated by the /O switch (along with the /Q switch):

```
.R F4
*MAIN<MAIN/O/Q        compiles MAIN with optimization
                      and produces MAIN.RL
```

OPTF4 may also be run from the monitor, to optimize a particular file of RALF code. In general, about 10 seconds per 100 Fortran statements are needed. I have never tested the program in a non-FPP environment, but I see no reason for trouble there. I will update to OS/8 V3D if necessary.

I also have a few additions to the Fortran IV library: DSINH, DCOSH, and DTANH double precision hyperbolic functions, a multiplicative congruential random number generator, and the display routines mentioned earlier (March 1977 Newsletter).

I will provide the above to users who send an OS/8 formatted DECtape or LINCtape. Return postage would be appreciated.

One final note, concerning FRTS. On a PDP-12 with FPP, it is occasionally useful to lockout the CPU; namely for long Fortran IV programs that don't have background tasks running. The CPU can be locked out by changing location 15776 in FRTS from 400 to 410. For convenience, I keep a second version of FRTS around for this purpose. I don't know how many users do heavy number crunching on a 12, but we do, and a 25% increase in execution times is valuable.

Sincerely,

Eric Swanson

Eric Swanson

From Dan Smith
Eye Research Institute
20 Staniford Street
Boston, Mass. 02114
617 742-3140

BATCHING PATCH

When in the course of human events it becomes necessary for a manufacturer to release a product (BATCH) containing extensive, internal device-dependent I/O, as part of a system (OS/8) that normally does most of its I/O through device-independent, user-suppliable handlers, a decent respect for users with non-standard devices indicates that they should explain what's going on and declare the patch areas and procedures that are available.

The only OS/8 handlers used by BATCH are SYS: and those co-resident with SYS:. Although "The BATCH input file may be a punched card [or] high-speed paper tape" and "The BATCH output file is a line printer listing," card reader, paper tape, and line printer support are all in the form of routines internal to BATCH. Thus, user-written or -modified LPT:, CDR:, or PTR: handlers will not be used by BATCH, and, incidentally, line printer SET commands (when and if they are released) will have no effect whatsoever on BATCH output. Another interesting point is that BATCH determines whether the line printer is available by issuing various IOT 66n commands and observing the response; so if there is a non-standard device on your system with a device code of 66, there could be some problems.

Line printer device patches

To implement a non-standard device for BATCH output, it is necessary to modify two routines.

LPTTST resides at 01517, is called by a JMS with clear AC, and returns with clear AC, skipping if a line printer is available. LPTTST presently occupies 1517-1545, but locations 1517-1575 are available. Example: if you merely wish to stop BATCH from fiddling around with device code 66,

```
01520/  JMP I 1517
```

will do it.

Routine BOSLPT resides at 06525 (in the SAVE image; it will be relocated to the highest field when it is actually called, and should be designed to run in any field). It presently occupies 06525-06533, but 06525-06563 are available. It must be the functional equivalent for the line printer of the routine:

```
TTYPE,  0
        TLS
        TSF
        JMP .-1
        CLA
        JMS I (CTRLC    /4773
        JMP I TTYPE
```

It will be called repeatedly with the AC containing the ASCII code for a single character to be output, in bits 4-11. Control codes 212, 214, and 215 (line feed, form feed, return), and perhaps others, may be present. CTRLC checks for ctrl-C on the keyboard, and aborts BATCH if present. CTRLC should be called with, and returns with, clear AC.

Other patches

In the interests of a neatly formatted BATCH log, BATCH outputs quite a few extra linefeeds, etc. Depending on your personality and your installation, you may find this useful, legible, and official looking; or you may find it irritating, cutesy-poo and wasteful of trees. At any rate:

When the BATCH log is sent to the teletype, a simulated form feed (four line-feeds) are output before the $JOB line. In any case, a $JOB line is always output to the teletype, followed by two extra line feeds; every $MSG line is output to the teletype, followed by an extra line feed; and an "#END BATCH" message is output. Thus, even if the /Q option is selected, the teletype can be fairly noisy.

```
06452/7200 CLA          suppresses the simulated form feed
07471/7200 CLA          suppresses the extra line feed after $MSG,
                        and the first extra line feed after $JOB
05671/7200 CLA          suppresses the second extra line feed after $JOB
05542/7610 CLA SKP      suppresses the "#END BATCH" message
```

When the BATCH log is output to the line printer, it prints a form feed, spaces down 36 lines, prints the $JOB line four times, and does another form feed.

```
05657/7200 CLA          suppresses the 36-line skip, and
05666/1356 TAD 5756     replaces the second form-feed with a line-feed,
                        positioning the $JOB line at the top of the
                        first page (rather than in the center of a
                        whole page to itself);
05663/7200 CLA
05664/7200 CLA
05665/7200 CLA          will cause the $JOB line to be printed once only.
```

Generally speaking, much of the BATCH code appears to be straightforward and space is not particularly tight.

NATLSCO
National Loss Control Service Corporation

Long Grove, IL 60049 • 312 | 540-2400
TWX 910 | 651-3571

a subsidiary of
Kemper Corporation

October 21, 1977

Mr. Robert Hassinger
Liberty Mutual Research Center
17 Franklin Road
Hopkinton, MA 01748

Dear Bob:

Newsletter No. 24 has prompted me to take dictating machine in hand and make a few comments. The first of these is that I think the appearance and readability of the Newsletters have been very good lately. As a reader, I thank you for the excellent job you are doing.

On page 26 a comment is made about BATCH not using the high end of core and not freeing it when it is done running. I am virtually certain that DEC published a patch for this quite some time ago. We had also encountered the problem when running large FORTRAN IV jobs. After running BATCH the program would not fit into the machine or it would not run under BATCH because BATCH would tie up the top field. The patch fixed this (at least the freeing up part). I'm sorry I can't recall the Software News which contained this, but it was some time ago.

On page 33 you printed a letter from Jim Kleckner, in which he asked about access level software for Tektronix TCS software. You may recall that I gave a talk on this at a DECUS meeting; I have all the necessary software. I called Jim and he is going to send me a tape. Apparently Tektronix is distributing some of my routines, as Jim mentioned that he had gotton two from someone at Tek who I had given the routines to.

In the event anyone is interested, we have two major routines. The first contains two entry points for the basic ADEIN and ADEOUT access level calls. These are the minimum necessary to do graphics. There are also four other entry points contained in another routine; these are used for character and string I/O through TCS. They are only necessary if you want a full implementation of TCS. By the way, they work for any Tektronix terminal, not just a 4010.

If anyone is connecting a Tektronix graphics terminal to their 8, and starts working with GIN mode (enables a cross-hair cursor for inputing screen coordinates) they will find that the system doesn't work too well. To make a very long story very short, it turns out that FRTS can't process the five incoming characters generated when you input coordinates via GIN mode; it looses some of them. The user must select a terminal straping option which allows the input (to the 8) transfer rate to be dropped down to a point at which FRTS can handle the characters; there is an adjustable "pot" for doing this. Once this is taken care of, there is no problem. Its still faster than you can ever type, but it might slow down CPU-CPU transfers through the terminal.

Finally, on page 41 some comments are made about the infamous date problem. I also like the idea of extending the directory block. Besides allowing for the date solution, having more space for a directory sure would be nice. We are continually filling the allotted directory space on our disc units. Since we are trying to use the PARAM facility, this means that the parameter block gets clobbered. Since we are using the version of DIRECT which supports the parameter block, when the block gets clobbered as the directory fills, DIRECT sometimes goes into a loop, making for interesting output. Let's hear it for larger directories!

My activity with the 8 has decreased considerably because we bought a new machine about nine months ago. It is a Harris system with a SLASH 6 CPU. It has a virtual operating system which allows us to run up to four megabytes of programs at the same time in virtual space. We currently have 288 Kbytes of "real" memory. This system also supports concurrent real time, time sharing, and batch. We have two 80 Mb discs on line, which has had the effect of solving any directory problems!

As a final comment, if anyone wants the TCS access routines, they should send me a DECTAPE and I will be glad to make the copies.

Best regards.

Norman R. Dotti, P.E.
Manager
Noise & Vibration Services

NRD/lm

Correlation-, Spectral-Density-, Transfer-Functions and Phase Angle On Two Analogue Stochastical Signals

Klaus Lickteig
Institut für Kerntechnik, Technische Universität Berlin

The dynamic functions like the correlation-, spectral-density-, transfer-functions and phase angle can be calculated from the analogue stochastical signals of a linear process. Such programs or program systems are normally encluded in the software package of 16-bit computers. These special computers (e.g. time-series-analyzer, Fourier-analyzer etc.) work very effective, which high accuracy and a short calculation time.

The above functions can be calculated with the cheaper 12-bit mini-computers (like PDP-8) too, but it takes longer calculation time, and it has not such a good resolution as the programs with 16-bit computers have - and you do not get such program systems in the standard sofware package of 12-bit computers.

For a great number of measuring problems the resolution and calculation speed of 12-bit computers is sufficient.To make use of a correlation program /6/ and a Fast-Fourier-Transform subroutine /5/ a program system was built which calculates the correlation functions on two analogue stochastical signals and the other dynamic functions with the FFT. All functions are listed below.

- x Auto-Correlation-Function
- x Cross-Correlation-Function
- x absolute Power-Spectral-Density-Function
- x squared PSDF
- real part of the PSDF
- imaginary part of the PSDF (theoretical zero)
- x absolute Cross-Spectral-Density-Function
- x squared CSDF
- real part of the CSDF
- imaginary part of the CSDF
- absolute Transfer-Function
- squared Transfer-Function
- Coherence-Function
- Phase-Angle
- Nyquist Plot

"Program System to Analyze Analogue Signals with the LAB-8 System" is published in the DECUS Program Library /3/; the listing is encluded in a German report /4/ which is available in iimited number only. The program system is built for a PDP-8/E (or 8/I) computer, an AXO8 A/D-Converter and 12 k memory (with 8 k memory You can calculate only the "x" functions).

The correlation- and spectral-density functi .s have some special parameters which are limited by the hardware and software of the computer.

The time interval or the sample time Δt between the data points $x(t_j)$ or between the data pairs $x(t_j)$, $y(t_j)$. has the range:

$$\Delta t = 0.1 \text{ msec}, 0.2 \text{ msec}, \ldots, 838860.8 \text{ msec} \approx 14 \text{ min}$$

The sample frequency $1/\Delta t$ has the range:

$$10.0 \text{ kHz} \geq 1/\Delta t \geq 0.00119 \text{ Hz}$$

The number of the ordinates N of a correlation function has to be:

$$8 \leq N \leq 512$$

The maximum time lag is:

$$\tau_{max} = (N-1) \cdot \Delta t$$

The frequency step within a spectral-density function is given by

$$\Delta f_1 = 1/(N \cdot \Delta t)$$

if you analyze the correlation function only for positive time lags; with positive and negative time lags, the correlation functions $R(\tau_j)$ have 2xN ordinates and the frequency step will be

$$\Delta f_2 = 1/(2xN \cdot \Delta t) \quad .$$

The functions in the frequency domain are calculated for the frequencies:

$$f = \omega/2\pi = f_j = j \cdot \Delta f$$

where

$j = 0, 1, \ldots, N/2-1$ if $\Delta f = \Delta f_1$
$R(\tau_j)$ has N ordinates

$j = 0, 1, \ldots, N-1$ if $\Delta f = \Delta f_2$
$R(\tau_j)$ has 2xN ordinates

BALL /1/ and BENDAT /2/ discussed the accuracy of correlation analyses in detail.

The accuracy of the "Program System to Analyze Analogue Signals with the LAB-8 System" is limited by

a) input parameters (Δt, N)
b) hardware-specified conditions (12-bit word etc.)
c) software-specified errors (1-word accuracy).

The maximum theoretical resolutions of all functions are listed in /3,4/; the correlation functions, spectral density functions and phase angle are calculated with accuracy.

The calculation speed is 0,2 msec per ordinate of a correlation function within continuous data taking /6/.

The program system allows an automatic calculation of all the functions

During off-line calculation an analogue magnetic tape with the measurement signals will be controlled automatically.

An output of all possible data and results is possible onto teletype, high speed punch, oscylloscope of the AX08, analogue plotter.

Literature:

/1/ Ball, G.A.: Korrelationsmeßgeräte
VEB Verlag Technik Berlin, (1972)

/2/ Bendat, J.S.; Piersol, A.G.: Random Data: Analysis and Measurement Procedures
John Wiley & Sons, Inc., New York (1971)

/3/ Lickteig, K.: Program System to Analyze Analogue Signals with the LAB-8 System
DECUS Program Library / DECUS-No. 8-826 (1976)

/4/ Lickteig, K.: Programmsystem für 12-bit Prozeßrechner (PDP-8) zur Analyse analoger stochastischer Signale
Report TUBIK-49, Berlin (Juni 1977)

/5/ Rothman, J.E.: FFTS-R - A Fast Fourier Transform Subroutine for Real Valued Functions (AX08 Version).
DECUS-No. 8-143 (1968)

/6/ Rothman, R.: The Auto- and Cross-Correlation Program for the LAB-8
DECUS-No. 8-340 (1970)

PROCESS CONTROL SYSTEMS, INC.
18130 S. Thornapple Lane • New Berlin, Wisconsin 53151 • (414) 782-3945

HARDWARE CONSULTATION
SOFTWARE DESIGN
PROCESS CONTROL SYSTEM DESIGN

October 24, 1977

Mr. Robert Hassinger
12 Bit SIG
c/o DECUS
146 Main Street
Maynard, Mass 01754

Dear Bob:

The Industrial 14 programmable controller and VT14 programming terminal are described in DEC's own words in the attached excerpt from the "1974 International Edition OEM Products and Services Catalog", pp33-36. Perhaps the TPL could provide you with some additional literature and drawings.

The rest of this letter is my opinion of the I-14 as it relates to the marketplace.

When the Industrial 14 came into the market about 1973 it was as good as or better than most controllers available from other manufacturers. However, no substantial improvements or enhancements other than a 211.2 KB interface system were ever offered for the I-14. Consequently, many other controllers eventually surpassed the I-14 in capability. Some features other controllers have today are analog input and output, TTL input and output, arithmetic operations, ability to generate reports about conditions arising in the controller. Some of these features were planned for the I-14, but they never got into the field.

During the winter of 1975-1976 the users of DEC Industrial 14's were informed that DEC was "de-emphasizing" the product. Prices went up about 30%, deliveries extended to an absolutely unreasonable length, and the message "DEC wants to dump the I-14" was explicit, loud, and clear. About 6 months later, since many of the present users didn't get the message or want to change, the prices went up again so those who were hard of hearing would understand. For example, the price of a VT-14 terminal went from $5990 to $9950 during this period. By early 1977 the I-14 was priced about 200% above, and the VT-14 programming terminal about 300% above similar offerings from DEC competitors. Deliveries were in the 6-9 month range, and sometimes were not met. The competition delivered from factory stock or from local distributor stock.

con't

But, as with the PDP-8, some customers wouldn't listen. Despite the attempt to kill it, the I-14 kept being ordered by customers. It is still available from TPL, and according to local DEC sources, orders are at a fairly high level.

I have observed with two of my customers, both >$1 billion corporations, that they held on to using the I-14 because of the tremendous difficulty in changing to anything else. I-14's are built into a control panel for one machine to control one machine. So once a company is committed to using them for all new machinery, they tend to proliferate rather rapidly. For example, one of these large companies has 32 I-14's in one plant, and a new one is installed every month or so. Since the I-14 is designed for troubleshooting by plant electricians using a VT14 terminal, the retraining required for using another programmable controller can be quite extensive and expensive. In both cases, despite the higher cost of the I-14, it still was cheaper to stay with it than retool and retrain.

Besides being a well known quantity, the I-14 has one outstanding redeeming social value that no other controller has: PDP-8's and PDP-11's talk to it with standard, DEC supplied hardware and software (INDUSTRIAL BASIC for the OS/8 and the ICCN package for RSX-11M).

The ability to monitor and control an I-14 from a computer can lead to some very interesting and powerful networks of processors. For example:

0) Multiple I-14's can be monitored and down line loaded from a single computer. I designed a system using a PDP-8 as a host for controllers on steel cutting machines. Each machine made several different parts from time to time. To change from one part to another a different program had to be put in the I-14. The new program was loaded from the PDP-8 to the I-14 on demand from the machine operator. The PDP-8 also kept track of the inventory of parts produced. Initially the system was designed for 6 machines, with extension to 8 in the future.
1) The I-14 can be used just as an I/O device for a computer. The computer can have A/D and other sensing devices connected to it. Depending on what the instrumentation tells it, the computer causes the I-14 to turn on or turn off outputs. PCS recently designed and built a temperature control system for 90 beer fermenting tanks configured this way. It was cheaper to use an I-14 than to hook up all that I/O to a PDP-8 using a DEC Industrial Control Subsystem (ICS). This is not idle speculation since 6 months earlier PCS shipped a functionally identical system for 145 beer fermenting tanks to the same customer using a PDP-8 with an ICS for the I/O.

con't

2) The I-14 can control a process in a "manual" type mode. An operator reads instruments, throws switches, etc. based on what he percieves the instruments tell him. Also, the instruments themselves may do temperature control, pressure control, etc. The machine is semi-automatic with the I-14 and instruments. A man is necessary to make some decisions. A computer would be connected to the instruments of the process, and also to the I-14. The computer reads these instruments and reads the state of I/O points in the I-14. Whenever a deviation from pre-programmed conditions occurrs in the system, the computer senses this by looking at the state of things, just like the operator would, and takes over the process from the I-14 to recover from the abnormal condition. Also, the computer could make periodic tests of the state of a process by forcing test conditions to appear in the system. A few years ago I built a large transformer vacuum inpregnating system which used a PDP-8 and I-14 in this way.

3) System 2 can be expanded to multiple I-14's with one computer (up to 8 for INDUSTRIAL BASIC, 16 for the RSX-11M stuff). This type of system represents a network with extraordinary control capabilities with very low demand on the computer because the I-14's really do all the grunt work, leaving the computer free to intervene when something goes wrong. A system like this is very much like timesharing, because the I-14's are connected over 20ma serial lines, and look somewhat like teletypes. PCS is involved with two multi I-14 networks. In one, the customer has installed 4 I-14's and will install more. Now he wants some information on what the machines are doing. A PDP-8 running INDUSTRIAL BASIC will be used to monitor the 4 existing I-14's with expansion for 4 more. This example demonstrates that a network can be built a piece at a time, whenever economics justify, without really compromising the final result. A second multi I-14 network was conceived because the process (a brewhouse) was so large and so complex that a computer, even of the dual 11/70 variety, would be too slow to act as the single controller. The reliability of having one system controlling the entire process was also questionable. Also, it would have been almost impossible to debug such a system. Using a computer-controller network, the digital I/O stuff was thrown on 12 I-14's, each one with about 500 I/O points, and the analog control loops were put on 12 analog programmable controllers. All 24 controllers will be supervised by a PDP-11/34 RSX-11M system. Small parts of the process equipment can be brought on line independently of the others, with the final debugging taking place when all of the pieces are performing properly.

One very nice feature of the I-14 to host computer communications is the ability to program transitions of specific I/O points in the I-14 to cause an interrput in the host computer and send a message telling what happened. Both INDUSTRIAL BASIC and RSX-11/M support

con't

this. So, rather than continually poll important points in an I-14, the computer can just sit and wait for somethine to happen. This feature can be used to unload a processer that, for several I-14's, could easily become I/O bound.

I don't mean to infer that other controllers are incapable of talking to computers. Even the Texas Instruments 5TI can be configured to do that. But no other programmable controller vendors offer the package deal that DEC does with its 8's, 11's, and 14's. The usual reason given for this lack of support is that there are too many types of computers, types of operating systems, etc. to justify the effort. Also, the concept of the controller-computer network is just as "new" as computer networks, meaning not enough buyers of controllers are doing it yet. DEC, on the other hand, decided to do it several years ago, and developed the hardware and software.

Well, Bob, I hope that gives you some idea of what an Industrial 14 is, and how it can be used with a computer.

Yours truly,

Michael E. Mazzoni
President

MEM:blm
encl.

HARRIS SEMICONDUCTOR
A DIVISION OF HARRIS CORPORATION

REF7532-JGL78-001
10/26/77

MR. ROBERT HASSINGER
LIBERT MUTUAL RESEARCH CENTER
71 FRANKLAND ROAD
HOPKINTON, MA 01748

SUBJECT: PDP-8 MICROPROCESSOR (HM6100)

Dear Bob,

I am a new Applications Engineer at Harris Semiconductor for the 6100 Microprocessor and its family of support devices. I am writing this letter as the first of many for the DECUS 12- Bit SIG newsletter for the purpose of sharing ideas about uses of the Microprocessor and to inform users about new 6100 products. I am planning to attend the Fall Symposium and I hope to present a paper about interfacing the 6100 to non-volatile MNOS memories.

Before proceeding with these objectives, let me review my credentials. I joined Harris ESD (Electronic Systems Division) in 1973 as an Electrical Engineer, and since then I have been a logic designer, programmer, systems designer and project supervisor. I have worked with the 6100 Microprocessor since March 1976 and have learned both its strengths and, painfully, its weaknesses. In my last position I was the manager of a Gen Rad 1792 and 1797 Logic Test Facility at ESD and obtained experience with the OS-8 operating system.

Mr. Dan Smith stated on page 17 of the September 1977 Newsletter that Intersil was "the only sizeable company making PDP-8 Micros". That is not so!!! We at Harris are very alive and well, thank you. I admit that in the past we have kept a low profile, but that is going to change. For the present we also plan to limit ourselves to 6100 devices and support chips. However, we are open to suggestions and do heartily encourage them. Some ideas that we are thinking about are PC boards, hybrids, leadless packages, and leadless packages on ceramic substrates as a module. We are currently working on a new, all CMOS SIMON that would have an In-Circuit-Emulator and would use a DEC STATION 78 as the terminal and software development system. We are looking for any suggestions regarding this activity.

Jonathan Lockwood

Jonathan Lockwood
Application Engineer
M.S. 54-70
(305) 724-7481

Vitramon

VITRAMON, INCORPORATED
BOX 544 • BRIDGEPORT, CONNECTICUT 06601
TELEPHONE: (203) 268-6261
TWX: 710-450-3640 • Cable: VITCOR

October 26, 1977

Mr. Robert Hassinger
12 Bit SIG
c/o DECUS
146 Main Street
Maynard, MA 01754

Dear Bob,

As a new 12 Bit SIG member, I enjoyed reading the first of the newsletters received a few days ago.

We are operating a PDP-8/a 500 here using Fortran IV to develop some complicated numerical control routines for special built equipment.

In addition, for certain data applications we are using Basic. On one particular job we are experiencing a system hang-up that seems not to have been covered by any of these published Software Up-dates, at least none that I have seen myself. I wonder if any of your readers can help. The problem is most concisely stated as follows:

A program was written in Basic to:

1.) Read a 40 character line of alpha-numeric data from a file (which contains about 400 lines of data).
2.) Print the data with appropriate headings and spacing on the line printer.
3.) Read and print the next line, etc.
4.) Calculate and print an average (considering a six character segment from each line).

The problem is that an occasional line of data not only causes the program to die, but also causes the computer to fall out of the "run" mode.

The data in the questionable lines appears normal and no obvious similarity exists among the various lines which have caused the problem to occur.

When the "problem line" is edited out of the data file, the program invariably runs to completion.

If PIP is used to print the file, the problem does not occur.

We will appreciate any help you can give us.

Sincerely yours,

VITRAMON, INCORPORATED

Robert Swart

Robert Swart
Technical Director

RS/mc