Contributions and correspondence should be sent to:

> Robert Hassinger, Coordinator - 12 Bit SIG
> c/o DECUS   ..or..   Liberty Mutual Research Center
> 129 Parker Street, PK-3/E55   71 Frankland Road
> Maynard, MA 01754   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 SUBMISSIONS

The Newsletter is currently published bi-monthly in the odd months. The deadline for each issue is the last Friday of the preceding even numbered month. Submissions are accepted at all times and are normally used in the next issue to go to press regardless of date of receipt. The deadline for ready-to-use material for the next Newsletter is 28-April-1978. Material requiring editing/re-typing should 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.

SIG COMMITTEES AND WORKING GROUPS

Steering Committee:

Robert Hassinger - address above - (617) 435-3452

Jim Crapuchettes   Lee Nichols
Menlo Computer Associates, Inc.   E. I. Du Pont
P.O. Box 298   Experimental Station
Menlo Park, Calif. 94025   Building 357
                                    Wilmington, DE 19898
(415) 323-3009   (302) 772-3839

Special Steering Committee Advisors:

Tom W. McIntyre   Stan Rabinowitz

Micro-8 Working Group   ## RTS-8 Working Group

Jonathan Lockwood   Lee Nichols - see above
Harris Semiconductor
PO Box 883
Melbourne, F: 32901
(305) 724-752-   M.S. 54-40

Symposium Program Exchange Committee

Send copies of software you wish to exchange at the next US symposium
to the appropriate committee member for preparation:

DECtapes   Russell Overbey
DEC floppys   PO Box Y Bldg. 92101-2
AED floppys   Oak Ridge National Laboratory
                  Oak Ridge, TN 37830

DECtapes   Earl T. Elis, Jr.
Magtapes   USCG R & D Center
DEC floppys   Avery Pt.
                  Groton, CT 06349
                  (615) 483-8611 Ext. 3-5176

LINCtapes   Larry Alber
                  FDA Room 1222
                  433 W. Van Burren
                  Chicago, IL 60607
                  (312) 353-5863

SPRING SYMPOSIUM

I recently participated in the Symposia Committee's Planning Meeting for
the Spring Symposium. Users interested in 12 bit subjects will find
this meeting interesting and varied.

A "Road Map" session will provide orientation to the 12 bit program and
establish coordination for the media conversion and program copying
activities. Tutorial sessions will be given on the new Commercial
version of BASIC, MACREL and TECO for new users. The PDP-8 product line
will conduct a product panel to discuss its products and plans with the
users. Workshop sessions include DECstation-78 applications, OS/8 and
RTS/8 product workshops where current and future plans will be discussed
and a String Processing Workshop that will cover the user written
languages SNOBOL 8.2 and STAGE 2. Papers on the MULTI-8 real time, time
shared operating system, the DECsystem-8 update for V3D, SLANG, a
structured language, an application of a micro-8 based system, and the
design of a Serial Interprocessor Buffer for the 8e are scheduled.

Two free format sessions are planned to provide opportunities to give
short discussions of projects and problems, and to exchange ideas.
Contributions to these sessions will be accepted up to the start of the
meeting. Details of how to take part will be included in the meeting
anouncements and mini-papers.

The FOCAL and TECO SIG sessions will also be of interest to many 12 Bit
users. Among the poster papers are ones on a Magtape File System for
OS/8, a Real Time Music Generation System, and one on PDP-8 Software
Diagnostics. The poster papers went quite well last time and I am
looking forward to expanded interest in this aspect of future symposia.

We expect that DEC will bring a respectable collection of PDP-8 family
hardware. The PDP-12 that TPL used to bring is no longer available so
no LINCtape facilities will be available at the meeting site. With
luck, someone in the local area might help out if the need comes up.
Among the hardware I expect to see are a variety of DECstation-78s and
VT-78s and a large PDP-8a based system. The large system will be the
host system for the DEC's demonstrations and the user's activities. If
all goes as planned the system will be configured with the demos running
in an RTS-8 foreground while OS/8 runs in the background for the users
so some user activity can take place even when DEC is showing off their
hardware. The configration will probibly support the following media:
RK05 hard disc, RX01 floppy disc, and TD8e DECtape. This system looks
like it will have a number of very interesting things about it. DEC is
also going to provide a PDP-11 system that will be coordinated by the 16
bit SIGs. Unfortunately, no means of communication between the 8 and
the 11 is currently available. We, the users, should be working to ard
mutual sharing of facilities. For example, we might like to get copies
of some of the FORTRAN software available on 11s or maybe access a
magtape. The 11 world might like to be able to access DECtape or
floppys.

In addition to our regular session rooms, the 12 Bit SIG will share a
small room (the Pan American A Room near our session room) with the
RT-11 SIG throught the meeting. We expect that it will serve as a
headquarters for our activities. At the last Symposium the RSTS and RSX
SIGs had such rooms and made good use of them as gathering spots and
places to coordinate and control program exchange activities. I hope we
will do the same to bring some control and order to what otherwise can
become an uncontrolled madhouse around the machines. Our committee will
be working on this and looking for help during the meeting.

MACREL NOTES

I understand that the next release of MACREL is being done at this time.
Various shortcomings in the first release are to be corrected and a few
features that DEC needs to have for things like RTS-8 are being put in.
As I understand it, a "wish list" of all the nice features that people
have suggested exists, and in the time remaining in the project, some
fraction of them may also get into the release. For example, Stan
Rabinowitz suggested that if users felt that shared page zero literals
(i.e. literals on page zero shared by more than one module to minimize

use of page zero locations) were important, they needed to speak up now.
Stan suggests writing to Ed Steinburger at DEC to do this. Send a copy
to me also.

My pet projects are to try to get FORTRAN II and/or FORTRAN IV to work
to some degree with MACREL. If anyone is interested in this area let me
know. Another pet project is to encourage people implementing languages
and other projects of general interest to use MACREL. If you come
across some one like that let me know.

DECUS PROGRAM LIBRARY NEWS

Chuck Conley, DECUS's Program Library Director, feels that the Library
cannot continue to support the full cataloging of the over 800 PDP-8
items and still do a good job. He is considering placing most of the
old 8 material in some sort of an archive status and removing the
listings from the regular catalog. We talked about the possibility of
putting all the old write ups and listings on a set of microfiche and
maybe putting the current version of the PDP-8 Volume 1 catalog on fiche
so it would be available to anyone who really wanted it. Presumably,
the machine readable versions of the programs would still be available
but the catalog could be streamlined, and the cost to DECUS reduced so
that updates to the catalog could be more frequent. Lars Palmer and
Chuck have selected about 35 items from Volume 1 that they feel should
be retained in the streamlined system because they seem to be of
continuing interest and value in today's hardware and software
environment. Other items can be added to this list if people feel they
meet these criterion. If you have any candidates to nominate, write to
Chuck at the DECUS address on page 1 and please send me a copy also.
The proposed list of Volume 1 programs (i.e. up to 8-599) to be retained
in future catalogs is as follows:

5-25	A Pseudo Random Number Generator
5/8-32a	A Program to Relocate and Pack Programs in Binary Format
8-84	One Pass PAL III
8-89	XOD - Extended Octal Debugging Program
8-91	MICRO-8: An On-Line Assembler
8-102a	A LISP Interpreter for the PDP-8
8-103	Four Word Floating Point Package
8-122	SNAP (Simplified Numerical Analysis)
8-127	XDDT Extended Octal - Symbolic Debugging Program
8-143	FFTS-R - A Fast Fourier Transform for Real Data
8-144	FFTS-C - A Fast Fourier Transform for Complex Data
8-168	Calcomp Plotting Package
8-178	Reverse Assembler
8-213	4K ALGOL
8-257	UCONN-EAP, Editor-assembler
8-261	QUBIC
8-335	COLPAC
8-375	Three Page Floating Point Package
8-397	8K Editor
8-400..	Execute Slow
8-446	A Patch to FFTS-R (8-143) for Use Without EAE
8-466	The RL Monitor System

8-497   8BAL - PDP-8 Macro Language
8-527   XDDT8E
8-530   8BALIB - 8BAL Macro Library Generator, etc.
8-554   ANOVA and DUNCAN
8-555   MULTC Multiple Correlation Program
8-556   CHISQ - Chi Square Program
8-557   CLUSTR - Cluster Analysis Program
8-558   CORREL - Correlation Program and FCCMP-VARMX Factor Analysis
8-570   BIN4SV
8-594   FP8 - Floating Point Arithmetic
8-597   N.I.H. OS/8 Package
8-598   CRT: - An OS/8 Handler for Tektronix 611 Storage Scope
8-599   DIBILD - Directory Rebuilder

Recent Program Revisions

ALPHA: Sort OS/8 Directories (Version 2xx) - DECUS 8-825
This revision updates ALPHA to handle the extended date from OS/8 V3D
plus the additional ability to print a header showing the way the
directory was sorted and the DECsystem-8 parameter block.

Price codes: Writeup - A01, DECtape - H12.

SSP: Scientific Subroutine Package - DECUS 8-802
I recently submitted the fully commented version of the complete package
on an RK05 disc. The SSP is now available on an RK05 disc, DECtapes and
RX01 floppy dics. I wrote up an explanation of the various ordering
options for DECUS. If you have any questions, check with them.

Recent Program Submissions

CONVERT: 4K Disk/DECtape Monitor to PS/8 Conversion - DECUS 8-856
This is the old CONVERT program from the original release of PS/8. It
was designed to move ASCII files from 4k Disc/DECtape Monitor DECtapes
to PS/8 (i.e. OS/8) files. The program only works with the TC08 DECtape
controller (the TD8e had not been invented back then !). In the current
form it is undocumented, if you do not have the PS/8 manual you will
have to figure it out from uncommented sources. The main reason it was
put in the library was to be sure it was not lost now that DEC has no
interest in it.

Price code: DECtape - H12

TECOVL: TECO overlay for display on VC8/e - DECUS 8-863
This is an overlay for TECO to allow it to display part of the text
buffer on a VC8/e as it does with the PDP-12 scope.

Price codes: Writeup and listing - D01, Binary paper tape - F02.

SNOBOL-8.2 Compiler - DECUS 8-864
SNOBOL-8.2 is a SNOBOL-3 compiler which runs in 8k under OS/8. SNOBOL
is a powerful string processing language based on pattern matching.
Major restrictions are the absence of functions and extended arithmetic
operations. Extensions include access to PAL code within the SNOBOL

program, and OS/8 file manipulation operations.   An extensive user's
manual is includ'd in the documentation.

Price codes: Write up - A02, DECtape - H12

SNOBOL-8.2 Demonstration Programs -- DECUS 8-865
A collection of sample SNOBOL-8.2 programs including source listing
program for SNOBOL programs, working documentation extraction system,
and a source library management system.

Price codes: Write up - A02, DECtape - H12

OS/8 Magtape Handler DECUS 8-869
A handler for writing ASCII files or output onto 7-track tape.
Character packing and unpacking and tape blocking are done in the
handler.   A further program for reading the tape into, for example, an
TDM computer is included.   Both programs have the capability of handling
variable record length files.

Price codes: Write up and listing - D01

Interupt Bus Testing Program - DECUS 8-870
This program is designed to deal with the situation were an unknown
device keeps causing interupts which results in interupt driven programs
(such as FORTRAN IV programs) to hang up in the interupt service skip
chain.   It helps locate and identify the source of the problem.

Price codes: Write up and listing - D01, DECtape - H12, Source paper
tape - G06

U: A Program to Type Out CCL Recollections - DECUS 8-871 & 12-210
"Although the CCL 'recollection' feature (OS/8 Handbook p. 1-54) is
convenient, it is sometimes annoying that the recollections are totally
'invisable'.   For example if a command such as '.UA' produces puzzling
results, it may be because the user himself has forgotten exactly what
he specified.   Program 'U' provides a convenient way of examining CCL's
recollections."

I typed the short program in from the listing in the writeup, got it to
work with V3D and added V3D commands like MAC and LINK with no
particular problems.   DEC really ought to support "remembering" with a
feature like this but until they do this program gets the job done. - RH

Price codes: Listing and writeup - D01, Binary paper tape - F02.

RSMON: RSTS terminal monitor on a PDP-8 - DECUS 8-873
This program allows a PDP8 with 12k to communicate with with a PDP-11
RSTS/E timesharing system as a terminal and to access the OS/8
environment.   Written for KL8 type interfaces on the 8.   The PDR: and
PDP: handlers mentioned below are required.   It is not clear if they are
included under this order number or if they must be ordered separately
under their own number.

Price codes: Listing and write-up - D01, Source paper tape - G02. See
note with next item.

PDP & PDR: Serial I/O handlers for interprocessor communications
        DECUS 8-874
Device handlers for use on any OS/8 system with a KL8e type serial
interface to do ASCII I/O.

Price code: Source code and write up on paper tape - G06. (Note: I am
working on adding hard copy listings and a DECtape with both these items
on it to the library.)

GAME: VT-52 or VT-78 Pattern Game - DECUS 8-878
An enhanced version of a popular TV pattern game. Creates a pattern of
light that evolves as it runs. Pattern editing keys and built-in
instructional displays.

Price codes: No write up, Binary paper tape - F02, DECtape - H12, Floppy
disk - K09

MTFOTP and TM8EZ - DECUS 8-879
Magtape FOTP and TM8EZ handler for the TM-8e Magtape. See writeup
elsewere in this issue.

Price codes: Write up - A01, DECtape - H12

RK8ESTORE: Data Acquisition and Storage Subroutines - DECUS 12-211
"RK8ESTORE is comprised of a set of subroutines callable from FORTRAN
IV. Subroutine DATSTO performs continuous single channel A-D conversion
with subsequent storage of digitized data on an RK8e disc system.
Digitization rates in excess of 10kHz are possible and the number of
samples which may be taken is limited only by disc size. Subroutine
DATGET retrieves the data from disc and returns it to the calling
FORTRAN program. Although PDP-12 automatic priority interrupt hardware
has been employed, it should be possible to use the normal interrupt
system with a slight decrease in maximum sampling rate." A minimum of
16k is required and EAE and API hardware.

Price codes: Writeup and listing - D01, Sources on paper tape - G06

LTFRMT: LINCtape Formatter - 12-212
A significantly enhanced replacement for all current versions of MARK12
that can be used under either DIAL or OS/8. Formats provided are the
standard 1, B, P, and X; plus the addition of a new Q option allowing
operator choice of both the number of blocks and the words per block.
Under OS/8 you can also: Zero directories and set up the DECsystem-8
parameter block with automatic volume number sequencing if desired. The
program must be assembled with PAL12.

Price codes: Write up - A01, OS/8 LINCtape - J11.

QCHESS: Quigley's Algebraic Chess Program - BASIC8-104
QCHESS is a Chess game written in BASIC that uses an algorithmic model

of the game of Chess.

Price code: Write up and listing - D01

YORK2: Two Error Linear Regression with Correlated Errors - BASIC8-105
A linear regression program that allows for errors in both X an Y and
positive and negative correlation of the X and Y errors. The
mathematics are identical to FOCAL8-315. Requires at least 12k of
memory.

Price codes: Write up and listing - D01, Source paper tape - G02

HOW TO WRITE TWO-PAGE SYSTEM HANDLERS FOR OS/8

The following is from an advance copy of material for DIGITAL SOFTWARE
NEWS provided by David Spector. I have added some additional
clarifications provided by Stan Rabinowitz.

This tutorial explains how to write two-page system handlers for OS/8
for those rare occasions when a device handler cannot be written to fit
in one page.

The remainder of this discussion will use the term "handler" to refer
only to two-page system handlers.

Such a handler has code and/or variables in the last page of fields 0
and 2. The last page of field 1, it will be recalled, contains resident
monitor tables.

All DIGITAL-supplied handlers have a one-letter version designation,
starting with "A". The version letter, truncated to 6 bits, must be
stored in (or immediately preceding) each entry point. (i.e. The
algorithm for determining the version of a handler is to start at the
entry point of interest and scan back towards lower addresses until a
location is found that contains a value less than 40 octal. This allows
handling the type of multiple entry points used in handlers like the one
for TC08 DECtape. RH) One entry point must be called "SYS"; any other
names or entry points are termed "coresident with SYS". All entry
points must be located in the first page (in field 0). All unused
location within the handler portions of each page must contain zeros.
(This is actually more of a good practice than an absolute requirement.
BUILD discards anything that would go in 07644-07777 and 27774-27777. It
is believed that 07600-07606 is handled the same way but to be safe all
these locations should be zeroed as documented. RH)

Your handler must contain the following items:

1. *0   (This tells BUILD that a header block is
                    starting.)

2. -n   (Where n is the number of device names = entry
                    points.)

3. DEVICE xxxx   (Where xxxx is the "group name" of your handler;
xxxx should also be the file name: xxxx.BN)
(Actually the file naming convention usually
used by DEC is xxxxSY.BN for system handlers and
xxxxNS.BN for non-system handlers. RH)

4. DEVICE SYS   (Device name)

5. 4xxp   (Where xx is a "device type" of your handler,
used by PIP to ZERO the device directory, and p
is the number of platters or other indication of
device size. If the device has only one size,
p=0.) (i.e. if p is defined equal to 0 it is
taken by BUILD as equal to 1. RH)

6. SYS&177+6000   (entry point address offset)

7. 0

8. xxxx   (where xxxx is the octal number of blocks = size
of your device.) (In the case of multi-platter
devices this is the size of <u>one</u> platter only.
The value of p from item 5 is used to multiply
xxxx to determine the total size of the device.
This value is used to establish the size of the
free space if you answer yes to the question
about zeroing the new system device when you
BOOT it at the end of BUILD. RH)

9. Repeat items 3 thru 8 for each entry point = device name,
changing item 4 for each name, and changing item 6 for each
entry point (use 5000 instead of 6000).

10. -n   (Where n is the number of words in your
secondary bootstrap routine.)

11. RELOC x   (Where x is the first location of your secondary
bootstrap when it is in memory.)

12. Your secondary bootstrap code (see below). NOTE: This code must
not contain origin statements.

13. RELOC

14. *200   (This tells BUILD to start the first page of the
handler.)

15. RELOC 7600
    ZBLOCK 7   (Contains monitor code.)

16. SYS, VERSION   (Entry point for SYS must be at 7607.)

17. The first page of your handler may occupy locations 7607 through
7743. Locations 7744-7777 are used by the monitor.

18. RELOC
    *400   (This tells BUILD to start the second page of
                            the handler.)

19. RELOC 7600

20. The second page of your handler may occupy locations 7600
    through 7773. Locations 7774-7777 must be reserved for use by
    BATCH.

These 20 items, in order, comprise the handler code.

Bootstrapping (starting up the monitor) involves the following steps:

1. The primary bootstrap code (loaded from the console switches,
    Read-Only-Memory, BOOT.SV, or other method) reads at least the
    first half of OS/8 Block 0 (one page) from your device into
    memory, at any desired location. This page will contain your
    secondary bootstrap code.

2. The secondary bootstrap reads the second half of Block 0 into
    the last page of field 0.

3. The secondary bootstrap reads the first half of Block 66 (octal)
    into the last page of field 1.

4. The secondary bootstrap reads the second half of Block 66
    (octal) into the last page of field 2.

5. The secondary bootstrap jumps to location 07605.

Your handler must obey the following restrictions:

1. Location 7612 of the first page must contain a 3. This is a
    flag to OS/8 that this is a two-page handler. (This has been
    established as the standard indicator of a two page system
    handler for the future. All programs should use this flag if
    they need to determine whether the system handler occupies one
    or two pages. In future releases at least MACREL, FRTS, and
    BRTS will use this flag. RH)

2. Location 7642 of the first page must contain either CIF 20, CDF
    20, or CIF CDF 20. This is used by FRTS when relocating the
    second page to or from the highest memory field. This is a
    temporary restriction. (The current released versions of FRTS
    and BRTS look at several explicit locations in the field zero
    page of the system handler and they alter those locations to
    move the second page to an alternate field. Patches will be
    released soon that will eliminate explicit references in the V3D
    releases except for the one to this location. Therefore,
    handlers that conform to this requirement should work with the
    patched versions of the currently released software. The
    explicit locations previously referenced were the ones in the

TD8e handler because that was the only one DEC had to support.
RH)

3. If you have any "once-only" code, it must appear only in the
    first page, since BUILD restores only the first page when
    building the monitor. (When you do a BOOT to finish BUILDing a
    system, BUILD swaps a copy of the system device handler you have
    designated for the object system device into 07600-07777 and
    27600-27777. It then uses that handler to write out copies of
    all the system head blocks, handlers and, if requested, a zero
    directory on the object device. When this is done, the first
    write to the new SYS: will cause any initialization code in SYS:
    to be executed so the copy of the handler in 07600 and 27600 may
    be altered from the virgin state. When it comes time to write
    out the system handler itself, therefore, a fresh copy of the
    07600 page is taken from BUILD's buffers but the 27600 page is
    taken from the initialized version. Note also that the
    secondary bootstrap is not executed during the initialization so
    it should not contain anything that would keep the above
    operations from working. RH)

4. There must be no instructions of the form 62nx, where n>0 and x
    is anything, in the second page or from 7607 through 7634 in the
    first page. This refers to CIF 20, CDF 20, and CIF CDF 20.

5. All desired instructions of the form 62nx in the first page must
    appear only in locations 7635-7743.

6. No constants (non-instructions) of the form 62nx are allowed in
    locations 7635-7743 of the first page.

7. Restrictions 4,5 and 6 are to be ignored in the case of
    instructions and constants which are used once only. (Once-only
    code is not executed on succesive calls to the handler once it
    is resident.)

(Notes: The conventions outlined in 4, 5, and 6 are to conform to the
future design for OS/8. BASIC's BRTS, FORTRAN IV's FRTS and perhaps
some future programs need to establish as large as possible a contiguous
address space. The usual procedure is to move the top of field 1
(17600-17774 - the USR tables, etc.) to some convenient, out of the way
place, if the system handler is a two page handler, the second page is
moved from 27600-27774 to 7600 of the highest available field and the
CDF 20, CIF 20 and CDF CIF 20 instructions in the first page are patched
to refer to the new field, and a patch is installed in locations
07600-07606 so that any attempt to exit or branch to the monitor will be
trapped to a short routine in the program that restores the system to
its normal state before actually exiting to the monitor. In the past,
as noted above, the moving of the second page of the system handler was
done in an ad-hoc manner critically dependent on absolute locations in
the TD8e handler. Now that DEC is generalizing the procedure, these new
conventions have been established. Apparently the relocation procedure
will be simply to recognize the presence of a two page handler from the
3 in 07612 then to scan over locations 07635-07743 looking for

instructions of the form 62nx. These locations will be patched to 62fx
where f is the highest available field to which the second page of the
handler is then moved. Note that for best results only 27600-27774
should be moved to avoid destroying the BATCH words in f7775-f7777 so
that if the program is used under BATCH it will not wipe it out. As I
understand it, this convention is not nessessarily implemented in the
V3D releases but it will be in future releases so handlers written now
should be designed to work with it. RH)

PATCH FOR TWO PAGE SYSTEM HANDLERS

Ian Templeton sent a copy of an SPR he has submitted to DEC. On a 32k
PDP/8e system with a Sykes Floppy Disk as the system device he found
that the following patch is mandatory for two page system handlers:

```
        .GET SYS SET
        .ODT
        713/0000 66
        725/0000 66
        ^C
        .SAVE SYS SET
```

Without this patch he says that SET SYS INIT will not work, and the
system bootstrap may be corrupted.

FUTIL PATCHES

The following is from an advance copy of material for the DIGITAL
SOFTWARE NEWS provided by David Spector.

This patch upgrades FUTIL V7A to V7B. It corrects the following
problems:

1. Typing CTRL/U crashes FUTIL if the current partially typed line
    contains a semicolon.

2. Overlay mapping (in SAVE mode) is not done correctly.

PATCH:

```
        .GET SYS FUTIL
        .ODT
        310/3523 3536
        333/1523 1536
        3342/3357 3362
        3343/1357 1362
        3351/1361 1357
        3354/2357 2362
        12520/0100 0200
        ^C
        .SAVE SYS FUTIL
```

PATCHES FOR DTFRMT

I recently submitted an SPR (8-2471) regarding a couple of old bugs in
DTFRMT, the TC08 type DECtape formatting program. On my 32k PDP/8-I a
control-C causes the "SYSTEM ERR" message to be printed, then an exit to
OS/8 which overprints the message rather than starting on a new line.
Also, after formatting a tape, I find that location 07777 has been
changed. This causes the date to be wrong (under V3D) and does other bad
things depending on which bits have been changed.

On the first problem, DEC could not reproduce the problem but they
suggested patching the CAF at 00300 to a NOP (7000). I tried it and it
worked. The only problem left is that no "^C" is printed now. On the
second, more important bug, DEC provided the following patch which seems
to work fine. I noticed that the patch they provided includes an update
of the version from V4A to V4B so I assume this patch will be published.

```
.GET SYS DTFRMT
.ODT
1763/1732 1765
1010/6401 6402
^C
SAVE SYS DTFRMT
```

Incidentlly, the total turn-around time to service this SPR from the
date the report was typed and mailed from here until the answer arrived
here was just 32 days. Considering that DTFRMT is an old, obscure CUSP
that must have a low priority for maintenance, I think the response
shows a clear improvment in 12 bit SPR servicing. Thanks and keep up
the good work DEC.

MORE ON PATCHING V3C DATES

Friedemann Brauer writes as follows:
"We have definitely decided not to buy OS/8 V3D; instead we've solved
the date problem as follows. CCL was modified to (print - rh) '78,
allowing for a patch every 8 years. Second, it chains to 'DATE.SV' if
the date word equals zero. This is a FORTRAN II program, asking for the
date and calling the SABR subroutine SETDAT, which sets up the date word
and the extension bits. (The new version of 'DIRECT V5' in DECUS will
allow him to print the new dates in directories as well as giving him
other features. - rh) A FORTRAN IV patch for the listing dates is:

```
.GET SYS PASS3
.ODT
561/ 6760 6770
^C
.SAVE SYS PASS3
```

How about fixes for RALF, PAL8 and 12, and CREF?"

Listings of DATE.FT and SETDAT.SB and a SRCCOM listing of his patches to
CCL are attached further on in the Newsletter.

QDP 78

QUODATA (EDUCOMP) recently sent me information on the announcment of a
new product based on the DECstation 78 called the QDP78. You get a
DECstation 78 with dual floppy disc, 16k PDP8, etc. and an LA35 for hard
copy plus the OS/78 version of OS/8 complete with FORTRAN IV and BASIC.
What distinguishes the QDP78 is that you also get QUODATA's QBOL and
SORT/8 software. QBOL is a business criented language almost identical
to DEC's DIBOL but it runs in an OS/78 environment rather than COS 308.
SORT/8 is a sort that runs in the OS/78 environment which has provisions
for both OS/8 style files and QBOL files which are packed differently.
The documentation for the sort makes it sound attractive. It is a key
sort so that it minimizes the scratch storage needed which can be an
important consideration on media with limited space such as floppy
disks. The interface between QBOL and its file format and the rest of
OS/8 is not as well developed as I would like, but overall the idea that
a varient of DIBOL is available for the OS/8 environment is very
interesting. The really nice aspect of the QDP78 is that you get the
entire hardware - software package including QBOL and SORT/8 for about
the same price DEC charges for just the hardware and OS/78. There might
also be some advantages in the area of more personalized and responsive
marketing, account servicing and delivery. For further information
contact Richard L. Prendergast at 196 Trumbull Street, Hartford,
Connecticut 06103 (203) 728-6777.

REAL TIME EXTENSIONS FOR U/W FOCAL

Dan Smith has been working on an extension for U/W FOCAL that he calls
FOCLOK. The purpose of Dan's work is to provide a facility for
dynamically loading, starting, terminating and communicating with an
arbitrary number of page-wise relocatable assembly language modules
which can be executed at each clock tick.

The modules are written in PAL8 (or PAL12) following certain
conventions. The user must be sure the conventions are followed and
that the modules are page-relocatable. Modules are not restricted to a
single page and features are provided to simplify cross page
communications.

The modules are assembled independently of FOCAL. At run time, the
FOCAL program can dynamically allocate space in core and load modules
into it. Once space has been allocated for a module, it cannot be
recovered even though the module is deactivated (i.e. there is no way to
"unload" a module execpt to reload FOCAL).

Each module can contain one or more tasks. When initially loaded, tasks
are placed on an inactive list; they can be dynamically activated and
deactivated under program control.

The clock rate is set in the usual way via the U/W FOCAL "Hesitate"
command. On each tick of the clock, all active tasks are executed once.
The order in which the tasks are called each time is determined by their
priority which is originally determined at assembly time but may be
altered at run time.

The package includes several instructions added to U/W FOCAL via the "@" command to implement fetch, start, kill, wait and list status of tasks. An FTSK function provides the various forms of communication between the tasks, FOCLOK, and the main FOCAL program such as passing data back and forth, checking and modifying priorities and checking on the active/inactive status of a task.

Presently this work is in a state of flux with very active development of new features and concepts. Dan is interested in input from others. Because it is a patch to the internals of U/W FOCAL it will only be useful to others who have that version of FOCAL for the foreseeable future and for the moment it is only configured for Dan's 32k PDP-12. Others with an 8 or 12 and at least 20k could adapt it fairly easily. Dan feels it is at the experimentation stage rather than being a full fledged tool ready to go. If you are interested, the address is: Eye Research Institute, 20 Staniford Street, Boston, Mass. 02114 (617) 742-3140.

RESTORING /E,/F AND /L TO V3D PIP

Jim van Zee sent the following note.
"Avid fans of OS/8 PIP may have noticed that the V3D release is missing a few favorite features: /E/F & /L. I don't suppose that these options will be badly missed since DIRECT is clearly a superior program for listing (i.e. directories of - rh) one's files, but I have always found PIP useful for short listings, especially when one is alternating between listings, deletions and squishes in order to create a hole big enough for some monster output file."

"At any rate I have whipped together a little patch which restores this feature which I will gladly distribute to any one else who might like to have it. I fixed the dates to observe the new date algorithm and also arranged them so they all line up. At the moment only numeric dates are printed, but someone with a bigger interest in the project could easily add the alpha-numeric format. My incentive to carry the project beyond this point is partly limited by the fact that I have a little program for restoring directories from a PIP /E listing with the old-style dates. ..."

(Note: I have always wished that some one would write a patch for FOTP that would give a directory listing capability because FOTP is so much nicer for doing the moves, deletes, and so on. On slower system devices like DECtape and maybe floppys this would give definitely faster file maintenance than alternating between PIP or FOTP and DIRECT. It would even be faster than Jim's patched PIP due to minimization of directory accesses, wild card file specifications and the ability to say things like "delete all files on dev1: that appear on dev2:". RH)

INPUTS FROM BRIAN CONVERSE

This material relates to newsletter issue 26, which was received after the enclosed document (see attatched discussion of FPP use later in this issue - RH) was produced. I'm not happy about the basic FPP-12

documentation, given my adventure with ALN. Certain nagging questions
occur every time I look up an instruction. The documentation is correct
and logical; it's not very reassuring! Another item on the FPP-12
simulator wish list is a new mode: single-word, entered via STARTS.
This would please Dan Smith (whose FORTRAN IV QUIZ was a real riot)
except that FORTRAN IV would need rewriting...it'd sure make me happy.

Kudos to 12-bit short notes and Bill Lennon. Microfiche is VERY neat IF
you have a reader. Otherwise, it's a pain. I have a reader and want
everything on fiche, but have no wish to ram it down other people's
throats; telling somebody he can go to the library to read his
newsletter is a cop-out. The first order of business is to see how many
have fiche reader, or access to one, and how many want fiche. Note that
the PDP-11 DECUS catalog went onto a single fiche and you can bet that
isn't the case for the PDP-8! Part of the fun of the 11 fiche was
roaring around thru the entire catalog without having to "page". It
will be interesting to see if user-DEC cooperation can gel before OS/8X
does. Hopefully, some way can be found to let the earlier 8s and the
PDP-12, as well as the Intersil/Harris IM6100 (how about some more
powerful support chips?) hang in there as was done with FORTRAN IV for
non-FPP machines. One good sign would be for DEC to ask current OS/8
users what the major problems with OS/8 are (not that they aren't
well-known) and go about making the new version consistent and then
consider fancy new features. It's sad that the 12-bit SIG is so small-
perhaps all of us readers can contribute by seeking out other PDP-8
users who would benefit as 12-bit SIG members.

Addendum:
The ability to make ANSI labeled OS/8 magtapes, as presented by Mike
Kelley brings up the exciting possibility that users with magtape drives
might be able to get most of the DECUS OS/8 library in the future
(should they have or get RTS-8) on a single magtape. The DECUS end of
things might be improved as well since at least the OS/8 portion of the
library could be manipulated on PDP-11s or PDP-10s.

As far as comments, it is unpleasant to see such fine software require
still another DEC software package when we are still reeling from the
likes of V3C & V3D! The logic used to justify Northwestern's choice is
impeccable, but as with Peter Lemkin's magtape package, our installation
would have to buy hardware; additionally, we'd now have to get more
software. Undoubtedly, RTS-8 is going to become more and more
important, it's just not time for the switch at this installation.
Further, to run OS/8 under RTS-8, a PDP-12 needs the special timeshare
hardware. We happen to have 16K of memory; the other useful-looking
magtape package is written in FORTRAN II and requires 28K... This is
really sort of wistful thinking: it'd be nicer if the Northwestern
package used less core and didn't require RTS-8; given the possibilities
of the software as described in their excellent abstract, it is
impossible to fault the implementation chosen.

INDUSTRIAL BASIC PATHES

Michael E. Mazzoni has written with the following offer:
"For those who are interested in the enhancements to Industrial Basic I
have been writing about, I have prepared a DECtape with all the
enhancements in sorce form. ... The software is available for a fee of
$20 to cover cost of DECtape, postage, and handling."
The following is a summary of the list of patches:

```
MULTI    Corrects a multiple statement/line with terminal input problem
CORSIZ   Corrects software core size bug
CHAIN    Change to loader to implement chain to .SV file instead of BASIC
STRLEN   Corrects a string length error message problem
DATE     Set OS/8 date word
ZERUDC   Turns off UDC outputs when exiting INBRTS
2NDTTY   Allows string input/output on 2nd TTY
DK8EP    Use DK8-EP clock instead of DK8-EC
PWRF8A   Power fail/instant restart for PDP-8/A clock
PWRFEP   Power fail/instant restart for DK8-EP programmable clock
PWRFEC   Power fail/instant restart for DK8-EC crystal clock
BIGLIN   Type out more than 72 characters per line
UDC      Replaces UDC with M-1705 outputs and DR8-ED inputs
ZM1705   Turns off M-1705 outputs when exiting INBRTS
PNT      Allows PNT(X) to print 8 bit ASCII instead of 7 bit
MESSAG   Prints message after a power failure
BITSET   Sets Digital Output 13 on when power fails
TABLE    Cross field symbol referencing for CREF
```

The address is Process Control Systems, 18130 S. Thornapple Lane, New
Berlin, Wisconsin 53151 (414) 782-3945.

The Use of TM8E Magnetic Tape on the PDP-8

Robert W. Phelps
University of Rochester Medical Center
601 Elmwood Avenue
Rochester, New York 14642

Industry compatible magnetic tape remains the most widely supported and cost efficient means for storing large data bases. Although character codes and record length often vary, 9-track tapes written on one computer can generally be read on any other computer equiped for 9-track tape. In order to use DEC's TM8E controller and 9-track TU-10 drive, it is essential that rather extensive software changes be made in order to write tapes that can be efficiently referenced on the PDP-8 and which are compatible with other computers (including other DEC computers). This writeup describes some of the hardware/software deficiencies of DEC's TM8E/TU-10 hardware and accompanying software, and describes the use of the MTFOTP program and TM8EZ handler written to circumvent these deficiencies.

The TM8E allows the user to write on 9-track tape in either of two modes: "Core-dump" mode which essentially treats the 9-track tape as if it were 7-track, and "9-track" mode which only writes the eight right-most bits from each twelve bit word. The latter mode does not allow conventional OS/8 buffers to be written, and then read without the loss of one-third of the characters in the buffer! The former produces tapes which cannot be read on other 7- or 9-track drives because 9-tracks are actually written, but without industry compatible parity and CRC information. Thus, data written on such a system is retrievable only by a PDP-8 and cannot be read by a different computer for analysis. In addition, MCPIP and PIP, the only OS/8 programs that are reasonably designed to reference magnetic tape, are extremely awkward to use. They require extensive tape motion to keep track of what is on a tape (MCPIP), or demand the user assume total responsibility for keeping track of what data is where (PIP). CAMP only allows for positioning of the magnetic tape and does not transfer information.

We have solved these difficulties as follows:

I.   TM8EZ magnetic tape handler.

A new handler was written which uses the calling sequences as documented in the OS/8 Software Support Handbook, and reads and writes standard OS/8 buffers. However, because the TM8E hardware will not write standard OS/8 buffers in 9-track mode, it is necessary that the handler reformat the data into a form compatible with the hardware. (Standard buffers can be written in "core-dump" mode but then tapes cannot be read on industry compatible drives.) This requires that (1) more than two pages of code are necessary for the handler and (2) an additional buffer is necessary between the OS/8 buffer and the magnetic tape. This additional code is stored in system block 66 (the block used for the extended TD8E handler) and executed in the first two pages of

the field defined in the source file. The bulk of the remainder of this same field is then used as buffer area. In general, the highest available field of core is dedicated to this purpose with programs that use the TM8EZ handler. System block 66 is loaded by the program TM8ESU, which may be run explicitly, or which will be run automatically the first time TM8EZ is called following rebuilding (not rebooting) of the system. TM8EZ is loaded with BUILD in exactly the same manner as any other handler.

The TM8EZ handler is only necessary if the user wants to write machine independent 9-track magnetic tapes. Otherwise DEC's TM8E handler is sufficient. Note that a hardware "bug" makes it impossible to read or write location 0 of field 7 with either handler without generating an error. (This is because the EMA-7 increment error is not gated with the EMA-7 increment enable bit.) To implement the TM8EZ handler, in the file TM8ESU.PA, set MYFLD=<the field for the extended handler and buffer>, then compile, load and save TM8ESU.SV. In the file TM8EZ.PA, set XFIELD=<same field as MYFLD>, compile, then using BUILD, load TM8EZ and insert TM8X,MTA0.

Use of the TM8EZ handler produces default records of 384 eight bit bytes (256 12 bit words or one OS/8 block), each frame being decoded from the twelve bit PDP-8 words according to OS/8 ASCII convention.

II. MTFOTP (Mag-Tape-File-Oriented-Transfer-Program)

MTFOTP is a modification to DEC's FOTP which treats magnetic tape (MTA0: only) as a special directory oriented device while at the same time retaining other FOTP functions. Indeed, by using the name FOTP.SV for MTFOTP.SV, CCL commands equivalent to those used to copy to and from other directory devices may also reference magnetic tape. Four additional features have been added for convenience in using magnetic tape, and are invoked using the appropriate option switches:

/P -- Causes the mag-tape to be positioned before the first record of the specified file. The file can then be read by any program as if it came from a non-file structured device.

/S -- Use a sub-file directory on the mag-tape.
On output, the output file name (or the first input file name if no output name is given) is the only entry made in the main mag-tape directory. All input files (wildcards are legal) are copied and their names are entered into a "sub-file directory".
On input from mag-tape, if a /S is specified, the first input file name is the name of the sub-file directory and is not transferred. The remaining input files are transferred from this sub-file. If no second file is specified, *.* is assumed and all files in the sub-file are transferred.
This feature allows similar files to be grouped together and referenced with a single name. It also allows a mechanism to circumvent the OS/8 maximum file limit.

/Z -- Zero the mag-tape directory. Note that MTAO: must be the
output device. Input files are ignored. A zero directory
must exist before any files can be transferred.

=nn - This option transferrs nn files beginning with the first
file in the input list if input is from mag-tape.
Subsequent files are those in sequential order immediately
following the specified file.

MTFOTP requires 12K of core. It will work with DEC's TM8E handler
if the handler is placed in file mode (use SET). For a 7-track TU-10,
the DEC supplied TM8E handler must be used. If the TM8EZ handler is
used, an additional field is required and TM8EZ and TM8ESU must be
assembled accordingly. As supplied, TM8EZ and TM8ESU will use field 3.
The following commands show an example of how to backup device DSK: on
mag-tape sub-file BACK.01:

COPY MTAO:</Z   /First zero the tape's directory if necessary.
                     Note that this is a different technique than
                     for zeroing directories on other devices.

COPY MTAO:BACK.01<DSK:*.*/S

To restore the device, use the command:

COPY DSK:<MTAO:BACK.01/S

When using MTFOTP on magnetic tape, the first file on the tape is
essentially a dump of an OS/8 directory with an extra padding record to
protect the file mark. The length of every file (except sub-files) is
exactly one "block" so that the "start block" of a file is really the
file number. Sub-files are N+1 files long where N is the number of
files in the sub-file and the additional file is for the sub-file
directory. Sub-files and sub-file directories cannot be changed once
they are written.

III. DIROVL directory overlay for mag-tape

DIROVL is an overlay to DEC's DIRECT program which allows it to
access directories on magnetic tapes and output the directories in the
same format as for other file-oriented devices. The CCL command

DIR MTAO:

will produce a listing of a magnetic tape's directory. The command

DIR MTAO:<sfd name>/S

will produce a listing of a sub-file directory from the mag-tape. The
/A option cannot be used with /S.

Dear Bob

        Hereby some material for the newsletter. In April we are
going to try to start an local users group in UK for the 12bit SIG.   At
the same time we are going to start a EUROPE wide group project to
create support material for MACREL and I'll try to summarize our present
thoughts about this project.

        I sincerely hope that anyone that has any thoughts on the
subject will let me know so we can incorporate their ideas into the
work.

        As I can see now we need three support modules for MACREL:

        1) A rather simple librarian. I definetly think such a
program is nessicary to utilise the library feature of MACREL. It
should at a minimimum be able to:
List the external reference points in a library.
Add to ,remove from and change modules in the library.   As this means
rewriting the library in most cases it will entail a lot of io and be
rather slow but try working with a library that starts reaching 30-40
modules and you see what I mean.

        2)A library of subroutines to perform commonly used
functions (typing strings on the console e.g.). We have a good
tradition to stand on in the PDP8 Cookbook already in the library.
Several questions come to mind when discussing this. Some of my
thoughts are set down here in an ustructured way,critisism is wellcome:

        1)How should the field questions be handleded?   My
opinion is that the best usage of the library is in fast not
too well structured programs were core and overhead 's not too
much a question ,and that for maximally efficient programs one
will have to do more detailed programming. If this is so then
all rutines should be crossfield callable.

        2)There must be no backward references in the
library.

        3)To be unique I propose the following scheme of
naming of external names:

        A)All user callable entry points start with . and users
should never use such names in their programs.
        B)Section names start with .$ and a entrypoint .abcde
normally recides in section .$abcd. The exeption is when
several entry points are in the ﹍me section when only one
entry point can correspond to the sect� name.
        C)Other external names in the library start with ..

4)The library is supplied in wellcommented source
form (of course) with a small page documentation for each
module(see appendix). However to simplify handling the file
begins with a TECO macro that seperates the file into seperate
modules and creates a batch file to build the library. as not
all have TECO and BATCH the distribution media also contrains
the ready to run library (+librarian)

3)The third support component is a macro library. The same
general rules as to names apply here to but all symbols generated inline
are local symbols so the user can if he wishes use local symbol blocks.
Any macro that need a temporary location must be able to take the
location from the argument list ie if a given macro normally is called
with:
.MACRO A,B
Calling it instead with:
.MACRO A,B,TEMP
Will make it use TEMP instead of generating a new temp location. Temp
locations nessisary and not passed by the user will be generated inline
e.g.
SKP
TEMP,   0

The macro library will also be supplied as a wellcommented
source but with a pre-amble macro to delete all comments (believe me
when MACREL scans a DECKTAPE library 10th time you wish it was half as
long !)

well do I hear any comments ?

Some not MACREL related notes.

I received a note from George Vele at Manchester on a
microprocessor emulation system that runs on a PDP8. I have looked at
the documentation and it looks quite nice. I enclose his note and I
think you can include is it is.

I enclose the call that the European steering committe for
the SIG sent out. Please reprint if you have room. And let me at the
same time extend a hearty wellcome to any PDP8 users in USA to come and
see us in Europe. If you come to this part of the word do contact me or
Earnst. (Earnst will by the way come to the Chicago meeting to help to
broaden the contacts between our SIGS.)

I sent some time ago 2 pieces of material to people at
Maynard and as I have not heard any reaction from there I shall have to
push on them by sneaking in a notice here.

To Gary Cole I sent a tape containing full support for the VT55 in PDP8 FIV, using the high level calls in the OS8 handbook. The reason I sent them to him and not to the library (they are ready for library distribution) is that they contain modified versions of DEC copyright material and I think that DEC have to ok it before they can be realeased. Please if you have need for support for the VT55 under FIV, do not deluge me with requests for the material I can handle a few requests but I have already had quite a few ,contact the PDP8 product line at Maynard and specially Gary Cole and ask them to release the material to the DECUS library.

I also sent to Chuck Connely a very preliminary draft of some thought on the revision of the PDP8 DECUS library. I told him I would do no work on it until I heard from him so I have not done any. I know that my draft will cause discussion (thats the idea) but maybe we can get somewhere if we get started now. I shall not repeat my ideas in detail here but just summarise very shortly

1)stop all rutinely support on programs that use non-current DEC hard ware and software. All such programs are lumped into one package and only made available as such (i do not mean genera` `oose modifiable programs but things like special versio. .ps for the 8k disc monitor). No new programs are accepteo .. this class.
2)Sort out as much junk as possible from the library,both bad programs and duplications. Maybe put them on one common media in one package.
3)Paper tape handling is very expencive but some programs are needed on paper tape(hobby type applications etc). Make several paper tape packages containing all those programs that are of use to such installation.
4)Continue ana press ahead with the work of collecting all other programs into a few large packages and make these available on as many media as is practical.
I also enclosed a review of about 400 of the programs.(in very short note form)

Now some thought generated on reading the last newsletter I have seen (number 25)

Page 12;the bat` handler can be used with FIV. There is a small 2 location patcn available,and i have published it in the newsletter. In this connection I have all the USER patches for FRTS for the new FIV release running (the error trap patch, the patch for USR and some other small ones, if you have anytrouble moving any of these from the previous FIV you can contat mee)

Page 15;i am very interested in the pfiles of mr Gardner(if he reads this I hope he sends them to me I have not had time to write to him yet). However I do not think they supplant my approach to with the mult handler rather they

compliment it very nicely. The fine thing with the mult
handler isthat files do not have to be moved. I can have
8*200(actually the most I do have is about 500) FORTRAN
sources available and can compile any of them by a simple
assign without moving them !
I also think that SRCCOM lists are about the best way to
document modifications to DEC suftware.

        Well I really think thats enough for now. Best whishes to
you all

example macro library documentatation
Name
Purpose
Usage
Expands to
Creates or adds to sections
Calls binary (from maclib)
Restrictions

example MACHEL binary library documentation
Section
Entry points
Purpose(for all entry points)
Size
External rutines called
Restriction(specially field d:o)

To all PDP8 users

As members of the Steering Committee of the PDP8 Special Interest
Group we would like to encourage you to participate in the next
DECUS Europe Symposium,. as speaker, chairman or training seminar
instructor.

Traditionally the annual Symposium provides good opportunity to talk
to DEC management, exchange information and experience with otuer
users and learn a lot about current and future systems. We feel that
the PDP8 users have had especially fruitful sessions during the last
symposia and we are planning further improvement.

For the 1978 symposium to be held in Copenhagen you may expect
training seminars on MACREL/LINKER (the new macro- and relocatable
assembler) and an introductory course on OS/78 - OS/8. Probably
these will be 'user-for-user' seminars, which means high quality and
low price.

Topics for the symposium itself will most probably include user
experience with MACREL, with the VT78 (a microprocessor PDP8 built
into a VT 52 terminal), the new release of RTS8 (with 64K memory
support?) and networking, both DECNET and others. Every effort will
be made to install a nice complement of new hardware and software
in the exposition room. Also, special sessions will be planned for
OEM and Commercial users.

However, you should realise that these plans can only become true
if we have the cooperation of YOU, the PDP8 user. It is your
experience, good or bad, we seek to communicate, they are your
problems and solutions that make these gatherings worthwhile.
Please take the effort to condense your expertise in a clear paper,
voice your needs and be with us!

Any remarks, ideas or suggestions are welcome to

Lars Palmer	Ernst Lopes Cardozo
Hässle	Physiology Laboratory
Mölndal 43130	Vondellaan 24, Utrecht
SWEDEN	HOLLAND

NOTE FROM GEORGE H VALE

George sent a note and user handbook on an interactive M6800 simulation program he is marketing to PDP-8 users. The program simulates all M6800 instructions and operates in a similar way to the firmware MIKBUG plus extra features such as break-points, memory display and I/O via Keyboard/ printer. A trace-mode feature prints instruction, effective address, register state and cycle count for each instruction executed. Interrupts are simulated from the keyboard.

More detail from Mr. G. H. Vale, Department of Mechanical Engineering, UMIST, PO BOX 88, Manchester, M60 1QD, U.K.

Hässle

AB Hässle Subsidiary of Astra Pharmaceuticals AB

Date	Our reference
Your date	Your reference

Dear Bob

Having just received newsletter no 26 I would like to add some remarks.

I dont think people know that there exists a very powerful reverse assembler from holland called DCP.I use it myself a lot (i left disasm behind when i found DCP)I enclose the users documentation to the 24k version (there is also a smaller 16k version)note specially things like
1)labels and literals are assigned autamåtically (if possible)on first pass
2)the s-sect is very powerful once argumenst to a subrutine is known

I appreciate very musch the problem of not being able to obtain backupp issues of the newsletter. I have all numbers and very often go back in the older numbers. I think that being able to get them all cn a set of microfish would be very useful. However I am sceptical to the newsletter only being on fiche I think òur compromise is the best possible combination of both worlds.

best wishes

Lars Palmer

Harris Semiconductor
PO Box 883
Melbourne, Fl. 32901

NOTES FROM MICRO-8 WORKING GROUP
By Jonathan Lockwood   Phone: (305) 724-7542   M.S. 54-40

SPRING SYMPOSIUM

Plans for the Spring Symposium are moving along. I have submitted
abstracts for four sessions covering Micro-8 applications. The
first session is a product panel on the DECstation-78 to be pre-
sented by Gary Cole of DEC. Next I will present a paper that
tries to explain the various options that are available for the
DECstation by exploring a hypothetical application that uses all
of these options. After this will be the paper about "Twenty
Thousand Leagues under the sea with a Micro-8 computer". It will
be presented by Mark Hernandez and Gordon Williams of the Scripps
Institution of Oceanography. This paper discusses the develop-
ment and use of a system that monitors ocean weather with acous-
tic measurements. Finally there will be a 1 1/2 hour hardware &
software workshop for short papers and open discussion. If you
would like to present a short paper during this time please let
me know. I will accept contributions up to the actual day of the
meeting; no abstracts are required.

MICRO-8 HARDWARE

In the last newsletter I mentioned that there were several 4K
CMOS RAMS available. Now there are also 2K RAMS avaliable.
These are a fallout from 4K's, ie. only half the memory matrix
works. The importance of the 2K RAMS is that they are more
readily available. The part numbers and relative pricing fol-
lows:

PART NUMBER	ORGANIZATION	PRICE (100'S)
HM6508	1K X 1	$ Y
HM6513	512 X 4	2Y
HM6514 & HM6533	1K X 4	5Y
HM6503	2K X 1	2Y
HM6504 & HM6543	4K X 1	5Y

I have developed an application note that utilizes these new RAMS
in a 32K word memory system. This application note is attached
for your information.

Some of the future components that are being considered are
various ROMs that contain commonly used routines. The design of
a 4k x 12 ROM is underway and thus the attractiveness of a ROM as
a standard component is increased. Some routines that are under
consideration are: 1) a floating point package; 2) some type of
real time executive; and 3) a standard monitor with a binary
loader, ODT, and automatic bootstrap. No firm commitments have
been made yet and thus any comments or suggestions regarding what
direction to take would be greatly appreciated.

PDP-8 FOR 1980

NOTE: The following product description is only the hypothetical conjecture of the author. Any similarity to existing or planned systems is pure conincidence. It is intended to be used as a seed for accademic discussions and possibly as a foundation for the next generation of systems. It is based on currently avail-able technology and thus is feasible to build today. The primary concept that is used is distributed multiprocesors to achieve the effect of a large, multitasked operating system.

DATE   APRIL   1980
ANNOUNCING THE NEW VAX-8/880
with VT-8X and NANO-8 computers

VAX-8/880 is the new PDP-8 computer system with RTS-8X, a multi-user, multi-tasked operating system.   VAX-8 is un-ique in its system architechture in that is based on mul-tiple PDP-8's sharing the overall processing tasks.   The foremost of these is the SUPER-8 computer consisting of a SUPER-8 processor, 128k words of high speed memory, and bulk core for the mass storage unit.   The processor itself utilizes a bit slice approach to emulate the instruction set and to provide a user micro-programable working store. It also utilizes a new floating point processor for sci-entific calculations.   This computer handles all input/ output operations via direct memory access.

The multi-tasking and overal system management is provided by the system controller subsystem.   This subsystem like the other subsytems, consists of a NANO-8 computer to ccn-trol the various devices within itself and to implement the various algorithms needed for system management.   The NANO-8 computer is a small module containg an LSI-8 micro-processor, 12k words of ROM, and 4k words of RAM.   The system controller monitors and controls all data transfer between the SUPER-8 computer and the various other sub-systems.   It also provides the function of interfacing to operator's terminal, a VT-88, for the VAX-8/880 system.

Hard copy output control is provided by the line printer subsystem.   It   simultaneously manages up three letter quality printers, a graphics printer (eg. Printronix), and a high speed line printer (eg. a 900 lpm Data Products printer).   When data is sent to this subsystem, it is buffered internally if the printers are not busy or it is spooled to the high speed disk.   The spooling is initiated by this subsystem.

Various types of devices can be controlled by the mass storage subsystems. There is a low speed controller for such devices as floppy disks, DEC tape, or 3M style cartridges. The high speed controller is used for 5 to 300 megabyte disks and for large reel magnetic tapes. The handlers for the various devices are part of the NANO-8 computer for each subsystem. In addition, the high speed controller has built in optimizing routines to minimize overall access time and large buffers to smooth out the data transfers. One application of the low speed controller would be a mass storage unit based on 3M cartridges in addressable carrousels.

Communications to the VAX-8/880 system is handled by two subsystems, one for asynchronous data and the other for synchronous protocols. The asynchronous subsystem, which can handle 8 full duplex channels at speeds up to 19.2k baud, is used to communicate with the various VT-8X's that are the prime input devices for the overall system. Several asynchronous subsystems can be the prime input devices for the overall system. Several asynchronous subsystems can be used within each VAX-8/880. Fully independent control for three channels is provided by the synchronous subsystem. These channels can be used to interface with an IBM-370, a PDP-10, a packet data network, or other VAX-8/880's.

The VT-8X's is a family of devices used to input data to the system. The first of these is the VT-80, a 16k word PDP-8 micro-computer that can be integrated into a user's special peripherial or into a hard copy terminal such as a LA-36. The next device is the VT-82, an upgraded version of the VT-80 with 32k words of memory, a hardware extended arithmetic unit, and bubble memory to store system programs and to provide some mass storage capability. The VT-84 utilizes the VT-82 in a VT-52 style crt terminal. It includes two progamable serial ports, two programable parallel ports, and an electonic program module for automatic bootstraping. With the addition of dual -double sided, double density- floppy disks, which slide into a slot provided for them in the crt terminal, the VT-84 is upgraded to a VT-86. The final version it the VT-88, a VT-86 with a bu.ld in, 80 cps thermal printer for limited hard copy output. The VT-88 is thus a complete, stand-alone computer with all the capability of early PDP-8's in a package the same size as a VT-52.

MICRO-8 COMPUTERS

Dan Smith has been mentioning various microcomputers that utilize the HM6100 microprocessor. I would appreciate any feedback on the these systems, both good and bad, as well as a discription of their applications. In a future issue of this newsletter I will write up the results of this feedback. I have found another system to add to Dan's list. This is a model 6600 made by Adapative Systems, Inc., PO Box 1481, Pompano Beach, Florida 33061, (305)-942-4000. It is a briefcase style portable microcomputer with dual 40 key keyboards.

OS-78 AND WPS-8

I have just received a Word Station-78 that I will also run OS-78 on. Since the Fall Symposium I have been working on a conversion program that will convert word processing files to CS-78 files and vice-versa, but so far have not had the time to finish this. After investigation, I found that there exists conversion routines for WPS-8 to and from COS-308. Also within COS-308 there are conversion routines to and from OS-78. There must exist, somewhere, conversion routines for OS-78 to and from WPS-8. If anyone knows of these, please call me!

HIGH LEVEL LANGUAGES

There is a need for a high level language that produces efficient ROM based code. After talking with Bob Hassinger, I learned that FORTRAN II has the potential for this. The first requirment is that the user's special peripheral interface can easily be integrated into the the high level language. FORTRAN II allows you to code in SABR in line to achieve this. The next requirement is that the language generates a final object file that needs no run time system to execute. FORTRAN II produces a save (.sv) file that could be made into a ROM pattern. One problem with the resulting code is that it needs to execute from RAM because the JMS instruction won't work in a ROM. Since MACREL exists, I believe that FORTRAN II should be upgraded to take advantage of its improved linking loader and at the same time, the hooks for producing ROM based code be added. The problem with a run time system could be solved by putting it in a standard ROM that could be sold as a component.

INTERFACING HM6504 OR HM6503 MEMORIES TO
HM6100 MICROPROCESSOR

This application note examines one method of using the HM6504
(4K X 1) or the HM6503 (2K X 1) with the HM6100 microprocessor.
The design objective is to obtain an all CMOS, 32K word, micro-
processor system that runs at 2.5MHz from a 5 volt power supply.
The assumed temperature range for this evaluation is -40° to
$+85^{\circ}C$, and thus all the parts will be industrial grade devices.
Worst case propagation delays will be used some of which may be
preliminary (e.g., the bus drivers). The design will allow use
of the 6503's because they are more plentiful and thus are easier
to obtain. Figure 1 shows the memory subsystem. This subsystem
is partitioned into 2 boards of 16K words each or 4 boards of 8K
words each. The switch positions to obtain this partitioning is
shown in Table 1. The partitioning is needed to distribute the
capacitance of the memory array which is tabulated as follows:

Signal Name	Total Capacitance for 16K	
	Typ(pf)	Max(pf)
ADDRESS = 12*CI*(4 fields)	240	384
DATA IN = 1*CI*(4 fields)	20	32
DATA IN + ADDRESS	260	416
WRITE = 12*CIX(4 fields)	240	384
DATA OUT = 1*CO*(4 fields)	24	40

Since the HD6433 drivers are rated at a 300pf load and the HD6440
latched decoder is rated at a 200pf load, this partitioning is
reasonable.

Figure 2 shows a simplified CPU subsystem that would be used with
the memory subsystem. It is assumed that the HD6102 will be used
only for extended memory addressing and thus we can use _XMAR
directly from the microprocessor rather than LXMAR* from the
HD6102. It is assumed that the total load on the motherboard
connecting the various PC cards is less than or equal to 300pf.

Figure 1.

MEMORY SUBSYSTEM FOR HM-6100 MICROPROCESSOR
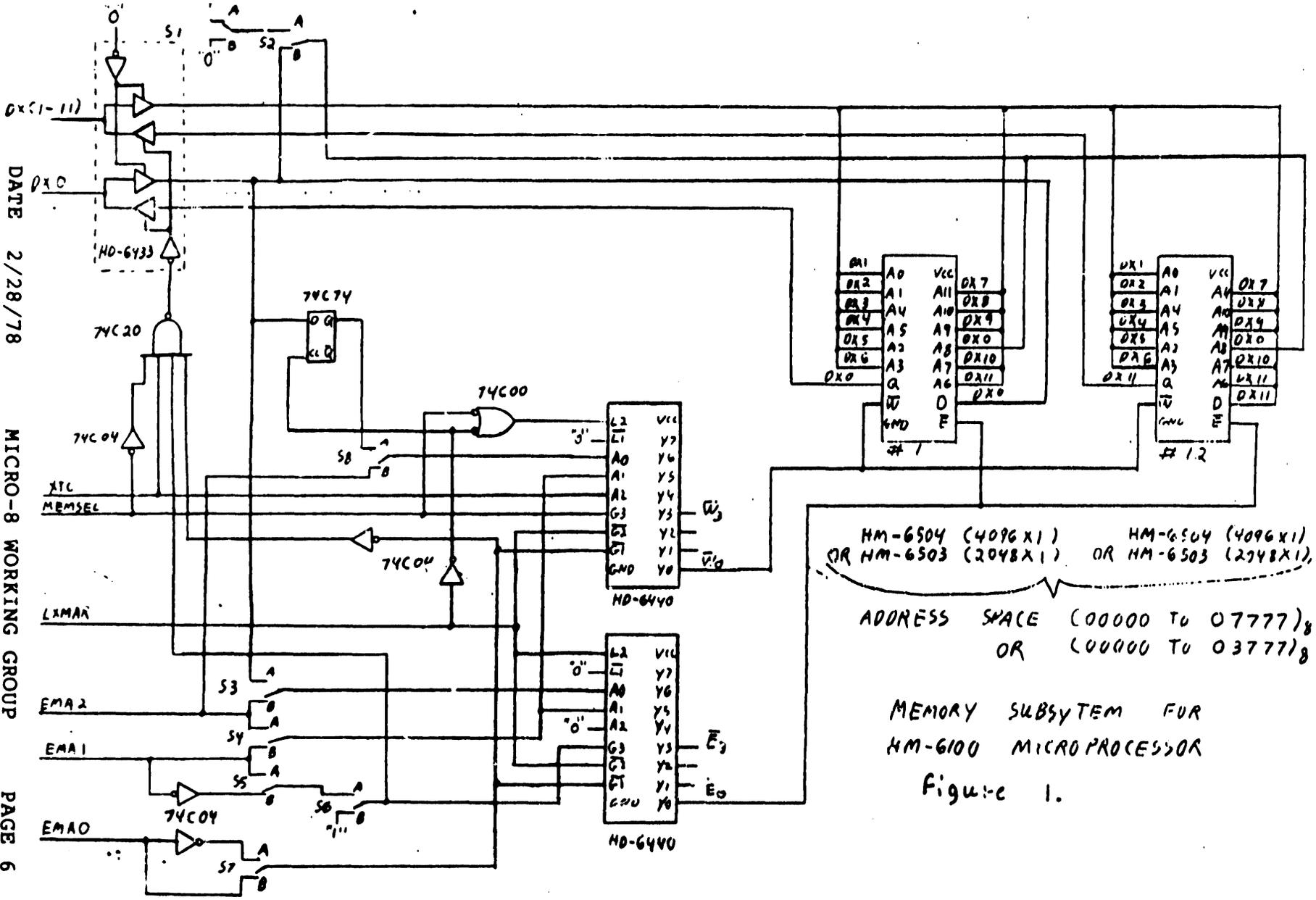
TABLE 1

Switch Positions For Memory Subsystem

Switch	A	B	FUNCTION
	X		For 8K system with HM6503H
S1		X	For 8K system with HM6503L
	X		For 8K system
S2		X	For 8K system
	X		For 8K system
S3		X	For 16K system
	X		For 8K system
S4		X	For 16K system
S5	X	X	Field select for 8K system
	X		For 8K system
S6		X	For 16K system
S7	X	X	Field select for both 8K & 16K systems
	X		For 8K system
S8		X	For 16K system

Field Select For 8K System

S5 A	S5 B	S7 A	S7 B	FIELD
X			X	FIELD 0 & 1
	X		X	FIELD 2 & 3
X		X		FIELD 4 & 5
	X	X		FIELD 6 & 7

Field Select For 16k System

S7 A	S7 B	FIELD
	X	FIELDS 0 - 3
X		FIELDS 4 - 7

SIMPLIFIED CPU SUBSYSTEM

Fig 2
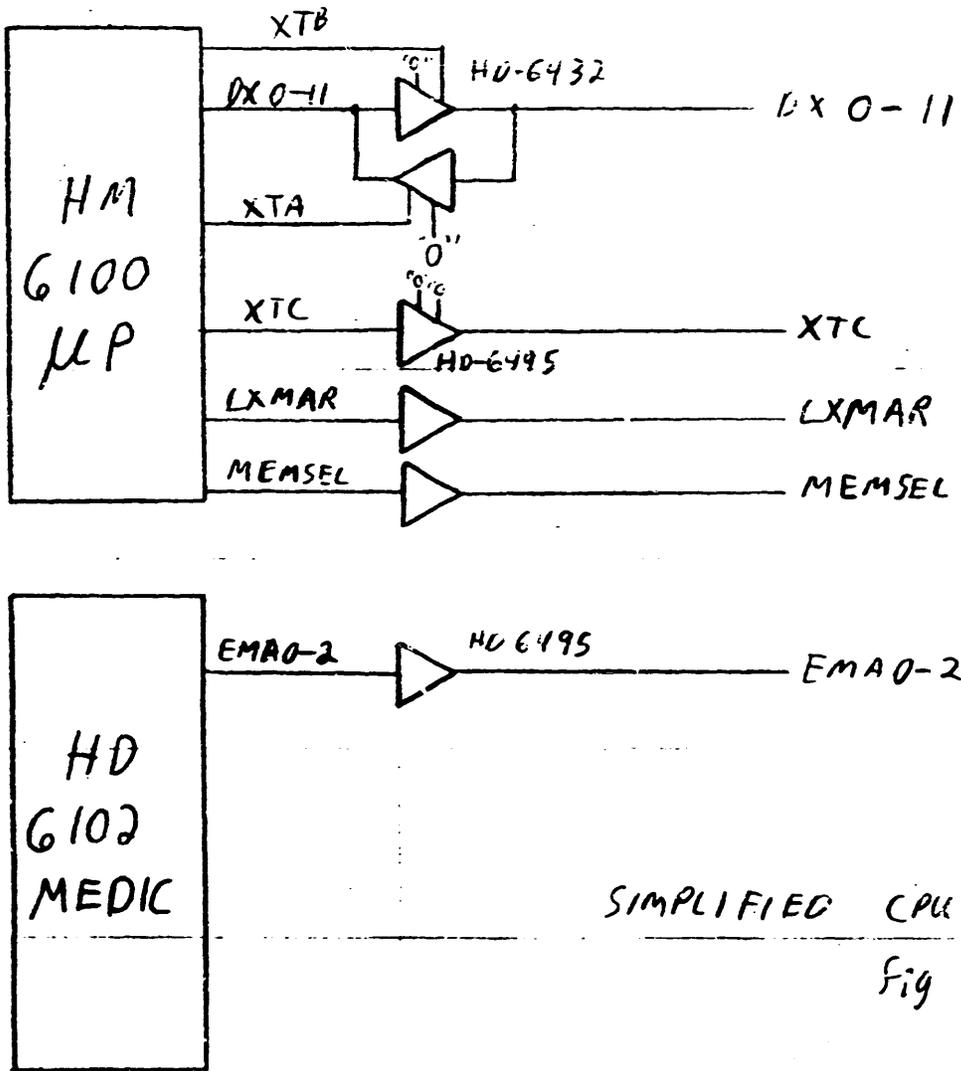
Now let's look at the various timing considerations. First, the
address setup and hold times are determined. The equations that
are used are from the HM 6100 Design Guide Lines. These equa-
tions are needed to calculate the AC parameters when changing the
state time of the microprocessor, ie. from 3.3 MHZ of the data
sheet to 2.5 MHZ of this example.

   ADDRESS SETUP

        HM 6100   HM 6504
   @ 3.3 MHZ   @ 2.5 MHZ   From the data sheet
From data sheet TAS'=3/4(TS'-TS)+TAS
----
      85   235   20

   ADDRESS HOLD

        HM 6100   HM 6504
   @ 3.3 MHZ   @ 2.5 MHZ   From the data sheet
From data sheet TAH'=1/4(TS'-TS)+TAH
----
      125   175   50

Since both LXMAR and the DX bus are both buffered, the setup and
hold times that the memory sees is approximately the same as from
the 6100 itself, albeit delayed.

Next is the access time from the falling edge of LXMAR:

	ACCESS TIME (ns)	COMMENT
From 6100 data sheet	520	
TAL @2.5MHZ	770	TAL'=5/4(TS'-TS) +TAL
Minus delay from 6495	100	LXMAR delay on CPU card
Minus delay from 6440	100	LXMAR delay to mem enable
Minus delay from 6504	300	TELQV chip enable access time
Minus delay from 6433	100	Delay for data on mem card
Minus delay from 6432	100	Delay for data on CPU card
Access time margin	70	

From this tabulation and the the timing diagram (fig. 3), it can
be concluded that the memory should perform a valid read oper-
ation over the full temperature range.
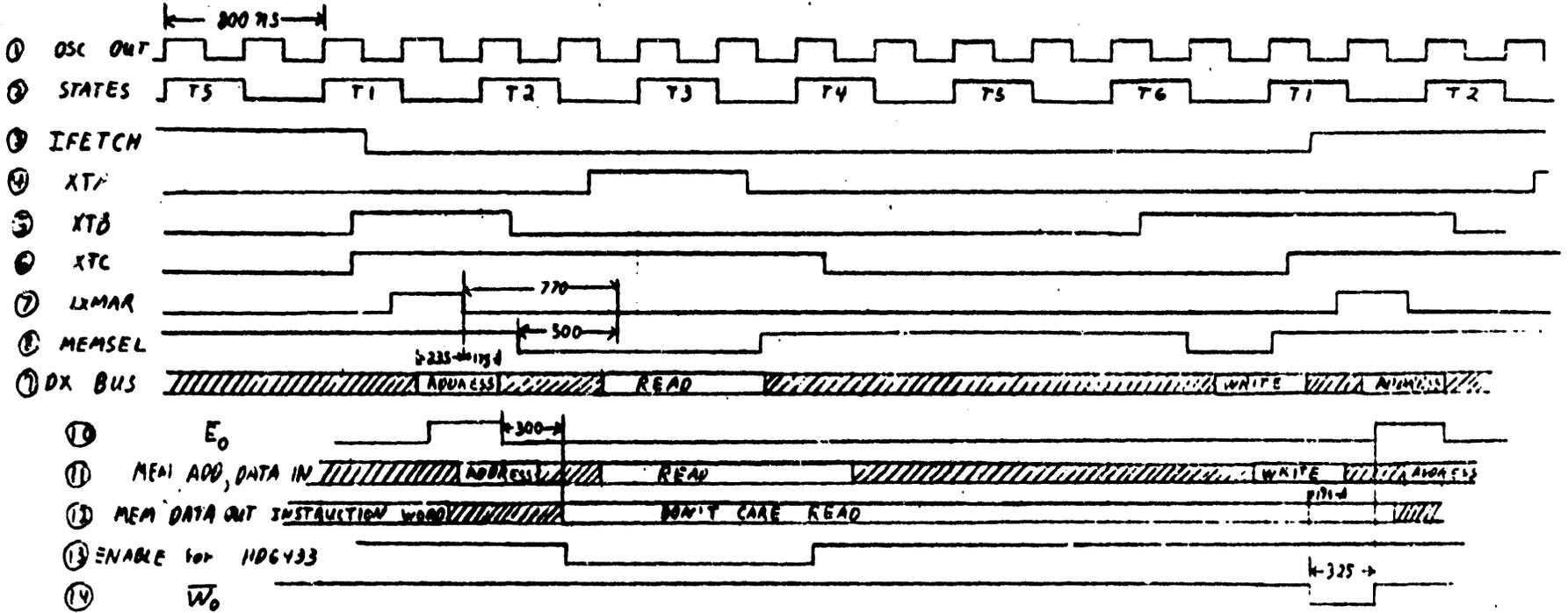
Notes: a) Signals ①,③-⑨ at pins of microprocessor
b) Signal ③ is internal to microprocessor
c) Scale = nano-seconds

0   250   500   750   1000

TIMING DIAGRAM FOR
DCA 0 (3000) INSTRUCTION
FREQ = 2.5 MHZ

Figure 3

Since the memory strobes in data on the negative edge of the
write signal and the microprocessor uses the positive edge of
MEMSEL, the write operation is more difficult. The approach that
is taken here is to synthesize a write signal after MEMSEL and
before the next LXMAR. With the configuration shown in fig. 1,
the only delay involved is the enable for the latched decoder,
which was chosen for both its speed and high capacitance drive.
The memory needs no setup time for data, but it needs 80 ns of
hold time. The micprocessor provides a minimum of 175 ns of hold
time. Since the data is delayed by the bus drivers, there is
ample margin for the hold time. Another parameter that the
memory is concerned with is the time from WRITE going low to
ENABLE going high; this time is 200 ns minimum. As shown in
fig. 3, this time is also met.

CONCLUSION

The resulting memory subsystem should work for all memory oper-
ation over the full temperature range. The primary reason why
the HM6504 memories were chosen was the minimum package that is
obtained with the 18 pin DIPs. A side benefit of this approach
is a reduction in overall system power because LXMAR does not go
to all memory devices simultaneously. If the temperature range
is restricted to 25°C, and/or if the system voltage is kept at or
above 5 volts, then the system could operate faster than 2.5 MHZ.

GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA. GEORGIA 30332

(404) 894-5201

SCHOOL OF PHYSICS

January 30, 1978

Mr. Bob Hassinger, Coordinator
Liberty Mutual Research Center
71 Franklin Road
Hopkinton, Massachusetts 01748

Dear Bob:

Here is a summary of the 12-bit hardware session in San Diego
(finally through belatedly). I hope it is in a format which is
directly reproducible and will get in the next newsletter.

Sincerely,

N. S. Kendrick

NSK:ec

1977 Fall Decus Symposium, San Diego, Ca.

This tutorial-workshop provided a forum for a lively exchange of ideas at the fall symposium. The session was orgainzed around a series of short (supposedly 10 minute) presentations on specific hardware problems for the benefit of the "expert" and novice alike.

Mike Allen of Livermore Labs led off with a series of "Ten Straight-8 Hints" below which some of us old timers appreciated. (A note to the younger set: Those are real logic drawings!).

CPU HINTS & KINKS #2

REF: PRINT BS-D-8P-0-9 TIMING, KEYS, SWITCHES, AND RUN CONTROL.

PROBLEM: MEMORY LOCATIONS ALTERED DURING POWER-UP.

A. DON'T POWER-DOWN!

B. PRINT COORD. 2 ½ D.

PWR STATUS — PWR OK
10µf
20V
1800pf MICA
FILTERS ADDED
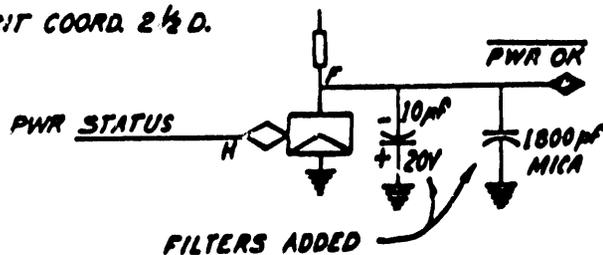
PROBLEM: SPURIOUS OPERATION WHEN USING KEY START, EXAMINE, DEPOSIT, LOAD ADDR

PRINT COORD. 2 A ½

WS01 PA30
SPO
50µf 50V
CAPACITOR ADDED TO INCREASE T.C. TO 100 MS.
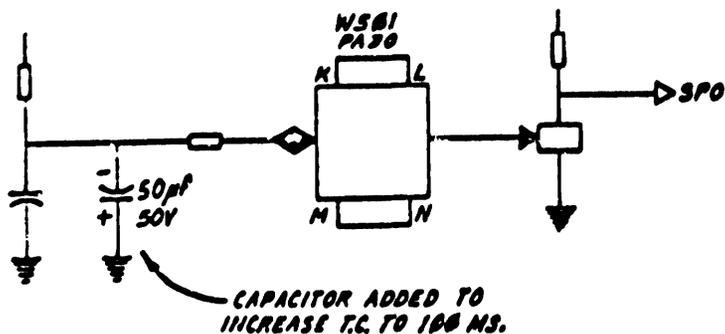
CPU HINTS & KINKS #1

1. QUICKIE MEMORY TEST

A. USING FRONT PANEL SWITCHES, DEPOSIT 7777 IN 20 OR MORE LOCATIONS IN EACH MEMORY FIELD.

B. USING EXAM KEY, EXAMINE THOSE LOCATIONS.

C. REPEAT TEST USING 0000 PATTERN.

CPU HINTS & KINKS #2

PROBLEM: 30 CPS TERMINALS REQUIRE DELAY FOR CARRIAGE RETURN.

NOTE 1: SINCE CPU SLOTS ME? AND MF29 ARE BLANK, ADD THE FOLLOWING CIRCUIT:

NOTE 2: DISCONNECT THE SIGNAL "OUT ACTIVE (0)" FROM PIN N OF ME21. IT IS REPLACED BY SIGNAL "SET TTY FLAG".

NOTE 3: ONE SHOT TIME DELAY IS SET TO THE MAX. TIME FOR A CARRIAGE RETURN.

TO PIN N OF ME21   SET TTY FLAG

R302

195 MS ONE SHOT

OUT ACTIVE (0)

R121

OUT ACTIVE (1)

TTO 0 (1)   S
TTO 1 (0)   T
TTO 2 (0)   U
TTO 6 (0)   V

R121
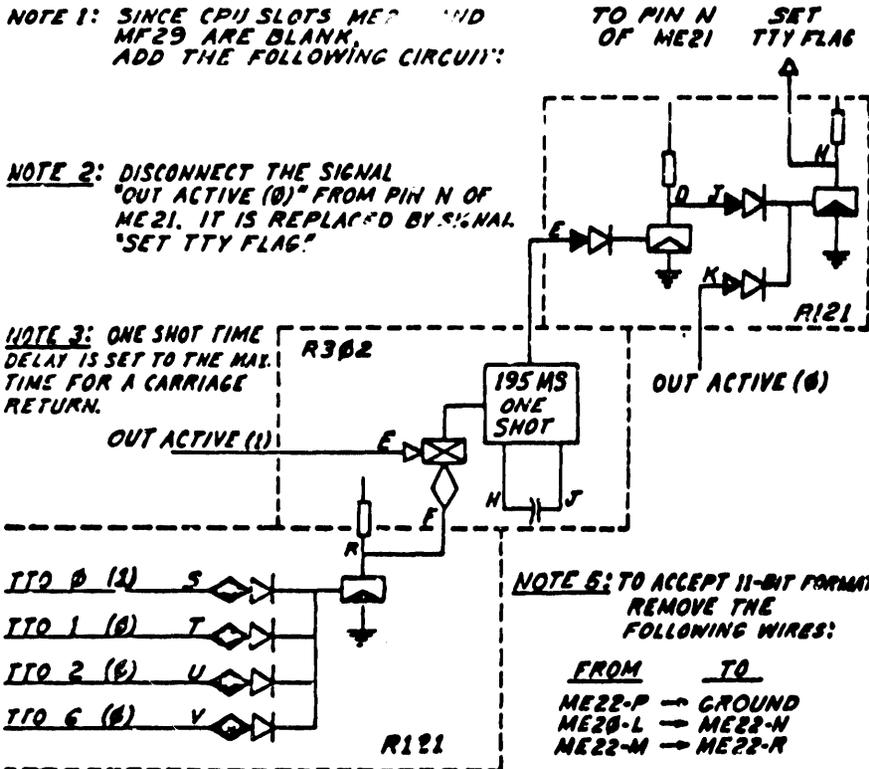
NOTE 5: TO ACCEPT 11-BIT FORMAT REMOVE THE FOLLOWING WIRES:

FROM	TO
ME22-P	GROUND
ME20-L	ME22-N
ME22-M	ME22-R

ADD THE FOLLOWING WIRES:

FROM	TO
ME22-F	ME20-L
ME22-U	GROUND

NOTE 4: REPLACE MODULE R405 IN SLOT ME15 WITH R401 MODULE ADJUSTED FOR 21.12 KHZ.

CPU HINTS & KINKS #3

REF: PRINT BS-D-8P-0-10 INPUT/OUTPUT CONTROL

COORD: 6C

PROBLEM: SPURIOUS SKIP SIGNALS WHEN USING LONG I/O CABLES.

SKIP BUS IN

W005 PE9

D   E

100

.01

FILTER ADDED

R603 PB21

PA

COUNT PC ENABLE

T28   R

TT SET
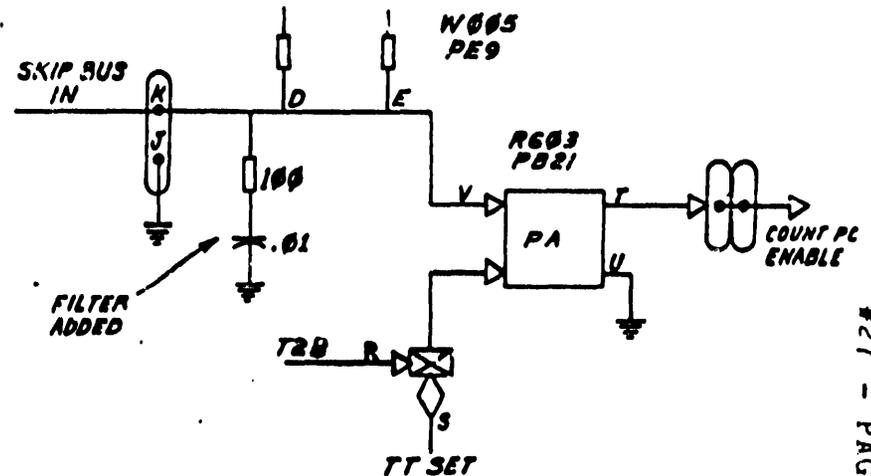
RF∅8 HINTS & KINKS #1A

PROBLEM: EAE CONFLICTS WITH RF∅8 DISK DATA
BREAK CYCLES.

REF: PRINT RF∅8-∅-1 INPUT OUTPUT CONTROL

COORD: B 7½

DMA LDMAP ——————

DIODE ADDED

1N4148

EMA∅6 (∅)

IOC DRE (1)

MRS
S262
B15

IOC SCLP1
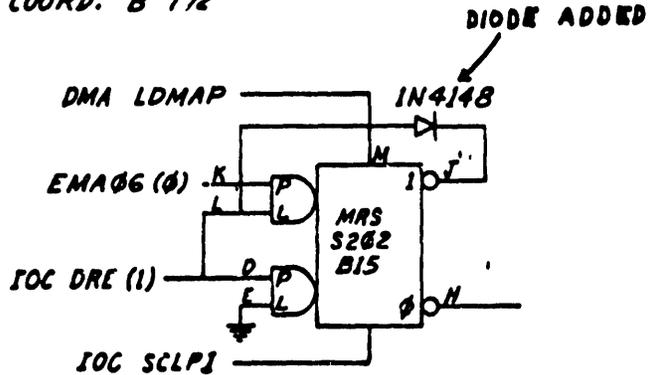
NOTE: DIODE CLEARS MRS WHENEVER DRE IS SET. THIS
MOD. WON'T WORK IF ANOTHER RS∅8 IS ADDED.

REF: PRINT RF∅8-∅-5 STATUS REGISTER

AC DEP (1)

DC DRL SYNC (1)

DRL
S2∅3
A15

SIGNAL NAME
CHANGED

IOC SCLP1
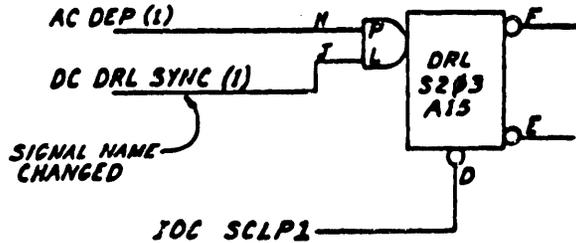
CPU HINTS & KINKS #5

REF: PRINT BS-D-182-0-3 EAE CONTROL.
COORD: 1D AND 5D

PROBLEM: EAE CONFLICTS WITH DATA BREAK CYCLES.

CHANGE CIRCUIT TO THAT SHOWN BELOW.

EAE INST.

PC → MA
ENABLE

B SET

R101
PE3∅

1N4148

DIODE ADDED
TO NODE
TERMINAL

EAE LINK COMP

EAE LINK COMP

R107
PE3∅
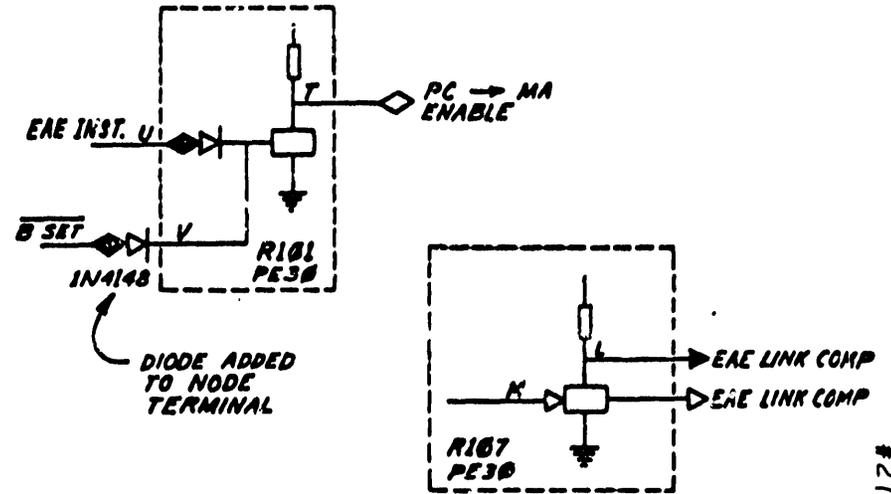
NOTE: PIN CHANGES IN
BOTH CIRCUITS.

RF∅8 HINTS & KINKS #2

PROBLEM: DISK DATA DIAGNOSTIC FAILS ADDRESS
         TEST 2.
  REF: PRINT RS∅8-∅-1 TIMING CONTROL
COORD: A ½ 3 ½

NOTE: ADJUST RS3∅2 ONE SHOT TO APPROX.
      11∅ μSEC DURATION.

PROBLEM: TIMING OR ADDRESS TRACK WITH
       NO OUTPUT

A. GET DEC TO RE-FORMAT TIMING AND
   ADDRESS TRACKS.

     -OR-

B. REMOVE MODULE IN SLOT A∅1 AND
   RE-INSERT IT UPSIDE-DOWN. THIS
   PICKS OFF DUPLICATE SET OF TIMING
   AND ADDRESS TRACKS ON OPPOSITE
   SIDE OF DISK. BE SURE AND RE-ADJUST
   TIMING AND ADDRESS AMPLIFIER AND
   SLICE LEVELS.

RF∅8 HINTS & KINKS #1B

REF: RF∅8-∅-4 DATA CONTROL.
    ADD CIRCUIT AS SHOWN:

EMA CLMAP
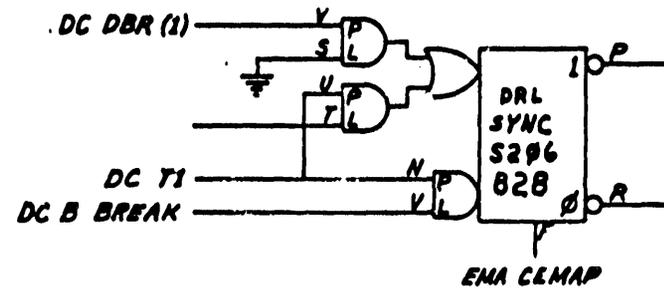
NOTE 1: DISCONNECT WIRE FROM S2∅2 MODULE PIN D AT
     SLOT B13. PRINT COORD. C ½ 6.

NOTE 2: CHANGE THE FOLLOWING SIGNAL NAMES AND RE-WIRE:

FROM	TO	COORD.
IOCON ADDR. ACC	IOCON T1	C7
AC DEP (1)	AC DEP D	B8 / C8
DC R/W (1)	DC B BREAK	B3
TC BTCS-	AC DEP (1)	B ½ 2

NOTE 3: THIS MODIFICATION CAUSES DRL TO SET WHEN
     EAE CAUSES A BAD WORD TO TRANSFER
     FROM MEMORY TO DISC OR VICE-VERSA AS
     WELL AS WHEN A WORD IS LOST ENTIRELY.

POSITIVE I/O INTERFACE   HINTS & KINKS

WHEN CONNECTING A CONTROLLER DESIGNED FOR
A PDP8I (OR LATER MACHINE) TO A STRAIGHT-8
WITH POSITIVE I/O INTERFACE, BE AWARE OF
THE FOLLOWING TIMING RELATIONSHIPS:

STRAIGHT-8 SIGNAL NAME	POSITIVE I/O INT. SIGNAL NAME
BT 1	BTS3
BT2A	BTS1

NOTE: APPROPRIATE CONTROLLER MODIFICATIONS
MAY BE NECESSARY TO ACCOMODATE
I/O TIMING RELATIONSHIPS REVERSED BY
THE POSITIVE I/O INTERFACE.

PCØ1 READER/PUNCH   HINTS & KINKS #1

PROBLEM: WON'T RELIABLY READ OILED AND
UNOILED TAPES.

A. DON'T USE BOTH TYPES!

B. RE-DESIGN THE PHOTO-DIODE READ
AMPLIFIER TO PERMIT INDIVIDUAL
ADJUSTMENT OF ... V FROM EACH
PHOTO-DIODE TRACK.

PROBLEM: PERIODICALLY READER INTERFACE LOSES
THE READER FLAG SIGNAL.

A. MECHANICALLY ADJUST THE READER
SPROCKET WHEEL ON IT'S SHAFT IN
ORDER TO ASSURE THAT THE TAPE
CHARACTER JUST READ HAS PASSED
COMPLETELY BEYOND THE PHOTO-DIODE
READ HEAD.

B. MECHANICALLY ADJUST THE STEPPING
MOTOR MOUNTING POSITION TO ASSURE
PROPER STROBE TIMING.

If one needs to make the cheap equivalent of a one-shot the following circuit performs admirably. The curve was plotted from off-the-shelf components assuming a single TTL load (the extra inverter) on the output.
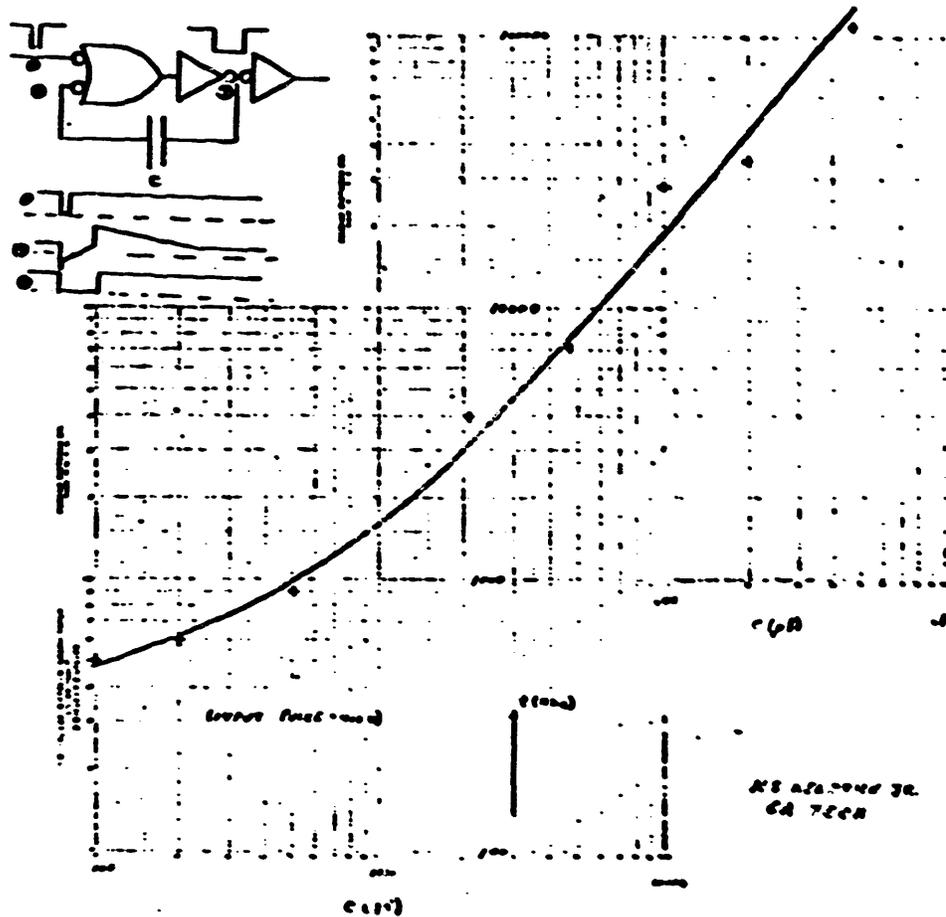
The concepts of AC coupling to pick off either positive or negative level transitions are shown below.
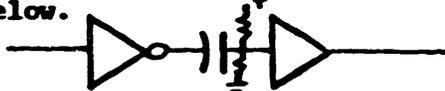
Delays up to a few hundred nsec are easily obtained by the following technique.
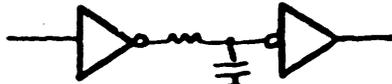
The above ideas were presented by Neb Kendrick of Georgia Tech.

Tom McIntyre, late of W.V.U. - now with DEC, gave a nice tutorial on device selectors and also described a method of mapping bits in the AC of a Straight-8 which among other things can be used to swap the AC.

Jim Mahaffee of Georgia Tech described an RS-232 interface employing the type 1488 or 1489 line drives available from several manufacturers. Jim also presented some details on using the GE H11A2 opto-isolator transceivers and a very interesting collage pattern for the omnibus which allows a high chip density and the mixing of 14, 16, 18, 20, 24, 40 etc. pin chips on the same board. This artwork is available free.

A large group assembled to hear Bill Lennon of Northwestern U. expound on "Contraceptive Jellies and PDP8/e Fuse Holders". Several people commented on difficulties with both of these.

John Lockwood of Harris Semiconductor (Melbourne, Fla.) presented a lot of useful information on MNOS EAROMS, not the least of which was what all the letters mean. For the uninitiated these are Metal Nitride Oxide Silicon Electrically Alterable Read Only Memories. These devices can be reprogrammed by purely electrical means (as opposed to U-V erasable PROMS). They should find applications as look-up tables, program storage and auxilary storage devices. They are manufactured by NCR and GI. Level shifters for use with them are available from Harris.

Earle Ellis of the U.S. Coast Guard R. & D Labs contributed the following.

20 mA. CURRENT TO CURRENT ADAPTOR

This circuit was designed to facilitate the interconnection of two processors connected with a 500 foot cable using the current loop portion of a standard teletype interface. All the power required for proper operation of this circuit is provided by the "current" loops.

Although communication with a standard model 33 or 35 Teletype is considered as being done using a 20 mA. current loop, in practice it is a voltage circuit. Since the circuit designed is sensitive to the polarity of the current loops, one of the first measurements that must be made is the voltage levels and polarity sense of the loops being connected.

The Teletype uses a mechanical contact closure to "transmit" bits of information to the processor interface. In this pair of lines, one will be found to be more positive than the other. In our application, using a DEC KL8-J and PCM 12060, the voltage was between 19 and 20 volts for both. The polarity of this voltage must be observed for the proper operation of the optically coupled isolator. Almost any isolator can be used. In our case an HP 5082-4371 was used because it was easily obtained and had current gain.

The second pair of lines provides a current to the Teletype. The voltage and current characteristics on this pair varied between interfaces. This is the reason the circuits are not symmetric. Resistors R1, R2, R3 and R4 were selected to provide the proper operating current to the isolator input and maintain a "20 mA." current in the line. These values will vary depending on the interfaces being used and the isolator characteristics.

Resistors R5 and R6 were needed to "speed-up" the operation of the HP isolator and may or may not be needed in other designs.

To date the circuit has been used in a 1200 baud line. An attempt to use the line at 2400 baud was not successful. The problem is believed to be in the PCM 12060 interface. In theory the line should be useable to 4800 baud. The maximum baud rate is line length dependent. Errors at high baud rates can preclude the use of that bit rate.
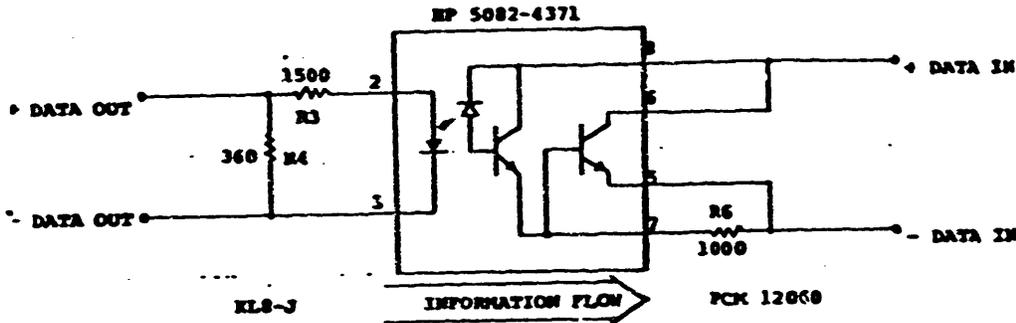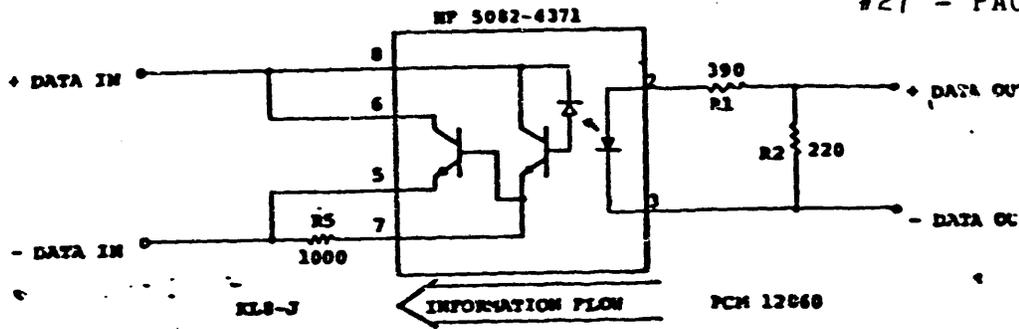
20 mA. CURRENT TO CURRENT ADAPTOR

Jack Donner of W.V.U. described an interface between an LSI-11 and a PDP-12 he has been working on. It consists essentially of a DRV11-P (?) and M1703 and M1705 devices. The details of how to get from these Omnibus Modules to the PDP-12 can be gotten from Jack.

John Alderman from Digital Communications Associates provided the following:
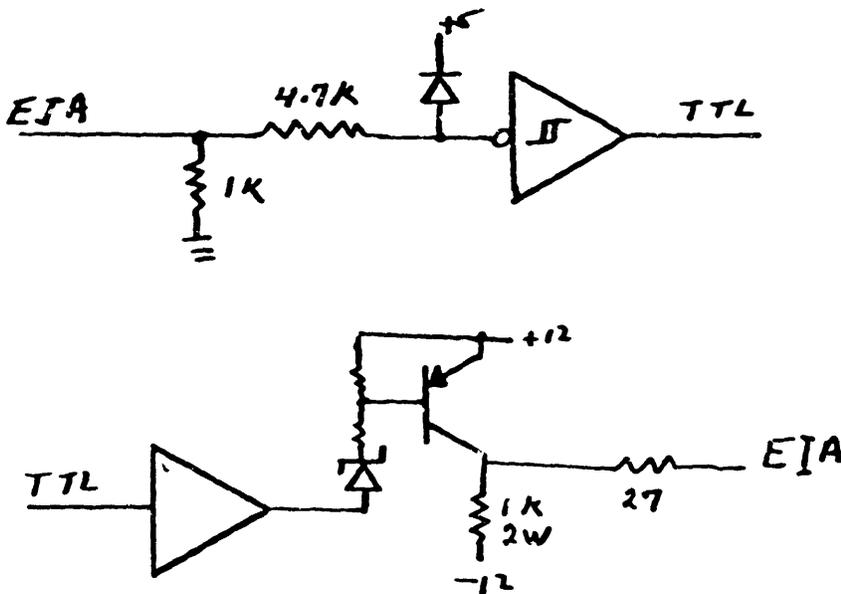
John also suggested that the 8T26 and 8T28 (tristate quad bus trans-cievers) were extremely useful. He also uses the National DM8136 as a device selector on the Omnibus. At this point considerable discussion on "Chips I Like To Use" ensued and some of the devices mentioned were the 8T37 hex receivers, an 8-bit bidirectional I/O port (latch and transciever) the AMD2905 which is a four bit slice with D flops, recieves, drivers, etc. Some people strongly suggested that the Signetics 8T10 shcu'd be avoided at all costs. The CMOS 4049 and 4050 usable as hex recievers though they have limited drive capability.

Lastly Don Harmer of Gerogia Tech presented the following method of expanding the unexpandable 8's.

An often overlooked method of providing bus expansion for PDP8s or PDP8M, F, or E is the use of the KA8E bus conversion module. This module was originally designed to provide for PDP8E "OMNIBUS" conversion to the positive (TTL) external bus of the PDP8/L, I or PDP-12 processors for compatibility with older perepherals. The module provides a measure of isolation from the "OMNIBUS" tm and, as such, prevents damage to the CPU modules by inadvertant wiring errors in external interfaces. The external bus may be utilized in a parallel bus fashion, similar to the "OMNIBUS" by wiring two or more H 803, H 8030 or H 863 module connector blocks in parallel such that "QUAD" or "Hex" modules can be accommodated. This permits the use of quad and wire-wrap modules for user interfacing. The H 033 frame on a home-built frame may be used for four slot configurations and the H 034 frame for nine-slot configurations, alternates include the H 933 C or H 934 CB with sockets.

External power supplies may be used; often only +5-Volts will be needed for many "home-built" interface modules.

This method of expansion may be indicated for 10, 12 or 20 slot 8a's where the DEC supplied interfaces can be accommodated in the main-frame and user-built interfaces are to be accommodated externally.

Interfacing to the external bus is simple and straight-forward, and the details are shown in the 8 E and 8 M, F "SMALL COMPUTER HANDBOOKS" (1973). Most standard AC transfer input-output operations can be accomplished on the external bus. Notable exceptions are: The 6XX0 instructions which is not allowed and

the operations involving the PROGRAM COUNTER, and the LINK which are allowed
on the "OMNIBUS".

The external bus is not bidirectional, that is, separate lines are provided
for AC OUT (BAC), ACIN, and the MB lines are supplied with MBn (1)H and
MBn(o)H for bits 3 thru 8. Interface timing and strobes are derived from the
IOP pulses, which are generated if MB 9, 10, or 11 are (1) in the IOT instruction.
Since no IOPAUSE or equivolent signal is generated on the external bus, the
interface must provide an AND between device address selected and IOP pulse to
uniquely identify an IOT for a specific device. Full decoding of the instruction
from MB 9, 1C, 11 may be accomplished however (with the above mentioned exception
of CXX0, for which no IOP is generated). Three slots are required for the ex-
ternal bus. In some cases, an external "OMNIBUS" may be built in a similar
fashion to permit the use of "HEX" modules with the 8 E, F, M which do not have
room for these modules. Note that some "hex" modules may be accomodated in an
8 F or M chassis by moving the fans outboard and that at lease one may be in-
serted immediately behind the front panel. This "trick" permits the use of
some later DEC modules on the quad-slot machines.

In any case, use of the external bus convertor KA8-E may have some useful
applications for the expansion of bus-slot limited or non-expandable PDP-8's.

All of the above contributors will be happy to supply additional in-
formation on their topics.
Bill Lennon actually said that the main problem with the fuse holder
on an 8/e was one of heat dissipation. The use of silicon heat sink com-
pound greatly reduced failures.

FROM DANIEL SMITH
EYE RESEARCH INSTITUTE
OF RETINA FOUNDATION

20 Staniford Street
Boston, Massachusetts 02114

#27 - PAGE 49

(617) 742-3140
24-FEB-78

BATCH

My patches and notes (#25:25-26) were for version 5, and I can't say yet whether any of them are applicable to the current version distributed with OS8 V3D. Thanks for the new /H option, DEC.

KL8E problems

The teletype handler distributed with V3D (at least on the DECtape system, AL-4711C-BA and AL-4712C-BA) is configured for the VT78. It omits features desirable for the ASR-33, such as expanding tabs into spaces, and the missing features cannot be reinstated via SET.

Everybody should list the beautifully documented file KL8E.PA if they are not familiar with the implementation of the teletype handler. There is a problem with the current version, however. One would expect that setting "VT78=0" would produce a standard non-VT78 handler. It is, however, also necessary to set "PAUS=0;" as distributed, merely setting VT78=0 includes options which take up too much space, and an assembly error is produced.

To summarize for inexperienced users: if you don't like the way your TTY: handler acts, edit KL8E.PA to set VT78=1 and PAUS=0; assemble via PAL8 to produce a new version of KL8E.BN; then .RUN SYS BUILD, $PR, $UPLOAD TTY, $LOAD KL8E, $IN TTY, $PR, $BO, .SAVE SYS BUILD. (Needless to say, all this should be done on a copy of the system--not the original!)

V3D documentation

My biggest gripe about V3D is that the handbook update is a) unusable, and b) incomplete. I realize the pressures DEC was working under. According to the MOBY MUNGER (TECO SIG newsletter) #1, there have been extensive changes to TECO, only a few of which are documented in the update. Many new features are not described (in many cases, they're not described in the MOBY MUNGER, either--frustration!) There seem to be some new features in CCL which are not documented. For example, it appears that if you have BATCH, it is now possible to type a string of CCL commands separated by semicolons (e.g. ".DATE;.DIR;.PAL PROG-LS;.LIST PROG.LS" all on one line, and have CCL turn it into a BATCH stream and run it.

Everybody should list CCL.PA. Although change note 66 warns that the format of the main table has been modified, the notes explaining the format (on the following page) have not been, and are presently incorrect. Apparently the change is that, instead of a fixed 3-word list of options to be or-ed with those given by the user, there is now an indefinite-length list of form (address, contents) terminated by 0. The only apparent use of this is in .DIR, where it permits CCL to give a default number of columns for the directory listing.

WISH LIST

Let me add my heartfelt agreement to Bryan Fredricks (#25:13), who wants a SET CONSOLE command which would do for CCL, CD, KBM, and ODT what SET CODE presently does for the TTY handler. We are going to encounter this problem in a few months, but have been postponing it. I would appreciate it if Bryan Fredricks or any other knowledgeable people could state:

1) are there known words in CCL, CD, KBM, ODT which are of form 603x
or 604x but are <u>not</u> terminal I/0? 2) Where and in what CUSPs is
terminal input performed directly (rather than by calling CD?) 3) Pre-
sumably all CUSPS that do terminal output are capable of doing the
output on the BATCH log under BATCH. Any known exceptions? And does
this lead to a an algorithmic method of patching terminal output on them?
Over the next few months I will be accumulating answers to some of these
questions, but would appreciate any clues (especially from those with
OS/8 source).

A FOTP wish list
     Particularly when it's system update time, there are options I
wish were in FOTP. (Some of the suggested functionality existed in
DIAL; why, oh why isn't progress monotonic?) 1) "Move;" i.e. after
a file has been successfully copied to an output device, delete it from
the input device. 2) "Existing;" copy a file only if the output filename
is already present on the output device; 3) "Superseding;" copy a file
only if the output filename is already present on the output device <u>and</u>
has an earlier date than the file being transfered; 4) "Additional;"
copy a file only if the output filename is <u>not</u> present on the output
device. (Note that Superseding and Additional might usefully be combined).
     Now that we have a really nice DIRECT thanks to Jim van Zee and
others, maybe it's time to start thinking about FOTP.

The ADS MICRO-8
     arrived just before New Years' and is presently alive and well
at the Retina Foundation. I will report more fully on it later, but
for now will just say that it seems to be a solid unit, has lived up
to our expectations (which were high), and seems to be well worth
considering for laboratory control applications where the available
person is a "software type." That is, it is easy to interface to and
test out in simple laboratory applications. On the other hand, if you
are looking for a sort of test bed and prototyping system and expect to
do extensive hardware development on your own, this may not be the
best system: the interface between ADS's cards and the laboratory is
well documented, but the interface between the cards and the CPU, i.e.
the bus structure, is not.

Has FORTRAN IV's load format changed?
     We are still using FORTRAN IV V3C, for reasons noted below.
Recently, I tried to used the version of FUTIL distributed with V3D
to patch some RALF code (SET MODE LOAD, OUTPUT FPP) in an overlay
load module, and I couldn't get it to work. I kept getting error
?40 at 06 BAD ADDRESS/OVERLAY (ODT COMMANDS), even though I knew my
addresses were good. FUTIL version 6.11 worked fine. I'm not going to
send in an SPR, but would like to know whether version 07A does, in fact,
work with FORTRAN V3D .LD modules. If it does not, there is a bug in
FUTIL. If it <u>does</u>, than the format of .LD files has changed, a fact
of some importance since it presumably implies that V3C .LD files might
not load correctly under FRTS V3D. If this is true, people updating to
V3D are required to re-LOAD all of their programs, and not just to fix
the date!

FOCAL vs. FORTRAN, round N

We now plan to shift completely from OS/8 FORTRAN IV to U/W-FOCAL.

This decision has not been made lightly. We have had FORTRAN IV for nearly three years, and have an investment of over $20CC in the initial license, updates, and sources. We also have an extensive programming investment in FORTRAN IV, including several thousand lines of RALF code, much of it concerned with real-time laboratory processing.

Many of the differences between FORTRAN and FOCAL as languages have already been aired in this newsletter. FOCAL's greater "friendliness" and ease of use is very important to us; on the other hand, with a PDP-12 we cannot get a mode B EAE and cannot afford an FPP, so we cannot take advantage of FORTRAN's potential speed advantage.

I wish here to emphasize some differences between OS/8 FORTRAN IV V3C and Jim van Zee's U/W-FOCAL v4A as products. The chief reason why we are making the transition is that FORTRAN IV, compared to FOCAL, is much less well integrated into the OS/8 operating environment. It is far, far more difficult to write and debug RALF SECT8 modules than it is to add new commands and functions to U/W-FOCAL. This is especially true of modules that must reference the system in any way. (I am still lost in admiration for Robert Phelps' courage in attempting to write USR and getting it to work.)

Furthermore, FOCAL is a far more fully debugged product than FORTRAN IV, and, at least at the moment, it is being more conscientiously maintained.

DEC's apparent policy, as nearly as I can determine by observing their responses to my SPR's, and from informal conversations with Ron Jansen, is to support FORTRAN IV only as a pure FORTRAN product. That is, they respond to problems that are encountered in pure FORTRAN programs, and (to a limited extent) to problems in FORTRAN library routines (i.e. to DEC-written RALF code). They do not, however, seem to see it as a responsibility to handle problems that are encountered only in user-written RALF code; for example, differences in behavior between the FPP simulator and the hardware FPP. Moreover, although this is less tangible, there still seems to be a feeling that users should not really expect to be able to use a cute little PDP-8 for big, hairy-chested work. That is, problems seen as encountered by "advanced users," e.g. the long-standing problems with arrays of more than about 1000 elements, problems resulting from failure of FORTRAN IV to implement the ANSI standard correctly, etc. get short shrift.

This policy, whatever its merits, ought to be more clearly stated and better publicized by DEC. What it means is that, unlike the majority of FORTRAN systems, OS/8 FORTRAN IV is not a good vehicle for projects that require integration of substantial amounts of assembly-language code into a high-level-language mainline. It is also not a good vehicle for programs that want to do file manipulation, look at directories, call the command decoder, etc. FORTRAN IV runs under OS/8, but at runtime OS/8 is not only not present, it is nearly inaccessible.

HINTS FROM HELOISE (in memoriam)

I'd like to summarize some miscellaneous products, services, and ways
of life which took me a while to learn about, falling under the general
category of "housekeeping" of "better living in the computer room."

1) Supply sources. First of all, let us not neglect DIGITAL. Be sure
you have a copy of their "Supplies and Accessories" catalog. They are
probably the worst place to get anything you can find anywhere else, but
where else can you get those cute blue vinyl tabs that make the DECtape
stick to the reel? And if you have spare rack-mount space, the #18-09825
DECtape racks are outrageously priced ($14 and made out of thin sheet
plastic), but work better than anything else I know of. Keep in mind that
DEC has a $35 minimum order, slow delivery, balks at returns of defective
merchandise, and louses up the orders frequently. In their favor, they
tend to keep their prices stable.

Two very interesting firms are Minicomputer Accessories, 130 S. Wolfe
Road, P. O. Box 9004, Sunnyvale, Calif. 94086, (408) 737-7777; and MISCO,
963 Holmdel Keyport Road, Box 399, Holmdel, N. J. 07733. These are, in
effect, retailers. They have put together attractive catalogs with a
large variety of interesting and useful supplies aimed at the minicomputer
market; for example, printer ribbons for minicomputer printers, teletypes,
and terminals; floppy disks, etc. MCA offers 25-pin EIA RS-232 _extension_
cords, and a variety of DEC cables and connectors (e.g. 8-pin Mate-N-Loks).
Haven't yet ordered from Misco; service from MCA has been fairly fast,
so apparently they really do stock their items. Whenever I've compared
MCA's prices to others, MCA's have been a bit on the high side; I would
judge that they earn their markup legitimately.

For binders, files, media cabinets, etc: TAB Products, 2690 Hanover
Street, Palo Alto, California 94304; United States Tabulating Binder
Company, 7207 Melvina Avenue, Niles, Illinois 60648, 312/647-9310;
Visible Computer Supply, 9855 Derby Lane, Westchester, Illinois 60153,
(312) 681-6080; Wright Line Inc., 160 Gold Star Boulevard, Worcester,
Massachusetts 01606 (617) 852-4300; Devoke Co., 3788 Fabian Way, P. O. Box
10011, Palo Alto, California 94303 (415) 494-8844 (also microfiche supplies);
Engineered Data Products, Inc., 37200 Amrhein Road, Livonia, Michigan 48150
(313) 261-9550 (cabinets). These are big, hairy-chested suppliers for the
macho raised-floor crowd, and it may be hard figuring out which is the
right binder from a catalog that includes hundreds of them--but it sure
beats trying to read unburst printout in a binder intended for burst printout!

2) VELstraps. The cabling in our laboratory is constantly changing, and
we did not want to 1) trip, 2) install a raised floor, or 3) have to get
into the walls or ceilings to make a change. Our lab _does_ have a dropped
ceiling; so what we have done is to buy VELCRO straps, wrap them around
the ceiling supports, and run our cables under the ceiling, held by the
VELstraps. Cables are easily inserted or removed, do not need to be
threaded or pulled, are out of the way, and (believe it or not) are held
very securely. It's a very nice system if you have our kind of dropped
ceiling. Velcro Corporation, 681 Fifth Avenue, New York, N. Y. 10022;
212 751-2144. Ask for their "comprehensive product guide" and the price
list for "Nylon Velstrap Fasteners." They're made to order in almost any
size, length, and color, and are fairly cheap (around 35¢ for a 3/4" by 14"
strap), but there is a minimum order problem and I don't know of any retailers

3) VECTOR patch panels. Vector Electronic Co., 12460 Gladstone Avenue,
Sylmar, California 91342, (213) 365-9661. This company has very nice
products for packaging circuitry and very bad catalogs and instruction
sheets. I want here to call attention to their "pre-programming patchboard
systems," rather like what you find in old IBM accounting machines,
except that they have a complete line of patchcords including 1- and 2-con-
ductor shielded patchcords, making the system suitable for analog signals.
Three labs use our PDP-12, and before we got the patchboards each user had
to connect or disconnect dozens of individual cables in various locations,
with resulting problems because user B would disconnect one or two of user
A's cables and user A would then run, not noticing that a couple of cables
were disconnected, while user C was very polite about restoring the previous
cabling when he left but did not always do so correctly, etc. Now each
user just puts in his own patchboard. On the whole, it works, and we havn't
had any obvious noise or connection problems. (Footnote: see also MAC/PANEL
Co., P. O. Box 5027, High Point, North Carolina 27262, (919) 882-8138)
4) Brookstone connectors. (There must be another name and another source
for these, but I don't know it.) S-4650 quick-connectors, Brookstone
Company, 125 Vose Farm Road, Peterborough. New Hampshire 03458. In our
lab, these is a constant need for miscellaneous connectors with varying
numbers of contacts. Maintaining a connector inventory is both difficult
and expensive. A particularly annoying situation occurs when there is a
need for a few new lines on an existing cable. The interesting property
of the Brookstone connectors is that the connectors (which each contain a
single contact, and are hermaphroditic) have a keying arrangement on their
sides so that connectors may be snapped together laterally to form multi-
connector assemblies. That is, if you have built a ten-conductor connector,
you can add on two more and make it a twelve-conductor connector. The
contacts themselves are large and crimp on; assembly is rapid and easy.
They are extremely handy, although they are too big, expensive (about 60¢
each), and are not gold plated. If anybody knows Brookstone's source for
these, or knows of any other product of this type, I would very much like
to hear about it.

5) Controlled circulation publications--also known as free magazines.
They are completely supported by advertising and distributed free to
qualified individuals. Value of the editorial matter varies, but the
advertising is often useful. In most cases, honest answers will get you
the publication; you don't have to say you're the president of the firm; as
long as it sounds like you influence purchases they want to send it to you.
The usual procedure for getting them is to search around your organization
for somebody who's getting them already, and look for tear-out subscription
application cards bound into them; but presumably writing to the circulation
office will also work. DATAMATION, however, requires you to send them a
request for an application form on company letterhead, and for people of my
humble rank there is a waiting list.
    DATAMATION, Circulation, 35 Mason Street, Greenwich, Connecticut 06830.
Oldest, biggest, fattest, most prestigious, only one with really significant
editorial material. This is also the one that keeps you up to date on the
latest scuttlebutt about the Bell bill, antitrust stuff, what IBM is doing,
etc. and, of course, that very interesting annual DP salary survey . . .
    MINI-MICRO SYSTEMS, 5 Kane Industrial Drive, Hudson, Massachusetts 01749.
(Used to be "Modern Data," how's that for a dumb title? "Antique Data and
Computer Collectibles," now that would be a grabber). Mostly valuable for
its advertising, but lately there has been an effort to provide systematic,
tutorial information on mini- and microcomputers and components.
    COMPUTER DESIGN, Circulation, P. O. Box "A", Winchester, Mass. 01890.
Mostly at the "chip" and "board" level. A very interesting column by
John E. Buckley about communications (modems, Ma Bell, etc.)

ELECTRONIC PRODUCTS, Circulation, 645 Stewart Avenue, Garden City, N. Y. 11530. There is a great deal of interest in here for anybody that does interfacing of laboratory equipment; for example, this publication put me on to

6) the Philips stepping motor chip. Stepping motors are very attractive devices for a multitude of jobs around the laboratory, but if you looked into them a few years ago you probably noticed that the drive electronics for them was expensive. Typically, you might need a $50 motor, a $100 drive card, and maybe a 100V, 1 amp. DC power supply. Now Philips has a $12.50 chip that goes with their line of small, 12V stepping motors (which cost about $25 each). The most annoying thing is that the chip wants 12V signals, not TTL. They have some 5V motors, so maybe someday they'll have a 5V chip, too. Our first application is for reinforcing monkeys; when they do the right thing, the computer rewards them with a measured quantity of Tang. We do it by driving a tubing pump (the kind that pinches a plastic hose with moving cams) directly from a stepping motor. It seems to work a lot better and more reliably than a solenoid valve.

7) Hanging file folders. This is so dumb I really hate to mention it, but it took me years of fumbling around in stacks of paper before I discovered how to use file cabinets and file folders, and then it took me five more years before I discovered how very, very much better the hanging folders (e.g. "Pendaflex") are than the regular kind.

THE ROCKEFELLER UNIVERSITY

1230 YORK AVENUE · NEW YORK, NEW YORK 10021

February 23, 1978

Mr. Robert Hassinger, Coordinator
12 Bit SIG
Care of DECUS
129 Parker Street, PK-3/E55
Maynard, Mass. 01754

Dear Mr. Hassinger:

Enclosed is some material on FORTRAN II which you may want to enclose in the 12 Bit SIG Newsletter. Since the Newsletter is an absolutely indispensable source of information for the computer work in our laboratory, it seemed like a good idea to contribute some of what we have learned for others to use.

In the most recent Newsletter (No. 26) you ask for comment on the idea of using microfiche for the Newsletter. Such a change would be at best inconvenient and at worst disasterous for us. Our computer is located several miles from the nearest potential microfiche facility, and perhaps 80 miles to the nearest facility which we could use when programming evenings or weekends. The sophisticated readers of the Newsletter may not use it in the same fashion as some of us who program only when necessary. For many users of the latter type, I think the Newsletter serves as a reference and consultant when we want to learn how to use a new piece of software or when we run across a problem. At such a time, I reach for my box of Newsletters in order to take advantage of the experience others have gained. I suspect that there are a lot of copies of the Newsletter around with homemade tables of contents and marginal notes to help this process of looking back over the old issues when a question comes up. I certainly do not want to purchase a microfiche reader as a computer peripheral. I would also bet that others who find the Newsletter valuable would be willing to be charged for the extra cost of a Newsletter on paper.

Sincerely,

Ronald P. Larkin
Assistant Professor

RPL/rbk

FORTRAN II: Use with PAL8 and with Interrupts

Since FORTRAN II is the DEC OS/8 language which makes use of the speed of the CPU in doing integer arithmetic and logical operations, it is a natural candidate for use when speed is required, especially in on-line and data acquisition applications. The associated assembly language, SABR, however is awkward and relocatable to a fault. As a result, one desires to wed PAL8 to FORTRAN II in order to take advantage of the direct access to the hardware afforded by the former and the I/O, arithmetic ease, and general convenience of the latter.

The result of wedding FORTRAN II and PAL8 can be very pleasing and convenient to use, but the process of learning how to do so is fraught with problems of scattered and incomplete DEC documentation, costly sources or listings, and esoterica which can waste hours. This contribution is designed to share some lessons learned in writing two large (20k, 200-500 FORTRAN lines) interrupt-driven data acquisition programs and several smaller ones. Comments and corrections are welcome, since I am not a programmer by trade.

In the following discussion, I shall use the acronym "FORTRAN" in two ways: in the usual meaning of the language and to include the whole subsystem designed to support FORTRAN II, namely LOADER, FORTRAN II, and SABR. So read "FORTRAN II and SABR" or "all .RL programs" when I say FORTRAN below. You should be acquainted with the OS/8 Handbook, the SABR-FORTRAN Software Support Manual, and the past issues of the 12-bit SIG Newsletter[*], and of course with the PAL8, SABR, AND FORTRAN II languages.

I. Preliminary Considerations

A. Memory use by LOADER
   1. It loads programs in the order you specify
   2. It starts at the bottom of Field 0 and proceeds upwards
   3. COMMON lives in Field 1 starting at 10200.
   4. It fills a whole page if even one location on that page is used.
   5. The bottom of Field 0 is occupied by resident parts of LOADER.
      If the /I and/or /O options are used, each takes up a chunk of memory above the resident LOADER. In current software, locations 00000-02377 can be taken up this way if (IO) is specified.
   6. See the appended Exhibit A for a chart of memory use by LOADER.

B. Miscellaneous cautions and foibles

-LOADER does not treat COMMON as occupied memory. If you CALL CHAIN, don't expect to preserve COMMON across core images unless you specifically save 10200+ or put PAL8 code there (see below).
-LOADER allows only 63 decimal entry points unless you use the version mentioned in a recent 12-bit SIG Newsletter
-FORTRAN instructions start with and leave the ACC clear.
-FORTRAN allows the Data Field (DF) to vary widely during execution. Watch out.

---

[*]Especially #18 p. 7, #22 p. 2-3, and #24 pp. 21 and 28.

-Some SABR pseudo-operators such as OPDEF cannot be used in FORTRAN II
 programs unless they appear as the very first statements, before
 FUNCTION or SUBROUTINE statements, for instance. This is an un-
documented peculiarity of the FORTR pseudo-operator.
--Using PAL8 with FORTRAN requires a song-and-dance described at the
 top of page 7-7 in the OS-8 Handbook. Note that BATCH cannot
 help here and that you must specify a starting address (Location
 00200 is in LOADER).

II. Using FORTRAN with PAL8

   A. Where to put the PAL8 code. Three options follow and the rationales.

   1. At the top of the top field of memory. This way, all you
have to watch for is overflow of the FORTRAN programs into the PAL8
area. Just check the LOADER memory map.

   2. In the first available space in Field 0. One "reserves"
space here by writing a dummy SABR program which uses the BLOCK
and PAGE pseudo-ops to take up at least enough space for the PAL8
code. The dummy "program" must be the first program specified by
you to the LOADER. Then the .BN file containing the PAL8 code can
be placed in this space using ABSLDR. The absolute location of the
"reserved" space will be apparent from the LOADER memory map if
you include an entry or two in the dummy "program". The location
will vary with use of /I and /O as described above. The FORTRAN
programs can then be expanded to the limit without worry, but the
PAL8 code will be more difficult if you have reserved insufficient
space for expansion.

   3. In the COMMON storage area in Field 1. If you know how big
a COMMON area you need, you can expand it by 128 words for each page
of PAL8 code you have and start PAL8 code where the genuine data end.
For instance: COMMON X,Y,Z,ETC
                   COMMON JUNK
                   DIMENSION JUNK(512), X(5)

...will save four pages starting at location 10400. (Or 4+ starting
after ETC if you want to be miserly.) FORTRAN programs will be loaded
in any vacant space above the COMMON block.

   In general, one has to worry about access to Field 0 for interrupts
and possibly handlers, to Field 1 for COMMON (see below) and to Page
0 in whatever Field one chooses to put PAL8 in. Note that certain
locations in Page 0 of each field are used by FORTRAN in every Field.

   B. How to communicate between PAL8 and FORTRAN.

For the beginning user of PAL8 with FORTRAN II, and for everyone who
wants to play it safe, there are two rules:

   Rule 1: Call FORTRAN routines from PAL8, not the reverse.

   Write a SABR program to be loaded first in the LOADER program
string; it can contain PAGE and BLOCK specifications to reserve
space for PAL8 code as in option 2 above, if you like. The SABR

program will have the purpose of calling relocatable FORTRAN programs
from fixed positions in memory. (Since the LOADER always takes
the string of input programs in order, the SABR program in question
will always be the first program in Field 0.) An example such
program, "A2400.SB" is listed in Exhibit B, along with PAL8 code to
call the FORTRAN subroutines in this way.

Note that LOADER requires an entry named MAIN in the programs
it loads. It conventionally starts a core image at MAIN once the
relocatable files are in memory. To have a fixed address at which
to start a core image is handy; otherwise, one needs to find out
where the starting address really is every time the song-and-dance
described on page 7-7 of the OS/8 Handbook is performed. Also,
I/O is usually initialized in FORTRAN by the subroutine OPEN, called
in the program MAIN when a FORTRAN main program is used. OPEN or
a similar set of code is needed to initialize FORTRAN I/O flags
lest a device like the console device cause a hang on the first
data transfer statement. These considerations are all handled
by the entry MAIN which appears in the example program "A2400.SB".
Find out where MAIN is as an absolute address and specify this
address to ABSLDR or in a SAVE command to OS/8 keyboard monitor.
MAIN calls OPEN and immediately puts PAL8 in control. It must
have the starting address of PAL8 code in order to work.

Rule 2: Pass information back and forth via COMMON where possible.

This way of communicating between PAL8 and FORTRAN has a way
of doggedly stereotyping and documenting information transfers in a
way that is easier to debug than arguments or other more fancy means.
Lay out COMMON locations in PAL8, filling with zeroes or data. (In
a way, this is a great BLOCK DATA-type arrangement in FORTRAN II.)
Real variables need three locations.

If you wish to depart from Rule 1, calling PAL8 from FORTRAN
(OS/8 Handbook, page 4-60), watch your data field, instruction field,
and indirect references.

Two situations arise where Rule 2 may not be sufficient. First,
fixed locations in memory may need to be referenced by FORTRAN as in-
line SABR statements. To do this, use the implicit CDF in a DUMMY
statement rather than an explicit CDF. For instance, to store a
quotient in location 02600:

```
                        IQUOT = A/B
        S       JMP STOR        /BYPASS THE DUMMY
        S       DUMMY J
        SJ,     6201            /CDF 0
        S       2600
        SSTOR,TAD \IQUOT        /A FORTRAN ADDRESS
        S       DCA I J         /ACC WILL BE LEFT 0
```

Second, it may be more convenient to pass arguments than to use
COMMON. A method for passing an argument (an array first-word-address)
to FORTRAN is shown in Exhibit B as entry GOSUB.

III. Running FORTRAN II with interrupts on.

   A. There is a problem with file-oriented I/O.

   Locations 00000-00003 are used by routines OOPEN, OCLOSE, and
IOPEN. These four locations are used only during calls to the three
entries, not during actual ASCII I/O by a READ or WRITE. So there
are two ways to handle the obvious confilct with interrupts:

1). If the interrupt system can be off during opening and closing
files, then one can simply move instructions into locations 00001-
00003 in this way (in PAL8):

```
CDF 0
CLA
TAD (CDF CIF PALF      /WHERE PALF IS THE
DCA I (1               /FIELD WHERE PAL8 LIVES:
TAD (JMP I Z 3         /10, 20, ....
DCA I (2               /THESE MUST BE INDIRECT!
TAD (INT               /ADDRESS OF INTERRUPT-
DCA I (3               /HANDLING CODE IN PAL8
```

The above should be done before issuing an ION. An IOF must be
executed before calling any of the three offending subroutines.

2) If the interrupt system needs to be on during the opening and
closing of files (e.g. keeping time), then solution 1) is useless.
One needs to modify OOPEN, OCLOSE, IOPEN, and LOADER to use some
other locations for the transfer of the file name and extension from
relocatable code to LOADER. I have done this to my system and will be
willing to share the changes with anyone having a source of IOPEN.SB.
If there is demand, it will be submitted to DECUS.

   B. FORTRAN can run as interrupting code or as interruptable code,
but not both.

   Running .RL files in both "foreground" and "background" is im-
possible since all subroutine calls (including array references,
mathematical operations, and argument transfers) are handled by
one nonreentrant set of routines (See OS/8 Handbook, page 7-6).

   C. Under interrupt, FORTRAN I/O devices must be asynchronous.

   Providing that the device is asynchronous, FORTRAN I/O will
proceed normally when interrupted by a PAL8 routine. (The TD8-E is
not asynchronous.   RTAPE and WTAPE should work.   I don't know.)

   D. FORTRAN II I/O needs its flags to be initialized correctly.

   For instance, by OPEN, as discussed above under II, Rule 2.

   E. LOADER uses the USR for file opening and closing and for
chaining. Therefore, PAL8 programs which interrupt FORTRAN during
file operations may not have COMMON in memory in 10000-11777, but
rather the USR. Thus interrupts should not be enabled during file
opening and closing if the interrupting routines use this area
of memory.

THE ROCKEFELLER UNIVERSITY

1230 YORK AVENUE · NEW YORK, NEW YORK 10021

Exhibit A

RESERVED AND DISTURBED LOCATIONS USING FORTRAN II

All Fields	Locations	Reference*	Occupant (O) or Disturber (D)
	7	SABR 2-60	D:a scratch address
	10	"	D:OS/8
	20-32	"	O:Library Routines in their Field
	33-124	"	O:Device Independent I/O or CHAIN
Field 0	0-3	IOPEN listng	D:IOPEN unless modified
	176-177	SABR 2-60	D:I/O Handlers' temporary storage
	200-777	SABR 2-59-60	O:library linkage routines
	1000-1577	LOADER lstng	O,D: Either /I or /O but nct both
	1600-2377	"	O,D: Both /I and /O

free locations: 4-6,11-17   ,125-175,2400-7577 if /I/O used

Field 1	10020-10022	SABR 2-60	D:Floating point ACC
	10200-...	OS/8 Handb.	O,D:COMMON
	10000-11777	OS/8 2-2	D: USR
	17600-17777	OS/8 B-2	O: OS/8

free locations: 0-6,11-17,125-177,end of COMMON-17577

Field 2	27600-27773	OS/8 B-3	D:TD8E if present
Top Field	X7774-X7777	OS/8 B-3	O:BATCH

---

*SABR = 8k FORTRAN SABR assembler handbook, 1973
OS/8 = OS/8 Sortware Support Manual, 1974
OS/8 Handb. = OS/8 Handbook 1974

Exhibit B

Example Program to Communicate between PAL8 and FORTRAN

```
GOINI=1000      /PAL8 CODE TO CALL FORTRAN SUBROUTINES
GOSUB=GOINI+6   /(NEITHER /I NOR /O IS USED BY LOADER)

        CLA             /CALL FORTRAN SUBROUTINE "INIT.FT"
        CDF CIF 0       /VIA "A2400.SB" (SEE BELOW)
        JMS I (GOINI    /NOTE:   "A2400.SB" STARTS IN 01000

        TAD (CDF 1      /SET UP CDF TO PAL8'S FIELD
        DCA I (GOSUB+3  /PUT IN ARGUMENT
        TAD (ADDR       /AN ADDRESS IN PAL8'S FIELD
        DCA I (GOSUB+4  /PUT IN ARGUMENT+1
        JMS I GOSUB     /CALL FORTRAN "SUB.FT"
```

```
/ FILE:   A2400.SB   DEMO. OF PAL8-SABR COMMUNICATION

/ THIS ROUTINE MUST BE LOADED FIRST, INTO FIELD 0

OPDEF JMPI        5400
ABSYM FIELD       6213        /CDF CIF 1, WHERE PAL8 LIVES

GOINI,   HLT                  /ENTRY FROM PAL8
         CALL 0,INIT          /SUBROUTINE "INIT.FT" IS CALLED
         CLA                  /THIS "CLA" IS A SAFE CONVENTION
         FIELD
         JMPI GOINI           /RETURN TO PAL8 IN FIELD 1

GOSUB,   HLT                  /ENTRY FROM PAL8
         CALL 1,SUB           /SUBROUTINE "SUB.FT" WITH 1 ARG.
         ARG GOSUB            /USE ANY ADDRESS AS ARG. HERE
         CLA
         FIELD
         JMPI GOSUB

         ENTRY MAIN
MAIN,    NOP                  /AN ENTRY TO GIVE THE LOADER
         NOP                  /A STARTING PLACE AND SET UP
         CALL 0,OPEN          /I/O FLAGS, ETC.
         CLA
         FIELD
         JMP I (2400          /STARTING ADDRESS OF PAL8 CODE

         END
```

Earl T. Ellis Jr.
(203) 445-8501 x 296

USCG R&D Center
Avery Point
Groton, CT 06340

Dear Bob,

Since FUTIL is being distributed with OS/8 V3D and ETOS V5, it
is important that everyone learn how to use it. I have typed up 2 runs

1) Install SPR 8-2455 to CREF.sv as distributed in OS/8 V3D. This fixes
a well known bug. Everyone with V3D must fix CREF..

```
.R FUTIL

FILE CREF
CREF.SV     1675-1711  0015  (0013)  3.005  05-JUL-77
SET MODE SAVE
6063/ 2022  5270
6070/ 2025  2022                              LINEFEED
0000.6071/ 5252  5314
6114/ 0000  1363                              LINEFEED
0000.6115/ 0000  1025                         LINEFEED
0000.6116/ 0000  3025                         LINEFEED
0000.6117/ 0000  3425                         LINEFEED
0000.6120/ 0000  2025                         LINEFEED
0000.6121/ 0000  5252
2576/ 0301  302                               CHANGE VERSION
WRITE                            ....DON'T FORGET.....
EXIT              OR NEXT FILE OR CTRL C
```

.

The information FUTIL prints out between the FILE Name and the FILE
Date will vary according to the location of the file. All lines are
terminated with a RETURN unless otherwise noted.

2) Install Rev. Chases DIRECT patch under ETOS. FUTIL is a little differ-
ent under ETOS.

```
^VS                                  CTRL V, S
!PR 4040
!WE 0
!BOOT


.R FUTIL
FILE DIRECT                              I have Version 6...
DIRECT.SV  0165-0173  0007  (0007)  1.051  11-FEB-78
SET MODE SAVE
14000/ 0000                                   LINEFEED
0000.14001/ 3220  6046                        LINEFEED
0000.14002/ 4224  7200                        LINEFEED
0000.14003/ 0217  5600                        LINEFEED
0000.14004/ 3260
WRITE                         ....DON'T FORGET...
^VS              OR NEXT FILE OR EXIT (NOT BEST)
!WL 0                         WRITE LOCK ZERO
!BOOT
```

PATCH TO 'TECO', Vers. 5
PAS Sat. 11-Feb. 1978
Stand-alone OS-8 V. 3D

PURPOSE: To get rid of 'L'O'W'E'R 'C'A'S'E flagging.

.GET SYS TECO
--- --- ----
.ODT

02247/7000   5216   <C.R.>
------   ----   ----
^C
_
.SA SYS TECO
-- --- ----

Note: 5216 is the code immediately preceding code 7000.
       In the version shown, 02246 held 5216. The game of the
       name is to change 7000 to what immediately precedes it.

Is there some undocumented .SET for this, or what ?

----   underlines mark user's typing.

Dear Bob:

    Don't know if others have run across this problem or not.

Anyhow, here it is.

                    Yours,

                    Geb ·

                    (Rev. Dom) Geoffrey Chase OSB
                    Abbey / Portsmouth RI 02871

PORTSMOUTH ABBEY SCHOOL
PORTSMOUTH, R.I. 02871
(401) 683-2000

09-FEB-1977

Dear Bob:

Two more Band-Aids for OS/8 version 3-C & earlier.

1.   PAL-8:

   a.   On page 0 there is a constant 5070. Depending on
           version, it seems to live in location 00146 or 00147.

   b.   Location 01731 contains 1130, equivalently a TAD [260 .

   c.   Patch (.ODT) 01731 from old 1130 to new 1146 or 1147.
           This adds 8 ('5070' -- the upper byte doesn't count)
           to your base year. Should work through '79.

2.   MACREL, early release -X6C:

   a.   This is rather like patching CCL, from which it seems
           to have borrowed a few ideas.

   b.   (.ODT) 07365 from old 0106 to new 116 (base year 78)
                   07373 from old 0003 to new   6 (weekday offset)

                    Yours faithfully,

                    G.Chase osb

              Rev. Dom Geoffrey Chase

The following is refered to
at the bottom of page 13.   **DATE.FT**


```
C        OS/8 DATE , FEB 1978

C        FORMAT :  DD. MM. YY
C
         IBASEY=78

         COMMON IDA1, IDA2, IDA3, IDEXT

300      READ(1,302) IDA2, DUM, IDA1, DUM, IDA3
302      FORMAT('DATUM:  ', I2, A1, I2, A1, I2)
         IF (31-IDA2) 300,303,303
303      IF (12-IDA1) 300,304,304
304      IDA3=IDA3-IBASEY
         IF (IDA3) 300,34,34
34       IDEXT=(IDA3+8)/8

         CALL SETDA

         END
```

SETDAT.SB

```
        /  PROGRAM SETS UP OS/8 DATE
        /S. OS/8 SOFTW. SUPP. MAN. D-4
        /
IDA1,   COMMN 1    /MO
IDA2,   COMMN 1    /DAY
IDA3,   COMMN 1    /YR
IDEXT,  COMMN 1              /YR EXTENSION, OS8 V3D 1978
/
        ENTRY SETDA
        DUMMY DATE
        DUMMY DATEXT
SETDA,  BLOCK 2
        CLA CLL
        TAD IDA1
        AND (17
        CLL RAR; RTR; RTR
        DCA TEMP
        TAD IDA2
        AND (37
        CLL RAL; RTL
        TAD TEMP
        DCA TEMP
        TAD IDA3
        AND (7
        TAD TEMP
        DCA I DATE
/
        TAD IDEXT
        AND (3
        CLL RTR; RTR; RTR
        DCA IDEXT
        TAD I DATEXT
        AND (7177
        TAD IDEXT
        DCA I DATEXT
/
        RETRN SETDA
        CPAGE 4
DATE,   6211
        7666
DATEXJ, 6201
        7777
TEMP,   0
        END
```

```
1)066          JMP NODAT1
1)             DCA DATEM
****
2)066          JMP NODATE
2)             DCA DATEM


        ********

1)066               CLA STL IAC RAL /6
1)             TAD TM2
1)             TAD BASEYR
1)             AND (60
1)             SNA CLA
1)             TAD (66
1)             TAD TM2
1)             TAD B^SEYR
1)             DCA YEAR
****
2)066          TAD TM2
2,             TAD (6760
2)             DCA YEAR


        ********

1)066               TAD (6          /^78
1)             TAD DATEM
****
2)066          TAD (3
2)             TAD DATEM


        ********

        ********

1)066               JMP I DATE
1)      /
1)      NODAT1, JMS I (LOOK      /LOOKUP SYS:DATE.SV
1)              YDATE
1)              JMP NODATE       /MESSAGE IF IT'S NOT THERE
1)              JMP I (CHAIN     /CHAIN TO IT, IF IT'S THERE
1)      BASEYR, 6770             /^78
1)067   NODATE, JMS I (PRMESG
****
2)066           JMS I (LOOK      /LOOKUP SYS:DATE.SV
2)              YDATE
2)              JMP I DATE       /DO NOTHING IF IT'S NOT THERE
2)              JMP I (CHAIN     /CHAIN TO IT, IF IT'S THERE
2)067   NODATE, JMS I (PRMESG
```

FEBRUARY 15, 1978

MR. ROBERT HASSINGER
LIBERTY MUTUAL RESEARCH CENTER
71 FRANKLAND ROAD
HOPKINTON, MASS. 01748

DEAR ROBERT:

I AM WRITING AGAIN IN REGARD TO UPDATING THE PATCH FROM STEVE
LIGETT, DECUS NEWSLETTER OF APRIL 1976, NUMBER 16.
I HAVE DETERMINED THAT THE FOLLOWING PATCH TO BRTS V5A WILL
ADJUST THE COLUMN AND CHARACTER COUNTERS.

.GET SYS:BRTS

.ODT

```
2570/7774   7770        / 1 LESS THAN THE NUMBER OF COLUMNS
2573/7660   7600        / PAGE WIDTH
3375/7660   7600        / PAGE WIDTH
```

^C
SAVE SYS:BRTS

THIS SETS UP BRTS TO USE 9 COLUMNS, 128 CHARACTERS PER RECORD
(LOCATIONS 2570 AND 2573,3375). IT APPEARS THAT MR. LIGETT'S
PATCH TO BRTS HAS BEEN INCORPORATED IN THE NEW VERSION FOR
OS/8 3D.

I WOULD ALSO LIKE TO REQUEST THAT YOU ADD MY NAME TO THE MAILING
LIST FOR THE DECUS NEWSLETTER. THANK YOU.

                        SINCERELY
                        GREG WALTER

                        V.A. CTR. RESEARCH SERVICE
                        5000 W. NATIONAL AVE.
                        WOOD, WIS. 53193

<u>Concerning DIRECT, Headers, Labels, Dates and DECSYSTEM-8/78</u>

We have been revising some of the CUSP's for DECSYSTEM-8 to OS/8 V3D
standards. Two of these should be of general interest.

Jim Van Zee's version of DIRECT (V5G) has been revised, commented, and augmented
to aid in the "date" problem (creating V6A). Concerning the dates as now printed
by DIRECT: the fix for the OS/8 date problem, (only three bits in the date word),
was to add two bits in word 07777 (bits 3-4) as the extended date for the
.da command. The directory date word does not have these bits available. The
solution was to use the current system year as the base and print dates back
eight years. (e.g. in 1978 1970 dates show as 78, etc.) As an attempt to aid in
identifying old files the header prints the date it was created by(PARAM.FC or
HEADER.SV) The date in the header is printed as an absolute date. One may use
this date as an aid in identifying "old" files, if old files are kept together
on the same media. As an additional aid the /G option has been added. This
option prin ts directory file dates relative to the header date rather than
the system date. Currently in DIRECT V6A an offset is used so that these
relative to header dates are + or - 4 years. The sort options, /D and /T use
the appropriate relative date, either system or header if /G is selected. The
sort has also been modified so that undated files are the least timely. If
the reference date is 0, then no dates are printed. The header option (/H) now
prints "UNLABLED" if no header block exists.

The endless directory "bug" has been fixed in this version. An added option,
the /L option, prints the header only for identification and creation of labels
for the media (eg. DEC tapes, Floppies, etc). System I.D.'s have been added
for other systems at the suggestion of Earl T. Ellis of the coast guard.

The parameter block creation program (PARAM.SV) of DECSYSTEM-8 has been
extensively revised to include all DEC standard devices to date.

It has also been revised to fit the extended date algorithm. The creation
date word is in location 112 of the header, extended date in location 113. A true
"reference gear" is to be added to location 111 currently direct 6A optionally
uses the creation date as the reference date.

An additional option has been added to this program to permit creating header
blocks with the same label and only volume number's changed sequentially. The
name has been changed from PARAM to HEADER to avoid conflict with the RTS/8
cusp PARAM.SV, for DECSYSTEM-8/78. A summary of the options for direct is:

Direct Version 6A
Summary of the Command Decoder Options:

/A   Alphabetize directory before processing
/B   Print starting block numbers (octal)
/C   List only files with the current date
/D   Order files by date, then alphabetically
/E   Include empties in the Listing
/F   Fast mode (no lengths or dates)
/G   Use the header date as reference
/H   List contents of the Parameter block (header)
/I   Print additional information words (octal)
/L   Label, print header only
/M   List only the empties
/N   Print dates in numeric form (MM/DD/YY)
/O   List only files with other than today's date
/P   Eliminate Paging(No formfeeds in the output
/R   List Remainder of files after the first one
/T   List the most timely (recent) files first

```
/U    Treat each input specification separately
/V    List files not of the form specified
/W    Print the version number
/X    Alphabetize by Extension, then by name
=N    Use N columns - output is ordered by column
*     Wild name or extnesion
?     Wild character
```

```
Default input specification:     *.*
Default output device:           TTY:
Default output file extension:   .DI
```

The CCL for DECSYSTEM-8 was revised for)S8 V3C by D.E. Wrege using the preliminary V3D version. It will work with V3D (DS8CCL.PA) of 8-April-78) The major revision however was to provide a simpler means of updating it as the DEC version is changed, and to provide a continued means of adding user features via an "overlay" structure. The "true" V3D version will be available shortly.

In the current version any program can be run from SYS by just typing the name: eg: .R HEADER or .HEADER The safe SQUASH, ZERO, etc. and other features are retained.

Note that HEADER and DIRECT are "stand-alone" and will work with any version of OS/8, PS/8 DECSYSTEM-8, etc.

These updated programs will be submitted to DECUS and are available from:

> Don S. Harmer
> Schools of Physics and NE
> Small Computer Applications Lab
> Georgia Tech
> Atlanta, GA   30332

A presentation on the status of DECSYSTEM-8/78 is planned for the Spring DECUS meeting. Some discussions for standardizing the use of the HEADER block should be made. Also for solving the loss of the header if the directory extends into block 6. We have some proposals in this area.

> Submitted by
> Don S. Harmer and William H. Topliss
> Georgia Tech

FROM BRIAN CONVERSE

(See related material elsewere in this issue.)

        In reference to the FPP-12, I have done more tinkering with
it to see if it can be made to respond rationally.   One of the
nice things about the PDP-12 over most minis I've seen is that you
can really zip a program in there and play with it via the console
switches- much faster than you can on machines with a single input
register (especially since I always make mistakes entering things
into 8s and 11s and spend hours flipping switches).   The ability
to play with instructions via the "DO" switch has been of immea-
surable help sorting out shifts and testing wild ideas.   The
FPP-12 has absolutely no console and simply sits down in there
somewhere and laughs at you.   Testing out a wild idea means a
session with a text editor and a run of FLAP, a load and a start,
which means that one is less likely to bother to see if extention
code 13 of special format 3 does in fact do an undocumented base
16 logarithm of the FAC in floating point mode.   I've also glanced
through the hardware section of the FPP-12 manual and have been
struck by how similar the architecture shown in Fig 3-5 is to that
of a "bit-slice" microprocessor like the 6701 or 2900.   I'm not
sure if this is the tack taken on the PDP-8A's FPP or not, but it
looks like you could throw together one very simply because all
the logical complexity comes when talking to the PDP-8 I/O bus;
everything else in Fig 3-5 is now available on a "bit slice" chip
(you'd need 7 of them to do 28 bits).   If anyone should try this,
please put in a console! The advantage of the "bit slice" archi-
tecture is that one can tinker with the instruction set to add new
features simply by changing microcode.   The disadvantage would be
a new set of parts to stock and some maintenance software that
would need rewriting (or long hours devising microcode to emulate
FPP-12 signals).   With additional instructions, FLAP (and RALF)
might need work, but FLAP needs work already: Has anybody been
able to use FLAP's "REPEAT N" pseudo-op? If so, what's the trick?
FLAP seems to look at it, but won't do anything about it.
        There was some discussion of the FPP/FLAP SUPPORT LIBRARY in
#16, page 5.   First was mention of a RALF assembly with the FLAP
conditional enabled, resulting in a more modern version of FLAP.
Where does one obtain this "new" FLAP, failing access to FORTRAN IV
sources? Further was mentioned the suspicious resemblance between
the FPP SUPPORT LIBRARY and similar functions in the FIV library.
Could someone with FORTRAN IV sources and the FPP SUPPORT LIBRARY
document (which includes full listings) check this out, especially
with regard to those FIV functions that have been SPR'ed in the
last few years as being inaccurate or incorrect?
        I discussed FPP-12 I/O and memory use with Bob Massinger
some, and got the suggestion that much PDP-12 signal averaging
could be done in FPP fixed-point mode, reducing memory needs by
33% over floating-point mode as well as running faster.   It's not
quite that simple in fixed-point mode because everything is a

fraction and integer results and operands are verboten. Dividing
by N, for example, to calculate an average: N better well be a
fraction and the quotient better not be an integer or there is
trouble. This leads to some creativity in expressing N with the
benefit that it is possible to generate, in the limited case (such
as I'm working with) of 10-bit ADC samples and some other applica-
tions, a quotient which is essentially "fixed" for you without the
esoterica of ATX et al.
      The other hassle is feeding single-word values (e.G. ADC sam-
ples) to and from FPP fixed and floating values. If anybody ever
came up with a way of doing this not involving gallopng the FPP
index registers across the single-word data (via SETX) and XTA-ing
8 values at a time, I never found it. I think i've found a way
which may be faster. It involves picking up pairs of single-word
values in FPP fixed-point mode and using ALN to scoot the 12 most
significant bits into the 2nd word of the FAC "mantissa". This is
all that's needed for fixed point averaging, so I'm not sure what
to do next for floating point applications, perhaps FNORM. Since
the FPP-12 will effectively index by 2, two fixed-point FLDAs are
used with the same index register where the second FLDA uses the
buffer address+1:

```
          START0
          LDX -1000,0      /COUNTS 512 DOUBLE XFERS
          LDX -1,1         /1-WORD PICKUP PTR
          LDX -1,2         /FIXED-PT OUTPUT PTR
          LDX 14,3         /ALN CONST.(=DEC. 12)
LOOPLD,   FLDA BUFF,1+     /PICK UP 1ST 2 1-WDS
          ALN 3            /JUSTIFY 1ST, KILL 2ND
          FSTA FIXEDP,2+   /SAVE AS 2WD FIXED-PT
          FLDA BUFF+1,1    /NOTE OFFSET, NO INDEX
          ALN 3            /JUSTIFY AND KILL AS B4
          FSTA FIXEDP,2+   /SAVE 2ND VALUE AS B4
          JXN LOOPLD,0+    /DO 512 PASSES
```

      In the example, single-word values are brought from BUFF,
made into fixed-point FPP values via ALN, and stored as 2-word
fixed-point values in FIXEDP. The key to the example, aside from
the ALN, is the offset address BUFF+1 used in the 2nd FLDA.

2

```
FLAP V 50    FEB  3, 70 PAGE 1
             /ANOTHER FPP TEST PGM - BRIAN CONVERSE
                    ORG 400          /FITS IN WITH LINC-MODE TEST PGM ELSEWHERE
00400 0006          STARTD           /MUST DO IN 2-WORD INDEXING=FIXED POINT
00401 0101          LDX -1,1         /PICKUP POINTER FOR SINGLE-WORD VALUES
C0402 7777
00403 0102          LDX 14,2         /ALN INSTRUCTION CONSTANT=12 DECIMAL
00404 0014
CC405 0103          LDX -1,3         /OUTPUT POINTER
00406 7777
00407 0510          FLDA BUFF,1+     /GET THE "0001" AND "7776"
00410 0470
00411 0012          ALN 2            /"ASR" THEM UNTIL 0001 IN LSW
                                     /(NOTE FINE DIAGRAMS IN FPP-12 MANUAL
                                     /THAT SHOW HOW "ALN" WORKS SO YOU
                                     /WON'T HAVE TO RUN TEST PROGRAMS
                                     /TO DOPE IT OUT)
00412 6530          FSTA FIRST,3+    /SAVE AS PROOF
00413 0530
00414 0410          FLDA BUFF+1,1    /NOTE FANCY INDEXING NOW GETS
00415 0471
00416 0012          ALN 2            /2ND AND 3RD WORDS AND WE "ASR"
00417 6530          FSTA FIRST,3+    /TO 2ND IN LSW OF FAC "MANTISSA"
00420 0530
00421 0510          FLDA BUFF,1+     /NOW WE GET 3RD AND 4TH WORDS
00422 0470
00423 0012          ALN 2            /ETC., ETC...
00424 6530          FSTA FIRST,3+
00425 0530
00426 0410          FLDA BUFF+1,1
00427 0471
00430 0012          ALN 2
00431 6530          FSTA FIRST,3+
00432 0530
00433 0104          LDX -1,4         /JUST TOO WIERD, THIS INDEXING BIT
00434 7777
                                     /CAN'T JUST SWITCH MODES IN THE MIDDLE
                                     /OF A LIST, GOTTA USE ANOTHER INDEX
                                     /OR RESET CURRENT ONE OR USE ADDRESS
                                     /OFFSET.
00435 0510          FLDA BUFF,1+     /FLOATING-POINT SETUP TRY
00436 0470
00437 0012          ALN 2
00440 6530          FSTA FIRST,3+    /TO VERIFY FIXED-POINTNESS
0C441 0530
00442 0005          STARTF           /HELPS IF U WANT TO DO FL. PT.
00443 0004          FNORM            /THE KID GOES FOR BROKE
00444 6540          FSTA SECOND,4+
00445 0630
00446 0006          STARTD           /(GETS LIKE LINC/PDP MODE...)
00447 0510          FLDA BUFF+1,1+   /FL PT TRY ON POSITIVE VALUE
00450 0471
00451 0012          ALN 2
00452 6530          FSTA FIRST,3+
```

```
00453 0530
00454 0005          STARTF                /TOO BAD THERE ISN'T A "STARTS"
00455 0004          FNORM                 /INSTRUCTION: SINGLE WORD VALUES
00456 6540          FSTA SECOND,4+        /AND INTEGER MATH
00457 0630
00460 0000          FEXIT
00461 0000          FEXIT
00462 0000          FEXIT
00463 0000          FEXIT


                    ORG 470
00470 0001  BUFF,   0001                  /A TEST LIST OF FAKE SINGLE-WORD
00471 7776          7776                  /VALUES WE WANT TO FLOAT
00472 1234          1234                  /ALL 7 MAY NOT GET USED IN ABOVE
00473 5670          5670                  /CODE. PDP-12 ADC VALUES MUST BE
00474 5252          5252                  /CONVERTED TO 2'S COMPLEMENT OR
00475 2525          2525                  /CONVERTED (VIA JLT, ETC.) IN THE
00476 3567          3567                  /FPP CODE AS VALUES ARE LOADED
00477 0110          0110


                    ORG 530
00520 0000  FIRST,  0                     /THIS IS WHERE FIXED POINT FPP
00531 0000          0                     /INTEGERS APPEAR; APPROPRIATELY
00532 0000          0                     /STUFFED WITH ZEROS UNTIL VALUES
00533 0000          0                     /STORED BY FPP
00534 0000          ZBLOCK 40


                    ORG 630
00630 0000  SECOND, 0                     /WHERE OUR 3-WORD FL. PT.
00631 0000          0                     /VALUES SHOULD APPEAR
00632 0000          ZBLOCK 40

NO ERRORS
3 SYMBOLS, NO LINKS

  BUFF    00470   FIRST  00530   SECOND 00630
```

630/7764

00631 /5252
00632 /0000
00633 /7744
00634 /2200
00635 /0000

— floating point values (o.k. they need work- remember, ALN can run backward and should pull in zeros @ LSB - that should help)

..OD   results

530/0000
00531 /0001
00532 /7777
00533 /7776
00534 /0000
00535 /1234
00536 /7777
00537 /5670 — — — — values which follow
00540 /7777   also attempted → fl. pt.
00541 /5252
00542 /0000 } don't have time to figure it
00543 /0110 } out, but an indexing obviously occurred
00544 /0000
00545 /0000

16? 156/0000   APT locations
00157 /0461
00160 /0600
00161 /0400
00162 /0635
00163 /7744
00164 /2200
00165 /0000
↑C

This patch to USR.RA from Jeff Wyatt looks as though it applies to the early version which has been superseded by one that corrects this problem and handles the core size control better. I believe that the revised version is the one DECUS has been distributing for some time now. - RH

This modification to USR.RA allows it to work with a non-FPP 8e. The problem is that the FPP simulator does not drop the last bit it pushes over; therefore the ALN 1, ALN 2 does not mask as it should.

SRCCOM V4

```
1)         /        SUBROUTINE USR (UNIT, NAME, FUNCT, ERROR)
2)         /        SUBROUTINE USR (UNIT, NAME, FUNCT, ERROR)
1)002               SETX 7777        /LOCATION OF SOFTWARE CORE
1)                  XTA 0            /MUST HAVE BEEN SET BY CCL
1)                  SETX #SHIFT-1
1)                  ALN 1            /SHIFT RIGHT 3
1)                  FSTA #TMP
1)                  ALN 1            /THIS CODE MASKS OFF BITS 6-8
1)                  ALN 2
1)                  FNEG
1)                  FADD #TMP
1)                  ALN 2            /FIELD NOW IN BITS 6-8
1)                  JNE SKCONT       /DID USER SET IT?
****
2)002      /        MODIFIED HERE BY J. WYATT TO WORK ON NON FPP 8E
2)         /
2)                  TRAP4 FILE       /MASK OFF FIELD BITS
2)                  SETX X70
2)                  XTA 0
2)                  JNE SKCONT       /DID USER SET IT?
*******
1)004      FILE,    0;0;0;0;0;0;0;0;0
1)         END
****
2)004      FILE,    0                .  /MODIFIED FOR NON FPP 8E
2)                  CLA                 /THIS CODE IS ONCE ONLY
2)                  CDF 0
2)                  TADX X7777
2)                  AND X70
2)                  DCA X70
2)                  CDF CIF 0
2)                  JMPX FILE
2)         X70,     70
2)         X7777,   7777
2)         END
```

THE
GEORGE
WASHINGTON
UNIVERSITY
MEDICAL CENTER

576-2692

Office of Computer Assisted Education / 2300 Eye Street, N.W. / Washington, D.C. 20037 / (202)

February 22, 1978

Robert Hassinger, Coordinator
12 Bit Sig
Liberty Mutual Research Center
71 Franklin Road
Hopkinton, MA 01748

Dear Mr. Hassinger:

The Educomp BASIC RANDOMIZE command is rather ineffectual at presenting
the user with a different random number list each time a program is run.
Typically the first few numbers are almost identical and it is necessary
to call several numbers before you can be sure you have a different list.
If programs chain however, the RANDOMIZE has no effect at all and the
identical list will be accessed following each chain, see RANDOM and
REPEAT and the associated printout.

One solution to this is to use a data file to transfer information between
the chained programs, see RNDTST. RNDTST uses the file TEMP to hold the
transferred information. When RNDTST is chained to it opens the file,
reads the contents and deletes the file to avoid directory clutter. The
number read from the file is used to position the user at some 'RANDOM'
position down the random number list, lines 380-400. While it is a
nuisance, this method is clean in that all of this is transparent to
the user and the created files are destroyed by the program itself.

Incidently, the technique of having a program create a program to perform
a function which BASIC can't handle directly (RNDTST, lines 190-269) can
be quite useful. I have an abacus program, see CMPUTE, which can be very
convenient. CMPUTE can be incorporated into other programs by chaining
to CMPUTE when desired and adding a line 335 to chain back to the calling
program at the appropriate line number. The only restriction is that all
necessary variables in the calling program must be stored in a file
before chaining however, all multi-program projects must do this anyway.

Sincerely,
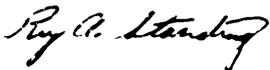
Roy A. Standing
Programmer

Enclosures

READY

OLD RANDOM
READY

LIST

RANDOM 2/23/70 EDUCOMP BASIC V3.4

```
   10  RANDOMIZE
   20  J=100:GOSUB 1000
   30  J=INT(V*20):GOSUB 1000
   40  PRINT J,V
   50  CHAIN "REPEAT"
 1000  FOR X=1 TO J:V=RND(V):NEXT X:RETURN
 4094  END
```

READY

OLD REPEAT
READY

LIST

REPEAT 2/23/70 EDUCOMP BASIC V3.4

```
   10  CHAIN "RANDOM", 10
   20  END
```

READY

RUN

REPEAT 2/23/70 EDUCOMP BASIC V3.4

```
9              .593391
9              .593391
9              .593391
9              .593391
^P
STOP AT LINE 1000
```
READY

RUN

REPEAT 2/23/70 EDUCOMP BASIC V3.4

```
14             .295044
14             .295044
14             .295044
14             .295044
14             .295044
14             .295044
^P
STOP AT LINE 1000
```
READY

READY

OLD RNDTST
READY

LIST

RNDTST 2/23/70 EDUCOMP BASIC V3.4

```
100 REM STANDING 1978 (GWUMC) - RNDTST
110 ! IF EDUCOMP BASIC PROGRAMS ARE CHAINED THE RANDOM NUMBER SEQUENCE
120 ! IS ALWAYS RESET TO THE SAME STARTING POSITION, SO YOU ARE REUSING
130 ! THE SAME LIST.  THE RUN COMMAND IS ALMOST, BUT NOT QUITE, AS BAD.
140 !
150 ! *****   START UP CODE   *****
160 RANDOMIZE
170 PRINT "HOW MANY TEST CYCLES DO YOU WANT TO RUN";
180 INPUT T
190 OPEN "RNDCHN.BA" FOR OUTPUT AS FILE 1  ! CREATE THE 'CHAIN' PROGRAM
200 PRINT #1, "10 CHAIN ";
210 PRINT #1, CHR$(34);
220 PRINT #1, "RNDTST";
230 PRINT #1, CHR$(34);
240 PRINT #1, ", 320"
250 PRINT #1, "20 END"
260 CLOSE 1
270 FOR I=1 TO 50 ! CALL ENOUGH TO OVERCOME INEFFECTIVE RANDOMIZE
280 N=RND(X)
290 NEXT I
300 GOSUB 450
310 GOTO 420
320 ! RE-ENTRY POINT FOR CHAINING
330 OPEN "TEMP" FOR INPUT AS FILE 1
340 INPUT #1, R,T
350 CLOSE 1
360 KILL "TEMP"
370 IF T=0 THEN 520
380 FOR I=1 TO R*20 ! USE R TO POSITION YOURSELF 'RANDOMLY' IN THE LIST
390 N=RND(X)
400 NEXT I
410 GOSUB 450
420 CHAIN "RNDCHN"
430 STOP
440 ! *****   SUBROUTINES   *****
450 OPEN "TEMP" FOR OUTPUT AS FILE 1
460 T=T-1
470 PRINT #1, N
480 PRINT #1, T
490 PRINT N
500 CLOSE 1
510 RETURN
520 KILL "RNDCHN.BA"
530 END
```

READY

RUN

RNDTST   2/23/70   EDUCOMP BASIC V3.4

HOW MANY TEST CYCLES DO YOU WANT TO RUN? 5
 .467011
 .963083
 .107712
 .728066
 .681433

READY

OLD CMPUTE
READY

LIST

CMPUTE   2/23/70   EDUCOMP BASIC V3.4

```
100 REM STANDING 1976 (GWUMC) - CMPUTE
110 DIM B$=50
120 PRINT "INPUT CALCULATION, NO TO STOP";:INPUT B$
130 IF B$="NO" THEN 330
140 FOR A2=1 TO LEN(B$)
150 B2=ASCII(MID(B$,A2,1))
160 IF B2=44 THEN 190 ! COMMA IS ILLEGAL
170 IF B2>39 THEN IF B2<58 THEN 220 ! NUMERICS AND SPECIAL CHARS
180 IF B2=94 THEN 220 ! EXPONENTIATION
190 PRINT "ILLEGAL EXPRESSION.  USE ONLY NUMBERS AND ALGEBRAIC OPERATOR
        S."
200 GOTO 120
210 IF B2=44 THEN 190
220 NEXT A2
230 OPEN "CALC.BA" FOR OUTPUT AS FILE 1
240 PRINT #1,"10 PRINT ";
250 PRINT #1,B$
260 PRINT #1,"20 CHAIN ";
270 PRINT #1,CHR$(34);
280 PRINT #1,"CMPUTE";
290 PRINT #1,CHR$(34)
300 PRINT #1,"30 END"
310 CLOSE 1
320 CHAIN "CALC.BA"
330 KILL "CALC.BA"
340 END
```

READY

RUN

CMPUTE   2/23/70   EDUCOMP BASIC V3.4

INPUT CALCULATION, NO TO STOP? 7^((4/5.372)*3.14)+9
 103.598
INPUT CALCULATION, NO TO STOP? (3*49+7/109)^7
 1.48781E+15
INPUT CALCULATION, NO TO STOP? NO

READY

GALLAUDET COLLEGE

KENDALL GREEN, WASHINGTON, D.C. 20002

**SENSORY COMMUNICATION
RESEARCH LABORATORY
HEARING AND SPEECH CENTER**

February 27, 1978

Mr. Robert Hassinger, Coordinator
12 BIT SIG
Liberty Mutual Research Center
71 Franklin Road
Hopington, Massachusetts 01748

Dear Bob:

I want to commend you for your valiant work on the SIG Newsletter.
I find it most informative and helpful.

There has been some discussion and lamenting that one cannot
read into Fortran IV from the batch stream. This is a very handy
feature implemented on many Fortran systems I am sure. I have found
a way to do this. It might seem a little expensive but can sure save
a lot of trouble making index or control files to simulate this action.
With large systems, i.e. 32K, if you make the Patch to batch to run in
the full 32K and then use PRTS as shipped (limited to 28K), there is no
conflict between batch and the Fortran run-time system. One can then
specify the BAT handler to a unit number, and read data from the batch
file passing it to the Fortran program being executed.

That was some good news. Now about the date, I bought the entire
OS8 V3D system a few days ago for a data analysis, I wanted to print
the system data, calling it from Fortran IV. Strangely it did exactly
as the book said and returned the date for 1970 instead of 1978!!! I
recognize that there are many parts to the system; however, when such
an obvious thing as a bad date has to be changed, shouldn't the entire
system be made to conform!!!

I would also like to mention Jim Van Zee's modification of direct--
I think it is a splendid bit of adaptation. All OS8 systems should not
be without it. I am willing to help distribute it; however, I cannot
produce paper tape. I can make copies on RK01, RK05 disks or DEC or
link tape.

One additional note, we have had an LA1₈₀ for about a year now
and on several occasions I have had trouble with it. It appeared to
die while printing a line. The print-head would just float back to the
normal position, sit there for a few seconds, and then continue printing,
usually overprinting a line. John Tucker who has a DEC COS 310 system
had the same problem. It was solved by grounding the wire frames that
are used to hold the paper and the line-printer itself. It seems that

when the humidity gets too low that static electricity is built up
and disables the line-printer for a short time. Since grounding the
LA180, I have not had a repeat of the problem.

Sincerely,

Fred B. Brandt

AMERICAN SIGN & INDICATOR CORPORATION

CORPORATE OFFICES. 2310 FANCHER WAY. SPOKANE. WASHINGTON • FOR MAIL : P.O. DRAWER 2727. 99220 • PHONE (509) 535-4101 • TWX-510-773-1839

17-Feb-78

Robert Hassinger, Coordinator
12-Bit SIG
c/o DECUS
129 Park Street, PK-3/E55
Maynard, MA   01754

Dear Bob:

This letter is in reply to your comments about Newsletters on mi-
crofiche in the January Newsletter.

It seems to me that distributing all of the back issues of the
12-Bit SIG Newsletter with an index on microfiche would be a very
good way to make back issues available and reduce costs. I'm not
sure, though, that distributing current newsletters on microfiche
would be such a good idea. I like the thought of getting it by
first class mail, but in our installation the newsletter is rout-
ed to about ten people, and it may be a hassle for each person to
go to the one viewer we have and spend the time there to read
through the newsletter.

I sure appreciate the work you put into getting out an excellent
newsletter.

                        Sincerely,

                        Gary B. Stebbins
                        Software Systems Engineer
                        R&D Engineering

```
┌─────────────────────┐
│ ┌───────────────┐   │
│ │   HUGHES      │   │   MICROELECTRONIC PRODUCTS DIVISION
│ └───────────────┘   │
└─────────────────────┘
```

HUGHES AIRCRAFT COMPANY

500 SUPERIOR AVENUE • NEWPORT BEACH, CALIFORNIA 92663 • 714-548-0671 • TWX: 714-642-1353

15 February 1978

Mr. Robert Hassinger, Coordinator
12 Bit SIG
c/o DECUS
129 Parker St., PK-3/E55
Maynard, MA 01754

Dear Mr. Hassinger:

Just thought I should let you know that I appreciate the work put in on the
12 bit newsletter, and say thanks for the various patches (such as the one for
the OS8 date word in the last issue).

I would also like to share an idea I had on reading about the number of PDP-8's
around. The system I am working with is part of a General Radio 1790 logic
tester. As originally configured the computer was in a dedicated application
(the 1790) with no reference at all to DEC software. Our application demanded
an expansion of the machine's capability (data logging and additional peripherals).
Our group included a very clever programmer, and gradually the system expanded
to include an RK8E disk, extra core, a probe table, and a video terminal. This
leads to the conjecture that many of the test systems in the field that use a
PDP8-E computer could also be improved, especially if hints on conversion were
made available. By the way, our system uses a positive I/O bus. The only time
we had trouble from it was in using RTS8, due to the presence of the disk.
Another source of trouble occurred when we tried to use the system cassette
recorders (SYKES CC-100's) as file structured devices. The handler available
from SYKES does not work under OS8-V3. We now have the bugs worked out
and use them for storing protected backup files.

So much for our system, but consider the possibilities. Our one system alone has
brought close to $10,000 in hardware and software sales to DEC. A few hundred
such conversions get into real money, and a foot in the door on larger systems.

Yours truly,

Frank J. Wilson

FJW:dw

STEVENS INSTITUTE OF TECHNOLOGY

Castle Point Station, Hoboken, New Jersey   07030

Area Code 201
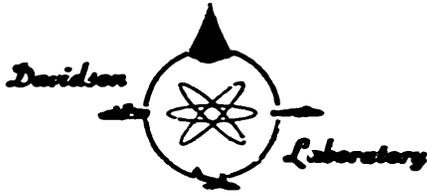
792 -2700

711 HUDSON STREET   10 February 1978

Mr. Robert Hassinger, Coordinator
12 Bit SIG
c/o DECUS
129 Parker St., PK-3/E55
Maynard, MA   01754

Dear Bob:

Reading the latest 12 bit SIG newsletter makes one aware
that we owe you a vote of thanks, both for all your work with this
group and for bearing on DEC to fix the Software News foul-up (and
other things).

At D.L. we have one of the early PDP8e's (# 384-1971) used
for on-line data acquisition at fairly high rates (250 scans/sec of
20 channels). We now have 32K core, ADO1, DECtapes and 2 RK05's.
We still use PAL8, SABR and FORTRAN II since the short integer word
is very convenient for this work, and FORTRAN II seems to have
advantages over FORTRAN IV in some other respects too. Using
interrupts with FORTRAN II has given us considerable trouble but
we think we have got over this now.

We have programs that will quickly give mean values and/or
Peak and Trough statistics, fairly elaborately massaged and in a
form ready to print in a report. Other programs will generate spectra
either by Blackman-Tukey or FFT methods. As there is a PDP10 on Campus
we often transfer data to it for further processing and we have a
program that will read a standard PDP8 DECtape on the PDP10 time-
sharing system.

I hope that you will meet my colleague Clarke Walker,
who is our PDP8 expert, at the Spring Symposium.

Sincerely yours,

DAVIDSON LABORATORY

Paul G. Spens, Head
Computing Division

PGS:as