```
/FOCAL SEMINAR NOTES







/FOCAL SEMINAR NOTES

/BY:       JIM CRAPUCHETTES
/          FRELAN ASSOCIATES
/          P/O.  BOX 298
/          MENLO PARK, CALIF. 94025
/
/
/          THESE NOTES ARE PROPRIETARY AND ARE NOT
/   TO BE COPIED WITHOUT THE WRITTEN APPROVAL OF
/   THE AUTHOR.  THEY ARE FURTHERMORE TO BE USED
/   ONLY BY THE PARTY TO WHICH THEY ARE SOLD AND
/   TO THEIR IMMEDIATE CO-WORKERS.
```

```
        /DEFINITIONS FROM FOCAL:

        /       THE FOLLOWING METHOD OF DEFINITION WILL WORK
        /  FOR THE FOLLOWING ASSEMBLERS: MACRO-8, PAL-D, PAL-D
        /  (8K VERSION) AND PAL8.
        /
        /       FOR THE VERSIONS OF THE PAL-III ASSEMBLER, ONE
        /  OF THE FOLLOWING MUST BE DONE:
        /
        /  1.   IF THE ASSEMBLER ALREADY HAS FLOATING-POINT
        /       INSTRUCTIONS DEFINED FOR THE STANDARD FLOATING-
        /       POINT PACKAGE, THEY MAY BE SIMPLY RE-DEFINED
        /       FOR THE PROPER OP-CODES.
        /
        /  2.   OTHERWISE, THE "EXPUNGE" PSEUDO-OP MUST BE USED
        /       FOLLOWED BY THE DEFINITION OF ALL MEMORY-REF.
        /       INSTRUCTIONS (AND, TAD, ETC...), THE FLOATING-
        /       POINT INSTRUCTIONS (BELOW), AND THE "FIXMRI"
        /       PSEUDO-OP.  FOLLOWING THIS, ALL THE REST OF THE
        /       PDP-8 INSTRUCTIONS MUST BE DEFINED!!!  UN-
        /       FORTUNATELY, THIS IS THE ONLY WAY...

  0000  FIXMRI  FGET=   0000    /PSEUDO-F.P. INSTRUCTIONS
  1000  FIXMRI  FADD=   1000
  2000  FIXMRI  FSUB=   2000
  3000  FIXMRI  FDIV=   3000
  4000  FIXMRI  FMUL=   4000
  5000  FIXMRI  FPOW=   5000
  6000  FIXMRI  FPUT=   6000

  7000  FNOR=   7000
  0000  FEXT=   0000
  4407  FINT=   JMS I 7

        IFNDEF  P13     <P13=   5>      /FOR DEBUG UNDER OS/8 ODT
  0011  XRT=    11
  0012  XRT2=   12
  0013  PDLXR=  13
  0014  FLTXR=  14
  0015  FLTXR2= 15
  0017  AXOUT=  17
  0020  XCT=    20
  0022  PC=     22
  0023  THISLN= 23
  0025  LASTLN= 25
  0030  PT1=    30
  0032  T1=     32
  0033  T3=     33
  0035  BOTTOM= 35
  0040  EX1=    40      /F.P. OPERAND
  0041  AC1H=   EX1+1
  0042  AC1L=   EX1+2
  0043  OVER1=  EX1+3
  0044  FLAC=   44
  0044  EXP=    FLAC    /F.P. AC
```

```
      0045  HORD=   FLAC+1
      0046  LORD=   FLAC+2
      0047  OVER2=  FLAC+3
      0051  MINSKI= 51
      0052  FISW=   52      /FORMAT DESCRIPTION WORD
      0053  INTEGE= 53
      0054  SORTCN= 54
      0057  CNTR=   57

      0063  OUTDEV= 63      /POINTER TO OUTPUT SUBROUTINE
      0064  INDEV=  64      /POINTER TO INPUT SUBROUTINE

      0065  NAGSW=  65
      0066  CHAR=   66
      0067  LINENO= 67
      0070  GINC=   70
      0071  T2=     71
      0077  CCR=    77      /C.R. CODE
      0102  PER=    102
      0104  P7600=  104
      0105  M20=    105
      0106  P177=   106
      0107  P17=    107
      0111  M2=     111
      0112  MINUSA= 112
      0113  C260=   113
      0114  M240=   114
      0115  MPER=   115
      0117  MFLT=   117
      0122  P77=    122
      0123  C200=   123
      0127  DOUBLE= 127
      0133  CFRS=   133
      0136  EFUN3I= 136
      0166  ERRPT=  166     /ERROR ROUTINE ADDRESS

      0177  START=  177
      0420  DO=     420
      0611  PROC=   611
      0614  PC1=    614
      0756  INPUT=  756
      1136  MCOM=   1136
      1377  GLIST=  1377    /SPACE (+ 'TLIST')
      1400  TLIST=  1400    /"," + ";" + C.R.
      1437  GS1=    1437
      1613  EVAL=   1613
      1743  EFUN=   1743
      1754  EFUN2=  1754
      2725  ERROR5= 2725
      5667  MULT10= 5667
      5712  REMAIN= 5712
      5713  DIGIT=  5713
      5715  MULT2=  5715
      5733  DUBLAD= 5733
      6623  ALIGN=  6623
      7335  DNORM=  7335
      7503  LIBRARY=7503


            /NEW INSTRUCTIONS:

      4540  PUSHJ=  JMS I 140       /RECURSIVE SUBROUTINE CALL
      1413  POPA=   TAD I PDLXR     /RESTORE AC
      5541  POPJ=   JMP I 141       /SUBROUTINE RETURN
      4542  PUSHA=  JMS I 142       /SAVE AC
      4543  PUSHF=  JMS I 143       /SAVE GROUP OF DATA
      4544  POPF=   JMS I 144       /RESTORE GROUP OF DATA
      4545  GETC=   JMS I 145       /UNPACK A CHARACTER
      4546  PACKC=  JMS I 146       /PACK A CHARACTER
      4547  SORTJ=  JMS I 147       /SORT AND BRANCH ON AC OR CHAR
      4550  SORTC=  JMS I 150       /SORT CHAR
      4551  PRINTC= JMS I 151       /PRINT AC OR CHAR
      4552  READC=  JMS I 152       /READ DATA INTO CHAR AND PRINT IT
      4553  PRNTLN= JMS I 153       /PRINT C(LINENO)
      4554  GETLN=  JMS I 154       /UNPACK AND FORM A LINENUMBER
      4555  FINDLN= JMS I 155       /SEARCH FOR A GIVEN LINE
      4557  RTL6=   JMS I 157       /ROTATE LEFT SIX
      4560  SPNOR=  JMS I 160       /IGNORE SPACES AND LEADING ZEROS
      4561  TESTN=  JMS I 161       /PERIOD; OTHER; NUMBER
      4562  TSTLPR= JMS I 162       /SKIP IF L-PAR: (, [ OR <
      4563  TSTGRP= JMS I 163       /SKIP IF C(AC) = C(LINENO)
      4564  TESTC=  JMS I 164       /TERM; NUMBER; FUNCTION; LETTER
      4566  ERROR2= JMS I ERRPT     /CALL ERROR SUBROUTINE
```

```
        /FREE LOCATIONS IN FOCAL (1969):

        / LOC(S)        COMMENTS

        /    2          USED ONLY BY PDP-5 INTERRUPT HANDLER.
        /    4          USED BY DEBUGGERS.  IF THIS LOCATION IS
        /               USED, THEN ODT CANNOT BE USED IN DEBUGGING.
        / 11,12,14,15   AUTO-INDEX REGISTERS 'XRT', 'XRT2',
        /               'FLTXR' & 'FLTXR2'.  THESE ARE USED ON
        /               A TEMPORARY BASIS BY FOCAL ROUTINES,
        /               AND CAN BE USED (ON A TEMPORARY BASIS)
        /               IF NO CONFLICTS ARISE WITH THESE ROUTINES
        /               [USE THE CREF LISTING TO DETERMINE THIS].
        / 32,33,71      'T1', 'T3' & 'T2'.  TEMPORARY WORKING
        /               LOCATIONS FOR FOCAL ROUTINES.  SEE THE
        /               COMMENTS ABOVE ON THE AUTO-INDEX REGS.
        /   37          'HINBUF', SEE NOTE 1.
        /  124          'P4000'.  USED ONLY AT TWO LOCATIONS.
        /               SEE PAGE 4 OF THIS LISTING.
        /  126          'PTCH', SEE NOTE 1.  MUST OVERLAY 3001
        /               AND 3002 WITH "NOP"S TO USE.  SEE FOCAL
        /               LISTING.
        /  167 - 174    USED BY 8K PATCH, FREE IN 4K VERSION.
        /  175          USED BY "MODV" (8K) PATCH, OTHERWISE FREE.
        / 1136          'MCOM', SEE NEXT LINE...
        / 1142 - 1157   "FDIS" FUNCTION, SEE NOTE 2.  IF THIS
        /               FUNCTION IS OVERLAID, 'MCOM' ABOVE CAN
        /               ALSO BE USED.
        / 1343 - 1353   "FADC" FUNCTION, SEE NOTE 2.
        / 1354 - 1362   'OUTL' SUBROUTINE, DIRECTLY FOLLOWING
        /               "FADC" CODING.  SEE NOTE 3.
        / 1530 - 1532   'RANO', SEE NEXT LINE...
        / 1553 - 1562   "FRAN" FUNCTION, SEE NOTE 2.  IF THIS
        /               FUNCTION IS OVERLAID, 'RANO' ABOVE CAN
        /               ALSO BE USED.
        / 2414 - 2424   'I33' SUBROUTINE.  SEE NOTE 3.
        / 5167 - 5177   FREE ONLY IF ALL FUNCTIONS ARE KEPT!!
        /               OTHERWISE PART OF THE TEXT-VARIABLE-
        /               P.D.L. AREA.  BE CAREFUL WHEN YOU USE
        /               THIS AREA!
        / 5352 - 5377   FREE ONLY IF "FSIN" & "FCOS" ARE KEPT
        /               OR IF 'BOTTOM' IS SET UP CORRECTLY.
        /               OTHERWISE PART OF THE STORAGE AREA.
        / 5774 - 5777   FREE
        / 6160 - 6177   USED BY 8K PATCH, FREE IN 4K VERSION.
        / 6311 - 6377   H.S. READER ROUTINE, SEE NOTE 1.
        / 7037 - 7062   PART OF THE MULTIPLY ROUTINE, 'DMULT',
        /               NOT USED IN THE 3-WORD FLOATING-POINT
        /               PACKAGE.  THIS AREA SHOULD BE USED ONLY
        /               IF IT IS KNOWN THAT THE 4-WORD PACKAGE
        /               WILL NEVER BE USED.
        / 7322 - 7334   PART OF THE DIVIDE ROUTINE, 'DUBDIV',
        /               NOT USED IN THE 3-WORD FLOATING-POINT
        /               PACKAGE.  NOTE THE RESTRICTION ABOVE.
```

```
                EJECT

          / NOTE 1:  IF NO HIGH-SPEED READER IS USED, THEN LOCA-
          /       TIONS 37, 126, 3000, 3001 AND 6311 - 6377 CAN BE
          /       USED FOR OTHER CODING.  IF THIS IS DONE,
          /       LOCATION 1201 ("*" ROUTINE IN 'COMGO') MUST BE
          /       CHANGED TO 177.  THIS WILL CAUSE AN "*" AS A
          /       COMMAND TO DO THE EXACT SAME THING THAT THE "Q"
          /       COMMAND DOES.

          / NOTE 2:  THESE TWO FUNCTIONS (AND PERHAPS "FSQT",
          /       LOCATIONS 7400 - 7467) CAN BE OVERLAID WITH
          /       OTHER CODING.  IN THIS CASE, THE ADDRESSES OF
          /       THESE ROUTINES IN 'FNTABF' (STARTING AT LOC
          /       374) MUST BE CHANGED TO "ERROR5" SO THAT THEY
          /       CANNOT BE CALLED.

          / NOTE 3:  'I33' AND 'OUTL' WERE SET UP TO ALLOW EASY
          /       DEBUGGING OF FOCAL WITH THE INTERRUPT OFF,
          /       AND WITH THE HELP OF A DEBUGGER.  THIS CAN BE
          /       DONE AS SHOWN IN THE "DEBUG" PATCHES.  IF THE
          /       SPACE IS NEEDED, 'XI33' AND 'XOUTL' COULD BE
          /       OVERLAID WITH ROUTINES EQUIVALENT TO 'I33' AND
          /       'OUTL', AND THE REST OF THE DEBUGGING PATCHES
          /       IMPLEMENTED TO RUN WITH IOF.
```

```
              /PATCHES TO USE "OPR 1" INSTRUCTIONS TO GENERATE
              / CONSTANTS IN THE AC RATHER THAN USE "TAD"S.  THE
              / "TAD"S WERE NEEDED FOR THE PDP-5 COMPUTER.  IF
              / YOUR PATCHES WILL NOT BE RUNNING ON A PDP-5, THESE
              / PATCHES WILL SIGHTLY INCREASE EXECUTION SPEED AND
              / FREE LOCATION 124 FOR OTHER USES.


       0245  *245     /OVERLAY "TAD P4000" - ONLY 2 REFS

00245  7330           STL CLA RAR     /= 4000

       1017  *1017    /OVERLAY "TAD M2"

01017  7344           CLL STA RAL     /= -2

       1631  *1631    /OVERLAY "TAD M2"

01631  7344           CLL STA RAL

       2052  *2052    /OVERLAY "TAD P3" - ONLY REF

02052  7325           STL CLA IAC RAL /= 3, PDP-8I,-8L & -8E ONLY

       3111  *3111    /OVERLAY "TAD M2"

03111  7344           CLL STA RAL

       7127  *7127    /OVERLAY "TAD P4000" - ONLY 2 REFS

07127  7330           STL CLA RAR

              /NOTE: THERE ARE ONLY 2 REFERENCES TO 'P4000' (LOC
              / 124).  WITH THE PATCHES ABOVE, THIS LOCATION ON
              / PAGE 0 IS FREED FOR OTHER USES.
```

```
        / INPUT FROM AND OUTPUT TO CHARACTER ORIENTED DEVICES
        /       OTHER THAN THE TELETYPE, AND ADDING DEVICES
        /       TO THE INTERRUPT SERVICE ROUTINE.

        /       TWO LOCATIONS ON PAGE 0, 'OUTDEV' [LOC. 63] AND
        /  'INDEV' [LOC. 64], CONTAIN THE ADDRESSES OF THE OUTPUT
        /  AND INPUT SUBROUTINES, RESPECTIVELY.  IF IT IS DESIRED
        /  TO DO I/O TO OTHER DEVICES, THESE LOCATIONS SHOULD BE
        /  CHANGED TO THE ADDRESSES OF THE NEW I/O SUBROUTINE(S).
        /  CHANGING THESE LOCATIONS UNDER PROGRAM CONTROL, PERHAPS
        /  WITH A MODIFICATION TO THE "L" COMMAND (OR PERHAPS WITH
        /  AN ADDITIONAL COMMAND), WOULD ALLOW FOR VERY VERSATILE
        /  I/O TO/FROM A FOCAL PROGRAM.
        /       AN EXAMPLE OF THE WAY THIS IS DONE IS SHOWN IN THE
        /  LISTING, THE HIGH-SPEED READER ROUTINE.  THIS IS A GOOD
        /  EXAMPLE NOT ONLY OF HOW TO DO IT, BUT ALSO OF THE
        /  PROBLEMS THAT CAN OCCUR.  THE PROBLEM WITH IT IS THAT
        /  THE READER CAN EASILY OVER-RUN THE 1 CHARACTER INPUT
        /  BUFFER, AT WHICH TIME THE INPUT STOPS AND AN ERROR IS
        /  GIVEN.  THIS WOULD OBVIOUSLY BE LESS OF A PROBLEM WITH
        /  A HIGH-SPEED OUTPUT DEVICE, BUT THE POINT IS THAT YOU
        /  SHOULD GIVE SOME THOUGHT TO WHAT YOU ARE DOING, FIRST.
        /
        /       IF NEW DEVICES ARE ADDED, A MODIFICATION SHOULD
        /  ALSO BE MADE TO THE ERROR HANDLER (STARTING AT 'RECOVR',
        /  LOCATION 2740) TO STOP ANY INPUT DEVICES IN MOTION AND
        /  RESET 'INDEV' AND 'OUTDEV' TO THE TELETYPE I/O ROUTINES.
        /  'OUTDEV' SHOULD BE RESET BEFORE THE ERROR MESSAGE IS
        /  OUTPUT SO THAT THE MESSAGE WILL BE TYPED ON THE TELE-
        /  TYPE.  ADMITTEDLY, IT IS NOT ABSOLUTELY NECESSARY THAT
        /  THIS RESETTING BE DONE, BUT IT WILL HELP ANYONE WHO IS
        /  USING YOUR MODIFIED VERSION OF FOCAL (PERHAPS YOU), AND
        /  MAY SAVE YOU SOME FRIENDS.


        /       IF THE ADDED I/O DEVICES REQUIRE SERVICE BY THE
        /  INTERRUPT ROUTINE, A FEW LOCATIONS ARE PROVIDED AT
        /  THE END OF THE ROUTINE, BEGINNING WITH 'EXIT' (LOC.
        /  2646).  THESE LOCATIONS ARE CURRENTLY USED BY THE H.S.
        /  READER, BUT THEY CAN EITHER BE REMOVED (IF NOT USED) OR
        /  MOVED SOME PLACE ELSE.
        /       AS WITH CHARACTER ORIENTED DEVICES AS DESCRIBED
        /  ABOVE, IT IS SUGGESTED THAT A MODIFICATION BE MADE TO
        /  THE ERROR HANDLER TO RESET ANY INTERRUPT-DRIVEN I/O
        /  DEVICES AT THE TIME THAT AN ERROR OCCURS.
        /       NOTE:  IF AN 8K VERSION OF FOCAL IS USED, THE
        /  "RMF" INSTRUCTION CURRENTLY AT LOCATION 2652 SHOULD
        /  BE MOVED TO LOCATION 2654.  IF AN 8K VERSION IS NOT
        /  USED, THIS INSTRUCTION IS NOT NEEDED.
```

```
             /       THESE PATCHES, WHICH ARE WORKING EXAMPLES, DO
             /  THE FOLLOWING:
             /
             /  1.   PROVIDE CHARACTER INPUT AND OUTPUT THROUGH
             /       "FIN" AND "FOUT" FUNCTIONS.
             /
             /  2.   PROVIDE BCD INPUT AND OUTPUT AS WOULD BE USED
             /       FOR INSTRUMENT INTERFACE THROUGH "FBI" AND
             /       "FBO" FUNCTIONS.
             /
             /  3.   GIVE AN EXAMPLE OF ARRAY OR COMMON STORAGE
             /       THROUGH AN "FA" FUNCTION.
             /
             /  4.   USE A MODIFICATION TO THE "L" COMMAND, INCLUDING
             /       CHANGING IT TO THE "O" COMMAND, TO SHOW HOW A
             /       FOCAL LINE OR GROUP CAN BE EXECUTED FROM
             /       WITHIN A FOCAL MODIFICATION.
             /
             /  5.   PROVIDE THE FOLLOWING CORRECTIONS TO FOCAL:
             /
             /       A.  PATCH THE "L" ROUTINE TO CONTINUE IN THE
             /       CURRENTLY EXECUTING FOCAL PROGRAM (DON'T GO
             /       TO 7600).
             /
             /       B.  PATCH THE FLOATING-POINT OUTPUT ROUTINE TO
             /       ROUND CORRECTLY, FLOAT THE SIGN TO THE IMMED-
             /       IATE LEFT OF THE MOST SIGNIFICANT DIGIT TYPED,
             /       AND REMOVE THE AUTOMATIC OUTPUT OF "=".


             /MISCELLANEOUS PATCHES:

       0035  *BOTTOM         /RESET BOTTOM OF P.D.L.

00035  5123        DATA-1


       1004  *1004   /CHANGE "L" TO "O" IN COMMAND LIST

01004  0317        "O

       1173  *1173   /OVERLAY OLD "L" ROUTINE ADDRESS

01173  6311        OCOM

       1201  *1201   /OVERLAY "*" ROUTINE ADDRESS

01201  0177        START


             /PATCHES TO THE FUNCTION NAME & ADDRESS TABLES:

       0377  *377    /REMOVE SOME FUNCTIONS, CHANGE OTHERS

00377  1343        XA      /"FA" - WAS 'XDYS'
```

```
00400  2725          ERROR5  /WAS 'XRAN'
00401  2725          ERROR5  /WAS 'XADC'
00402  2725          ERROR5  /WAS 'ARTN'
00403  5204          XBI     /"FBI" - WAS 'FEXP'
00404  5227          XBO     /"FBO" - WAS 'FLOG'
00405  1553          XIN     /"FIN" - WAS 'FSIN'
00406  5774          XOUT    /"FOUT" - WAS 'FCOS'


       2170  *2170   /CHANGE NAMES OF FUNCTIONS

02170  0301          "A      /A

       2174  *2174

02174  1115          "B↑2+"I         /BI
02175  1123          "B↑2+"O         /BO
02176  1140          "I↑2+"N         /IN
02177  2672          "O↑2+"U↑2+"T    /OUT
             /  NOTE:  "↑" IS THE MULTIPLY OPERATOR IN PAL8.  IT
             /  DOES NOT USE ANY HIERARCHY OF PRECIDENCE, BUT JUST
             /  MULTIPLIES THE VALUE ON THE LEFT BY THE VALUE ON
             /  THE RIGHT.  THUS A↑2+B↑2+C = (A*2+B)*2+C IN NORMAL
             /  ALGEBRAIC REPRESENTATION.

        /

             /       ALL OF THE "EXTENDED" FUNCTIONS AND SOME OF
             /  THE OTHER FUNCTIONS HAVE BEEN REMOVED TO MAKE ROOM
             /  FOR THE PATCHES THAT ARE TO BE ADDED.  NO PARTICULAR
             /  REASONING WAS USED TO DETERMINE THE ORDER OF THE NEW
             /  FUNCTION REFERENCES EXCEPT THAT THE "FA" FUNCTION WAS
             /  PUT AS CLOSE TO THE BEGINNING OF THE TABLE AS WAS
             /  POSSIBLE TO MINIMIZE THE TIME IN THE "SORTB" SUB-
             /  ROUTINE WHEN LOOKING IT UP.
       1142  *1142   /OVERLAY OLD "FDIS" FUNCTION.

             /       THIS SUBROUTINE, CALLED BY "PUSHJ; CHKCOM",
             /  CHECKS TO SEE IF TERMINATOR OF THE LAST EXPRESSION
             /  (I.E., THE CONTENTS OF 'CHAR') IS A ",".  IF IT
             /  IS NOT, CHKCOM EXITS TO THE WORD FOLLOWING THE
             /  CALL.  IF IT IS, CHKCOM EVALUATES THE NEXT EXPRESSION
             /  AND EXITS TO THE SECOND WORD FOLLOWING THE CALL.
             /       CHKCOM NEEDS TO BE A RECURSIVE SUBROUTINE
             /  BECAUSE THE CALL TO 'EVAL' MAY END UP CALLING
             /  THIS SUBROUTINE AGAIN.

01142  1066  CHKCOM, TAD CHAR        /IS C(CHAR) = ","?
01143  1336          TAD MCOM
01144  7640          SZA CLA
01145  5541          POPJ            /NO, EXIT NOW TO CALL+1
01146  4540          PUSHJ           /YES, EVALUATE NEXT ARGUMENT
01147  1612                  EVAL-1  /(SKIP THIS CHAR)
             /
             /       THE FOLLOWING CODE COULD BE REPLACED BY
             /  "CLA IAC; POPJ" IF IT IS DESIRED TO USE THE MIN-
             /  IMUM SPACE AND THE "JMS I INTEGER" IS NOT USED.
             /
01150  1413          POPA            /GET RETURN ADDRESS
01151  3015          DCA FLTXR2      / & STORE IT HERE
             /
             /       IF IS IS DESIRED TO FIX THE FLAC AND RETURN
             /  WITH THE INTEGERIZED VALUE, REMOVE THE FIRST "/"
             /  FROM THE FOLLOWING LINE OF CODE:
             /
             /       JMS I INTEGER   /FIX FLAC & RETURN WITH VALUE
01152  5415          JMP I FLTXR2    /EXIT TO CALL+2



       1343  *1343   /OVERLAY OLD "FADC" FUNCTION & 'OUTL' ROUTINE.

             /       NEW "FA" FUNCTION, ACCESSES DATA STORED IN AN
             /  ARRAY.  IT IS EQUIVALENT TO THE WAY ARRAYS ARE
             /  HANDLED BY HIGH-LEVEL COMPILER LANGUAGES, I.E., THE
             /  INDEX INTO THE ARRAY IS MULTIPLIED BY THE NUMBER OF
             /  WORDS PER ITEM AND ADDED TO THE BASE ADDRESS OF THE
             /  ARRAY TO GET THE ITEM ADDRESS.  THIS IS AN ADAPTA-
             /  TION TO THE "FNEW" FUNCTION FIRST WRITTEN BY JOHN
             /  ALDERMAN.
             /       CALL:  ...FA(I)...       GET ITH VALUE
             /              ...FA(I,V)...     PUT V IN ITH LOCATION
             /
             /       WHERE I WILL BE TAKEN TO BE AN INTEGER IN THE
             /       RANGE 0.LE.I.LT.C('LIM') AND V IS ANY VALUE
             /       (INCLUDING ANY EXPRESSION).

01343  4453  XA,     JMS I INTEGER   /FIX FLAC
01344  7104          CLL RAL         /MULTIPLY BY 2 = N*2
01345  1046          TAD LORD        / & ADD = N*2+N = N*3
```

```
01346  1361          TAD BASE        /ADD BASE ADDRESS OF ARRAY
01347  4542          PUSHA           / & SAVE ITEM ADDRESS
01350  1045          TAD HORD        /TEST INDEX FOR PROPER RANGE:
01351  7650          SNA CLA         /(BAD IF HORD.NE.0)
01352  1362          TAD LIM         /(BAD IF LORD.GE.LIM)
01353  7141          CIA CLL
01354  1046          TAD LORD
01355  7630          SZL CLA
01356  4566          ERROR2          /SORRY, CHARLIE!
01357  5760          JMP I .+1       /OK, GO TO REST OF ROUTINE
01360  6160            XA1
01361  5124  BASE,   DATA            /BEGINNING OF DATA ARRAY
01362  0020  LIM,    DSIZE           /LENGTH OF ARRAY


       1553  *1553   /OVERLAY OLD "FRAN" FUNCTION.

             /       NEW "FIN" FUNCTION, INPUTS A SINGLE CHARACTER
             /  FROM THE INPUT DEVICE AND RETURNS IT AS A NUMBER
             /  (INTEGER).
             /       CALL:   ...FIN()...   ANY ARG IS IGNORED

             /       TO READ THE CHARACTER WITHOUT ECHOING IT,
             /  REPLACE THE "READC" WITH "JMS I INDEV".  THE
             /  CHARACTER CAN THEN BE READ & ECHOED BY
             /  ...FOUT(FIN())...

01553  4552  XIN,    READC           /READ & ECHO CHARACTER
01554  1066          TAD CHAR        /GET THE CHAR
             /       AND P177        /TO IGNORE PARITY BIT
01555  3045          DCA HORD        /PUT IN HIGH WORD (MAX 8 BITS)
01556  3046          DCA LORD        /SET REST OF FLAC TO 0
01557  3047          DCA OVER2
01560  1005          TAD P13         / & SET UP EXPONENT
01561  3044          DCA EXP
01562  5536          JMP I EFUN3I


       2414  *2414   /OVERLAY OLD 'I33' SUBROUTINE.

             /LISTS FOR THE NEW "O" COMMAND:

02414  0314  OCLIST, "L
02415  0304          "D
02416  7777          -1      /TERMINATOR IS NEG #

02417  7503  OBLIST, LIBRARY
02420  6326          ODO
```

```
               /DEFINE PARAMETERS FOR DATA ARRAY:

        0020   DSIZE=  20      /CHANGE THIS ONLY FOR DIFFERENT SIZE
        0060   DWORDS= DSIZE↑3 /3 WORDS PER F.P. ITEM


        5124   *5377-173-DWORDS        /BEGINNING OF DATA ARRAY

        5124   DATA=

05124   0000   ZBLOCK DWORDS   /INITIALIZE ALL LOCS TO 0

               /       THE "173" ABOVE IS THE HAND-COUNTED NUMBER OF
               /  LOCATIONS THAT THE FOLLOWING CODING TAKES.  THIS
               /  IS THE ONLY WAY THAT THE ORIGIN CAN BE SET
               /  PROPERLY.

               /       THE FOLLOWING TWO FUNCTIONS "FBI" AND "FBO",
               /  FOR "BCD IN" AND "BCD OUT", ARE GIVEN AS EXAMPLES
               /  OF POSSIBLE TECHNIQUES FOR INTERFACING FOCAL WITH
               /  BCD I/O DEVICES.  THE ACTUAL IMPLEMENTATION AS
               /  GIVEN DOES NOTHING EXCEPT SHOW AND TEST THE TECH-
               /  NIQUES.


               /       NEW "FBI" FUNCTION.  THE ARGUMENT IS CONVERTED
               /  FROM FLOATING-POINT TO PACKED BCD AND THEN BACK
               /  TO FLOATING-POINT.  ONLY THE INTEGER PART OF THE
               /  ARGUMENT IS USED.
               /       CALL:  ...FBI(V)...
               /       WHERE V IS ANY EXPRESSION, THE INTEGER PART OF
               /       WHICH IS CONVERTED TO BCD & BACK.
               /
               /       IN ACTUAL USAGE, THE EXPRESSION MIGHT BE A
               /  COMMAND TO AN INSTRUMENT, AND THE VALUE RETURNED
               /  MIGHT BE THE READOUT FROM THE INSTRUMENT.

05204   4262   XBI,    JMS BIN2BC      /CONVERT ARG TO PACKED BCD
05205   1363           TAD DIGS1       /GET FIRST 3 DIGITS
05206   4235           JMS BCD2BN      /CONVERT TO BINARY
05207   3047           DCA OVER2       / & STORE IN LOWEST WORD
05210   3046           DCA LORD
05211   3045           DCA HORD        /SET REST OF FLAC, 'DIGIT'
05212   3737           DCA I DIGI
05213   3736           DCA I REMI      / 'REMAIN' TO 0
05214   4740           JMS I M10I      /MULTIPLY BY 10
05215   4740           JMS I M10I      / & AGAIN
05216   1364           TAD DIGS2       /GET 2ND 3 DIGITS
05217   4235           JMS BCD2BN      / & CONVERT THEM
05220   3737           DCA I DIGI      /STORE RESULT IN 'DIGIT'
05221   4740           JMS I M10I      / [1ST 3 DIGS]*1000+[2ND 3 DIGS]
05222   1226           TAD XP43
05223   3044           DCA EXP         /SET UP EXPONENT
05224   4742           JMS I DNORI     /NORMALIZE BEFORE OVER2 CLEARED
05225   5536           JMP I EFUN3I
```

```
05226  0043  XP43,   43


             /       NEW "FBO" FUNCTION.  THE ARGUMENT IS CONVERTED
             /  FROM FLOATING-POINT TO PACKED BCD AND THEN UNPACKED
             /  AND SENT TO THE OUTPUT DEVICE (TELETYPE).  AS ABOVE,
             /  THE PRIMARY PURPOSE IS TO SHOW TECHNIQUES.
             /       CALL:  ...FBO(V)...
             /       WHERE V IS AS FOR "FBI" ABOVE.

             /       SINCE THIS FUNCTION IS NOT SET UP TO RETURN
             /  ANY VALUE, IT MIGHT JUST AS WELL HAVE BEEN CALLED
             /  BY ANOTHER MODIFICATION TO THE "L" (OR "O")
             /  COMMAND, I.E. "O OUT EXP;" [WHERE "OUT" WOULD BE
             /  THE OPTION TO CALL THIS ROUTINE, WHICH WOULD HAVE
             /  HAD TO CALL 'EVAL' TO EVALUATE "EXP"].

05227  4262  XBO,    JMS BIN2BC      /CONVERT ARG TO PACKED BCD
05230  1363          TAD DIGS1       /GET FIRST 3 DIGITS
             /
             /       I/O INSTRUCTIONS WOULD BE INSERTED HERE TO
             /  SEND DATA TO A DEVICE, UNDER NORMAL USAGE.
             /
05231  4343          JMS UPACKD      /UNPACK & PRINT DIGITS
05232  1364          TAD DIGS2
05233  4343          JMS UPACKD      /SAME AS ABOVE
05234  5536          JMP I EFUN3I
```

```
                / 3 DIGIT PACKED BCD TO BINARY CONVERSION
                /       BY RADIX DEFLATION.
                /
                / FROM: CARL LOWENSTEIN, U.C.S.D., "DECUSCOPE",
                /       VOL. 10, NO. 3.

05235  0000  BCD2BN, 0
05236  3044          DCA TEM1        /= 256*H + 16*M + L (DECIMAL)
05237  1044          TAD TEM1        /  400*H + 20*M + L
05240  0260          AND C7400       /  400*H
05241  7110          CLL RAR         /  200*H
05242  3040          DCA TEM2
05243  1040          TAD TEM2
05244  7012          RTR             /   40*H
05245  1040          TAD TEM2        /  240*H
05246  1044          TAD TEM1        /  640*H + 20*M + L
05247  0261          AND C7760       /  640*H + 20*M
05250  7012          RTR             /  150*H +  4*M
05251  3040          DCA TEM2
05252  1040          TAD TEM2
05253  7010          RAR             /   74*H +  2*M
05254  1040          TAD TEM2        /  234*H +  6*M
05255  7041          CIA             / -234*H -  6*M
05256  1044          TAD TEM1        /  144*H + 12*M + L
05257  5635          JMP I BCD2BN    /= 100*H + 10*M + L (DECIMAL)

05260  7400  C7400,  7400
05261  7760  C7760,  7760

       0044  TEM1=   EXP
       0040  TEM2=   EX1      /USE PAGE 0 TEMPORARIES


05262  0000  BIN2BC, 0        /CONVERT F.P. ARG TO PACKED BCD
05263  1045          TAD HORD
05264  7710          SPA CLA
05265  4566          ERROR2          /NEGATIVE ARGS ARE ILLEGAL
05266  4407          FINT            /ENTER F.P. PACKAGE
05267  1330          FADD FHALF      /ADD 0.5 TO ROUND UP
05270  4333          FMUL SCALE      /MULTIPLY BY 1.0E-6
05271  0000          FEXT
05272  1044          TAD EXP         /IS THE RESULT < 1?
05273  7740          SMA SZA CLA
05274  5301          JMP BINERR      /NO, ARG IS > 999999; TOO BIG!
05275  7001          IAC
05276  3043          DCA OVER1       /YES, MAKE OPERAND NON-0
05277  3040          DCA EX1         /ALIGN BINARY POINT TO THE LEFT
05300  4741          JMS I ALINI     / OF THE MOST SIGNIFICANT BIT.
05301  4566  BINERR, ERROR2          /COULDN'T ALIGN--ARG WAS TOO SMALL!
05302  4527          JMS I DOUBLE    /NOW SHIFT 1 LEFT TO END OF WORD
05303  3736          DCA I REMI      /SET 'REMAIN' TO 0,
05304  3737          DCA I DIGI      / SET 'DIGIT' TO 0,
05305  1327          TAD DCCON       / & SET UP THE "DCA"
05306  3322          DCA BIN2B3
05307  7344          CLL STA RAL     /= -2
```

```
05310  3032          DCA T1          /USE PAGE 0 TEMPORARIES
05311  7346  BIN2B1, CLL STA RTL     /= -3
05312  3071          DCA T2
05313  7106  BIN2B2, CLL RTL         /SHIFT DIGITS 4 BITS
05314  7006          RTL
05315  3033          DCA T3          / & SAVE RESULTS
05316  4740          JMS I M10I      /POP OUT A DIGIT
05317  1033          TAD T3          / & MERGE WITH THE OTHERS.
05320  2071          ISZ T2          /DONE 3?
05321  5313          JMP BIN2B2      /NO
05322  7402  BIN2B3, HLT             /WILL BE A "DCA" TO STORE DATA
05323  2322          ISZ BIN2B3      / BUMP THE INSTRUCTION
05324  2032          ISZ T1          /DONE 2 GROUPS?
05325  5311          JMP BIN2B1      /NO
05326  5662          JMP I BIN2BC    /YES, EXIT

05327  3363  DCCON,  DCA DIGS1       /DUMMY INSTRUCTION

05330  0000  FHALF,  0       /= 0.5, ROUNDING CONSTANT
05331  2000          2000
05332  0000          0

05333  7755  SCALE,  -23     /= 1.0E-6
05334  2061          2061
05335  5737          5737

             /       THE SCALING OF THE NUMBER FOR OUTPUT IS DONE
             /  AS FOLLOWS:
             /
             /  1.   0.5 IS ADDED TO THE NUMBER TO ROUND UP ANY
             /       FRACTIONAL PART.  THIS IS NEEDED DUE TO THE
             /       FACT THAT 0.1 CANNOT BE REPRESENTED EXACTLY
             /       BY A FIXED-LENGTH BINARY FRACTION.
             /
             /  2.   THE NUMBER IS MULTIPLIED BY 1.0E-6.  THIS
             /       CONVERTS ALL NUMBERS LESS THAN 1,000,000 TO
             /       A FRACTION (100,000 => 0.1, 1 => 0.000001).
             /
             /  3.   THE RESULT IS THEN UN-FLOATED TO A FIXED-
             /       POINT FRACTION WITH THE BINARY POINT TO
             /       THE LEFT OF THE MOST SIGNIFICANT BIT.  THIS
             /       PLACES THE BINARY POINT BETWEEN 'REMAIN'
             /       (THE OVERFLOW FROM THE FLAC WHEN 'MULT10'
             /       IS CALLED) AND 'HORD' (THE HIGH ORDER WORD
             /       OF THE FLAC).
             /
             /  4.   EACH OF THE DIGITS IS NOW DEVELOPED BY
             /       SIMPLY CALLING 'MULT10' TO MULTIPLY THE FLAC
             /       BY 10.  AFTER EACH MULTIPLICATION THE
             /       NEXT DIGIT WILL HAVE "POPPED" OUT OF THE
             /       FRACTION IN THE FLAC, AND WILL BE RETURNED
             /       IN THE AC.
```

```
05336  5712  REMI,   REMAIN
05337  5713  DIGI,   DIGIT
05340  5667  M10I,   MULT10
05341  6623  ALINI,  ALIGN
05342  7335  DNORI,  DNORM


05343  0000  UPACKD, 0       /UNPACK & OUTPUT PACKED BCD
05344  3044          DCA TEM1        /SAVE DIGITS
05345  7346          CLL STA RTL     /= -3
05346  3040          DCA TEM2        /NOTE: CANNOT USE T1, T2 OR T3!
05347  1044  UPACK1, TAD TEM1        /GET DIGITS
05350  7106          CLL RTL         /ROTATE HIGH ONE TO AC 8-11
05351  7006          RTL
05352  3044          DCA TEM1
05353  1044          TAD TEM1
05354  7004          RAL
05355  0107          AND P17         /MASK TO 1 DIGIT
05356  1113          TAD C260        /MAKE A CHARACTER
05357  4551          PRINTC
05360  2040          ISZ TEM2        /DONE?
05361  5347          JMP UPACK1      /NO
05362  5743          JMP I UPACKD


05363  0000  DIGS1,  0
             DIGS2,          /USE 'DPRINT'


             /       ROUTINE TO FLOAT THE SIGN OF THE FLOATING-POINT
             /  NUMBER BEING OUTPUT TO THE IMMEDIATE LEFT OF THE
             /  MOST SIGNIFICANT DIGIT.  THIS ROUTINE AND THE TWO
             /  PATCHES AT LOCATIONS 5532 AND 6007 ARE ALL THAT IS
             /  NECESSARY TO ADD THIS FEATURE.

05364  0000  DPRINT, 0       /PRINT DIGIT & SIGN (IF NEEDED)
05365  7510          SPA             /PRINTING A SPACE?
05366  5375          JMP DPRIN1      /YES, JUST DO IT
05367  3343          DCA UPACKD      /NO, MUST BE A DIGIT, SAVE AC
05370  1566          TAD I ERRPT     /GET THE SIGN CHARACTER
05371  7440          SZA
05372  4551          PRINTC          /PRINT IF NOT PRINTED YET
05373  3566          DCA I ERRPT     / & RESET CHAR TO 0
05374  1343          TAD UPACKD      /RESTORE AC
05375  1113  DPRIN1, TAD C260        /MAKE AN ASCII CHARACTER
05376  4551          PRINTC
05377  5764          JMP I DPRINT
```

```
       5525  *5525   /CORRECT THE ROUNDING CONSTANT

05525  0004          4

       5532  *5532   /OVERLAY 'OPUT'; PART OF FLOATING SIGN

05532  5364          DPRINT  /DIGIT & SIGN PRINTER



       5774  *5774   /USE THE SMALL HOLE

             /       NEW "FOUT" FUNCTION, OUTPUTS THE ARGUMENT
             /  AS A CHARACTER TO THE OUTPUT DEVICE.  ALLOWS
             /  CHARACTERS TO BE OUTPUT.
             /       CALL:  ...FOUT(V)...
             /       WHERE V IS ANY EXPRESSION IN THE RANGE
             /       1.GE.V.LE.4095.  NOTE THAT THE NUMBER IS
             /       ACTUALLY OUTPUT MODULO 256 (8 BIT CHARACTER).
             /       TO OUTOUT A NULL TO THE TELETYPE, USE
             /       ...FOUT(256)...  THIS MUST BE DONE BECAUSE
             /       0 IS THE EMPTY FLAG FOR THE TTY OUTPUT
             /       BUFFER.

05774  4453  XOUT,   JMS I INTEGER   /FIX THE FLAC
05775  7440          SZA             /CANNOT USE 0!
05776  4551          PRINTC          /OUTPUT IT
05777  5536          JMP I EFUN3I


       6001  *6001   /REMOVE OUTPUT OF "=" IN F.P. OUTPUT

06001  7610          CLA SKP

       6007  *6007   /OVERLAY "PRINTC"; PART OF FLOATING SIGN

06007  3566          DCA I ERRPT     /SAVE SIGN CHARACTER



       6160  *6160   /REST OF "FA" FUNCTION

06160  4540  XA1,    PUSHJ           /CHECK FOR & EVALUATE 2ND
06161  1142                  CHKCOM  / ARG, IF ANY
06162  7332          STL CLA RTR     /ONLY 1, AC = 2000 => FGET
06163  1373          TAD FPCON       / 2 ARGS => FPUT
06164  3370          DCA XAOP        /SET UP OPERATION
06165  1413          POPA
06166  3030          DCA PT1         /SET UP ADDRESS
06167  4407          FINT
06170  0000  XAOP,   FEXT    / FGET OR FPUT
06171  0000          FEXT
06172  5536          JMP I EFUN3I

06173  6430  FPCON,  FPUT I PT1
```

```
        6311  *6311   /OVERLAY OLD H.S. READER ROUTINE

              /       MODIFIED "L", "NOW "O", COMMAND.  CAN STILL
              /  BE USED TO OUTPUT LOCATIONS USED BY FOCAL, AND HAS
              /  AN ADDED ACTION TO ALLOW THE EXECUTION OF A LINE OR
              /  GROUP IN A FOCAL PROGRAM.
              /
              /  SUBCOMMANDS:
              /
              /  L    LOCATIONS:  AS WITH OLD "L" COMMAND
              /
              /  D    DO EXP:  THE EXPRESSION IS EVALUATED AND THEN
              /                MADE INTO THE FORM OF A LINE #.  THIS
              /                LINE (OR GROUP, IF THE STEP PART IS 0)
              /                IS THEN EXECUTED AS WITH THE "DO"
              /                COMMAND.  THIS IS GIVEN FOR AN EXAMPLE
              /                OF HOW TO CALL A FOCAL LINE OR GROUP
              /                FROM A PATCH.

06311  4560  OCOM,   SPNOR              /"O" COMMAND HANDLER
06312  1066          TAD CHAR           /SAVE CHAR
06313  4542          PUSHA
06314  4545  OCOM1,  GETC               /GET NEXT CHAR
06315  4550          SORTC              /LOOK FOR TERMINATOR
06316  1376                  GLIST-1
06317  7410          SKP
06320  5314          JMP OCOM1          /NOT FOUND
06321  1413          POPA
06322  4547          SORTJ              /GO DO SUB-COMMAND
06323  2413                  OCLIST-1
06324  0003                  OBLIST-OCLIST
06325  4566          ERROR2             /ILLEGAL CHARACTER
              /
              /
06326  4540  ODO,    PUSHJ              /EVALUATE THE EXPRESSION
06327  1612                  EVAL-1  /(SKIP CHAR)
06330  4453          JMS I INTEGER      /FIX THE FLAC
06331  7450          SNA                /TEST RANGE: 1.LE.#.LE.31
06332  5337          JMP ODERR          / = 0, BAD
06333  0373          AND P7740
06334  7450          SNA                / .GE. 31, BAD
06335  1045          TAD HORD           / .NE. 0, BAD
06336  7640          SZA CLA
06337  4566  ODERR,  ERROR2             /ILLEGAL NUMBER
06340  1046          TAD LORD           /OK, GET GROUP PART & MOVE
06341  4557          RTL6               / TO PROPER BITS
06342  7004          RAL
06343  3067          DCA LINENO         /SAVE GROUP PART
06344  4451          JMS I MINSKI       /NEGATE FLAC
06345  4407          FINT
06346  7000          FNOR               /RE-NORMALIZE FLAC (NEGATIVE)
06347  1430          FADD I PT1         /SUBTRACT OFF INTEGER PART
06350  4374          FMUL F100          /MULTIPLY BY 100(10)
06351  0000          FEXT
06352  4453          JMS I INTEGER      /FIX THE STEP PART
```

```
06353  1067          TAD LINENO      /MERGE IT WITH THE GROUP
06354  3067          DCA LINENO      /..LINE NUMBER...
06355  1046          TAD LORD        /WAS THE STEP PART = 0?
06356  7640          SZA CLA
06357  7130          STL RAR         /NO, AC = 4000; DO 1 LINE
06360  3065          DCA NAGSW       / OTHERWISE = 0; DO GROUP
06361  1066          TAD CHAR        /MUST SAVE CHAR
06362  4542          PUSHA
06363  1077          TAD CCR         / & SET IT TO A C.R.
06364  3066          DCA CHAR        / FOR PROPER EXIT
06365  4540          PUSHJ           /NOW GO "DO"...
06366  0421                  DO+1
06367  1413          POPA            /RESTORE SAVED CHAR
06370  3066          DCA CHAR
06371  5772          JMP I .+1       /NOW GO DO NEXT COMMAND
06372  0611            PROC

06373  7740  P7740,  7740

06374  0007  F100,   7       /= 100.0
06375  3100          3100
06376  0000          0


       7522  *7522   /DON'T GO TO SYSTEM AFTER "L"

07522  5541          POPJ            /CONTINUE WITH NEXT LINE

             $$$
```

```
AC1H    0041        FLTXR   0014        PT1     0030
AC1L    0042        FLTXR2  0015        PUSHA   4542
ALIGN   6623        FMUL    4000        PUSHF   4543
ALINI   5341        FNOR    7000        PUSHJ   4540
AXOUT   0017        FPCON   6173        P13     0005
BASE    1361        FPOW    5000        P17     0107
BCD2BN  5235        FPUT    6000        P177    0106
BINERR  5301        FSUB    2000        P7600   0104
BIN2BC  5262        F100    6374        P77     0122
BIN2B1  5311        GETC    4545        P7740   6373
BIN2B2  5313        GETLN   4554        READC   4552
BIN2B3  5322        GINC    0070        REMAIN  5712
BOTTOM  0035        GLIST   1377        REMI    5336
CCR     0077        GS1     1437        RTL6    4557
CFRS    0133        HORD    0045        SCALE   5333
CHAR    0066        INDEV   0064        SORTC   4550
CHKCOM  1142        INPUT   0756        SORTCN  0054
CNTR    0057        INTEGE  0053        SORTJ   4547
C200    0123        LASTLN  0025        SPNOR   4560
C260    0113        LIBRAR  7503        START   0177
C7400   5260        LIM     1362        TEM1    0044
C7760   5261        LINENO  0067        TEM2    0040
DATA    5124        LORD    0046        TESTC   4564
DCCON   5327        MCOM    1136        TESTN   4561
DIGI    5337        MFLT    0117        THISLN  0023
DIGIT   5713        MINSKI  0051        TLIST   1400
DIGS1   5363        MINUSA  0112        TSTGRP  4563
DIGS2   5364        MPER    0115        TSTLPR  4562
DNORI   5342        MULT10  5667        T1      0032
DNORM   7335        MULT2   5715        T2      0071
DO      0420        M10I    5340        T3      0033
DOUBLE  0127        M2      0111        UPACKD  5343
DPRINT  5364        M20     0105        UPACK1  5347
DPRIN1  5375        M240    0114        XA      1343
DSIZE   0020        NAGSW   0065        XAOP    6170
DUBLAD  5733        OBLIST  2417        XA1     6160
DWORDS  0060        OCLIST  2414        XBI     5204
EFUN    1743        OCOM    6311        XBO     5227
EFUN2   1754        OCOM1   6314        XCT     0020
EFUN3I  0136        ODERR   6337        XIN     1553
ERROR2  4566        ODO     6326        XOUT    5774
ERROR5  2725        OUTDEV  0063        XP43    5226
ERRPT   0166        OVER1   0043        XRT     0011
EVAL    1613        OVER2   0047        XRT2    0012
EXP     0044        PACKC   4546
EX1     0040        PC      0022
FADD    1000        PC1     0614
FDIV    3000        PDLXR   0013
FEXT    0000        PER     0102
FGET    0000        POPA    1413
FHALF   5330        POPF    4544
FINDLN  4555        POPJ    5541
FINT    4407        PRINTC  4551
FISW    0052        PRNTLN  4553
FLAC    0044        PROC    0611
```

```
             /FOCAL DEBUG PATCHES #1

             /DEBUGGING PATCHES FOR RUNNING FOCAL WITH "IOF":

       0063  *63     /CHANGE I/O ROUTINE POINTERS

00063  1354          1354    /"OUTL"
00064  2414          2414    /"I33"

       2732  *2732   /ERROR HANDLER PATCHES FOR IOF

02732  5342          JMP 2742        /NO WAITING FOR OUTPUT

       2762  *2762

02762  7000          NOP             /REMOVE THE "TLS"


             /ADDITIONAL PATCHES TO ALLOW THE USE OF OS/8 ODT:

       0001  *1      /MOVE TWO CONSTANTS FROM LOCS 5 & 6

00001  0013  P13,    13
00002  0100  C100,   100

             /NOW CHANGE ALL REFERENCES TO THEM

       1462  *1462   /"GS2+1"

01462  1001          TAD P13

       2277  *2277   /"UTE+1"

02277  1002          TAD C100

       2545  *2545   /"PCK1+7"

02545  1001          TAD P13

       3044  *3044   /"RUB2+2"

03044  1002          TAD C100

       5065  *5065   /"STARTL+1"

05065  1001          TAD P13
```

```
              /FOCAL DEBUG PATCHES #2

              /DEBUGGING PATCHES FOR RUNNING FOCAL WITH "IOF"
              /  WITH THE STANDARD "IOF" ROUTINES OVERLAID:

       0063   *63      /CHANGE I/O ROUTINE POINTERS

00063  2676            DOUTL   /DEBUG LOW-SPEED OUTPUT
00064  2704            DI33    /DEBUG LOW-SPEED INPUT

       2732   *2732    /ERROR HANDLER PATCHES FOR IOF

02732  5342            JMP 2742        /NO WAITING FOR OUTPUT

       2762   *2762

02762  7000            NOP             /REMOVE THE "TLS"


              /ADDITIONAL PATCHES TO ALLOW THE USE OF OS/8 ODT:

       0001   *1       /MOVE TWO CONSTANTS FROM LOCS 5 & 6

00001  0013   P13,     13
00002  0100   C100,    100

              /NOW CHANGE ALL REFERENCES TO THEM

       1462   *1462    /"GS2+1"

01462  1001            TAD P13

       2277   *2277    /"UTE+1"

02277  1002            TAD C100

       2545   *2545    /"PCK1+7"

02545  1001            TAD P13

       3044   *3044    /"RUB2+2"

03044  1002            TAD C100

       5065   *5065    /"STARTL+1"

05065  1001            TAD P13
```

```
                            EJECT


            2676  *2676     /OVERLAY 'XOUTL'

02676  0000  DOUTL,    0
02677  6046            TLS
02700  6041            TSF     /SLOWEST WAY
02701  5300            JMP .-1
02702  7200            CLA
02703  5676            JMP I DOUTL

02704  0000  DI33,     0
02705  6031            KSF
02706  5305            JMP .-1
02707  6036            KRB
02710  0106            AND P177        /IGNORE PARITY, NULLS
02711  7450            SNA
02712  5305            JMP DI33+1
02713  1123            TAD C200
02714  5704            JMP I DI33
```