VOLUME 2 : #6

SOFTWARE PERFORMANCE SUMMARY

FOR

THE PDP-8 & PDP-12

SOFTWARE INFORMATION SERVICE

Software Performance Summary

This Software Performance Summary is intended to supplement your set of Family of 8 software. It contains all current information on known software problems, patches, manual corrections and programming notes. Each article is coded sequentially by system program in the lower right hand corner. As new versions correct software problems and reprinted manuals include programming notes and manual corrections, new articles will announce the revised software and specify by this code which articles should be removed from your Software Performance Summary file. Articles may also be replaced when new information becomes available; such as, a procedure to circumvent a problem may replace the original report of the problem. This information, as well as newly discovered problems, methods, etc., is printed in Digital Software News for the PDP-8 & PDP-12. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your local Digital Office.

Additional copies of the current Software Performance Summary and updated copies of the PDP-8/12 software price list are available at no charge upon request. Please order by mail from the Program Library in Maynard.

SOFTWARE PERFORMANCE SUMMARY

TABLE OF CONTENTS

FOCAL
Patch to accept more common variables — FOCAL 2

FOCAL
Patch to correct Input-Buffer-Overflow — FOCAL 3

FOCAL
Patch to correct one of the Nine Constants — FOCAL 4

FOCAL
Patch to 8K FOCAL with DISK/DECtape Monitor — FOCAL 5

FOCAL
Note to users of FOCAL — FOCAL 6

FOCAL
Patch to use LINC-8 display — FOCAL 7

FOCAL
Machine Language Subroutine — FOCAL 8

FOCAL
Programming Note — FOCAL 9

FOCAL
Patch to Correct Rounding Error — FOCAL 10

4K FORTRAN
Procedure to restore locations destroyed by SYMBOL-PRINT — 4K FORTRAN 1

4K FORTRAN
Operating System with DECtape Option — 4K FORTRAN 2

4K FORTRAN
Programmer's Reference Manual — 4K FORTRAN 3

4K FORTRAN
Programmer's Reference Manual — 4K FORTRAN 4

4K FORTRAN
Compiler and Operating System — 4K FORTRAN 5

4K FORTRAN
Problem with mode of variables — 4K FORTRAN 6

4K FORTRAN
Patch to correct diagnostic for overlap — 4K FORTRAN 7

DF32/RFØ8 PIP PIP 3
Program halt due to unexpected interrupt

PDP-8 SABR ASSEMBLER (DEC-08-A2D2-PB)
Problem in Listing Pass with OFF-PAGE Externals SABR 1

LAP6-DIAL EDITOR LAP6-DIAL EDITOR 5
Problem with the work area

LAP6-DIAL EDITOR LAP6-DIAL EDITOR 6
Preserving pointers upon restart or
exist for LAP6-DIAL-V2 or LAP6-DIAL-MS.

LAP6-DIAL EDITOR LAP6-DIAL EDITOR 7
Allowing input into a line containing
60 or more characters.

LAP6-DIAL-MS EDITOR LAP6-DIAL EDITOR 8
Incorrect placement of a comma

LAP6-DIAL-MS EDITOR LAP6-DIAL EDITOR 9

PDP-12 LAP6- DIAL-MS EDITOR
To Correct problem with EDITOR software
flag LAP6-DIAL EDITOR 10

PDP-12 LAP6-DIAL-MS EDITOR
ADD Program by line number problem LAP6-DIAL EDITOR 11

PDP-12 LAP6-DIAL-MS EDITOR
ADD Program by block number problem LAP6-DIAL EDITOR 12

PDP-12 LAP6-DIAL-MS EDITOR
ADD Program by block number LAP6-DIAL EDITOR 13

DIAL-MS SOURCE TAPES (version 1) DIAL-MS SOURCE TAPES(ver.1) 1

MISCELLANEOUS MISCELLANEOUS 1

PDP-8 Binary Loader (DEC-Ø8-LBAA-PM)

Procedures to restore destroyed instructions.

The current version of the Binary Loader (DEC-Ø8-LBAA) dated May 10, 1967 is not compatible with either the Bootstrap Loader for the TCØ1 DECtape Library System (DEC-Ø8-LUAA) or with any program using the DISK Data Break (Locations 7750 & 7751).

The Binary Loader uses locations 7614-7616 as temporary storage of variables thus destroying the three instructions of the TCØ1 Bootstrap which should be in these locations. To restore the Bootstrap after reading in or using the Binary Loader the user should restore the contents of locations 7614-7616 as follows:

```
7614    6766    DCTA    DTXA
7615    3354    DCA     WCOUNT
7616    6771    DTSF
```

The TCØ1 bootstrap may then be started as usual at location 76ØØ. This version of the Binary Loader does not harm the Bootstrap Loader for the 552/555 DECtape Library System (Digital-8-3-U).

The Binary Loader also uses locations 7750 and 7751 for instructions. These two locations are used by the DF32 Mini Disk as its Word Count and Current Address registers for 3 cycle Data Break I/O transfers. Any disk I/O the user may do will, therefore, destroy the two instructions of the Binary Loader contained in locations 7750 and 7751. To restore the Binary Loader, restore the contents of these locations as follows:

```
7750    1355    TAD     WORD2
7751    5743    JMP     I ASSEMB
```

The Binary Loader may then be started at 7777 and used as usual.

PDP-8

Problem in DECtape Copy (DEC-08-YPTA-PB)

DECtape Copy (DEC-08-YPTA-PB) does not work if the field bits in DECtape Status Register B are not Ø on entry to the program. The START key clears these bits, however, if the program is loaded by PS/8 from a disk file these bits may be set.

The following patch will correct this problem:

LOCATION	CHANGE FROM	TO
Ø242	7450	744Ø
Ø243	5247	1Ø44
Ø244	1Ø44	766Ø
Ø245	762Ø	52ØØ
Ø246	52ØØ	6774

DISK DDT (DEC-D8-CDE1-PB and DEC-D8-CDE2-PB)

Patch to use high speed reader for reading symbol table tapes.

The patch given below will be of use to users who have no low speed paper tape reader and who wish to use Disk DDT but who are hampered by the fact that only the low speed reader may be used to read in user symbol table tapes. The patch is used as follows:

1) Load the program to be debugged into core.

2) Call DDT into core by typing DDT↵ in response to a Monitor "."

3) Stop DDT by depressing STOP on the console; then load the patch into core using the Binary Loader (system LOADER will destroy DDT). Note that the patch overlays one location in DDT (loc 3374) and thus the patch must be read in after DDT is in core.

4) Place the symbol table tape in the high speed reader with leader code under the read head.

5) Start the patch at location 4600[1]. The high speed reader will be initialized and control will pass to DDT, which will be expecting a command.

6) Type ALT MODE (or ESCape) R (echoes as [R) - the symbol table tape will read in.

7) After the last symbol has been read, DDT will type the new lower limit of the external symbol table, followed by an up-arrow (↑) to show that it is waiting for CONTROL P (↑P).

8) Type CONTROL P (↑P) - DDT will now be ready to accept commands, and the user may begin debugging, using the symbols he has just read in.

1 Users may change the location of the patch by changing the *4600 in the source and reassembling, being careful not to overlay any locations in his own program or any locations in DDT other than loc 3374, which must contain the constant READHS.

NOTE: Once a symbol table has been read in via the high-speed reader, DDT is restored to its normal state. If the user types [R again, DDT will expect symbols from the teletype/low speed reader. If the user wishes to read in more than one symbol table tape with the high speed reader, he may NOP locations 4640 and 4641 of the patch, which restore the overlaid DDT location, and DDT will always expect additions to the symbol table to be input via the high speed reader. This patch cannot take full advantage of the high speed reader, for DDT still handles the input characters one at a time rather than buffering them as they are read.

An assembly listing of the patch follows:

```
                        /ROUTINE TO ALLOW DISK DDT TO READ
                        /SYMBOL TABLE TAPE FROM HIGH SPEED
                        /READER
                        /
                        COMM=2513
                        START2-24Ø3
                        COMM34=3374
                        PUNOUT=3455
                        READKB=34Ø1
                        *COMM34
3374    46Ø3            READHS              /CHANGE DDT SYM TAB READ ROUTINE ADDR
        *46ØØ
46ØØ    6Ø14            RFC                 /INITIALIZE HS READER
46Ø1    56Ø2            JMP 1  .+1          /JUMP TO
46Ø2    24Ø3            START2              /START2 IN DDT
46Ø3    ØØØØ    READHS,Ø
46Ø4    72ØØ            CLA
46Ø5    1244            TAD COMM1A          /GET ADDR OF NEW TABLE READ ROUTINE
46Ø6    3645            DCA I INCOMM        /STORE IT IN DDT
46Ø7    6Ø11            RSF                 /WAIT ON FLAG
461Ø    52Ø7            JMP  .-1
4611    72ØØ            CLA
4612    6Ø16            RRB RFC             /GET CHARACTER
4613    3251            DCA TFM1            /STORE IT
4614    1251            TAD TFM1
4615    125Ø            TAD MIN2ØØ          /IS IT LEADER-TRAILER (2ØØ)
4616    745Ø            SNA
4617    52Ø4            JMP READHS+1        /YES - IGNORE IT
462Ø    1252            TAD CON1            /IS IT LINE FEED (212)
4621    765Ø            SNA CLA
4622    5234            JMP EXITA           /YES - GET CHAR AND EXIT
4623    1251            TAD TFM1
4624    1253            TAD CON1+1          /IS IT ROBOUT (377)
4625    745Ø            SNA
4626    52Ø4            JMP READHS+1        /YES - IGNORE IT
4627    1254            TAD CON1+2          /IS IT EOT (2Ø4)
463Ø    765Ø            SNA CLA
4631    5236            JMP FIXUP           /YES - RESTORE CHANGED DDT LOCS
4632    1251    RETN,   TAD TEM1            /NO - GET CHAR
4633    4647            JMS I OUTPUN        /ECHO IT
4634    1251    EXITA,  TAD TFM1            /EXIT WITH CHAR IN AC
4635    56Ø3            JMP I READHS
4636    1243    FIXUP,  TAD COMM1
4637    3445            DCA I INCOMM
4640    1243            TAD COMM1
4641    3646            DCA I INCOMM+1
4642    5232            JMP RETN
                /
                /
4643    34Ø1    COMM1,  READKB
4644    46Ø3    COMM1A, READHS
4645    2513    INCOMM, COMM
4646    3374            COMM34
4647    3455    OUTPUN, PUNOUT
465Ø    76ØØ    MIN2ØØ, 7600
4651    ØØØØ    TFM1,   Ø
4652    7766    CON1,   -12
4653    74Ø1            -377
4654    Ø173            173
```

PDP-8 DISK DDT (DEC-D8-CDE1 & DEC-D8-CDE2)

Failure to recognize overflow of the user symbol table

The Library version of Disk DDT does not correctly test for overflow of the user symbol table. The following patch will cause DDT-D, on occasion of symbol table overflow, to print a carriage return and line feed, and the new lower limit of the user symbol table, followed by a "↑" to indicate that it is waiting for a Control P (↑P).

Patch .DDT (DEC-D8-CDE2-PB) before saving it on the system device as follows:

LOCATION	FROM	TO
33Ø6	1361	5777
3353	Ø4Ø1	716Ø
3377	unused	4554
4554	"	4757
4555	"	176Ø
4556	"	5761
4557	"	3463
456Ø	"	3361
4561	"	33Ø7

PDP-8 DISK EDITOR (DEC-D8-ESAD-PB)

Two Problems in the DISK EDITOR have been found and corrected.

The first problem concerns rubbing out past the beginning of a line while in Search mode. Because it is caused by the contents of a previous text buffer. the problem occurs rarely. The correction to this problem is as follows:

LOCATION	OLD CONTENTS	NEW CONTENTS
1332	1014	5733
1333	3114	3342
3342	(unused)	1014
3343	(unused)	3114
3344	(unused)	3526
3345	(unused)	5746
3346	(unused)	1334
0053	3342	3347
0126	3342	3347

The second problem occurs when in Append mode. If an odd number of characters are put in a line terminated with a CR, and the user immediately starts rubbing out, the EDITOR will rub out even though no characters are on the current line. Again, this causes damage to the text buffer. If this occurs, please make the following correction:

LOCATION	OLD CONTENTS	NEW CONTENTS
1323	1173	1116
1362	5345	5344

PDP-8 DISK LINKING LOADER (DEC-08-A2C7)

Patch to allow the Linking Loader to recognize CONTROL C (↑C) when it is waiting for a FIELD specification.

The library version of the PDP-8 Disk Linking Loader fails to recognize a CONTROL C (↑C) when it is waiting for a FIELD specification following user selection of the L or O loading options; it treats ↑C as an illegal FIELD designation and responds with:

```
?
*OPT-
```

The following patch will correct this error and recognize ↑C as a call to the Monitor.

LOC	OLD CONTENTS	NEW CONTENTS
6065	(7200)	5374
6066	(5654)	0055
6174	(unused)	1266
6175	(")	7650
6176	(")	5451
6177	(")	5654

The patch may be toggled in after loading the Linking Loader into core. Once the patch is made the Loader may be resaved on the disk. Users should also note that restart for the Linking Loader is at Loc. 6ØØØ FIELD Ø and not at Loc. 6ØØ as was specified in the Cumulative Software Manual Update of January, 1970 (page H-4 of 8-K SABR ASSEMBLER MANUAL, Line 7 from the bottom).

Patching programs before storing on the system device

The following steps apply to patching any disk/DECtape system program.

1. Set up the program (to be patched) for loading with the Disk System Loader. Answer all of its questions as usual with the exception of the starting address.

2. When the Loader requests the starting address (ST=), type 7636 for disk or 76Ø6 if the system device is DECtape. This will provide the halt referenced in step 4.

3. Load the program to be patched.

4. When loading is complete, the system will halt.

5. Now the desired patches may be made to the program in core.

6. Load address 7600, press START and control will return to the monitor.

7. Type the SAVE command as usual.

Programming note on chaining in the PDP-8 Disk Monitor System

The following procedure illustrates the chaining of programs under the Disk Monitor System for the 8 Disk/DECtape system. First, save each overlay as a system program in the normal manner. Second, a call from one program to bring in the next program as an overlay must be done as follows:

1. In locations 74ØØ and 74Ø1 deposit the ASCII codes for the name of the system program to be loaded.
(Note: These ASCII codes are the eight bit code minus 24Ø)

2. In order to enter the routines which bring in the overlay, execute the following code:

```
                JMS I DYSKIO        /LINK TO MONITOR I/O
                3                   /READ
                1Ø                  /BLOCK #10 (DECTAPE #13)
                72ØØ                /CORE ADDRESS
                Ø
                HLT                 /ERROR RETURN
                JMP I  .+1          /LINK TO LOADER
                72Ø1
DYSKIO,         7642
```

The following example of a program that only brings in an overlay, called "TEST", may be helpful to many users.

```
                *2ØØ
Ø2ØØ    7ØØØ    NOP
Ø2Ø1    7ØØØ    NOP
Ø2Ø2    4612    JMS I DYSKIO
Ø2Ø3    ØØØ3    3
Ø2Ø4    ØØ1Ø    1Ø
Ø2Ø5    72ØØ    72ØØ
Ø2Ø6    ØØØØ    Ø
Ø2Ø7    74Ø2    HLT
Ø21Ø    5611    JMP I .+1
Ø211    72Ø1    72Ø1
Ø212    7642DYSKIO,7642

                *74ØØ
74ØØ    6445    6445    /TE WITH 24Ø SUB. FROM EACH CHAR
74Ø1    6364    6364    /ST
```

PDP-8 SYMBOLIC EDITOR (DEC-08-ESAC)

Problem with missing line feed

When using the Next (N) command for output to the high speed punch, a timing problem causes the final line feed (ASCII code 212) to be lost, ending the tape with a carriage return (ASCII code 215). This difficulty occurs in the current version of the Editor (DEC-08-ESAC). The problem may be corrected with the following patch:

Change: COM1,JMP FORM to COM1,JMP FORM+1

LOCATION	FROM	TO
13ØØ	5216	5217

PDP-8 SYMBOLIC EDITOR

Problem with rubbing out past beginning of lines

There is a problem with the current library version of the Symbolic Editor such that under certain conditions, the user can delete characters from right to left past the beginning of a line. This usually results in the Editor losing track of the proper line sequences. Making the following patch before starting the Editor will eliminate the problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
ØØ57	(1624)	1631
Ø115	(1624)	1631
1324	(1Ø13)	5725
1325	(3124)	1624
1624	(unused)	3515
1625	(")	1Ø13
1626	(")	3124
1627	(")	563Ø
163Ø	(")	1326

PDP-8 FLOATING POINT PACKAGE USING EAE (DIGITAL-8-25-F)

Patches to Correct ALIGN Routine and Check for Largest Negative Mantissa

The current version of the PDP-8 Floating Point Package using the Extended Arithmetic Element (EAE), has an incorrect constant in the ALIGN routine for addition and subraction. The constant only makes a difference when the two numbers being added vary greatly in magnitude. The following patch corrects the problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
6167	(7747)	775Ø

The EAE Floating Point Package also fails to check for the largest negative mantissa (OCTAL 4ØØØ ØØØØ) after normalization. This number could cause erroneous results in subsequent calculations. The patch to correct this problem requires several changes to the EAE package itself, plus a fifteen word patch which can be inserted anywhere in the same memory field as the EAE Floating Point Package. In the listing that follows, the patch has been placed just before the start of the Interpreter (i.e. just before loc. 56ØØ). This patch would have to be moved for use with other versions of the EAE Packages (i.e. for packages 2,3, or 4). The necessary changes to the EAE package are:

LOCATION	OLD CONTENTS	NEW CONTENTS	
6607	(326Ø)	3367	
6655	(226Ø)	2367	
6657	(5600)	5766	
6660	(0000)	5600	
6766	(unused)	5561	(address of first word of patch)

Following is the 15 location patch which may be placed anywhere in the appropriate memory field. Note that location 6766 must contain the address of the first word of the patch.

LOCATION	CONTENTS
5561	1Ø46
5562	764Ø
5563	5777
5564	1045
5565	7510
5566	7041
5567	77ØØ
5570	5777
5571	2044
5572	7000
5573	1045
5574	7130
5575	3045
5576	5777
5577	6660

FOCAL 1969 (DEC-08-AJAE)

Failure to read user programs on a PDP-8/S

FOCAL 1969 sometimes fails to read user generated tape program via the high speed reader on an 8/S. This is due to a timing problem in the reader delay loop. The following patch will correct the problem:

Change Location 6322 after executing the dialogue:

LOCATION	FROM	TO
6322	7Ø4Ø (CMA)	112Ø (TAD M5)

Note: The dialogue attempts to reduce the time-out delay by changing TAD M2Ø to CMA but this is not correct.

PDP-8 FOCAL LIBRA System (Multi-user overlay)

Patch to accept more common variables

FOCAL LIBRA can be modified to accept more common variables by using the following patch after the initial dialogue:

LOCATION	FROM	TO
0035	4560	4553
6362	4565	4560
6363	7773	7771
6364	0005	0007

FOCAL 1969 (DEC-08-AJAE)

Patch to correct Input-Buffer-Overflow

To avoid input-buffer-overflow while reading in a tape, remove the echo:

LOCATION	TO
2163	7ØØØ

To restore the keyboard echo:

LOCATION	TO
2163	4551

FOCAL 1969 (DEC-08-AJAE)

Patch to correct one of the SINE Constants

To correct one of the SINE Constants, use the following patch:

LOCATION	FROM	TO
5333	24Ø1	25Ø1

PDP-8 FOCAL (DEC-08-AJAE-PB) with 8K Overlay

Patch to 8K FOCAL with DISK/DECtape Monitor

After loading the 8K Overlay use the following patch if you are having problems saving, restoring or running 8K FOCAL Programs with the DISK/DECtape Monitor System.

LOCATION	FROM	TO
7525	6213	6212

FOCAL 1969 (DEC-08-AJAE)

Note to users of FOCAL

To read data tapes on the TTY reader, use the following patch to remove the interrupts:

LOCATION	FROM	TO
63	2676	1354
64	2666	2414
2732	6ØØ1	5336
2762	6Ø46	7ØØØ

NOTE: Disables high-speed reader unless the following patch is also made:

LOCATION	FROM	TO
6324	1Ø37	6Ø11
6325	77ØØ	741Ø

PDP-8 FOCAL '69 (DEC-08-AJAE)

Patch to use LINC-8 display

To use the LINC-8 display with FOCAL '69 use the following code which should be assembled with PAL III. This binary tape must be loaded after loading FOCAL.

```
                    /DISPLAY PATCH FOR FOCAL  69

                    START=4500
                    /VALUE OF START IS DETERMINED BY TYPING   L   TO FOCAL
                     AFTER
                    /INITIAL DIALOGUE.  THE LAST VALUE TYPED MINUS 31 OR
                     GREATER
                    /THAT DOES NOT CROSS PAGE BOUNDARY
                    XDYS=1142
                    *XDYS +1
1143      3757      DCA I X
                    *XDYS+11
1153      6167      6167
1154      7200      CLA
1155      5756      JMP I PATCH
1156      4551      PATCH,LINDIS
1157      4550      X,SAVX
                    *BOTTOM
0035      4547      START -1
                    *START
4550      0000      SAVX,0
4551      1371      LINDIS,TAD LINCPC /LINC DISPLAY ROUTINE
4552      6165      6165
4553      7200      CLA
4554      1775      TAD I SAVFOC
4555      3374      DCA SAVLOC
4556      1350      TAD SAVX
4557      3775      DCA I SAVFOC
4560      1376      TAD ENABL
4561      6141      6141
                    EFUN3I=136
4562      1377      TAD TWLV
4563      6141      6141
4564      7000      NOP
4565      7200      CLA
4566      1374      TAD SAVLOC
4567      3775      DCA I SAVFOC
4570      5536      JMP I ENFUN3I /RETURN TO FOCAL
4571      4572      LINCPC,NEXT
4572      0141      NEXT,141      /LINC CODE DISPLAY 1
4573      0000      0             /LINC HALT
4574      0000      SAVLOC,0
4575      4001      SAVFOC,4001
4576      0010      ENABL,10
4566      0002      TWLV,2
                    BOTTOM=35
```

BOTTOM	0035
EFUN3I	0136
ENABL	4576
LINCPCP	4571
LINDIS	4551
NEXT	4572
PATCH	1156
SAVFOC	4575
SAVLOC	4574
SAVX	4550
START	4550
TWLV	4577
X	1157
XDYS	1142

PDP-8 FOCAL '69 Machine Language Subroutine

FNEW functions must be written in assembly language in patch form and loaded over the FOCAL program. These may be loaded after FOCAL has successfully loaded, and do not affect the "FOCAL" program the user has previously entered.

Subprogram to operate on an indefinite number of arguments.

Called from FOCAL by: SET = FNEW (A,B,C,...,N); TYPE Z where A,B,etc. have been previously defined and are to be operated upon to yield Z. Subprogram must be assembled as a patch and loaded over FOCAL.

```
START = (SEE NOTE)
*BOTTOM
 FNEW-1
*FNTAB+14
 FNEW
*START
 FNEW,TAD LIST-1
      DCA 16              /PUT ADDRESS OF LIST INTO AUTOINDEX REG.
      DCA KOUNT
      ISZ KOUNT
      JMS I INTEGER       /FETCH NEXT ARGUMENT
      DCA I 16            /STORE IN LIST
      PUSHJ
      ARG                 /GET NEXT CHARACTER
      SKP                 /RETURN IF NOT A COMMA
      JMP FNEW+3          /RETURN IF COMMA WITH NEXT ARG. IN FLAC
      CLA CLL             USER CODING TO OPERATE ON N ITEMS
      .                   STORED IN LIST.  KOUNT CONTAINS THE NUMBER
      .                   OF ITEMS,N.  THE RESULT MUST BE STORED
      .                   IN INTEGER FORM IN ANS.

      CLA CLL
      TAD ANS             /CODING TO SET ANS INTO FLAC
      CLL RAR
      DCA FLAC+1
      RAR
      DCA FLAC+2
      TAD C14
      DCA FLAC
      JMP I EFUN3I        /CHECK FOR RIGHT PAREN,NORMALIZE FLAC,RETURN
C14,14
KOUNT,Ø
ANS,Ø
      LIST-1
LIST, Ø
      Ø
      .
      .                   /AS LONG AS ARGUMENT LIST WILL BE
      .
```

```
ARG,    TAD CHAR          /CODING TO FETCH NEXT CHAR.
        TAD MCOMMA        /RETURN TO CALL+3 (CALL+2)
        SZA CLA           /IF CHAR. IS (IS NOT) A COMMA
        JMP  .+4
        PUSHJ
        EVAL-1
        IAC
        POPJ
MCOMMA,7524

INTEGER=53                /THESE VALUES MAY VARY
PUSHJ=4540                /WITH THE VERSION OF FOCAL USED
EVAL=1613                 /FOR FOCAL '69 DEC-08-AJAE-PB
FLAC=44                   /THESE DEFINITIONS APPLY
CHAR=66
EFUN3I=136
BOTTOM=35
FNTAB=374
```

NOTE: CODING MAY OCCUPY UPPER END OF STORAGE USE FOR TEXT AND PUSH-DOWN LISTS. THIS AREA OCCUPIES CORE LOCATIONS:

3220-4577	with all function
3220-5177	with FEXP,FLOG,FATN,
3220-5232	FSIN,FCOS and above deleted.

TO DETERMINE THE VALUE FOR "START" USE THE "L" COMMAND IN FOCAL. "START" = LAST VALUE PRINTED MINUS LENGTH OF PATCH STARTING AT *START.

On the following page is a subprogram which sums an indefinite number of arguments. It is called from FOCAL by:
SET Z = FNEW (A,B,C,.....,N) :TYPE Z where A,B, etc. have been previously defined. This subprogram must be assembled as a patch loaded over FOCAL.

FOCAL Subroutines (Continued)

```
START = (SEE NOTE)
*Bottom
FNEW-1
*FNTAB+14
FNEW
*START
FNEW,DCA SUM              /CLEAR SUM
       JMS I INTEGER      /BRING FIRST (NEXT) CHAR. INTO ACCUMULATOR
       TAD SUM
       DCA SUM            /UPDATE SUM
       PUSHJ
       ARG                /GET NEXT CHAR.
       SKP                /RETURN IF NOT A COMMA
       JMP FNEW+1         /RETURN IF COMMA WITH NEXT ARG IN FLAC
       CLA CLL
       TAD SUM            /FINAL SUM SET INTO FLAC FOR
       CLL RAR            /RETURN FROM SUBPROGRAM
       DCA FLAC+1
       RAR
       DCA FLAC+2
       TAD C14
       DCA FLAC
       JMP I EFUN3I       /CHECK FOR RIGHT PAREN,NORMALIZE FLAC,RET.
C14,14                    /FLAC,RETURN
SUM,   Ø
ARG,   TAD CHAR           /CODING TO FETCH NEXT CHAR.
       TAD MCOMMA         /RETURN TO CALL+3 (OR CALL+2)
       SZA CLA            /IF CHAR IS (IS NOT) A COMMA
       JMP  .+4
       PUSHJ
       EVAL-1
       IAC
       POPJ
MCOMMA,7524

INTEGER=53                /THESE LOCATIONS MAY VARY
PUSHJ-4540                /WITH THE VERSION OF FOCAL
EVAL=1613                 /FOR FOCAL '69 DEC-08-AJAE-PB
FLAC=44                   /THESE DEFINITIONS APPLY
CHAR=66
EFUN3I=136
BOTTOM=35
FNTAB=374
```

NOTE: CODING MAY OCCUPY CORE FROM 3220-4577. THIS AREA IS OCCUPIED BY STORAGE LISTS AND PUSH-DOWN LISTS, SO CODING SHOULD OCCUPY THE TOP OF THIS REGION IN CORE SO AS NOT TO SEVERELY LIMIT THE WORKING STORAGE AREA. IF FEXP, FLOG, AND FATN ARE DELETED, THE USER HAS AVAILABLE LOCATIONS 3220-5177. IF IN ADDITION TO ABOVE, FSIN AND FCOS ARE DELETED, THE USER HAS AVAILABLE 3220-5232. TO DETERMINE THE VALUE FOR "START" USE THE "L" COMMAND IN FOCAL. START=LAST VALUE PRINTED MINUS LENGTH OF PATCH STARTING AT *START.

FOCAL 1969 (DEC-08-AJAE)

Programming Note

TYPE #" issues a CR plus a FORM feed to allow the carriage to get back to the left margin. On certain models of teletypes, multiple line feeds are generated upon reception of a form feed. The following patch should be used if you have this type of TTY:

LOCATION	TO	FROM
125Ø	7Ø4Ø	7ØØ1

FOCAL 1969 (DEC-08-AJAE)

Patch to correct rounding error

A rounding error has been discovered in FOCAL where a remainder of 4 in the first digit discarded causes the last digit printed to be rounded upwards. To correct this problem, the following patch should be used:

LOCATION	FROM	TO
5525	ØØØ5	ØØØ4

FOCAL 1969 (DEC-08-AJAE)

Patch to allow automatic restart with L command

The following patch will automatically restart FOCAL when using the L command (upon completion of the four locations printout).

LOCATION	FROM	TO
7526	55Ø4	5177

Note: This patch must be used when the Disk Monitor System is not in use.

FOCAL 1969 (DEC-08-AJAE) with CLINE & PLOTR

Patch to correct circle

If the radius of a circle being plotted is an exact power of 2, it is mistakenly interrupted as Ø and a string of dots is drawn in place of the circle.

The following patch should be used (according to the overlay in use) :

For CLINE:

LOCATION	FROM	TO
5Ø6Ø	1Ø46	1Ø45

For PLOTR:

LOCATION	FROM	TO
5Ø32	1Ø46	1Ø45

FOCAL 1969 (DEC-08-AJAE)

Patch for use on LAB-8

A problem has been discovered when using FOCAL on a LAB-8 system. A possible enabling of the clock interrupt may occur. To prevent this, the following patch should be made before the initial dialogue:

LOCATION	FROM	TO
4377	6Ø12	7ØØØ

PDP-8 4K FORTRAN Compiler (DEC-08-AFC1)

Procedure to restore locations destroyed by SYMBOLPRINT (DEC-08-AFA2)

The PDP-8 4K FORTRAN compiler is designed to be restarted for successive compilations and not require reloading after compiling each source program. The PDP-8 FORTRAN Symbolprint program, which is read in over the compiler to use the information left in its tables and counters, resides in the compiler's input statement so that this restart capability will not be lost.

However, the use of Symbolprint imposes two restrictions on the restart facility:

1. The compiler must be reloaded after each use of Symbolprint if the user chooses the switch option to indicate that his FORTRAN source program contains DECtape I/O statements. This is a design restruction due to lack of free space in the compiler. A portion of the additional code to compile DECtape I/O statements is destroyed by Symbolprint.

2. The contents of three locations in the compiler are changed by Symbolprint during execution. They must be restored or subsequent compilations will yield incorrect results, possibly without producing any error messages.

   The locations and their proper contents are:

LOCATION	CONTENTS
1Ø	ØØØØ
11	Ø6Ø3
12	Ø6Ø4

   These values must be restored by the following procedure after running Symbolprint and before starting the next compilation.

Set	switch	register	to	0010	depress	LOAD ADDRESS
"	"	"	"	0000	"	DEPOSIT
"	"	"	"	0603	"	DEPOSIT
"	"	"	"	0604	"	DEPOSIT

   Restart the compiler as usual.

PDP-8 4K FORTRAN Operating System with DECtape Option (DEC-08-AFC3)

Initialization of READ/WRITE Routines

When using PDP-8 FORTRAN (4K version) with the DECtape option, a physical READ or WRITE must precede any pseudo READ or WRITE in a program to initialize the READ/WRITE routines. Contrary to an example given on page 46 of the FORTRAN manual (DEC-08-AFAC-D), this is essential or the program will fail during execution and yield a "TILT 76" error diagnostic.

Chapter 8: FORTRAN with DECtape option

Several points concerning DECtape I/O with 4K FORTRAN demand clarification:

1. Logical READs and WRITEs, performed by specifying unit 0, must include a list of one or more data items.

2. Physical READs and WRITEs, performed by specifying a unit number other than 0, may include a list, but need not. If a list is given, a logical data transfer of the named data items takes place, as well as the 128-word buffer-DECtape transfer; on input, the buffer pointer is set to the next untransferred item in the buffer upon completion; on output, the pointer is reset to the beginning of the buffer.

3. Some physical I/O operation must logically precede any logical I/O in the user's program; the following sample program illustrates such a usage:

```
C;      SAMPLE PROGRAM FOR DECTAPE
C;
C;      DO A DUMMY PHYSICAL READ
C;      TO INITIALIZE THE DECTAPE ROUTINES
C;
        READ 1,1,703
703;    FORMAT (E)
C;
C;      PUT EIGHT ITEMS INTO RESIDENT BUFFER
C;

        DO 12 K = 1,4
        ACCEPT 703, TEMP
        ACCEPT 703, PRES
        WRITE 0,0,703,TEMP,PRES
12;     CONTINUE
C;
C;      WRITE BUFFER TO DECTAPE
C;
        WRITE 7,100,703
        STOP
        END
```

FORMAT Interaction with DECtape WRITE Statement:

In the following program sequence, let I=3, J=6, and K=9; then:

```
1ØØ     FORMAT (/,I,I,I,/)
        WRITE Ø,Ø, 1ØØ, I, J, K
```

will put the following into the DECtape buffer:

```
        Ø215        (a carriage return)
        ØØØ3
        ØØØ6        the three data words
        ØØ1Ø
        Ø215        (a carriage return).
```

Had the program sequence been:

```
1ØØ     FORMAT (/,I,I,I,/)
        TYPE 1ØØ, I, J, K
```

then each slash in the FORMAT statement would have delivered a carriage return and a line feed to the teleprinter.

PDP-8 4K Fortran Compiler and Operating System
(DEC-08-AFC1-PB and DEC-08-AFC3-PB)

Patch to Compiler Input Overflow and Input Conversion Limitations

The library version of 4K Fortran Compiler (DEC-08-AFC1-PB) uses a mask of 74ØØ to check for overflow in the high order mantissa word during floating point input conversion. The correct mask is 76ØØ. To patch the compiler, load it using the binary loader, then DEPOSIT 76ØØ into location 5515 (old contents of 5515 was 74ØØ). The compiler may now be started as usual.

Users should not attempt to input floating point constants of more than six decimal digits, either in the Fortran source program or via the run-time ACCEPT statement, as in general such constants will not be handled correctly. The results of including a constant of the form XXXXXXX. in the source program are unpredictable, while a constant of the form .XXXXXXXE±XX will be represented correctly to six decimal places. Attempting to input a constant of the form XXXXXXX. via the run-time ACCEPT statement, may cause an error in the calculation of the Exponent, while inputting the same constant as XXXXXX.XE+1 via the ACCEPT statement, will cause the exponent to be calculated correctly, and the mantissa to be correct to 6 decimal places. Users should be especially aware of the fact that different representations of the same constant (7 or more digits) may cause different results when these representations are input to the compiler, or when input via the run-time ACCEPT statement.

4K FORTRAN COMPILER (DEC-08-AFC1-PB)

Problem with mode of variables.

The current version of the 4K FORTRAN COMPILER (DEC-08-AFC1-PB) is inconsistent in its handling of variables when the mode (integer of floating point) associated with the variable name differs from the mode specification in a FORMAT statement. Several examples will illustrate this inconsistency:

A)
```
1;    FORMAT (E)
      ACCEPT 1,I
      TYPE   1,I
```

In the above example, if the number 1.23 followed by a terminator is typed in response to the ACCEPT statement, the decimal part will be lost and the number will be stored as an integer on input (only one word of storage will be allocated the variable I - the number will be input as a floating point number, but then it will be converted to fixed point and stored as an integer). On output, the print-out will be:

```
+0.100000E+1
```

so the integer mode associated with the variable name I has taken precedence over the format specification on input but not on output.

B) In the case

```
1;    FORMAT (I)
      ACCEPT 1,A
      TYPE   1,A
```

the integer mode specification in the FORMAT statement takes precedence on input and output.

C) Inputting the number 1.33 followed by a terminator to the following:

```
1;    FORMAT (I,E)
      ACCEPT 1,A,B
      TYPE   1,A,B
```

will yield as output: +1 +0.330000E+2

for the decimal point is recognized only as a terminator of the input for the integer variable A (FORMAT statement has precedence as in example B, above), and 33 is taken as input for the floating point variable B. Had the program looked as follows:

```
1;    FORMAT (E,I)
      ACCEPT 1,I,J
      TYPE   1,I,J
```

then inputting 1.33 followed by a terminator would not satisfy the ACCEPT statement, for 1.33 would be read in as a floating point number, then converted to fixed point and stored as integer +1 (as in example A), the operating system would then still be waiting for more input; if the number 2 is now typed, followed by a terminator, the ACCEPT statement would be satisfied and output would be as follows:

```
+0.100000E+1   +2
```

(again output for the variable I reacts as described in example A).

4K FORTRAN Compiler (DEC-08-AFC1-PB)

Patch to correct diagnostic for overlap of program and data requirements.

The library version of the 4K Fortran Compiler incorrectly prints out the diagnostic Code Ø6 when the user's program is too large and overlaps storage set aside for user data. The correct diagnostic is 1Ø. The compiler may be patched as follows to type out the correct code:

LOCATION	OLD CONTENTS	SHOULD BE
4275	(7001)	7006

PDP-8 FORTRAN-D (DEC-D8-AFA1-6)

Suggested method for saving FORTRAN-D object programs.

The following method for saving FORTRAN-D object programs enables the user to call his program with the FORTRAN-D operating system as a Monitor system program.

1. Compile the FORTRAN-D source program as usual.

2. Load and run the compiler output under FOSL.

3. At the end of a successful run, control will return to the Monitor. The user should then call FOSL to reload the compiler output as if to run the program again.

4. When FOSL has loaded the program (and subroutines, if any) it will type

   READY

   The user should type CONTROL C (↑C) to return to the Monitor.

5. A core image of the user's program in FOSL may not be saved on disk as a system program by typing

   SAVE NAME!Ø-7577;5Ø43

   Note that the starting address must be 5Ø43.

6. When the Monitor has returned control to the user, the core image may be called at any time as a system program. On starting, the program will type

   READY
   ↑

7. CONTROL P (↑P) will initiate execution. When execution is complete the program will type "!" and return to the Monitor.

Users who run FORTRAN-D with binary subroutines will find this method especially time saving.

PDP-8 FORTRAN-D (DEC-D8-AFA2)

Patch to correct problem in floating point input routine.

The floating point input conversion routine in .FT. fails to correctly compile decimal numbers which have more than six significant digits and are above a certain magnitude. This problem results from an error in masking to check whether accepting another decimal digit will cause an overflow. The mask is now set at 7400 (octal) and, therefore, applies to the top four bits of the double precision word. However, the top five bits should be interrogated as the leading sign bit must not be changed. To correct this situation, the mask at location 4715 (octal) should be set to 7600 (octal). A similar conversion routine in the operating system does use a mask of 7600. This patch will be implemented in the next revision of FORTRAN-D. Until such time see the article DISK MONITOR SYSTEM 4 for a description of how to make the patch to location 4715 of the Compiler (.FT.).

PDP-8 FORTRAN-D (DEC-D8-AFA3 and AFA4-PB)

Patch to allow non-system DECtape I/O.

In order to perform I/O on a DECtape other than that being used as the system device, it is necessary to make the patch given below to FOSL and .OS. of the Fortran-D Operating System.

Without this patch, I/O can only be handled via the teletype, the high speed reader/punch, or the system device. With the patch, I/O is enabled on DECtape unit 7 rather than on the DECtape specified as the system device.

The procedure for loading the Operating System, patching it, and saving it is outlined as follows.

1. Load the operating system loader (FOSL) using Loader in one pass with a 7636 argument to ST=.

```
.LOAD ↵
*IN-R: ↵
*
*OPT-1
*ST=7636 ↵
↑↑
```

After FOSL has been completely loaded, the Loader will halt at location 7636 enabling you to enter the following patch through the switch register.

Instructions to allow reading from DECtape 7.

LOCATION	NEW CONTENTS	
166	27ØØ	
27ØØ	1377	
27Ø1	3766	
1224	7Ø3	/DECtape unit 7 input (603 for input from unit 6)

Instructions to allow writing onto DECtape 7.

27Ø2	1375	
27Ø3	3774	
27Ø4	5773	
1324	7Ø5	/DECtape unit 7 output (605 for output to unit 6)

Storage cells for the above changes:

LOCATIONS	NEW CONTENTS
2773	263
2774	5353
2775	705
2776	5331
2777	7Ø3

Once the patch has been made, load address 76ØØ and press START to return to the monitor. FOSL may then be saved in the following manner.

```
.SAVE FOSL! Ø-1577,27ØØ;2ØØ↵
```

2. The operating system interpretive and arithmetic package (.OS.) should now be loaded by using the Loader in one pass and specifying 7636 as the argument to ST=.

```
.LOAD ↵
*IN-R:↵
*
*OPT-1
*ST=7636↵
↑↑
```

The loader will again halt at location 7636 and the following patch must be made to allow writing on DECtape 7.

LOCATIONS	NEW CONTENTS
5155	1377
5177	7Ø3

When the patch has been made, load address 76ØØ and press START. Control will return to the monitor and you may now save .OS. as .SAVE .OS.! Ø-5177; The following page shows an example program and its results to prove this new method of DECtape I/O.

```
*L
            DEFINE DECTAPE
            WRITE 3,9
            FAC=1
            DO 1Ø N=1,2Ø
            WRITE 3,2Ø,N,FAC
            EN=N+1
            FAC=EN*FAC
1Ø          CONTINUE
            STOP
2Ø          FORMAT (,E)
9           FORMAT ("FACTORIAL PROGRAM"/)
            END
*
.FORT
*OUT-S:TE
*
*IN-S:TES1
*
↑                   /COMPILATION FINISHED.  CNTRL C WAS THEN TYPED
.FOSL
*IN-S:TE
*
*OPT-S
*OUT-D7:IO           /DECTAPE 7 MUST ALWAYS BE SPECIFIED FOR I/O
*
*IN-
*
*READY
↑                   /PROGRAM WAS EXECUTED BY TYPING CNTRL P
!
.PIP                /CALLED IN PIP TO LIST THE FILE WRITTEN ON TAPE 7
*OPT-A              /BY OUR FORTRAN PROGRAM
*OUT-T:
*
*IN-D7:IO
*FACTORIAL PROGRAM

1           Ø.1ØØØØØE+1
2           Ø.2ØØØØØE+1
3           Ø.6ØØØØØE+1
4           Ø.24ØØØØE+2
5           Ø.12ØØØØE+3
6           Ø.72ØØØØE+3
7           Ø.5Ø4ØØØE+4
8           Ø.4Ø32ØØE+5
9           Ø.36288ØE+6
1Ø          Ø.36288ØE+7
11          Ø.399168E+8
12          Ø.479ØØ1E+9
13          Ø.6227Ø2E+1Ø
14          Ø.871782E+11
15          Ø.13Ø767E+13
16          Ø.2Ø9227E+14
17          Ø.355686E+15
18          Ø.64Ø236E+16
19          Ø.121644E+18
2Ø          Ø.243289E+19
*OPT-
```

PDP-8 FORTRAN-D (DEC-D8-AFA1-PB)

Problem with storage of SAM BLOCK numbers.

In Disk Fortran both the Compiler and the Operating System exist as two part tapes which are stored on the Disk as four unique files. The Compiler Loader, or driver, is saved on the system device under the file name FORT while the main body of the Compiler is saved under the file name .FT. and is located and loaded by FORT whenever necessary. Similarly, the Operating System loader, or driver, is saved under the file name FOSL while the main body of the Op. Sys. is saved under the file name .OS. and it is located and loaded by FOSL whenever necessary.

In order to locate their associated files, FORT and FOSL do a directory look up for the required name (FORT will look up .FT. and FOSL looks up .OS.). In doing this both FORT and FOSL assume that the first SAM BLOCK number is stored in the third word of each DN (directory name) block. The Monitor, however, stores this number in the third word of the first DN block only (block 177).

If .FT. is not one of the first twenty-five (decimal) files saved on the system, its Directory Name entry will not be in the first DN block. When FORT finds the DN entry for .FT. it will assume the third word of the same DN block to contain the block number of the first SAM BLOCK when in truth it does not. FORT will therefore be unable to load .FT. and the compiler will not run. A similar situation is encountered by FOSL if .OS. is not among the first 25_{10} files saved.

To correct this situation both FORT and FOSL should be made to use a constant 2ØØ as the first SAM BLOCK number (which it always is). This is done as follows:

Load the Fortran Compiler Loader (FORT) and change locations:

LOCATION	FROM	TO
1456	1672	1272
1472	7402	Ø2ØØ

Save the Compiler Loader as usual.[1] Load and save the Compiler main body as usual.[1] Load the Fortran Operating System Loader (FOSL) and change:

1Ø56	1672	1272
1Ø72	46Ø2	Ø2ØØ

Save the OP. Sys. Loader as usual.[1] Load and save the OP. SYS. main body as usual.[1]

Both the Compiler and the Operating System will now function properly no matter which DN block they are entered in when saved.

[1] See DEC-D8-SDAB-D for complete instructions.

(See the article DISK MONITOR SYSTEM 4 for a complete description of how to patch a system program, also see the Disk Monitor Manual DEC-D8-SDAB-D.)

PDP-8 FORTRAN-D (DEC-D8-AFA2)

Patch to correct two error conditions during compilation time.

The following two error conditions are not properly flagged during compilation time.

1. In a GO TO statement the absence of a comma between the closing parenthesis and the operand may or may not produce an error message, depending on the second character of the variable. If this character is a floating point designator, the error will be flagged as such. But if it is a fixed point designator, no error will be detected.

   The following patch will correct this problem by adding a new syntax error code Ø2.

```
6Ø24    5357            JMP PATCH1

6157    15Ø1    PATCH1,TAD I Z END      /GET NEXT CHARACTER
616Ø    21Ø1            ISZ Z END      /INDEX TO NEXT
6161    1367            TAD MCOMMA     /CHECK FOR
6162    765Ø            SNA CLA        /COMMA
6163    5225            JMP BACK       /O.K., CONTINUE
6164    4454            ERROR          /NOT FOUND
6165    4ØØ2            4ØØ2           /SYNTAX ERROR
6166    5446            JMP I Z NEWL   /GET NEXT LINE
6167    7524    MCOMMA,-254            /COMMA
```

2. If an apostrophe (single quote) accidentally appears in any statement, an error message (usually 14, system overflow) may or may not be generated depending on where the apostrophe appears. The apostrophe at the end of a line signifies a continuation of the statement to the next line. (Maximum number of characters must not exceed 128.) However, if it appears anywhere except before the carriage return at the end of a line it should be flagged as an illegal character (code 12).
   The following patch corrects this problem:

```
Ø277    5365            JMP PATCH

Ø365    1363 PATCH      TAD M215       /CHECK TO SEE IF NEXT
Ø366    765Ø            SNA CLA        /CHAR IS A CARRIAGE RETURN
Ø367    5217            JMP FILB       /YES, O.K., CONTINUE
Ø37Ø    2145            ISZ Z STMT     /NO, INDEX STATEMENT NO.
Ø371    7ØØØ            NOP
Ø372    4454            ERROR          /SINGLE QUOTE IS
Ø373    4Ø12            4Ø12           /ILLEGAL CHARACTER
Ø374    5342            JMP OVFL+4     /IGNORE REST OF STATEMENT
```

PDP-8 FORTRAN-D COMPILER (DEC-D8-AFA2-PB)

Notice of temporary restriction in Subscript Expressions

Subscript expressions which appear in input/output statements, such as:

```
READ  3,1ØØ,     LIST (expression)

WRITE 2,77,      PACK (expression)
```

may only be of the form (V), (C), (V+C or (V-C), where V is a simple integer variable and C is an unsigned interger constant.

Notice that more elaborate subscript expressions may be used outside of I/O statements, permitting the user to redefine the above sequence in a manner such as:

```
L=(expression 1)
K=(expression 2)
READ  3,1ØØ,     LIST (L)
WRITE 2,77,      PACK (K)
```

or even:

```
LL=LIST (expression)
KK=PACK (expression)
READ  3,1ØØ,     LL
WRITE 2,77,      KK
```

MACRO (DEC-08-CMA1 and CMA2)

Note on reading symbol table tapes

Pages 11 and 25 of the DDT-8 manual describe how to enter a user's symbol definitions into the DDT-8 external symbol table for use in a debugging run.

The description on page 11 pertains to reading in the symbol table tape which is part of the output from PAL III or MACRO-8. This description is correct and the user will encounter no difficulty in reading in the symbol table tape if it was produced by PAL III.

However, MACRO-8 symbol table tapes do not contain the proper code to signal the end of tape to DDT-8. If the user wishes to read symbol table tapes produced by MACRO-8 into DDT-8's external symbol table, he may manually type in the EOT character, after the tape has been read through the reader and the reader has stopped, by depressing and holding the CTRL key while hitting the D key. The user should then turn off the reader and hit the CONTINUE key to proceed with the debugging.

MACRO-8 may be patched as follows to produce the proper end of tape code.

a) load MACRO-8 with the Binary Loader as usual
see (DEC-08-LBAA-D or DEC-08-NGCC-D)

b) set switch register to 2516 press LOAD ADDRESS

c) set switch register to 1376 press DEPOSIT

d) set switch register to 2576 press LOAD ADDRESS

e) set switch register to 0204 press DEPOSIT

PDP-8 MASTER TAPE DUPLICATOR (Digital 8-16-S)

Note to users

The Master Tape Duplicator (Digital 8-16-S) does not check for extra blank frames in the duplicate tape.

Users having difficulty with binary tapes which load and verify properly but do not run properly, should order tape and document number 5-10 from the DECUS Program Library. The program name is Paper Tape Reader Tester. It is a program for the PDP-5 but will run on the PDP-8 and should be used as a second verifying operation. Programs which are too long for the space left in core should be broken into two or more shorter tapes for this operation.

ıission of tag in Direct Assignment Statement

The omission of the tag in a direct assignment statement will cause various errors in PAL III depending upon the placement of the statement. If a statement like =3 occurs before any symbol has been used in a program which is assembled by PAL III, PAL will type out a REDEFINITION diagnostic during PASS 1, with a meaningless printout for the symbol being redefined. If the statement =3 occurs after a symbol has been used, PAL assumes that the last symbol referenced is being redefined. For instance

```
.
.
.
.
*2ØØ
CLA
=3
.
.
.
```

will cause the diagnostic

```
RD      CLA      AT      Ø2Ø1
```

during PASS 1. In either case PAL III may be restarted after PASS 1, but attempting to continue to PASS 2 will leave PAL in a state from which it can neither be continued for PASS 3 nor restarted for another PASS 1.

PAL-D (DEC-D8-ASAC-PB)

Recognition of memory reference instructions.

The newest version of the PAL-D Disk Assembler uses a slightly different criterion for recognizing memory reference instructions than was used by previous versions.

Given a line of code like:

```
SYMB1          SYMB2
```

where neither SYMB1 nor SYMB2 is an assembler pseudo-op, PAL-D will treat this line as a memory reference instruction only if SYMB1 is a permanent symbol and SYMB2 is not a permanent symbol. Formerly if SYMB2 was not a permanent symbol, the line was treated as a memory reference instruction, regardless of whether or not SYMB1 was a permanent symbol. Note that the category "permanent symbol" includes user-defined symbols that have been added to the PAL-D permanent symbol table via the FIXTAB pseudo-op.

All other combinations of permanent and non-permanent symbols for SYMB1 and SYMB2 will cause the line to be treated as a microinstruction. The values of SYMB1 and SYMB2 will be inclusively ORed together and this resultant value will be generated as code for the line. Because of these considerations, users should be cautious in their use of the FIXTAB pseudo-op.

Note especially that the floating point instruction (FADD,FSUB,...., FPUT) must be defined at the beginning of a program and must be followed by a FIXTAB pseudo-op, as follows:

```
FADD = 1ØØØ
FSUB = 2ØØØ
  .
  .
  .
  .
FPUT = 6ØØØ
FIXTAB
```

PDP-8 DISK MONITOR SYSTEM REFERENCE MANUAL (DEC-D8-SDAB-D)

Additional information on copy functions of PIP

The Peripheral Interchange Program of the Disk/DECtape Monitor System (PIP) will transfer files between any of the various devices recognized by the system (i.e. Disk, DECtape, High Speed Paper Tape Reader, High Speed Paper Tape Punch, Teletype Reader, Teletype Punch).

All of the transfer or copy functions use the same conventions to specify the destination (output) and the origin (input) of the file to be copied, and to start the actual transfer once the necessary information has been given. These conventions are essentially:

1) Use the device name followed by the file named to reference any file which exists (input) or is to exist (output) on a file structured mass storage device. For the system device (Disk if present or DECtape) this name is S:
For any non-system DECtape this name is D1:, D2:, D3: etc. depending upon the physical unit setting of the hardware.
The file name consists of up to four ASCII characters.

2) Use only the device name to reference any file which exists or is to exist on a non-file-structured device. For both the high speed paper tape reader and punch this name is R:
For both the teletype reader and punch this name is T:

3) To indicate more than one file (legal only as input to A option; used for merging ASCII files), reference the files by the above rules and use commas as separators.

```
ex      S:FILA,    S:FILB,    D4:FILN
        R:, R:,    T:, R:
```

4) Type CTRL/P (↑P) to start the actual transfer of each file. This is done by depressing the CTRL key while striking the P. PIP will type ↑'s to remind you to type CTRL/P to start transfers if the copy operation is between disk and DECtape, disk and high speed paper tape, DECtape and high speed paper tape, or if it uses the teletype for input but not for output.

If, however, output is to the teletype punch, it is imperative that the user remember the convention and type CTRL/P to start the transfers. PIP cannot type an ↑ as a reminder because it would be punched on the output tape as well as typed. (CTRL/P is not echoed so it won't be punched.)

EXAMPLE:

```
*OPT-A
*OUT-T:↵

*IN-S:FILE↵

A pause occurs here as PIP is waiting for the user
to type CTRL/P

*OPT-A
*OUT-R:↵
*
*IN-S:FILE↵
*↑

A pause occurs here after PIP has reminded the user
it is waiting for CTRL/P.
```

The purpose of the pause is to allow the user time to turn on the punch he wants to use, or to take a DECtape off write lock, or to place a tape in the reader. The copy routine will not proceed until a CTRL/P has been typed.

PDP-8 DF32/RFØ8 PIP (DEC-D8-PDAD and DEC-D8-PDZE)

Information on performance of latest versions of PIP.

PIP can only copy relocatable binary tapes (output of the SABR Assembler) from reader to punch. PIP cannot copy relocatable binaries to or from a mass-storage device (disk or DECtape).

PIP will only copy the first section of a multi-section binary tape when output is to the disk or DECtape, even though the entire tape will go through the reader. A multi-section binary tape is one that has leader/trailer (code 2ØØ) imbedded. The binary copy of such a tape to the mass-storage device will cease once the leader/trailer is encountered, even though PIP will not stop reading the tape until it has received a physical end of tape signal from the reader.

PIP may insert an extraneous character when merging a paper tape with a disk or DECtape file, when the output is going to disk or DECtape. This can happen when the end of the paper tape is jagged and it is interpreted as punches in the tape. It can also be caused by the fact that PIP treats input, character by character when it is from a non file-structured device (teletype or high speed reader) and word by word when it is from a file-structured device (disk or DECtape). Thus, if there is an odd number of characters from the non file-structured device, one half of a word in the output file will be filled with either an @ character or a control character. This can be avoided by first transferring the paper tape to the system device before merging the files.

PDP-8 PIP (DEC-D8-PDAD and DEC-D8-PDZE)

Program halt due to unexpected interrupt

The above noted library versions of PIP will halt at location Ø226 with PC=Ø227, MA=Ø226, MB=7412 and AC=ØØØØ, upon receipt of an Unexpected Interrupt. An example of this may be an interrupt from a device other than Disk, DECtape, High Speed Reader or Punch or Keyboard Reader or Punch. Users should be able to eliminate the problem by starting the Monitor at 76ØØ and calling PIP again (start key will clear the flags).

8K SABR ASSEMBLER (DEC-08-A2D2-PB)

Problem in Listing PASS with OFF-PAGE Externals

There is a problem in the library version of the SABR assembler (version 16), which can cause SABR not to list a linkage to an off-page external symbol. The binary output of pass 1 contains the correct link-word, but the word is not listed in the pass 2 output. The following patch will correct this problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
3735	(ØØ16)	Ø377
3777	(unused)	Ø4Ø3

Problem with the work area

The EDITOR, as currently implimented, allows a source file of more than 1ØØ blocks of teletype input. If the user types in such a file, block 1Ø1 will be written onto LINCtape block 47Ø; block 1Ø2 onto 471, etc. In doing this, the Editor will destroy any previously stored data in blocks 47Ø+.

A. If the user has created a LAP6-DIAL-MS tape, read block 3Ø2 into core. Change the 23rd word of that block (word Ø = Ø, word 1 = 1, etc).

FROM: Ø312

TO: 1311

then write the block back again.

B. If the user has not created a LAP6-DIAL-MS tape, display the index of DEC-12-SE2D-UO (the LINCtapes containing the binaries of DIAL-MS) and find the starting block number of DIAL-MS 1. Call it FB. Read into core, block FB+2, and change the 23rd word of that block:

FROM: Ø312

TO: 1311

then write the block back again.

PDP-12 LAP6-DIAL EDITOR

Preserving Pointers Upon Restart or Exit for LAP6-DIAL-V2 or LAP6-DIAL-MS.

The routine that preserves the integrity of the EDITOR's pointers upon restart or exit, are improper. To correct this problem, the following must be done:

FOR LAP6-DIAL-V2:

Read the LINCtape block 312 of a LAP6-DIAL-V2 tape, and change the following words:

WORD	FROM	TO
132	1766	7ØØØ
133	741Ø	7ØØØ
134	1767	7ØØØ

Now write the block back out.

FOR LAP6-DIAL- MS:

Read in the LINCtape block 312 of a LAP6-DIAL-MS tape, and change the following words:

WORD	FROM	TO
136	15Ø1	7ØØØ
137	741Ø	7ØØØ
138	15Ø3	7ØØØ

Now write the block back out.

Allowing input into a line containing 60 or more characters

Both the LAP6-DIAL-MS and the LAP6-DIAL-V2 EDITORS, will not allow input into the beginning of a line which has 60 (sixty) or more characters. To correct this problem, do the following:

LAP6-DIAL-V2

Read in block 313 of a DIAL-V2 LINCtape and change word 217 from Ø467 to 7633 and write the block out.

LAP6-DIAL-MS

Read in block 313 of a DIAL-MS LINCtape and change word 22Ø from Ø456 to 7634 and write out the block.

Incorrect placement of a comma

Assume that while editing a LAP6-DIAL-MS program, a mistake is discovered. After turning the cursor back to the mistake in line five (5), the scope appears as follows:

```
1  DIAL,    SET I 16
2           RCG
3           SET I 17
4           73ØØ
5           TAG LDA I
```

Attempting to insert a comma after the G in TAG will cause the comma to be placed after the I in LDA I, and the scope will appear as follows:

```
1  DIAL,    SET I 16
2           RCG
3           SET I 17
4           73ØØ
5  TAG, LDA I,
```

To correct this problem, the following should be done:

> Set the left switches to Ø7ØØ; set the right switches to Ø313; press I/O preset; press the DO key. Change location 4115 from 74Ø1 to 7237. Set the left switches to Ø7Ø4 and the right switches to Ø313; press the DO key.

The above assumes you have already created a DIAL-MS system using GENASYS.

The basic list command for DIAL is:

→ LI (Line Number 1, Line Number 2) (Name, Unit)

However, if the user tries to input a non-standard command, (ie. LI 7ØØØ,3ØØØ) the check for this error will fail.

To correct this problem, change the following locations in block #306 of a DIAL-MS tape:

WORD	FROM	TO
51	Ø641	ØØØ2
52	Ø6Ø2	5653
53	62Ø7	7741

The basic list command for DIAL is:

→ LI (Line Number 1, Line Number 2) (Name, Unit)

However, if the user tries to input a non-standard command, (ie. LI 7ØØØ,3ØØØ) the check for this error will fail.

To correct this problem, change the following locations in block #306 of a DIAL-MS tape:

WORD	FROM	TO
51	Ø641	ØØØ2
52	Ø6Ø2	5653
53	62Ø7	7741

LAP6-DIAL-MS EDITOR

To Correct problem with EDITOR software flag

To correct a problem with one of the EDITOR'S software flags, make the following changes to DIAL-MS or DIAL V2 system tape.

DIAL-MS source change--EDITOR2 (DEC-12-ZR7A)

Change:

LINE NUMBER	FROM	TO
4353	JMP CHKBUF	JMP PATCH1

Add:

LINE NUMBER	FROM	TO
4654	unused	LMODE
4655	unused	PATCH1, JMP CHKBUF
4656	unused	LDA I
4657	unused	1
466Ø	unused	STC PLAFLG
4661	unused	JMP H8+2

A created DIAL-MS system tape

Change:

TAPE BLOCK	WORD	FROM	TO
321	111	6112	7753

Add:

TAPE BLOCK	WORD	FROM	TO
321	353	unused	6112
321	354	unused	1Ø2Ø
321	355	unused	ØØØ1
321	356	unused	4347
321	357	unused	7512

DIAL-V2 Source change-- ADD PROGRAM (DEC-12-ZW8A)

Change:

LINE NUMBER	FROM	TO
Ø277	JMP CHKBUF	JMP PATCH1

Add:

LINE NUMBER	FROM	TO
Ø542	unused	PATCH1, JMP CHKBUF
Ø543	unused	LDA I
Ø544	unused	1
Ø545	unused	STC PLAFLG
Ø546	unused	JMP H8+2

DIAL-V2 system tape

Change:

TAPE BLOCK	WORD	FROM	TO
321	111	6017	7733

Add:

TAPE BLOCK	WORD	FROM	TO
321	333	unused	6107
321	334	unused	1Ø2Ø
321	335	unused	ØØØ1
321	336	unused	435Ø
321	337	unused	7512

LAP6-DIAL-MS EDITOR

ADD PROGRAM by line number problem

If two or more sucessive "ADD PROGRAM BY LINE NUMBER" commands are given using the LAP6-DIAL-MS EDITOR, the EDITOR(DEC-12-ZR7A) will crash. To correct this problem, make the following changes.

DIAL-MS source change -- EDITOR2:

Change:

LINE NUMBER	FROM	TO
4345	SAE	NOP
4346	LN2	NOP
4347	SKP	NOP
435Ø	JMP J8	NOP
4361	JMP F8	JMP PATCH2
4372	JMP F8	JMP PATCH2

Add;

LINE NUMBER	FROM	TO
4662	unused	PATCH2, LDA
4663	unused	LN1
4664	unused	SAE
4665	unused	LN2
4666	unused	JMP F8
4667	unused	JMP J8

A created DIAL-MS system tape:

Change:

TAPE BLOCK	WORD	FROM	TO
321	104	1440	ØØ16
321	105	1745	ØØ16
321	106	Ø456(or 0467)	ØØ16
321	107	7536	ØØ16
321	116	7465	776Ø
321	127	7465	776Ø

Add:

TAPE BLOCK	WORD	FROM	TO
321	36Ø	unused	1ØØØ
321	361	unused	1744
321	362	unused	144Ø
321	363	unused	1745
321	364	unused	7465
321	365	unused	7536

DIAL-V2 source change-- ADD PROGRAM (DEC-12-ZW8A)

Change:

LINE NUMBER	FROM	TO
0271	SAE	NOP
0272	LN2	NOP
0273	SKP	NOP
0274	JMP J8	NOP
0305	JMP F8	JMP PATCH2
0317	JMP F8	JMP PATCH2

Add:

LINE NUMBER	FROM	TO
0547	unused	PATCH2, LDA
0550	unused	LN1
0551	unused	SAE
0552	unused	LN2
0553	unused	JMP F8
0554	unused	JMP J8

A DIAL-V2 system tape:

Change:

TAPE BLOCK	WORD	FROM	TO
321	104	1440	ØØ16
321	105	1731	ØØ16
321	106	0467	ØØ16
321	107	7536	ØØ16
321	116	7465	774Ø
321	127	7465	774Ø

Add:

TAPE BLOCK	WORD	FROM	TO
321	34Ø	unused	1ØØØ
321	341	unused	173Ø
321	342	unused	144Ø
321	343	unused	1731
321	344	unused	7465
321	345	unused	7536

LAP6-DIAL-MS EDITOR

ADD PROGRAM by block number

As currently implemented, the LAP6-DIAL-MS EDITOR does not check the following command:

→ AP XXXX,U,

In this case, XXXX represents a four digit block number rather than a three digit block number; U is unit number. To correct this problem, make the following changes.

DIAL-MS Source change--EDITOR2

LINE NUMBER	FROM	TO
4274	E8, LDA	E8, JMP PATCH3

ADD:

LINE NUMBER	FROM	TO
4670	unused	PATCH3, LDA
4671	unused	E6+2ØØØ
4672	unused	BCL I
4673	unused	-7ØØØ
4674	unused	AZE
4675	unused	JMP K8
4676	unused	LDA
4677	unused	E6+2ØØØ
47ØØ	unused	JMP E8+2

Change for a created DIAL-MS system tape

Change:

TAPE BLOCK	WORD	FROM	TO
321	Ø34	1ØØØ	7766

Add:

TAPE BLOCK	WORD	FROM	TO
321	366	unused	1ØØØ
321	367	unused	2371
321	370	unused	156Ø
321	371	unused	Ø777
321	372	unused	Ø45Ø
321	373	unused	7661
321	374	unused	1ØØØ
321	375	unused	2371
321	376	unused	7436

(LAP6-DIAL-MS EDITOR CONT'D)

DIAL-V2 Source change-- ADD PROGRAM (DEC-12-ZW8A)

LINE NUMBER	FROM	TO
Ø22Ø	E8, LDA	E8, JMP PATCH3

ADD:

LINE NUMBER	FROM	TO
Ø555	unused	PATCH3, LDA
Ø556	unused	E6+2ØØØ
Ø557	unused	BCL I
Ø56Ø	unused	-7ØØØ
Ø561	unused	AZE
Ø562	unused	JMP K8
Ø563	unused	LDA
Ø564	unused	E6+2ØØØ
Ø565	unused	JMP E8+2

Change a DIAL-V2 system tape -- ADD PROGRAM (DEC-12-ZW8A)

TAPE BLOCK	WORD	FROM	TO
321	Ø34	1ØØØ	7746

ADD:

TAPE BLOCK	WORD	FROM	TO
321	346	unused	1ØØØ
321	347	unused	2371
321	35Ø	unused	156Ø
321	351	unused	Ø777
321	352	unused	Ø45Ø
321	353	unused	7655
321	354	unused	1ØØØ
321	355	unused	2371
321	356	unused	7436

ADD PROGRAM by block number

As currently implemented, the ADD PROGRAM feature of the EDITOR fails to check for the end of the tape while doing an ADD PROGRAM near or at block 777. This problem can not be corrected by a binary patch for DIAL-MS version 1. Shown below are both the source and binary patch for DIAL-V2.

DIAL-V2 source change --ADD PROGRAM (DEC-12-ZW8A)

Change:

LINE NUMBER	FROM	TO
0334	JMP DTEXTR-1	JMP PATCH4

Add:

LINE NUMBER	FROM	TO
0565	unused	PATCH4, LDA
0566	unused	TPWD
0567	unused	BCL I
0570	unused	-7ØØØ
0571	unused	SAE
0572	unused	MBLK2
0573	unused	JMP J8
0574	unused	JMP DTEXTR-1
0575	unused	JMP MAGTP

A DIAL-V2 system tape

Change:

TAPE BLOCK	WORD	FROM	TO
321	151	77ØØ	7757

Add:

TAPE BLOCK	WORD	FROM	TO
321	357	unused	1ØØØ
321	36Ø	unused	1553
321	361	unused	156Ø
321	362	unused	Ø777
321	363	unused	146Ø
321	364	unused	2ØØØ
321	365	unused	7536
321	366	unused	77ØØ
321	367	unused	7552

PDP-12

DIAL-MS (Version 1) SOURCE TAPES (DEC-12-SEZA-UO)

The DIAL-MS (Version 1) Source Tapes (available by request from the Program Library) will not generate the same binary as the binaries which are being distributed by the Program Library (DEC-12-SE2D-UO).

In order to assemble the DIAL-MS Sources, the pseudo-op CHAIN, is necessary and is present in the sources. However, in the DIAL-MS System Tape (DEC-SE2D-UO), the CHAIN pseudo-op was NOP'ed out of block #343.

WORD	SE2D	SEZA
0	7777	6167
1	7777	ØØ56
2	7777	1ØØ6
3	7777	2536

Also, the patch to correct the improper placement of the comma in the LAP6-DIAL Editor (article #LAP6-DIAL EDITOR 8) was left in the source tapes (not the SYSTEM'S tape). DIAL-MS-System tape block #313

WORD	SE2D	SEZA
115	7401	7237

Both of these problems will be corrected in the next submission of DIAL-MS to the Program Library.

PDP-8 Hardware design note for the DF32 disk

Programming note concerning the DEAL instruction.

It is now possible for bit Ø of the AC to be set by the photocell sync mark when using DF32 Dick instruction, DEAL (6615). If the user expects the AC to be unchanged, this could affect his program.

Therefore, the user should not write his program such that it depends on the contents of the AC remaining unchanged after a DEAL instruction. Page 2-4 of the DF32 Disk File and Control Instruction Manual specifies only the contents of AC bits 1-8 and claims only these bits will remain unchanged. If the information in the AC is to be used after a DEAL, bits 0 and 9-11 should first be masked out.