IDENTIFICATION

PRODUCT CODE:	DEC-11-GGRSA-A-LA1
PRODUCT NAME:	LISTING OF FOCAL-GT
DATE CREATED:	MARCH 19, 1973
MAINTAINER:	SMALL SYSTEMS SOFTWARE ENGINEERING
AUTHOR:	ROBERT FRIEDENTHAL

```
                        ;FOCAL-GT

                        ;DEC-11-GGRSA-A-LA

                        ;COPYRIGHT 1972,1973

                        ;DIGITAL EQUIPMENT CORPORATION
                        ;MAYNARD, MASSACHUSETTS 01754

                        ;GT40 CODE BY BOB FRIEDENTHAL

                        ;EDIT 2
                        ;MARCH 17, 1973


                        ;FOR GT40 CONDITIONALIZATION INSTRUCTIONS, SEE DNEW CODE
                        ;LABELLED **DNEW** NEAR END OF THIS LISTING



                        ;DOCUMENTATION NOTES:
                        ;DOUBLE QUOTE MARKS DENOTE TRAP-INSTRUCTION MODULES
                        ;(X) MEANS THE CONTENT-OF-X,
                        ;ASTERISKS DENOTE COMMAND MODULES,
                        ;"C.R."MEANS "CARRIAGE RETURN",
                        ;SINGLE QUOTE MARKS DENOTE A SUBROUTINE,
```

```
                                ;ASSIGNMENTS OF REGISTERS


                                ;AS USED GENERALLY
000000                          TEMP=%0         ;SCRATCH
000001                          AC=%1           ;ACCUMULATOR
000002                          PTR=%2          ;VARIABLE POINTER
000003                          AXOUT=%3        ;TEXT READER
000004                          CHAR=%4         ;CHARACTER
000005                          R5=%5           ;EXCEPTIONAL USE REGISTER AND RUBOUT PROTECTION

000006                          SP=%6           ;STACK POINTER
000007                          PC=%7           ;PROGRAM COUNTER PDP-11

                                ;AS USED BY OUTPUT CONVERSION
000000                          P=TEMP          ;PLACES BEFORE "."
000001                          AC=AC           ;TOTAL NO. OF DIGITS.
000002                          E=PTR           ;NO. OF INTEGER DIGITS
000003                          F=AXOUT         ;TOTAL NO. OF PLACES.
000004                          CHAR=CHAR       ;NO. OF DECIMAL POINTS
000005                          R5=R5           ;SCRATCH


                                ;AS USED BY MAIN FLOATING POINT
000000                          TEMP=TEMP       ;SCRATCH
000001                          AC=AC           ;INPUT EXP;MAY CONTAIN OP-CODES
000002                          BH=PTR          ;FLAC HORD;MAY CONTAIN ADDRESS
000003                          BL=AXOUT        ;FLAC LORD;MAY BE NEEDED BY FREAD
000004                          AH=CHAR         ;INPUT HORD;MAY BE NEEDED BY FREAD
000005                          AL=R5           ;INPUT LORD


000200                          ONE=200         ;SWITCH ASSIGNMENTS
000001                          ALL=1
000020                          NALPHA=20       ;0=TERMINATE ON ASCII CODES
                                                ;1=TERMINATE ON ;;C,R,ALSO

000216                          CR=216          ;INTERNAL CODE
005015                          CRLF=05015      ;FOR USE IN "PRINT2, CRLF"

177560                          TKS=177560      ;TELETYPE KEYBOARD
177564                          TPS=177564      ;TELETYPE PRINTER
177514                          LPS=177514      ;LINE PRINTER
177550                          PRS=177550      ;H.S. READER
177554                          PPS=177554      ;H.S. PUNCH
177776                          STATUS=177776
```

```
                                        ;MAIN VECTORS

            000000                      .=0             ;RESERVED FOR MANUAL RESTART VECTOR
000000  000137  003072                          JMP  @#INIT2
                                        ;.=4            ;ERROR TRAP VECTOR: STACK OVERFLOW TRAP
000004  001706                                  STACKO
000006  000340                                  340
                                        ;.=10           ;RESERVED INST
000010  005472                                  FEMT
000012  000100                                  100
                                        ;.=14           ;ODT
000014  001706                                  STACKO
000016  000340                                  340
                                        ;.=20           ;IOT
            000004                      DELETE=IOT      ;REMOVE A LINE OF TEXT
000020  003272                                  XDELET
000022  000340                                  340
                                        ;.=24           ;PWR-FAIL/AUTO-RESTART
000024  005752                                  PWRDWN
000026  000340                                  340
                                        ;.=30           ;EMT
000030  001706                                  STACKO
000032  000340                                  340
                                        ;.=34           ;TRAP
000034  001714                                  TRAPH
000036  000340                                  340
                                        ;.=40           ;SYSTEM VECTORS

            000060                      .=60
000060  010110                                  KINT
000062  000340                                  340
                                        ;HERE TO 'BEGIN' IS ZEROED BY 'INIT',

                                                ;OTHER VECTORS?

                                        ;.=100          ;CLOCK OPTION

                                        ;.=200          ;LINE PRINTER


                                                ;PATCH AREA

                                        ;.=234          ;UDC VECTOR

                                        ;.=300          ;EXTRA TTY
```

```
                                ;<PUBLIC INFORMATION>

          001100                .=1100

          001100                BEGIN=.         ;FUNCTION TABLE BACKS INTO STACK AREA.


                                ;LIST OF FUNCTION ADDRESSES AND NAMES.

001100  000000                          0              ;END-OF-LIST MARKER.
001102  000000                          0
001104  006376                          XABS           ;FABS   -ABSOLUTE VALUE
001106  013353                          13353
001110  006402                          XSGN           ;FSGN   -SIGN PART
001112  014032                          14032
001114  011026                          XFSBR          ;FSBR   -USER DEFINED NEW FUNCTION
001116  014012                          14012
001120  010760                          XCHR           ;FCHR   -INPUT/OUTPUT OF CHARACTER DATA
001122  013442                          13442
001124  010704                          XRAN           ;FRAN   -RANDOM NUMBER
001126  013762                          13762
001130  011134                          XEX            ;FX     -EXPERIMENTAL UNIBUS CONTROL
001132  000560                          560
001134  011070                          XADC           ;FADC   -READ ANALOG TO DIGITAL CONVERTER
001136  013343                          13343
001140  010154                          XSQT           ;FSQT   -SQUARE ROOT
001142  014110                          14110
001144  010416                          FSIN           ;FSIN   -TRIG FUNCTIONS
001146  014042                          14042
001150  010426                          FCOS           ;FCOS   -COSINE
001152  013477                          13477
001154  006406                          XITR           ;FITR   -INTEGER PART
001156  013662                          13662
001160  010272                          XFCLK          ;FCLK   -CLOCK TIME
001162  013453                          13453


          001166                FNTABL=.+2      ;SPECIAL TOPSY-TURVEY TABLE.
                                ;FUNCTION TABLE BACKS INTO STACK AREA.
                                ;HASH CODE IS FORMED FROM 7-BIT ASCII WITH PLACE VALUE OF 4**N
                                ;E.G. FX=F*4+X=106*4+130=560
                                ;BITS OFF LEFT ARE LOST; ZEROS COME IN FROM RIGHT.
```

```
                    103414          JMS=104400-.+200            ;TRAP-INTERPRETIVE TABLE

                    104600          SORTJ  = JMS +,             ;SORT AND BRANCH ON (CHAR)
001164  002406                             SORTB
                    104602          SORTC  = JMS +,             ;SORT CHAR
001166  002440                             SORTD
                    104604          PRINTC = JMS +,             ;PRINT CHAR-S
001170  003002                             OUT
                    104606          READC  = JMS +,             ;READ DATA INTO CHAR AND PRINT IT-S
001172  002710                             CHIN
                    104610          OUTCH  = JMS +,             ;OUTPUT TO A DEVICE
001174  010052                             XOUT
                    104612          INCH   = JMS +,             ;INPUT FROM A DEVICE
001176  010022                             XI33
                    104614          GETC   = JMS +,             ;UNPACK A CHARACTER-S
001200  003232                             GETX
                    104616          PACKC  = JMS +,             ;SAVE A CHARACTER -S
001202  003136                             PACKX
                    104620          TESTC  = JMS +,             ;RETURNS ON (CHAR)=
                                                                ;TERM
                                                                ;NUMBER
                                                                ;FUNCTION
                                                                ;RETURN-ON-LETTER
001204  002334                             TESTX
                    104622          GETLN  = JMS +,             ;UNPACK AND FORM A LINENUMBER
001206  002454                             GETLNX
                    104624          FINDLN = JMS +,             ;SEARCH FOR A GIVEN LINE
001210  002566                             FINDX
                    104626          PRNTLN = JMS +,             ;PRINT (LINENO)
001212  003100                             XPRNTL
                    104630          COPYLN = JMS +,             ;READ NEXT LINE NUMBER
001214  004040                             COPYLX
                    104632          START  = JMS +,             ;RETURN TO COMMAND/INPUT MODE
001216  002034                             STARTX
                    104634          SPNOR  = JMS +,             ;IGNORE SPACES-S
001220  003126                             SPNORX
                    104636          ERASEV = JMS +,             ;ERASE AND SET VARIABLES,
001222  005100                             ERVX
                    104640          ERASET = JMS +,             ;ERASE TEXT
001224  004050                             ERTX
                    104642          PRINT2 = JMS +,             ;PRINT TWO CHARACTERS
001226  002626                             PRIN2A
                    104644          DIGTST = JMS +,             ;TEST FOR A DIGIT OF INDICATED PLACE VALUE
001230  002654                             DIGISA
                    104646          PARTST = JMS +,             ;CHECK FOR PARENTHESIS MATCH,
001232  005206                             PARTSA
                    104650          GROOVY = JMS +,             ;COMPARE GROUP NOS,
001234  002672                             GROVX
                    104652          SKPLPR = JMS +,             ;SKIP IF (CHAR) IS A LEFT PARENS,
001236  002636                             XTSTLP
                    104654          SKPNON = JMS +,             ;SKIP IF NOT A NUMBER
001240  002372                             SKPNOX
                    104656          TASK   = JMS +,             ;DO FORMAT CONTROLS FOR *ASK*TYPE*
```

```
   228  001242  005412                          TASKX
   229          104660                EVAL,X = JMS +,          ;"PUSHJ EVAL-2"
   230  001244  004664                          EVALUX
```

```
001246  104662                  JMS+.   ;SPARE FOR PATCHES.

        104400                  ERROR=104400            ;TRAP

                                ;PUSHJ X=JSR PC,X

        024646                  OPEN=024646     ;CMP -(SP), -(SP)

        022626                  CLOSE=022626    ;CMP (SP)+, (SP)+

        104400                  PRINT=104400    ;TRAP+(0,1,177) FOR ASCII CODES.

        000207                  POPJ=207        ;RTS PC
```

```

        001250             COMLST=.                     ;*FOCAL* COMMAND DECODING LIST

                                                        ;ENGLISH ;FRENCH    ;SPANISH    ;GERMAN

001250  005244                     ASK                  ;ASK     ;DEMANDE   ;INTERROGUE ;FRAGE
001252  003502                     ERRORC               ;BEGIN   ;LEVE      ;XECUTE     ;COMMENCE
001254  003524                     PC1                  ;COMMENT ;COMMENTE  ;COMENTARIO ;KOMMENTAR
001256  004066                     DO                   ;DO      ;FAIZ      ;HAGA       ;MACHE
001260  003744                     ERASE                ;ERASE   ;BIFFE     ;BORRE      ;LOSCHE
001262  004222                     FOR                  ;FOR     ;QUAND     ;PARA       ;DAFOR
001264  003436                     GOTO                 ;GOTO    ;VA        ;ADELANTE   ;GEHZU
001266  003502                     ERRORC               ;H-      ;K-        ;G-         ;U-
001270  003374                     IF                   ;IF      ;SI        ;SI         ;WENN
001272  003502                     ERRORC               ;J-      ;J-        ;J-         ;I-
001274  001704                     STOP                 ;KILL    ;HALTE     ;HALTE      ;HALT
001276  003502                     ERRORC               ;LIBRARY ;ENTERPOSE ;LAZO       ;BILBLIOTHEK
001300  003552                     MODIFY               ;MODIFY  ;MODIFIE   ;MODIFIQUE  ;ANDERE
001302  003502                     ERRORC               ;N-      ;G-        ;V-         ;J-
001304  010336                     PROGIO               ;OPERATE ;PRATIQUE  ;OBRARE     ;OBERATE
001306  003502                     ERRORC               ;P-      ;N-        ;N-         ;N-
001310  002034                     STARTX               ;QUIT    ;ARRETE    ;DETENGASE  ;ENDE
001312  003516                     RETURN               ;RETURN  ;RETOURNE  ;RETOURNE   ;QUITTE
001314  004222                     SET                  ;SET     ;ORGANIZE  ;UBIQUE     ;SETZE
001316  005224                     TYPE                 ;TYPE    ;TAPE      ;TIPPEE     ;RECHNE
001320  003502                     ERRORC               ;U-      ;U-        ;W-         ;P-
001322  003502                     ERRORC               ;V-      ;W-        ;Q-         ;V-
001324  003646                     WRITE                ;WRITE   ;INSCRIS   ;ESCRIBA    ;TIPPE
001326  003544                     XECUTE               ;XECUTE  ;XECUTE    ;FLUIR      ;XECUTE
                           ;OTHERS UNUSED:              ;YZ      ;YZ        ;YZ       . ;YZ

                           ;TO CHANGE LANGUAGE, ALPHABETIZE ON THE APPROPRIATE COLUMN.
```

```
290
291
292
293                                     ;LISTS TO BE TESTED BY
294                                     ;"SORTJ" AND "SORTC"
295
296  001330  004300                     FLIST2: FLIMIT          ;,=STANDARD  *FOR*
297  001332  004266                             FINFIN          ;;=SHORT
298  001334  004252                             FINERR          ;CR=DUMB
299
300  001336  004254                     FLIST1: FINCR           ;,=STANDARD FORMAT  *FOR*
301  001340  003534                             TPR             ;;=SET;PLUS,,.GO TO PROCESS
302  001342  003540                             TPR1            ;C,R,=SET COMMAND---GO TO PC1
303
304           001344                     ATLIST=,                ;*ASK*TYPE* CONTROL CHARACTER TABLE
305  001344  005430                             TINTR           ;%=FORMAT DELIMITER
306  001346  005442                             TQUOT           ;"=LITERAL DELIMITER
307  001350  005404                             TCRLF           ;!=CARRIAGE RETURN AND LINE FEED
308  001352  005464                             TCRLF2          ;#=CARRIAGE RETURN ONLY
309  001354  010564                             TDUMP           ;$=DUMP THE SYMBOL TABLE CONTENTS
310  001356  005410                             TASK4           ;SP=TERMINATOR FOR NAMES
311  001360  005410                             TASK4           ;,=TERMINATOR FOR EXPRESSIONS
312  001362  003534                             TPR             ;;=TERMINATOR FOR COMMANDS
313  001364  003540                             TPR1            ;C,R,=TERMINATOR FOR STRINGS
314
315                                     ;$=FOR 'TDUMP' TERMINATES THE COMMAND!
316
317           001366                     INLIST=,                ;INPUT DATA CONTROL CODES,
318  001366  005372                             AGO             ;ALTMODE=LEAVE RESULT
319  001370  005340                             ASPACE          ;SPACE=CHECK FOR TERMINATOR FUNCTION
320  001372  005304                             ATAKE           ;IGNORE EQUALS SIGNS
321  001374  005330                             ARO             ;RUB OUT
322  001376  005304                             ATAKE           ;IGNORE LINE FEEDS
323
324           001400                     SRNLST=,                ;*MODIFY* CONTROL CHARACTER TABLE
325  001400  003572                             SCHAR           ;F,F,=CONTINUE
326  001402  003564                             SCONT           ;BELL=CHANGE SEARCH CHARACTER
327  001404  003566                             SCONL           ;L,F,=FINISH THE LINE AS BEFORE,
328
329           001406                     LISTGO=,
330  001406  002226                             SRETN           ;C,R,=END THE LINE HERE AS IS,
331  001410  003620                             SFOUND          ;CHAR=SEARCH CHARACTER
```

```
  333            001412                  TERMS=.         ;TERMINATOR TABLE FOR 'EVAL' AND 'GETVAR'
  334  001412      040                    .BYTE   040     ;SPACE  0 = (ASCII CODES) (INTERNAL CODES)
  335  001413      053                    .BYTE   053     ;+      1
  336  001414      055                    .BYTE   055     ;-      2
  337  001415      057                    .BYTE   057     ;/      3
  338  001416      052                    .BYTE   052     ;*      4
  339  001417      136                    .BYTE   136     ;UP ARR 5
  340  001420      050                    .BYTE   050     ;(      6 L=PARS
  341  001421      133                    .BYTE   133     ;[      7
  342  001422      074                    .BYTE   074     ;<      10
  343  001423      051                    .BYTE   051     ;)      11 R=PARS
  344  001424      135                    .BYTE   135     ;]      12
  345  001425      076                    .BYTE   076     ;>      13
  346  001426      054                    .BYTE   054     ;,      14
  347  001427      073                    .BYTE   073     ;;      15
  348  001430      015                    .BYTE   015     ;C.R.   16
  349  001431      075                    .BYTE   075     ;=      17 TO END GETARG FROM 'SET'
  350  001432      000                    .BYTE   000
  351
  352            001433                  ALIST=.         ;*ASK*TYPE* LIST OF CONTROLS (INTERNAL CODES)
  353  001433      045                    .BYTE   045     ;%
  354  001434      042                    .BYTE   042     ;"
  355  001435      041                    .BYTE   041     ;!
  356  001436      043                    .BYTE   043     ;#
  357  001437      044                    .BYTE   044     ;$
  358  001440      200                    .BYTE   200     ;SPACE
  359
  360            001441                  TLIST=.         ;TERMINATORS (INTERNAL CODES)
  361  001441      214                    .BYTE   214     ;,
  362  001442      215                    .BYTE   215     ;;
  363  001443      216                    .BYTE   216     ;CARRIAGE RETURN
  364  001444      000                    .BYTE   000     ;END LIST
  365
  366            001445                  SPECIAL=.       ;FOR INPUT DATA
  367  001445      175                    .BYTE   175     ;ALTMODE
  368  001446      200                    .BYTE   200     ;SPACE
  369  001447      217                    .BYTE   217     ;=
  370
  371            001450                  ECHOLST=.       ;TERMINATORS (ASCII)
  372  001450      177                    .BYTE   177     ;RUB-OUT (R.O.)
  373  001451      012                    .BYTE   012     ;LINE FEED (L.F.)
  374  001452      000                    .BYTE   000     ;END LIST
  375
  376
  377
  378            001453                  LIST6=.         ;*MODIFY*
  379  001453      014                    .BYTE   014     ;CONTROL-FORM
  380  001454      007                    .BYTE   007     ;CONTROL-BELL
  381  001455      012                    .BYTE   012     ;LINE FEED
  382
  383                                    ;CONTINUED
```

```
                        001456                  LIST3=.
001456    216                                   .BYTE   216             ;RETURN

                                                ;VARIABLE STORAGE AREA:******

001457    377                                   .BYTE   -1              ;SEARCH CHARACTER=**
001460    000                                   .BYTE   000             ;END OF LIST
                        001461                  COMBUF=.
                                                ;COMMAND/INPUT BUFFER
                        001601                  .=.+80.
001601    000           CCFLG:                  .BYTE 0                 ;CONTROL-C SEEN!
                                                .EVEN
001602  107654          LSPR:   107654          ;RANDOM NUMBER LOW PART.
001604  003526          PCF:    FLTZER-2        ;PROGRAM COUNTER FOR FOCAL = (START SAVE AREA)
001606  176172          THISLN: 0-.             ;LINE POINTER FROM 'FINDLN'
001610  000001          DEBG:   1               ;(ON-OFF, ENABLE) 0,0 = TRACE.
001612  176166  176164  FLARG:  0-.,,0-.        ;RESULT STORAGE
                                                ;FLOATING ACCUMULATOR
001616  176162          BE:     0-.             ;F.A.
001620  176160          HORD:   0-.             ;HIGH ORDER PART
001622  176156          LORD:   0-.             ;LOW ORDER PART
                                                ;
001624  176154          LINENO: 0-.             ;LINE NUMBER READ BY GETLN
001626  004012          FISW:   04012           ;OUTPUT FORMAT %8.04.
                        001631                  SWITCH=.+1              ;"NAGSW" ETC.
001630  000120          LINCNT: 80.             ;WIDTH OF TTY LINE
001632  177560          INDEV:  TKS             ;POINTER TO IN, DEV STATUS
001634  177564          OUTDEV: TPS             ;POINTER TO OUT, DEV STATUS, = (END SAVE AREA)
001636  176142          KIN:    0-.             ;(INTR DONE,TTY CHARACTER)
001640  000000          WHOOPS: 000             ;POWER FAIL/AUTO-RESTART SWITCH
001642  011450          AXIN:   BUFBEG+80.      ;STORAGE INDEX POINTER (STOPS 1ST "ERASEV"!)
001644  011330          BUFR:   BUFBEG          ;NEXT LOCATION IN BUFFER = LAST LOCATION USED.
001646  011450          STARTV: BUFBEG+80.      ;BEGINNING OF BUFFER AREA
001650  011330          TOP:    BUFBEG          ;BEGINNING OF TEXT BUFFER AREA.
001652  017500          BOTTOM: 17500           ;END OF ALL CORE (REALLY A CONSTANT)
                                                ;CONTINUED
```

```
                                
001654  176122                  CFRS:   .+2             ;TEXT DATA POINTER
001656  000000                          000             ;LINE ZERO.
001660  035103  047506  040503  .ASCII "C:FOCAL-11,LFOCA-A";VERSION IDENTIFIER
001666  026514  030461  046054
001674  047506  040503  040455
001702     216                  .BYTE   216
          001704                .EVEN

                              ;*KILL*

                              ;STOP ALL I/O

001704  000005               STOP:   RESET
```

```
                                        ;STACK OVERFLOW HANDLER
                                        ;
001706  012706  001130'         STACKO: MOV     #BEGIN,SP               ;RESET STACK IMMEDIATELY!
001712  104623                         ERROR+201+9.+9.                 ;THEN PRODUCE DIAGNOSTIC
                                        ;STACKO+2 IS PATCHED IF FUNCTION LIST IS CHANGED.
                                        ;
                                        ;
                                        ;TRAP HANDLER
                                        ;
                                        ;"PRINT"        0,1,177
                                        ;TRAP           200,2,376
                                        ;"ERROR"        201,2,377
                                        ;
                                        ;USE THIS FOR PRODUCTION:
001714  016667  000002  176054  TRAPH:  MOV     2(SP),  STATUS          ;RESTORE STATUS
001722  010566  000002                  MOV     R5,     2(SP)           ;SAVE R5
001726  011605                         MOV     @SP,    R5              ;GET PC
001730  016516  177776                 MOV     -2(R5), @SP             ;COPY CALL
                                        ;OR THE FOLLOWING FOR DEBUGGING:
                                        ;TRAPH: MOV     2(SP),  -2(SP)  ;KEEP THE STATUS-QUO FOR FIVE INSTRUCTIONS!
                                        ;       MOV     R5,     2(SP)   ;SAVE R5 ONTO STACK OVER STATUS BITS.
                                        ;       MOV     @SP,    R5      ;PICKUP THE RETURN ADDRESS.
                                        ;       MOV     -2(R5), @SP     ;GET THE CALL ITSELF ONTO THE STACK.
                                        ;       MOV     -2(SP), STATUS  ;RESTORE STATUS, T-BIT, ETC.
                                        ;
                                        ;SP=
                                        ;-CALL-
                                        ;-OLD R5-
                                        ;R5=OLD PC
001734  106216                         ASRB            @SP             ;EXAMINE LOW ORDER BIT OF CALL
001736  100167                         BPL     PRINTA                  ;GO PRINT ASCII CODES.
001740  103404                         BCS     ERR2                    ;USE ODDS AS ERRORS
001742  106116                         ROLB    @SP                     ;RESTORE WORD ADDRESS
001744  162716  103414                 SUB     #JMS,   @SP             ;COMPUTE ADDRESS OF THE POINTER.
001750  013607                         MOV     @(SP)+, PC              ;GOTO THE PROCESS.
                                        ;
                                        ;
                                        ;ERROR DIAGNOSTIC GENERATOR
                                        ;
001752  012767  177560  177652  ERR2:   MOV     #TKS,   INDEV           ;RESET DEVICE POINTERS
001760  012767  177564  177646          MOV     #TPS,   OUTDEV          ;...
001766  011767  177644                 MOV     @PC,    KIN             ;RESET KBD DATA/FLAG
001772  104642  037616                 PRINT2, 37616                   ;PRINTS AS "CR-LF"+"?"
001776  042716  177740                 BIC     #-40,   @SP             ;CLEAR THE HIGH ORDER BITS
002002  112602                         MOVB    (SP)+,  PTR             ;SAVE CODE NUMBER
002004  004767  005512                 JSR     PC,     PRNT2           ;AND PRINT IT.
002010  104642  040440                 PRINT2, " A                     ; "AT "
002014  104524                         PRINT+  'T                      ;...
002016  016701  177562                 MOV     PCF,    AC              ;WHAT TYPE LINE ARE WE POINTING TO?
002022  016101  000002                 MOV     2(AC),  AC              ;RETRIEVE THAT LINE NUMBER.
002026  104626                         PRNTLN                          ;PRINT IT OUT.
002030  104642  005015                 PRINT2, CRLF                    ;GO BACK TO COMMAND/INPUT MODE.
```

```
                                        ;"START"
                                        ;
002034  016706  177650          STARTX: MOV     STACK0+2,SP             ;INITIALIZE THE STACK POINTER
002040  012767  003526  177536          MOV     #PC1+2, PCF             ;INITIALIZE PC FOR FOCAL
002046  012767  177400  177534          MOV     #-400,  DEBG            ;ENABLE TRACE BUT TURN  IT OFF
002054  005067  177560                  CLR     WHOOPS                  ;UPDATE POWER FAIL SWITCH
002060  005067  175712                  CLR     STATUS                  ;ALLOW INTERRUPTS
002064  052737  000101  177560          BIS     #101,   @#TKS           ;ENABLE TELETYPE INTERRUPTS AND SET READER RUN
002072  026727  177534  177550          CMP     INDEV,  #PRS            ;DON'T ACKNOWLEDGE FOR H,S,R,
002100  001401                          BEQ     .+4                     ;,,,
002102  104452                          PRINT+  !*                      ;PRINT THE "READY" CODE,
002104  012705  001461                  MOV     #COMBUF,R5              ;INIT THE COMMAND BUFFER PROTECTION
002110  010567  177526                  MOV     R5,     AXIN            ;FOR PACKING AND RUBOUT LIMIT,
002114  104606          IGNOR:  READC                                   ;WAIT FOR KEYBOARD INPUT,
002116  120427  000012                  CMPB    CHAR,   #012            ;CHECK FOR TERMINATORS
002122  001774                          BEQ     IGNOR                   ;IGNORE LINEFEED
002124  104616                          PACKC                           ;PACK THE COMMAND STRING
002126  026727  177510  001600          CMP     AXIN,   #COMBUF+80,-1   ;AT END?
002134  002022                          BGE     BUFFUL                  ;YES!
002136  120427  000216                  CMPB    CHAR,   #CR             ;TEST FOR C,R,
002142  001364                          BNE     IGNOR                   ;NO, REPEAT,
002144  010503                          MOV     R5,     AXOUT           ;SETUP FOR READING THE COMMAND/INPUT STRING
002146  104614          GONE:   GETC                                    ;GET A CHARACTER
002150  104634                          SPNOR                           ;IGNORE SPACES BEFORE LINE NUMBERS,
002152  104654                          SKPNON                          ;BE SURE THAT IT IS A NUMBER
002154  002204                          INPUTN                          ;NOT A TERMINATOR, BEFORE STORING,
002156  004767  001272                  JSR     PC,     PROC            ;PROCESS IMMEDIATE COMMAND
002162  016703  177416                  MOV     PCF,    AXOUT           ;COMPUTE ADDRESS OF NEXT
002166  062303                          ADD     (AXOUT)+,AXOUT          ;LINE IN SEQUENCE,
002170  001721                          BEQ             STARTX          ;END FORMAT=RETURN TO C/I MODE,
002172  010367  177406                  MOV     AXOUT,  PCF             ;SAVE NEXT LINE ADDR,
002176  022323                          CMP     (AXOUT)+,(AXOUT)+       ;PREPARE TO INTERPRET,
002200  000762                          BR    GONE                      ;GO EXECUTE IT,
                                                                        ;
002202  104625          BUFFUL: ERROR+201+10,+10,                       ;ROOM ONLY FOR IMMEDIATE COMMANDS,
```

```
528
529  002204  011767  177400            INPUTN: MOV     @PC,    DEBG            ;DISABLE AND TURN OFF THE TRACE,
530  002210  104622                            GETLN                           ;READ THE DATA AS A LINE  NUMBER
531  002212  100040                            BPL     LINERR                  ;SINGLE LINE NUMBER?
532  002214  104634                            SPNOR                           ;IGNORE SPACES AFTER LINE NUMBER,
533  002216  004767  001404                    JSR     PC,     STLIN           ;PREPARE POINTER TO INSERT NEW TEXT AND
534  002222  000401                            BR              SRETN           ;SKIP TO THE TEST FOR END-OF- LINE,
535  002224  112304                            MOVB    (AXOUT)+,CHAR           ;UNPACK A CHARACTER W/O TRACE,
536  002226  104616                    SRETN:  PACKC                           ;SAVE THE BYTE ('TEMP' STILL CONTAINS (AXIN))
537  002230  122704  000216                    CMPB    #CR,    CHAR            ;TEST FOR END OF LINE,
538  002234  001373                            BNE             SRETN+2         ;GO BACK FOR ANOTHER,
539  002236  005200                            INC     TEMP                    ;ROUND UP TO
540  002240  042700  000001                    BIC     #1,     TEMP            ;EVEN ADDRESS,
541  002244  010067  177372                    MOV     TEMP,   AXIN            ;('TEMP' IS LEFT BY "PACKC")
542  002250  020067  177372                    CMP     TEMP,   STARTV          ;ANY INTERFERENCE WITH VARIABLE AREA?
543  002254  103401                            BLO     .+4
544  002256  104636                            ERASEV                          ;(USES 'AC')
545  002260  000004                            DELETE                          ;REMOVE THE OLD LINE, IF ANY, (NO BREAKS!)
546
547                                     ;AT THIS JUNCTURE
548                                     ;(PTR)=>LASTLN
549                                     ;(AXIN)=(BUFR)+NEW,LENGTH
550                                     ;(BUFR)=>THISLN
551
552  002262  010201                            MOV     PTR,    AC              ;COMPUTE THE NEXT
553  002264  061101                            ADD     @AC,    AC              ;LINE ADDRESS AS "C+2" THEN
554  002266  166701  177352                    SUB     BUFR,   AC              ;FORM "C+NEW+2",
555  002272  010177  177346                    MOV     AC,     @BUFR           ;SAVE NEW FORWARD LINKAGE,
556  002276  062701  000002                    ADD     #2,     AC              ;"C+NEW"
557  002302  160112                            SUB     AC,     @PTR            ;UPDATE OLD LINKAGE "NEW+A"
558  002304  016767  177332  177332            MOV     AXIN,   BUFR            ;POINT TO END OF LAST INSERTION,
559  002312  000650                            BR      STARTX                  ;RETURN TO COMMAND/INPUT MODE,
560
561  002314  104603                    LINERR: ERROR+201+1,+1,                 ;ILLEGAL LINE NUMBER,
```

```
                                ;"PRINT+'X"
                                ;USE LOW BITS AS ASCII CODES,
002316  106116                 PRINTA: ROLB    @SP                     ;RESTORE CODES,
002320  110466  000001                  MOVB    CHAR,   1(SP)           ;LEAVE OLD "CHAR" IN THE STACK FOR "OUTX-2"
002324  111604                          MOVB    @SP,    CHAR            ;(OLD, NEW)
002326  000316                          SWAB    @SP                     ;(NEW, OLD)
002330  000167  000456                  JMP     OUTZ                    ;PRINT IT,
                                ;"TESTC"
                                ;CALLING SEQUENCE:
                                ;TESTC  ;CALL WITH (CHAR)=TEST DATA
                                ;TADDR  ;"TERMINATOR
                                ;NADDR  ;"NUMBER
                                ;FPADDR ;"FUNCTION
                                ;       ;RETURNS IF "ALPHA
002334  004567  000566          TESTX:  JSR     R5,     SPNORX          ;TEST FOR SPACE AND IGNORE,
002340  100412                          BMI             TEX             ;MUST BE TERMINATOR
002342  005725                          TST             (R5)+           ;PREPARE SECOND RETURN
002344  104654  002366                  SKPNON, TEX                     ;IF NUMBER,TAKE EXIT
002350  122704  000056                  CMPB    #056,   CHAR            ;OR A POINT?
002354  001404                          BEQ     TEX                     ;YES, USE "NADDR,"
002356  005725                          TST             (R5)+           ;PREPARE FOR 3RD RETURN
002360  122704  000106                  CMPB    #'F,    CHAR            ;TEST FOR FUNCTION DESIGNATION
002364  001015                          BNE             SOX             ;RETURN,        MUST BE "ALPHA
002366  011505                  TEX:    MOV     @R5,    R5              ;GO BACK VIA POINTER
002370  000205                          RTS             R5              ;IN ARGUMENT LIST,
                                ;"SKPNON,YES=ADDR"
                                ;SKIP IF NOT A NUMBER
002372  120427  000060          SKPNOX: CMPB    CHAR,   #060            ;TEST 0
002376  002410                          BLT     SOX                     ;TOO SMALL
002400  120427  000072                  CMPB    CHAR,   #072            ;TEST 9
002404  000521                          BR      GROVZ                   ;NUMBER,USE YES=ADDRESS
                                        .EOT
                                ;"SORTJ"
                                ;CHARACTER TEST AND BRANCH ROUTINES
                                ;SORTJ,  LISTCHAR, LISTADDR, RETURN=IF=NOT=THERE
002406  012501                  SORTB:  MOV     (R5)+,  AC              ;PICKUP THE LIST POINTER AND
002410  120411                          CMPB    CHAR,   @AC             ;TEST WITH LIST CONTENTS,
002412  001404                          BEQ             SOUND           ;MATCH FOUND!
```

```
   617  002414  105721                  TSTB            (AC)+           ;TEST FOR END OF LIST OR
   618  002416  001374                  BNE             SORTB+2         ;REPEAT IF NOT AT END
   619  002420  005725          SOX:    TST             (R5)+           ;RETURN IF NO MATCH FOUND (SKIP OVER EXIT)
   620  002422  000205                  RTS       R5                    ;GO BACK VIA R5.
   621                                                                   ;
   622  002424  166501  177776  SOUND:  SUB     -2(R5), AC              ;COMPUTE THE INDEX FIRST THEN
   623  002430  006301                  ASL     AC                      ;MAKE EVEN AND
   624  002432  061501                  ADD     @R5,    AC              ;GET TABLE OF ADDRESSES AND FINALLY
   625  002434  011105                  MOV     @AC,    R5              ;SETUP ADDRESS FOR PC,
   626  002436  000205                  RTS             R5              ;GO BACK VIA NEW R5.
   627
   628
   629
   630                          ;"SORTC"
   631
   632                          ;SORTC,LISTCHAR,YESADDR,RETURN-IF-NOT-THERE
   633
   634  002440  012501          SORTD:  MOV     (R5)+,  AC              ;GET LIST ADDRESS
   635  002442  121104                  CMPB    @AC,    CHAR            ;COMPARE WITH CONTENTS
   636  002444  001750                  BEQ             TEX             ;FOUND IT,
   637  002446  105721                  TSTB            (AC)+           ;TEST NEXT FOR END
   638  002450  001374                  BNE             SORTD+2         ;REPEAT
   639  002452  000762                  BR              SOX             ;EXIT IF NOT THERE
   640
   641
```

```
                                ;"GETLN"
                                ;
                                ;LINE NUMBER FORMATION ROUTINE
                                ;
002454  105067  177151  GETLNX: CLRB            SWITCH          ;SET TO TERMINATE UPON ALPHA
                                                                ;CODES (ENTRY POINT #1)
002460  005001                  CLR             AC              ;FOR "ALL" USE ZERO,
002462  104634                  SPNOR                           ;IGNORE LEADING SPACES
002464  100416                  BMI     GALL1                   ;TERMINATOR=0="ALL"
002466  122704  000101          CMPB    #'A,    CHAR            ;TEST FOR "ALL
002472  001413                  BEQ     GALL1                   ;GO SET SWITCH AND RETURN OK
002474  104654  002562          SKPNON, GTESTN                  ;SOUNDS LIKE A VARIABLE NAME?
002500  010546                  MOV     R5,     -(SP)           ;SAVE RETURN
002502  004767  002166          JSR     PC,     GETARG          ;READ A NAMED LINE
002506  007001                  FPGET+IPTR                      ;PICKUP THE VALUE,
002510  012605                  MOV     (SP)+,  R5              ;RESTORE RETURN
002512  062767  000010  177076  GTESTW: ADD     #8.,    BE      ;MOVE 'POINT'RIGHT 8 BITS (ENTRY POINT #2)
002520  007071                  FPINT                           ;FIX
002522  010167  177076  GALL1:  MOV     AC,     LINENO          ;SAVE THE ANSWER IN "AC" AND "LINENO"
002526  001412                  BEQ             GALL            ;ZERO, CALL IT THE SAME AS "ALL"
002530  100671                  BMI     LINERR                  ;PREVENT USE OF BIT 15
002532  105701                  TSTB            AC              ;TEST THE STEP-NO,
002534  001411                  BEQ             GGROUP          ;MUST BE A GROUP,
002536  105767  177063          TSTB            LINENO+1        ;TEST THE GROUP-NO
002542  001664                  BEQ     LINERR                  ;MUST BE NON-ZERO
002544  152767  000200  177057  BISB    #ONE,   SWITCH          ;INDICATE A SINGLE LINE
002552  000205                  RTS             R5              ;RETURN
                                ;
002554  105267  177051  GALL:   INCB            SWITCH          ;SET BIT #0 FOR "ALL",
002560  000205          GGROUP: RTS             R5              ;RETURN TO PROCESS WITH STATUS BITS SETUP,
                                                                ;LINE NOS, >99,99 ARE NOT ERROR CHECKED,
                                ;
002562  007075          GTESTN: FREAD                           ;SET TO READ A NUMBER INTERNALLY,
002564  000752                  BR      GTESTW                  ;GO PROCESS THE RESULT
```

```
                                        ;"FINDLN"
                                        ;THIS ROUTINE LOOKS UP THE LINE WHOSE NUMBER
                                        ;MATCHES THE CONTENTS OF "LINENO"
                                        ;CALLING SEQUENCE: FINDLN,      NOTADDR,RETURN-IF-FOUND

                                        ;RESULTS IF FOUND:"THISLN"=FOUND LINE OR NEXT LARGER:
                                        ;"PTR" IS THE LAST LINE, I.E.,PRECEEDING OR SAME LINE,
                                        ;"AXOUT"IS SET FOR USE BY "GETC",

                                        ;RESULTS IF-NOT-FOUND: "THISLN"=ADDRESS OF NEXT IN LINE
                                        ;                       "AXOUT"=NEXT OR ZERO
                                        ;                       "PTR"=PRIOR LINE

002566  012703  001654          FINDX:  MOV     #CFRS,  AXOUT           ;LOAD STARTING ADDRESS OF TEXT
002572  010302                          MOV     AXOUT,  PTR             ;INIT FOLLOWING POINTER
002574  010367  177006          FINDN:  MOV     AXOUT,  THISLN          ;SAVE CURRENT POINTER
002600  026367  000002  177016          CMP     2(AXOUT),LINENO         ;TEST FOR MATCH
002606  101267                          BHI             TEX             ;PAST IT!=NOT FOUND
002610  001404                          BEQ     FINDO                   ;RIGHT ON!
002612  010302                          MOV     AXOUT,  PTR             ;COPY PRIOR POINTER,
002614  062303                          ADD     (AXOUT)+,AXOUT          ;GET NEW POINTER
002616  001663                          BEQ             TEX             ;END OF LIST=NOT FOUND
002620  000765                          BR              FINDN           ;TRY THE NEXT ONE,
002622  022323                  FINDO:  CMP     (AXOUT)+,(AXOUT)+       ;MAKE IT POINT TO TEXT
002624  000675                          BR              SOX             ;RETURN TO SEQUENCE

                                        ;DATA STRUCTURE OF LINES;

                                        ;WORD #1 :      NEXT-,-2                ;LAST IS 0-,-2
                                        ;WORD #2 :      LINE#                   ;GROUP, STEP
                                        ;WORDS #3-N:    7-BIT ASCII AND SPECIAL INTERNAL TEXT CODES
                                        ;LAST BYTE :     216            ;CARRIAGE RETURN
```

```
                                ;"PRINT2,ARGARG"
002626  112504                  PRIN2A: MOVB    (R5)+,  CHAR            ;COPY FIRST TRAILING BYTE
002630  104604                          PRINTC                          ;AND PRINT
002632  112504                          MOVB    (R5)+,  CHAR            ;COPY SECOND TRAILING BYTE
002634  000462                          BR      OUT                     ;AND GO PRINT IT


                                ;"SKPLPR,NOT-ADDR"

                                ;BRANCH IF LEFT-PARENS, NOT FOUND,

002636  120427  000210          XTSTLP: CMPB    CHAR,   #210            ;TEST FOR (<[,
002642  101251                          BHI     TEX                     ;RIGHT TERMINATOR? - YES, USE "NOT-ADDR",
002644  120427  000206                  CMPB    CHAR,   #206            ;OUT OF RANGE?
002650  103646                  GROVZ:  BLO     TEX                     ;YES, USE "NOT-ADDR"
002652  000662                          BR      SOX                     ;OK, SKIP ONWARDS,


                                ;"DIGTST,FIELD"

002654  012704  000060          DIGTSA: MOV     #60,    CHAR            ;INITIALIZE CHARACTER
002660  020215                          CMP     PTR,    @R5             ;TEST FOR POSSIBILITY
002662  002656                          BLT             SOX             ;LEAVE IF NO MORE POSSIBLE
002664  161502                          SUB     @R5,    PTR             ;MAKE CHANGE AND
002666  005204                          INC             CHAR            ;COUNT
002670  000773                          BR              DIGTSA+4        ;REPEAT


                                ;"GROOVY,NOT-ADDR"

002672  126763  176727  000003  GROVX:  CMPB    LINENO+1,3(AXOUT)       ;TEST FOR SAME GROUP
002700  001232                          BNE     TEX                     ;GO BRANCH OR
002702  005703                          TST     AXOUT                   ;CHECK FOR END OF TEXT
002704  001630                          BEQ     TEX                     ;TAKE NOT-ADDR
002706  000644                          BR      SOX                     ;JUST RETURN


```

```
                                        ;"READC" AND "PRINTC"

                                        ;I/O CONTROLS

002710  104612                 CHIN:  INCH                          ;READC = INPUT
002712  042704  177600                BIC     #-200,  CHAR         ;CLEAR HIGH ORDER BITS
002716  001774                        BEQ             CHIN         ;IGNORE NULLS
002720  022704  000003                CMP     #3,     CHAR         ;CONTROL-C?
002724  001462                        BEQ     INIT2                ;YES=RESTART
002726  104602  001450  003214        SORTC,  ECHOLST,RUBX2        ;TEST FOR NO-ECHO
002734  120427  000141                CMPB    CHAR,#141            ;##IF SMALL LETTER INPUT,
002740  002402                        BLT     .+6                  ;###
002742  042704  000040                BIC     #40,CHAR             ;###MAKE IT UPPER CASE,
002746  104602  001412  002756        SORTC,  TERMS,CHINX          ;CONVERSION TEST,
002754  000403                        BR              OUTW         ;GO ECHO
                                                                   ;
002756  062701  176566         CHINX: ADD     #200-TERMS,AC        ;FORM INTERNAL CODE
002762  110104                        MOVB    AC,     CHAR         ;AND SAVE IT
002764  022705  003566         OUTW:  CMP     #SCONL,R5            ;DON'T
002770  001511                        BEQ     RUBX2                ;ECHO SEARCH CHARACTER FROM *MODIFY*,
002772  026727  176634  177550        CMP     INDEV,  #PRS         ;DON'T
003000  001505                        BEQ     RUBX2                ;ECHO FOR H,S,R,
003002  110446                 OUT:   MOVB    CHAR,   -(SP)        ;SAVE THIS FORM ON THE STACK,
003004  100002                        BPL     OUTZ                 ;IF SPECIAL TERM,, REGENERATE BY
003006  116404  001612                MOVB    TERMS+200(CHAR),CHAR ;COMPUTING ASCII
003012  152704  000200         OUTZ:  BISB    #200,   CHAR         ;SET BIT8,
003016  104610                        OUTCH                        ;OUTPUT TO ANY DEVICE,
003020  122704  000215                CMPB    #215,   CHAR         ;(CHANGED BY !OUTCH!)
003024  001006                        BNE     OUTY                 ;JUST GO COUNT IT,
003026  112767  000111  176574        MOVB    #73.,   LINCNT       ;INITIALIZE THE LINE COUNT,
003034  121604                        CMPB    @SP,    CHAR         ;WAS THIS AN INTERNAL CR?
003036  103401                        BLO     OUTY                 ;IF NOT, JUST GO COUNT,
003040  104412                        PRINT+012                    ;ISSUE THAT EXTRA LINEFEED,
003042  026727  176566  177564 OUTY:  CMP     OUTDEV, #TPS         ;TEST FOR TTY OUTPUT,
003050  001005                        BNE     OUTX                 ;IF NOT, DON'T EDITORIALIZE,
003052  105367  176552                DECB    LINCNT               ;COUNT PRINT POSITIONS
003056  001002                        BNE     OUTX                 ;SKIP IF NOT NEAR THE MARGINS
003060  104642  005015                PRINT2, CRLF                 ;OUTPUT ONE OF EACH
003064  112604                 OUTX:  MOVB    (SP)+,  CHAR         ;RESTORE ORIGINAL DATA
003066  000205                        RTS     R5                   ;RETURN FROM TRAP


003070  104636                 ENIT2: ERASEV                  ;ERASE BUFFER ON INIT
003072  016706  176612         INIT2: MOV     STACK0+2,SP          ;RESET STACK FOR MANUAL RESTART
003076  104601                        ERROR+201+0,+0,
```

```
                                ;"PRNTLN"
                                ;PRINT A LINE NUMBER ROUTINE
003100  012702  002005          XPRNTL: MOV     #2005,  PTR             ;SET FORMAT TO %4,02
003104  010146                          MOV     AC,     -(SP)           ;ARGUMENT TAKEN FROM "AC"
003106  012746  177407                  MOV     #177407,-(SP)           ;...
003112  007003                          FPGET+FROM+STACK                        ;LOAD FLAC
003114  007076                          FPRINT                          ;PRINT RESULT
003116  022626                          CLOSE+STACK                     ;REPAIR HOLE IN THE STACK.
003120  104440                          PRINT+  ' '                     ;PRINT TRAILING SPACE
003122  000205                          RTS     R5                      ;RETURN


                                ;"SPNOR"
                                ;IGNORE SPACES
003124  104614                  SPNXT:  GETC                            ;MOVE ON TO NEXT CHARACTER CODE.
003126  122704  000200          SPNORX: CMPB    #200,CHAR               ;CHECK FOR SPACE SYMBOL.
003132  001774                          BEQ     .-6                     ;TRY AGAIN.
003134  000427                          BR      RUBX2                   ;LEAVE "CHAR" IN "STATUS" AND EXIT.
```

```
                                        ;"PACKC"

                                        ;TEXT BUFFER CONTROLS

003136  016700  176500   PACKX:  MOV     AXIN,   TEMP            ;COPY INPUT TEXT POINTER.
003142  122704  000177           CMPB    #177,   CHAR            ;TEST FOR RUBOUT
003146  001414                   BEQ             RUBIT           ;GO BACK UP ONE SPACE
003150  122704  000100           CMPB    #100,   CHAR            ;TEST FOR  AT SIGN
003154  001415                   BEQ     RUBX                    ;IGNORE IT
003156  122704  000137           CMPB    #137,   CHAR            ;LEFT ARROW
003162  001405                   BEQ     PBAR                    ;GO RESET.
003164  110420                   MOVB    CHAR,   (TEMP)+         ;SAVE CHARACTER CODE AND MOVE POINTER.
003166  026700  176460           CMP     BOTTOM, TEMP            ;TEST FOR END
003172  101006                   BHI     RUBX                    ;CONTINUE
003174  104625                   ERROR+201+10,+10.               ;C,F, INPUT;
003176                                                           ;
003176  011600           PBAR:   MOV     @SP,    TEMP            ;RESET INPUT POINTER.
003200  020016           RUBIT:  CMP     TEMP,   @SP             ;TEST FOR NULL LINE(OLD R5 IS "PACKST")
003202  001402                   BEQ             RUBX            ;IGNORE R,O, CODE COMPLETELY
003204  005300                   DEC             TEMP            ;BACKUP ONE PLACE
003206  104534                   PRINT+  '\                      ;AND ACKNOWLEDGE RECEIPT + USE.
003210  010067  176426   RUBX:   MOV     TEMP,   AXIN            ;SAVE POINTER
003214  110404           RUBX2:  MOVB    CHAR,   CHAR            ;SET CONDITION CODES BEFORE LEAVING.
003216  000205                   RTS     R5                      ;RETURN TO MAINLINE ROUTINES.

                                        ;"GETC"

                                        ;UNPACK A CHARACTER AND LEAVE IN 'STATUS'

003220  105767  176364   UTX:    TSTB            DEBG            ;TEST FOR TRACE ENABLED
003224  001373                   BNE             RUBX2           ;RETURN IF NOT ENABLED.
003226  105167  176357           COMB            DEBG+1          ;FLIP THE TRACE FLOP
003232  112304           GETX:   MOVB    (AXOUT)+,CHAR           ;PICK OUT NEXT BYTE
003234  122704  000077           CMPB    #'?,    CHAR            ;CHECK FOR TRACE FLIP-FLOP CODE
003240  001767                   BEQ             UTX             ;GO FLIP IT IF CODE FOUND PLUS ENABLED.
003242  005767  176342           TST             DEBG            ;TEST FOR BOTH DEBG+DMPS=0,
003246  001362                   BNE             RUBX2           ;NOT IN TRACE NOW.
003250  000654                   BR              OUT             ;GO PLAY-BACK THE BYTE.
```

```
                                ;"DELETE"
                                ; A LINE AND
                                ;GARBAGE COLLECTION IS DONE UP TO (STARTV);
                                ;(BUFR) IS CORRECTED;
                                ;(TEMP) IS AMOUNT OF CODE COLLECTED.
003252  012324                ECOLOGY;MOV     (AXOUT)+,(CHAR)+        ;STEP 3-COLLECT SPACE
003254  020367  176366                CMP     AXOUT,  STARTV          ;TEST FOR COMPLETION
003260  101774                        BLOS            ECOLOGY         ;CONTINUE UNTIL FINISHED.
003262  160067  176356                SUB     TEMP,   BUFR            ;UPDATE END OF TEXT POINTER.
003266  160067  176350                SUB     TEMP,   AXIN            ;...
                                                                        ;*** NO INTERRRUPTS!
003272  104624  006034        XDELET; FINDLN, PWRON                   ;SETUP LINE POINTERS FOR EXIT
003276  112304                XD3;    MOVB    (AXOUT)+,CHAR           ;READ THROUGH THE LINE
003300  120427  000216                CMPB    CHAR,   #CR             ;CARRIAGE RETURN MARKS END
003304  001374                        BNE             XD3             ;REPEAT UNTIL END REACHED.
003306  005203                        INC             AXOUT           ;ROUND OUT THE POINTER
003310  042703  000001                BIC     #1,     AXOUT           ;TO AN EVEN NUMBER.
003314  016704  176266                MOV     THISLN, CHAR            ;COPY POINTER TO THIS LINE.
003320  061412                        ADD     @CHAR,  @PTR            ;STEP 1-CREATE NEW RELATIVE
003322  062712  000002                ADD     #2,     @PTR            ;POINTER TO NEXT LINE IN LIST.
003326  010300                        MOV     AXOUT,  TEMP            ;COMPUTE DELTA POSITION
003330  160400                        SUB     CHAR,   TEMP            ;AS A POSITIVE, EVEN NO. OF BYTES.
003332  012701  001654                MOV     #CFRS,  AC              ;BEGIN AT TOP TO GARBAGE COLLECI.
003336  010102                XDOX;   MOV     AC,     PTR             ;STEP 2-FOLLOW + UPDATE LINKS ("THIS")
003340  001744                        BEQ     ECOLOGY                 ;GO COLLECT ALL
003342  062101                        ADD     (AC)+,  AC              ;TEST FOR LAST OF KIND.
003344  020203                        CMP     PTR,    AXOUT           ;TEST FOR ABOVE OR BELOW CHANGE AND
003346  103404                        BLO             XDTHIS          ;BRANCH TO FIXUP THIS ONE IF IT  IS  ABOVE
003350  020103                        CMP     AC,     AXOUT           ;TEST FOR NEXT-IS-BELOW AND
003352  103371                        BHIS            XDOX            ;BRANCH TO NEXT ONE IF ALSO BELOW
003354  060012                        ADD     TEMP,   @PTR            ;ADD THE CHANGE AND
003356  000767                        BR              XDOX            ;GO LOOK AT NEXT ITEM.
                                                                        ;
003360  020103                XDTHIS; CMP     AC,     AXOUT           ;IS NEXT ONE ABOVE THE CHANGE ALSO?
003362  103765                        BLO             XDOX            ;YES, CONTINUE.
003364  160012                        SUB     TEMP,   @PTR            ; NO,CHANGE "LINE".
003366  000763                        BR              XDOX            ;GO TO NEXT LINE

                                ;       HERE-THERE=CHANGE HERE
                                ;       A       A       0
                                ;       B       B       0
                                ;       A       B       -T
                                ;       B       A       +T
```

```
931
932
933                                     ;*IF*
934
935                                     ;CONDITIONAL TRANSFER PROCESS
936
937                                     ;THE FLAVORS OF *IF*
938                                     ;IF (EXP)=,0,+  [3-WAY]
939                                     ;         =,0;...[2-WAY OR,..]
940                                     ;         =,0    [2-WAY OR NEXT LINE]
941                                     ;         =;...  [1-WAY OR,..]
942                                     ;         =      [1-WAY OR NEXT LINE]
943
944  003370  104614                    LOST:   GETC
945  003372  000410                            BR      LOSE2
946
947
948
949  003374  104634                    IF:     SPNOR                           ;GOTO LPAR (ENTRY POINT)
950  003376  010446                            MOV     CHAR,   -(SP)           ;SAVE LPAR FOR "PARTST",
951  003400  104660                            EVAL,X                          ;EVALUATE THE EXPRESSION WITH PARENTHESES
952  003402  104646                            PARTST                          ;CHECK CLOSING PARENS AND DO 'GETC',
953  003404  005767  176210                    TST     HORD                    ;TEST SIGN
954  003410  100412                            BMI     GOTO                    ;READY FOR =
955  003412  001405                            BEQ     LOSE1                   ;SKIP LINE NUMBER(S)
956  003414  120427  000214            LOSE2:  CMPB    CHAR,   #214            ;TEST FOR END OF LINE NO
957  003420  103763                            BLO     LOST                    ;NOT YET
958  003422  101014                    LOSE0:  BHI     PROC                    ;SEMI, OR C.R, (OR =!)
959  003424  104614                            GETC                            ;COMMA #1
960  003426  120427  000214            LOSE1:  CMPB    CHAR,   #214            ;LOOK FOR COMMA
961  003432  001373                            BNE     LOSE0                   ;GO TEST OTHERS
962  003434  104614                            GETC                            ;SKIP THE COMMA
963
964                                     ;FALL THROUGH INTO *GOTO*
965
```

```
  967
  968                                   ;*GO*GOTO*COMMENT*CONTINUE*RETURN*XECUTE*
  969
  970                                   ;PRIMARY CONTROL AND TRANSFER
  971
  972  003436  104622                  GOTO:   GETLN                           ;READ THE ADDRESS AND
  973  003440  104624  003624                  FINDLN, SERR                    ;ATTACH TO NEW LINE.
  974  003444  016767  176136  176132  PSCAN:  MOV     THISLN, PCF             ;SET NEW LINE POINTER
  975  003452  104614                  PROCESS:GETC                            ;READ A CHARACTER IN LINE
  976  003454  120427  000216          PROC:   CMPB    CHAR,   #CR             ;TEST FOR END OF LINE
  977  003460  001421                          BEQ             PC1             ;GO-ON TO PROCESS NEXT LINE
  978  003462  110401                          MOVB    CHAR,   AC              ;COPY DATA
  979  003464  100772                          BMI     PROCESS                 ;IGNORE TERMINATORS
  980  003466  122704  000101                  CMPB    #'A,    CHAR            ;CHECK DATA
  981  003472  003003                          BGT     ERRORC                  ;TOO LOW,
  982  003474  122704  000130                  CMPB    #'X,    CHAR            ;TOO HIGH?
  983  003500  002001                          BGE     PC2                     ;OK
  984  003502  104611                  ERRORC: ERROR+201+4,+4,                 ;ILLEGAL COMMAND CODE,
  985  003504  104614                  PC2:    GETC                            ;IGNORE REST OF THE COMMAND'S
  986  003506  100376                          BPL     PC2                     ;CHARACTERS UNTILL TERMINATOR REACHED,
  987  003510  006301                          ASL     AC                      ;MAKE BYTE COUNT INTO WORD COUNT,
  988  003512  000171  001046                  JMP @COMLST-202(AC)             ;BRANCH TO THE COMMAND PROCESS,
  989                                                                          ;
  990  003516  012767  003526  176060  RETURN: MOV     #PC1+2, PCF             ;RETURN FROM SUBROUTINE (?)
  991  003524  000207                  PC1:    POPJ                            ;EXIT FROM LINE
  992  003526  174250                          =,-2                            ;DUMMY TERMINATOR
  993  003530  000000                  FLTZER: 00000                           ;DUMMY LINE NUMBER ZERO
  994  003532  000000                          00000                           ;AND DUMMY VALUE OF FLOATING ZERO,
  995                                                                          ;
  996  003534  005726                  TPR:    TST     (SP)+                   ;DUMP RETURN FOR 'TASK' UPON SEMICOLONS,
  997  003536  000745                          BR              PROCESS         ;GO END REST OF COMMAND LINE,
  998                                                                          ;
  999  003540  005726                  TPR1:   TST             (SP)+           ;DUMP RETURN UPON C.R,
 1000  003542  000207                          POPJ                            ;GO EXIT FROM A LINE,
 1001                                                                          ;
 1002  003544  004767  000750          XECUTE: JSR     PC,     EVAL            ;RUN THROUGH SOME FUNCTION CALLS,
 1003  003550  000741                          BR      PROC                    ;THEN GO GET NEXT COMMAND,
```

```
1005
1006                                    ;*MODIFY*
1007
1008                                    ;SEARCH FOR CHARACTER IN TEXT
1009
1010  003552  104622                    MODIFY: GETLN                           ;READ COMMAND ARGUMENT
1011  003554  104624  003624                    FINDLN, SERR                    ;LOOKUP THE INPUT DATA
1012  003560  004767  000042                    JSR     PC,     STLIN
1013  003564  104606                    SCONT:  READC                           ;READ SEARCH CHARACTER SILENTLY,
1014  003566  110467  175665            SCONL:  MOVB    CHAR,   LIST3+1         ;SAVE SEARCH CHARACTER
1015  003572  112304                    SCHAR:  MOVB    (AXOUT)+,CHAR           ;UNPACK AND PRINTOUT
1016  003574  104604                            PRINTC                          ;EXTRA OUTPUT FOR C,R,
1017  003576  104600  001456  001406            SORTJ,  LIST3,LISTGO            ;TEST FOR C,R, OR SEARCH CHARACTER
1018  003604  104616                            PACKC                           ;SAVE OLD CHARACTER,
1019  003606  000771                            BR              SCHAR           ;REPEAT
1020                                                                            ;
1021  003610  104606                    SFIND:  READC                           ;ABSORB AND ANALYSE
1022  003612  104600  001453  001400            SORTJ,  LIST6,SRNLST            ;THE INPUT TEXT
1023  003620  104616                    SFOUND: PACKC                           ;PACK NEW CHARACTER
1024  003622  000772                            BR              SFIND           ;REPEAT
1025                                                                            ;
1026  003624  104613                    SERR:   ERROR+201+5,+5,                 ;NONEXISTANT LINE OR LINE ZERO
1027
1028
1029
1030
1031
1032                                    ;START-UP-A-LINE SUBROUTINE
1033
1034  003626  016705  176012            STLIN:  MOV     BUFR,   R5              ;COMPUTE START OF NEW LINE
1035  003632  005025                            CLR             (R5)+           ;ZERO LIST LINK AND
1036  003634  010125                            MOV     AC,     (R5)+           ;SAVE LINE NUMBER
1037  003636  001772                            BEQ     SERR                    ;FLAG "M 0" ERROR IMMEDIATELY,
1038  003640  010567  175776                    MOV     R5,     AXIN            ;SETUP INPUT POINTER
1039  003644  000207                            RTS     PC
```

```
1041
1042                                    ;*WRITE*
1043
1044                                    ;OUTPUT COMMAND TEXT
1045
1046  003646  104622                    WRITE:  GETLN                           ;READ THE ARGUMENT
1047  003650  010446                            MOV     CHAR,   -(SP)           ;PERMIT FOLLOWING SEMICOLON.
1048  003652  010346                            MOV     AXOUT,  -(SP)           ;
1049  003654  104624  003712            WRITE2: FINDLN, WTESTG                  ;LOOKUP THE LINE
1050  003660  016301  177776                    MOV     -2(AXOUT),AC            ;TEST FOR LINE ZERO
1051  003664  001401                            BEQ             WRITEL          ;BRANCH TO PRINT TITLE ONLY
1052  003666  104626                            PRNTLN                          ;PRINT NON-ZERO LINE NOS, IN "AC"
1053  003670  112304                    WRITEL: MOVB    (AXOUT)+,CHAR           ;READ W/O TRACE
1054  003672  104604                            PRINTC                          ;PRINT ONE CHARACTER
1055  003674  122704  000216                    CMPB    #CR,    CHAR            ;TEST FOR END
1056  003700  001373                            BNE             WRITEL          ;REPEAT
1057  003702  016703  175700                    MOV     THISLN, AXOUT           ;COMPUTE NEXT LINE
1058  003706  062303                            ADD     (AXOUT)+,AXOUT          ;ADDRESS READY NOW.
1059  003710  001414                            BEQ             WGO             ;LEAVE IF LAST LINE.
1060  003712  105767  175713            WTESTG: TSTB            SWITCH          ;TEST FOR SINGLE LINE
1061  003716  100411                            BMI             WGO             ;YES=EXIT
1062  003720  104650  003730                    GROOVY, WRED                    ;SAME GROUP AS LAST LINE?
1063  003724  104630                    WRIG:   COPYLN                          ;COPY THIS NEXT LINE NUMBER.
1064  003726  000752                            BR      WRITE2                  ;GO FIND IT.
1065                                            ;
1066  003730  104604                    WRED:   PRINTC                          ;PRINT EXTRA CR AFTER GROUP.
1067  003732  132767  000001  175671            BITB    #ALL,   SWITCH          ;TEST FOR "ALL"?
1068  003740  001371                            BNE             WRIG            ;YES,KEEP IT UP.
1069  003742  000522                    WGO:    BR      DOXIT                   ;RETURN
1070
1071
```

```
1073
1074                                    ;*ERASE*ERASE ALL*ERASE TEXT*ERASE !GROUP,LINE!*
1075
1076  003744  104620                    ERASE:  TESTC                           ;TEST THE ARGUMENT, IF ANY.
1077  003746  004034                            ERVC                            ;ERASE "VARIABLES
1078  003750  004052                            ERL                             ;ERASE LINE OR GROUP OF TEXT
1079  003752  002314                            LINERR                          ;ERROR,
1080  003754  122704  000101                    CMPB    #'A,    CHAR            ;TEST FOR "ALL
1081  003760  001003                            BNE             ERT             ;WHY NOT USE A VARIABLE NAME ?
1082  003762  104640                            ERASET                          ;OUT THE TEXT
1083  003764  104636                    ERV:    ERASEV                          ;ERASE THE VARIABLES ALSO
1084  003766  104632                            START                           ;GO TO COMMAND/INPUT MODE
1085                                            ;
1086  003770  122704  000124            ERT:    CMPB    #'T,    CHAR            ;"ERASE TEXT" ONLY ?
1087  003774  001002                            BNE     ERL                     ;NO, DO LINE
1088  003776  104640                            ERASET                          ;YES
1089  004000  104632                            START                           ;GO BACK
1090                                            ;
1091  004002  104622                    ERL:    GETLN                           ;READ LINE NUMBER,
1092  004004  005701                            TST     AC                      ;DON'T ERASE LINE
1093  004006  001706                            BEQ     SERR                    ;ZERO!
1094  004010  000004                    ERG:    DELETE                          ;EXTRACT ONE LINE (NO BREAKS!)
1095  004012  105767  175613                    TSTB    SWITCH                  ;TEST FOR SINGLE OR GROUP
1096  004016  100762                            BMI             ERV             ;ONLY ONE
1097  004020  005703                            TST     AXOUT                   ;CHECK FOR END OF LIST TO
1098  004022  001760                            BEQ     ERV                     ;AVOID REALLY WILD LOOP!
1099  004024  104650  003764                    GROOVY, ERV                     ;TEST FOR SAME GROUP MEMBER
1100  004030  104630                            COPYLN                          ;MOVE AHEAD
1101  004032  000766                            BR              ERG             ;AND DO ANOTHER,
1102                                            ;
1103  004034  104636                    ERVC:   ERASEV                          ;*E* COMMAND IN TEXT IS OK,
1104  004036  000606                            BR      PROC                    ;GO TO NEXT COMMAND OF PROGRAM,
1105
1106
1107
1108                                    ;"COPYLN"
1109
1110                                    ;USED BY *WRITE*, *ERASE*, AND *DO*
1111
1112  004040  016367  000002  175556    COPYLX: MOV     2(AXOUT),LINENO         ;USE NEXT LINE NUMBER
1113  004046  000205                            RTS     R5                      ;RETURN TO *WRITE*ERASE*DO*,
1114
1115
1116                                    ;"ERASET"
1117
1118  004050  016767  175574  175566    ERTX:   MOV     TOP,    BUFR            ;ERASE ALL TEXT
1119  004056  012767  176122  175570            MOV     #0-CFRS-2,CFRS          ;INITIALIZE LINE ZERO POINTER DATA
1120  004064  000205                            RTS     R5
```

```
1122
1123                                    ;*DO*
1124
1125                                    ;RECURSIVE OPERATE
1126
1127  004066  104622                    DO:     GETLN                           ;READ LINE # ARGUMENT
1128  004070  010446                            MOV     CHAR,   -(SP)           ;SAVE THE NEXT CHARACTER,
1129  004072  010346                            MOV     AXOUT,  -(SP)           ;CHARACTER POINTER OF CURRENT LOCATION
1130  004074  016746  175504                    MOV     PCF,    -(SP)           ;SAVE ADDRESS OF LINE AND
1131  004100  104624  004106                    FINDLN, DOGR                    ;LOOKUP THE LINE
1132  004104  000407                            BR              DOGRP1          ;FOUND!
1133                                                                            ;
1134  004106  105767  175517            DOGR:   TSTB            SWITCH          ;TEST FOR SINGLETON
1135  004112  100442                            BMI     DOER                    ;YES, OUGHT TO HAVE BEEN THERE,
1136                                                                            ;C(THISLN)=C(AXOUT),
1137  004114  104650                            GROOVY                          ;COMPARE GROUP NOS,
1138  004116  004220                                            DOER            ;ERROR, NO SUCH GROUP
1139  004120  104630                    DOGRP2: COPYLN                          ;COPY FIRST LINE NO, OF THE GROUP
1140  004122  022323                            CMP     (AXOUT)+,(AXOUT)+       ;POINT FORWARD
1141  004124  116746  175501            DOGRP1: MOVB    SWITCH, -(SP)           ;SAVE FLAGS
1142  004130  016746  175452                    MOV     THISLN, -(SP)           ;SAVE ADDRESS OF LINE BEING DONE
1143  004134  004767  177304                    JSR     PC,     PSCAN           ;SCAN COMMANDS IN THAT LINE
1144  004140  012600                            MOV     (SP)+,  TEMP            ;RESTORE LINE LAST DONE ADDRESS
1145  004142  112667  175463                    MOVB    (SP)+,  SWITCH          ;RESTORE CORRECT SCOPE OF "DO"
1146  004146  100416                            BMI     DOCONT                  ;IF SINGLE LINE, WE ARE DONE NOW,
1147  004150  016703  175430                    MOV     PCF,    AXOUT           ;KEEP POINTER TO NEXT LINE TO BE DONE,
1148  004154  062303                            ADD     (AXOUT)+,AXOUT          ;COMPUTE NEXT ADDRESS IN GROUP,
1149  004156  001412                            BEQ     DOCONT                  ;LEAVE IF OUT OF TEXT ALTOGETHER
1150  004160  010367  175422                    MOV     AXOUT,  THISLN          ;SAVE POINTER
1151  004164  132767  000001  175437            BITB    #ALL,   SWITCH          ;TEST FOR "DO" OR "DO ALL"
1152  004172  001352                            BNE             DOGRP2          ;...
1153  004174  126063  000003  000003            CMPB    3(TEMP),3(AXOUT)        ;COMPARE GROUP NOS,
1154  004202  001746                            BEQ             DOGRP2          ;GO DO NEXT ONE,
1155  004204  012667  175374            DOCONT: MOV     (SP)+,  PCF             ;...
1156  004210  012603                    DOXIT:  MOV     (SP)+,  AXOUT           ;...
1157  004212  012604                            MOV     (SP)+,  CHAR            ;RESTORE THE LAST CHARACTER,
1158  004214  000167  177234                    JMP             PROC            ;CONTINUE THE STRING
1159                                                                            ;
1160  004220  104615                    DOER:   ERROR+201+6,+6,                 ;NO SUCH GROUP TO BE DONE,
1161
```

```
1163                                        ;*SET*FOR*
1164
1165                                        ;LOOP CONTROL STATEMENT
1166
1167            004222                      SET=.
1168  004222    004767  000446            FOR:    JSR     PC,     GETARG      ;LOCATE THE VARIABLE
1169  004226    104634                            SPNOR                       ;IGNORE TRAILING SPACES (0!D)
1170  004230    122704  000217                    CMPB    #217,   CHAR        ;TEST FOR "="
1171  004234    001006                            BNE     FINERR              ;ERROR TO LEFT OF = SIGN
1172  004236    010246                            MOV     PTR,    -(SP)       ;SAVE VARIABLE POINTER ON THE STACK.
1173  004240    104660                            EVAL,X                      ;EVALUATE RIGHT HAND EXP.
1174  004242    007064                            FPPUT+THROUGH+STACK         ;UPDATE+INDEX VALUE
1175  004244    104600  001441  001336            SORTJ,TLIST,FLIST1          ;TEST TERMINATOR
1176  004252    104617                    FINERR: ERROR+201+7,+7.             ;ILLEGAL FORMAT IN *SET* OR *FOR* COMMAND
1177  004254    104660                    FINCR:  EVAL,X                      ;EVALUATE EXPRESSION.
1178  004256    104600  001441  001330            SORTJ,TLIST,FLIST2          ;TEST TERMINATOR
1179  004264    000772                            BR              FINERR      ;ERROR CALL
1180                                              ;
1181  004266    012746  040000            FINFIN: MOV     #40000, -(SP)       ;SET THE
1182  004272    012746  000001                    MOV     #1,     -(SP)       ;INCREMENT TO UNITY
1183  004276    000403                            BR              FCONT       ;GO SAVE THE LIMIT.
1184                                              ;
1185  004300    024646                    FLIMIT: OPEN+STACK                  ;SAVE INCREMENT
1186  004302    007063                            FPPUT+INTO+STACK            ;...
1187  004304    104660                            EVAL,X                      ;EVALUATE LIMIT
1188  004306    024646                    FCONT:  OPEN+STACK                  ;SAVE THE LIMIT.
1189  004310    007063                            FPPUT+INTO+STACK            ;...
1190  004312    010346                    FCONT2: MOV     AXOUT,  -(SP)       ;SAVE TEXT POINTER ID AND
1191  004314    016746  175264                    MOV     PCF,    -(SP)       ;CURRENT LINE ADDR THEN
1192  004320    004767  177126                    JSR     PC,     PROCESS     ;GO EXECUTE THE REST OF THE LINE
1193  004324    012667  175254                    MOV     (SP)+,  PCF         ;RESTORE TEXT POINTERS
1194  004330    012603                            MOV     (SP)+,  AXOUT       ;...
1195  004332    016602  000010                    MOV     10(SP), PTR         ;GET VAR POINTER
1196  004336    005746                            TST     -(SP)
1197  004340    010616                            MOV     SP,     (SP)        ;CREATE INDEXED ADDRESS OF INCREMENT
1198  004342    062716  000006                    ADD     #6,     @SP         ;...
1199  004346    007001                            FPGET+IPTR                  ;LOAD FLAC WITH THE VARIABLE
1200  004350    007014                            FPADD+THROUGH+STACK         ;ADD THE INCREMENT AND
1201  004352    007061                            FPPUT+IPTR                  ;SAVE IT
1202  004354    005726                            TST     (SP)+               ;INDEX TO  THE LIMIT
1203  004356    007023                            FPSUB+FROM+STACK            ;COMPARE RESULT WITH LIMIT
1204  004360    005767  175234                    TST     HORD                ;AND DROP INDEXED POINTER.
1205  004364    003752                            BLE     FCONT2              ;REPEAT, IF LIMIT NOT EXCEEDED
1206  004366    062706  000012                    ADD     #12,    SP          ;UNLOAD THE STACK.
1207  004372    000207                            POPJ                        ;EXIT THE COMMAND.
1208
```

```
1210                                    ;'EVAL'
1211
1212                                    ;EVALUATE AN EXPRESSION
1213
1214  004374  110405                    ETERM:  MOVB    CHAR,   R5              ;COPY THIS OP.
1215  004376  120427  000211                    CMPB    CHAR,   #211            ;THISOP=RPAR?
1216  004402  103401                            BLO             ETERM2          ;NO
1217  004404  005005                            CLR             R5              ;YES, SET THISOP=ZERO
1218  004406  120516                    ETERM2: CMPB    R5,     @SP             ;COMPARE TWO OPERATORS
1219  004410  101023                            BHI             EPAR            ;LAST>THIS?-YES STACK AND CONTINUE
1220  004412  112600                            MOVB    (SP)+,  TEMP            ;SET LEFT HALF TO ONES OR STOP IF ZERO
1221  004414  010001                            MOV     TEMP,   AC              ;COPY THE OP CODE.
1222  004416  001407                            BEQ     EPURE                   ;END OF JOB?-YES, JUST GO LEAVE RESULTS
1223  004420  007003                            FPGET+FROM+STACK                ;NO, USE ITEM ON TOP OF STACK.
1224  004422  022626                            CLOSE+STACK                     ;REMOVE THE ITEM
1225  004424  106301                            ASLB            AC              ;MOVE OPERATOR CODE INTO POSITION
1226  004426  106301                            ASLB            AC              ;...
1227  004430  106301                            ASLB            AC              ;...
1228  004432  042701  177700                    BIC     #^C100, AC              ;MAKE POSITIVE.
1229  004436  062701  007001            EPURE:  ADD     #FPGET+IPTR,AC          ;LOAD BASE BINARY PLUS ADDRESSING MODE.
1230  004442  007200                            FCODE                           ;EXECUTE THE AC!
1231  004444  012702  001612                    MOV     #FLARG, PTR             ;LOAD POINTER
1232  004450  007061                            FPPUT+IPTR                      ;SAVE COPY OF RESULT
1233  004452  060500                            ADD     R5,     TEMP            ;CHECK THISOP-LASTOP
1234  004454  001354                            BNE             ETERM2          ;GO COMPARE PRIORITIES.
1235  004456  000207                            POPJ                            ;EXIT
1236                                                                            ;
1237  004460  120527  000206            EPAR:   CMPB    R5,     #206            ;CHECK FOR
1238  004464  002011                            BGE             EPAR2           ;LEFT PAREN,?-YES,
1239  004466  016246  000002                    MOV     2(PTR), -(SP)           ;OPEN STACK AND SAVE DATA.
1240  004472  012246                            MOV     (PTR)+, -(SP)           ;...
1241  004474  010546                            MOV     R5,     -(SP)           ;UPDATE LASTOP
1242  004476  004567  176422            ARGNXT: JSR     R5,     SPNXT           ;OUGHT TO BE AN ARGUMENT HERE
1243  004502  100007                            BPL     EVAL+2                  ;TEST FOR THE TYPE
1244  004504  104652  004572            ELPAR:  SKPLPR, OPERR                   ;ILLEGAL TO STAR AN EXPRESSION
1245                                                                            ;WITH RIGHT PARENS.
1246  004510  010446                    EPAR2:  MOV     CHAR,   -(SP)           ;SAVE THE LP CODE AND COMPUTE THE
1247  004512  012746  004656                    MOV     #EFUN3, -(SP)           ;LOAD TRANSFER FOR "POPJ" AT "EXIT"
1248  004516  104614                            GETC                            ;MOVE ONTO NEXT CHARACTER (ENTRY POINT #2)
1249  004520  005046                    EVAL:   CLR     -(SP)                   ;SET LASTOP=0 (ENTRY POINT #1)
1250  004522  104620                            TESTC                           ;TEST CHARACTER TYPE:
1251  004524  004544                                    ETERM1                  ;COULD BE A UNARY OPERATOR
1252  004526  004574                                    ENUM                    ;OR A NUMBER,
1253  004530  004614                                    EFUN                    ;OR A FUNCTION,
1254  004532  004767  000146                    JSR     PC,     GETVAR          ;OR A VARIABLE
1255  004536  104634                    OPNEXT: SPNOR                           ;IGNORE SPACES AROUND OPERATORS (O'D)
1256  004540  100412                            BMI     ETERMN                  ;IF NEGATIVE, THEN IT IS A LEGIT. TERM,
1257  004542  000413                            BR      OPERR                   ;OTHERWISE IT IS ILLEGAL FORMAT.
1258                                                                            ;
```

```
1260
1261  004544  012702  003530          ETERM1: MOV     #FLTZER,PTR             ;ASSUME PRESENT VALUE OF ZERO.
1262  004550  120427  000202                  CMPB    CHAR,   #202            ;MINUS?
1263  004554  001707                          BEQ             ETERM           ;YES, GO PROCESS IT
1264  004556  103747                          BLO             ARGNXT          ;PLUS?-YES, IGNORE IT
1265  004560  120427  000211                  CMPB    CHAR,   #211            ;SOME TYPE OF RIGHT TERMINATOR?
1266  004564  103747                          BLO             ELPAR           ;NO, GO TEST FOR LEFT PARENTHESIS SPECIAL CASE
1267  004566  104652  004374          ETERMN: SKPLPR,         ETERM           ;LEFT PAREN? - NO, CONTINUE TO PROCESS
1268  004572  104621                  OPERR:  ERROR+201+8,+8,                 ;MISSING OR DOUBLE OPERATOR ERROR
1269                                                                          ;
1270  004574  112767  000020  175027  ENUM:   MOVB    #NALPHA,SWITCH          ;USE INTERNAL DATA AND ACCEPT ALPHA DATA
1271  004602  007075                          FREAD                           ;READ IN A NUMBER.
1272  004604  012702  001612          ENUM2:  MOV     #FLARG, PTR             ;SET POINTER
1273  004610  007061                          FPPUT+IPTR                      ;SAVE RESULT
1274  004612  000751                          BR      OPNEXT                  ;GO GET OPERATOR
1275                                                                          ;
1276  004614  005005                  EFUN:   CLR     R5                      ;OLD "EFOP"
1277  004616  006305                          ASL             R5              ;HASH CODE
1278  004620  006105                          ROL             R5              ;...
1279  004622  060405                          ADD     CHAR,   R5              ;...
1280  004624  104614                          GETC                            ;READ THE MNEMONIC LETTERS
1281  004626  100373                          BPL             EFUN+2          ;REPEAT UNLESS TERMINATOR FOUND
1282  004630  010446                          MOV     CHAR,   -(SP)           ;SAVE LAST CHARACTER:PAREN.
1283  004632  010546                          MOV     R5,     -(SP)           ;SAVE HASH CODE
1284  004634  104660                          EVAL,X                          ;PROCESS FIRST ARGUMENT
1285  004636  012605                          MOV     (SP)+,  R5              ;GET BACK "EFOP"
1286  004640  012701  001166                  MOV     #FNTABL,AC              ;INIT THE SEARCH,
1287  004644  005741                          TST     -(AC)                   ;TEST FOR END,
1288  004646  001410                          BEQ     VERR                    ;NO SUCH FUNCTION
1289  004650  024105                          CMP     -(AC),  R5              ;TEST FOR MATCH
1290  004652  001374                          BNE     .-6                     ;REPEAT IF NOT FOUND,
1291  004654  004751                          JSR     PC,     @-(AC)          ;CALL THE FUNCTION,
1292  004656  104646                  EFUN3:  PARTST  stop                    ;CHECK CLOSING PARENS
1293  004660  007070                          FPNOR                           ;BE SURE THAT RESULTS ARE NORMALIZED,
1294  004662  000750                          BR      ENUM2                   ;GO SAVE RESULTS AND SET POINTER,
1295
1296  004664  010516                  EVALUX: MOV     R5,     @SP             ;SIMULATE "PUSHJ EVAL-2"
1297  004666  000713                          BR      EVAL-2
1298
1299
1300                                          .EOT
```

```
1302
1303                                    ;'GETVAR'
1304
1305                                    ;FIND OR CREATE A VARIABLE
1306
1307  004670  104605                    VERR:   ERROR+201+2,+2,             ;ILLEGAL VARIABLE OR FUNCTION NAME,
1308  004672  104607                    PERR:   ERROR+201+3,+3,             ;PAREN MISMATCH ERROR
1309  004674  104620                    GETARG: TESTC                       ;CHECK CAREFULLY
1310  004676  004252                            FINERR                      ;TERMINATOR IS NOT A VARIABLE,
1311  004700  004670                            VERR                        ;N
1312  004702  004670                            VERR                        ;F=? PTR=HOLE
1313  004704  012746  100000            GETVAR: MOV     #100000,-(SP)       ;CLEAR NAME SPACE ON STACK TO "SPACE",
1314  004710  110416                            MOVB    CHAR,   @SP         ;SAVE FIRST LETTER OF NAME IN STACK
1315  004712  005046                            CLR             -(SP)       ;CLEAR SUBSCRIPT SPACE ON STACK
1316  004714  104614                            GETC                        ;READ AND TRACE NEXT CHARACTER
1317  004716  100404                            BMI             GSERCH      ;TEST FOR TERMINATOR
1318  004720  110466  000003                    MOVB    CHAR,   3(SP)       ;SAVE SECOND LETTER OF NAME
1319  004724  104614                            GETC                        ;IGNORE UNTILL TERMINATOR FOUND,
1320  004726  100376                            BPL             .-2         ;---
1321  004730  104652  004776            GSERCH: SKPLPR, GS1                 ;CHECK FOR SUBSCRIPT
1322  004734  010446                            MOV     CHAR,   -(SP)       ;SAVE LEFT PARENS, CODE,
1323  004736  104660                            EVAL,X                      ;EVALUATE THE SUBSCRIPT AND
1324  004740  007071                            FPINT                       ;CORRECT IT TO AN INTEGER
1325  004742  010166  000002                    MOV     AC,     2(SP)       ;SAVE SUBSCRIPT ON STACK
1326  004746  120427  000214                    CMPB    CHAR,   #214        ;COMMA?
1327  004752  001004                            BNE     GS0                 ;SKIP IF ONLY ONE SUBSCRIPT,
1328  004754  104660                            EVAL,X                      ;GO READ THE SECOND SUBSCRIPT,
1329  004756  007071                            FPINT                       ;CONVERT IT TO 0-256,
1330  004760  110166  000003                    MOVB    AC,     3(SP)       ;COPY 2AND INTO LEFT HALF,
1331  004764  104646                    GS0:    PARTST                      ;CHECK FOR CLOSING PARENS MATCH,
1332  004766  000403                            BR      GS1
1333
1334                                    ;WHIP UP A VARIABLE &(C(TEMP))
1335  004770  012746  100046            WHIPV:  MOV     #100046,-(SP)
1336  004774  010046                            MOV     TEMP,   -(SP)
1337
1338                                    ;STACK NOW CONTAINS:
1339                                    ;*SUBS*   (SP)
1340                                    ;*NAME*   2(SP)
1341
1342  004776  016700  174644            GS1:    MOV     STARTV, TEMP        ;GET TEMPORARY POINTER (UPPER LIMIT)
1343  005002  016705  174644                    MOV     BOTTOM, R5          ;GET END OF "AREA",
1344  005006  010001                            MOV     TEMP,   AC          ;COPY THE STOP VALUE (LOWER LIMIT)
1345  005010  024545                            CMP     -(R5),  -(R5)       ;BE SURE LAST ENTRY DOES NOT PASS (BOTTOM),
1346  005012  160501                            SUB     R5,     AC          ;COMPUTE TABLE LENGTH NEGATED,
1347  005014  011602                            MOV     @SP,    PTR         ;COPY SUBSCRIPT
1348  005016  066602  000002                    ADD     2(SP),  PTR         ;COMBINE LETTERS PLUS SUBSCRIPT
1349  005022  000302                            SWAB    PTR                 ;MAKE LAST CODES FIRST!
1350                                                                        ;TAKE RESULT MODULO THE SIZE OF
1351  005024  040102                            BIC     AC,     PTR         ;THE SYMBOL TABLE AND
1352  005026  042702  000007                    BIC     #7,     PTR         ;MODULO 10(8)
1353
1354                                    ;CHANGE #7 TO #-1 FOR A CHRONOLOGICAL TABLE,
```

```
1356
1357
1358                                    ;TEMP=START
1359                                    ;AC=LENGTH
1360                                    ;R5=LOWER LIMIT
1361                                    ;PTR=HASH CODE
1362                                    ;SEARCH VARIABLES FOR MATCH OR AN UNUSED SPACE
1363
1364  005032  060002                            ADD     TEMP,   PTR             ;INITIALIZE POINTER "PTR"
1365  005034  010201                            MOV     PTR,    AC              ;SAVE THIS VALUE
1366
1367                                    ;'TEMP'  POINTS TO UPPER LIMIT
1368                                    ;'AC'    HOLDS HASH-CODE ADDRESS
1369                                    ;'PTR'   SCANS THE STORAGE
1370                                    ;'AXOUT' POINTS TO NEXT CHARACTER
1371                                    ;'CHAR'  HOLDS LAST TERMINATOR
1372                                    ;'R5'    HOLDS THE SCAN LOWER LIMIT
1373
1374  005036  004567  000072                   JSR     R5,     GTRY            ;SEARCH LOWER HALF
1375  005042  010002                            MOV     TEMP,   PTR             ;BEGIN AT THE TOP;
1376  005044  010105                            MOV     AC,     R5              ;END VALUE RESTARTED.
1377  005046  004567  000062                   JSR     R5,     GTRY            ;SEARCH UPPER HALF
1378
1379                                    ;SEARCH FOR A ZERO-VALUE VARIABLE AND SCRATCH IT
1380
1381  005052  016746  174574                   MOV     BOTTOM, -(SP)           ;END AT THE BOTTOM
1382  005056  010002                            MOV     TEMP,   PTR             ;BEGIN AT THE TOP
1383  005060  005762  000006          GSWIP:   TST     6(PTR)                  ;ZERO?
1384  005064  001441                            BEQ     GTAKE                   ;YES!
1385  005066  062702  000010                   ADD     #10,    PTR             ;BUMP.
1386  005072  020216                            CMP     PTR,    @SP             ;END?
1387  005074  103771                            BLO     GSWIP                   ;NOT YET!
1388  005076  104627                            ERROR+201+11,+11'               ;JUST NO ROOM AT ALL!
1389
1390
1391                                    ;VARIABLE STORAGE:
1392
1393                                    ;NAME:          (B,A)
1394                                    ;SUBSCRIPT:     (16) OR (8,8) : (2,1)
1395                                    ;LORD,EXP:      (FINAL VALUE OF PTR POINTS TO THIS LOCATION.)
1396                                    ;HORD:          (IF ZERO THEN ALL THESE PARTS ARE ASSUMED ZERO)
```

```
1398
1399                                    ;"ERASEV"
1400
1401  005100  016701  174546          ERVX:   MOV     BOTTOM, AC              ;CLEAR UP FROM THE BOTTOM
1402  005104  005041                          CLR     -(AC)                   ;CLEAR A WORD
1403  005106  020167  174532                  CMP     AC,     BUFR            ;TEST FOR END OF VARIABLE AREA
1404  005112  101403                          BLOS    .+10                    ;USE LARGER OF (AXIN) OR (BUFR)
1405  005114  020167  174522                  CMP     AC,     AXIN
1406  005120  101371                          BHI     .-14
1407  005122  062701  000120                  ADD     #80.,   AC
1408  005126  010167  174514                  MOV     AC,     STARTV
1409  005132  000205                          RTS     R5
1410
1411                                    ;GTRY
1412
1413                                    ;SCAN FOR A MATCH OR A VOID.
1414
1415                                    ;CALLED BY JSR  R5,GTRY
1416                                    ;STACK CONTAINS
1417                                    ;*LOWER LIMIT*  0(SP)
1418                                    ;*SUBS*  2(SP)
1419                                    ;*NAME*  4(SP)
1420                                    ;*OLD PC*6(SP)
1421                                    ;PTR CONTAINS START OF VARIABLE STORE
1422
1423  005134  026612  000004          GTRY:   CMP     4(SP),  @PTR            ;COMPARE TRUE NAME.
1424  005140  001407                          BEQ             GTEST           ;GO TEST SUBSCRIPT
1425  005142  005712                          TST             @PTR            ;LOOK FOR NULL
1426  005144  001411                  GTRY2:  BEQ             GTAKE           ;GO SWIPE IT!
1427  005146  062702  000010                  ADD     #10,    PTR             ;MOVE POINTER
1428  005152  020216                          CMP     PTR,    @SP             ;TEST LIMIT
1429  005154  103767                          BLO     GTRY                    ;REPEAT!
1430  005156  000205                          RTS     R5                      ;NOT IN THAT AREA, RETURN.
1431                                                                          ;
1432  005160  026662  000002  000002  GTEST:  CMP     2(SP),  2(PTR)          ;TEST SUBSCRIPT
1433  005166  000766                          BR              GTRY2           ;TRY AGAIN
1434                                                                          ;
1435  005170  016622  000004          GTAKE:  MOV     4(SP),  (PTR)+          ;SAVE NAME
1436  005174  016622  000002                  MOV     2(SP),  (PTR)+          ;SAVE SUBSCRIPT
1437                                                                          ;LEAVE PTR AS POINTER
1438  005200  062706  000006                  ADD     #6,     SP              ;FLUSH STACK DATA
1439  005204  000207                          POPJ                            ;RETURN FROM THE "GETVAR" ROUTINE.
1440
1441
1442
1443                                    ;"PARTST"
1444
1445                                    ;BE SURE PRESENT CHARACTER IS MATE TO PARENS IN THE STACK
1446
1447  005206  005726                  PARTSA: TST     (SP)+                   ;DUMP OLD 'R5'
1448  005210  062716  000003                  ADD     #3,     @SP             ;COMPUTE MATCHING PARENS
1449  005214  120426                          CMPB    CHAR,   (SP)+           ;COMPARE THE ACTUAL WITH COMPUTED.
1450  005216  001225                          BNE     PERR                    ;GO CALL "ERROR" IF THEY DON'T MATCH
1451  005220  104614                          GETC                            ;MOVE ON TO THE NEXT CHARACTER
```

```
  1452  005222  000115                    JMP     @R5                     ;RETURN TO THE SEQUENCE
```

```
1454
1455                                    ;*TYPE*ASK*
1456
1457                                    ;INPUT-OUTPUT STATEMENTS
1458
1459  005224  104656                    TYPE:   TASK                    ;CHECK FOR SPECIAL CODES
1460  005226  004767  177266                    JSR     PC,     EVAL    ;EVALUATE EXPRESSION
1461  005232  104475                            PRINT+  !=              ;MAKE LEADIN SIGNAL,
1462  005234  016702  174366                    MOV     FISW,   PTR     ;LOAD FORMAT DATA
1463  005240  007076                            FPRINT                  ;PRINT SAME
1464  005242  000770                            BR              TYPE    ;REPEAT
1465
1466          000030                            STRING=24,              ;NUMBER OF CHARACTERS ALLOWED ON INPUT,
1467  005244  104656                    ASK:    TASK                    ;CHECK FOR SPECIAL CODES
1468  005246  004767  177422                    JSR     PC,     GETARG  ;READ NAME AND SETUP PTR
1469  005252  104472                            PRINT+  !:              ;INDICATE READY FOR INPUT DATA,
1470  005254  010346                            MOV     AXOUT,  -(SP)   ;SAVE TEXT POINTER
1471  005256  110446                            MOVB    CHAR,   -(SP)   ;SAVE VARIABLE POINTER
1472  005260  012703  000030                    MOV     #STRING,AXOUT   ;MAKE COUNTER
1473  005264  160306                            SUB     AXOUT,  SP      ;OPEN AREA ON THE STACK
1474  005266  010605                            MOV     SP,     R5      ;SET RUBOUT STOP
1475  005270  010246                            MOV     PTR,    -(SP)   ;STACK THE DATA POINTER
1476  005272  112725  000200                    MOVB    #200,   (R5)+   ;PACK A LEADING ZERO,
1477  005276  010567  174340                    MOV     R5,     AXIN    ;USE FOR PACKING
1478  005302  005002                            CLR     PTR             ;SET SPACE - FLOP
1479  005304  104606                    ATAKE:  READC                   ;ACCEPT CHARACTER
1480  005306  104602  001442  005344            SORTC,TLIST+1,AFIX      ;TEST FOR PRIME TERMINATORS
1481  005314  005303                            DEC     AXOUT           ;COUNT CHARACTERS
1482  005316  003001                            BGT     .+4             ;SKIP IF OK,
1483  005320  104641                            ERROR+201+16,+16,       ;TOO LARGE AN INPUT STREAM,
1484  005322  104600  001445  001366            SORTJ,SPECIAL,INLIST    ;TEST FOR ALTMODE, SPACE, R,O,, L,F,
1485  005330  104616                    ARO:    PACKC                   ;PACK INPUT AND EDIT,
1486  005332  100764                            BMI     ATAKE           ;IF TERMINATOR, CONTINUE
1487  005334  011702                            MOV     @PC,    PTR     ;SET SWITCH SO THAT SPACE CODE
1488  005336  000762                            BR      ATAKE           ;WILL TERMINATE AND ALPHA IS ACCEPTED,
1489                                            ;
1490  005340  005702                    ASPACE: TST     PTR             ;CHECK STATUS OF SPACE,
1491  005342  001760                            BEQ     ATAKE           ;IGNORE,
1492  005344  112704  000214            AFIX:   MOVB    #214,   CHAR    ;PACK AN EXTRA COMMA
1493  005350  104616                            PACKC                   ;,,,
1494  005352  111767  174232                    MOVB    @PC,    DEBG    ;DISABLE TRACE
1495  005356  010503                            MOV     R5,     AXOUT   ;PICKUP DATA
1496  005360  105743                            TSTB    -(AXOUT)        ;BACKUP TO LEAD ZERO,
1497  005362  104660                            EVAL,X                  ;GO READ THE NUMBER OR EXPRESSION!
1498  005364  007064                            FPPUT+THROUGH+STACK     ;SAVE THE RESULT
1499  005366  105067  174216                    CLRB    DEBG            ;RE-ENABLE TRACE
1500  005372  062706  000032            AGO:    ADD     #STRING+2,SP    ;CORRECT THE STACK
1501  005376  112604                            MOVB    (SP)+,  CHAR    ;RESTORE TEXT SEQUENCE
1502  005400  012603                            MOV     (SP)+,  AXOUT   ;,,,
1503  005402  000720                            BR              ASK     ;CONTINUE *ASK* COMMAND,
```

```
1505
1506
1507
1508                                    ;"TASK"
1509
1510                                    ;AUXILLIARY PROCESSOR FOR INPUT-OUTPUT COMMANDS
1511
1512  005404  104642  005015            TCRLF:  PRINT2, CRLF                    ;PRINT THIS CODE FOR CR+LF,
1513  005410  104614                    TASK4:  GETC                            ;MOVE TO NEXT CHARACTER
1514  005412  104600  001433  001344    TASKX:  SORTJ,ALIST,ATLIST              ;TEST FOR SPECIAL CODES (ENTRY POINT)
1515  005420  122704  000210                    CMPB    #210,   CHAR            ;R=PARS AND = NOT VALID HERE ; IGNORE
1516  005424  101771                            BLOS    TASK4                   ;COMMA, SEMI, AND CR WERE TESTED ABOVE
1517  005426  000205                            RTS     R5                      ;RETURN
1518                                                                            ;
1519  005430  104614                    TINTR:  GETC                            ;PASS PRECENT SIGN
1520  005432  104622                            GETLN                           ;READ FORMAT CONTROL NUMBER
1521  005434  010167  174166                    MOV     AC,     FISW            ;SAVE CODE
1522  005440  000764                            BR              TASKX           ;CONTINUE
1523                                                                            ;
1524  005442  112304                    TQUOT:  MOVB    (AXOUT)+,CHAR           ;BYPASS TRACE
1525  005444  122704  000216                    CMPB    #CR,    CHAR            ;READ AND PRINT WITHIN QUOTES
1526  005450  001760                            BEQ     TASKX                   ;C,R,=NORMAL RETURN FROM 'TASK'
1527  005452  122704  000042                    CMPB    #'",    CHAR            ;QUOTE=CANONICAL RETURN
1528  005456  001754                            BEQ     TASK4                   ;GO SEE WHETHER THERE IS MORE
1529  005460  104604                            PRINTC                          ;PRINT MATERIAL,
1530  005462  000767                            BR      TQUOT                   ;REPEAT
1531                                                                            ;
1532  005464  104642  000015            TCRLF2: PRINT2, 00015                   ;#=CR ONLY
1533  005470  000747                            BR              TASK4           ;BUT EXTRA RUB OUT,
1534
```

```
1536
1537
1538
1539
1540                                ;FLOATING POINT HANDLER FOR FOCAL-11
1541
1542                                ;EXTERNAL DATA IS (LORD, EXP)(HORD)
1543                                ;INTERNAL DATA IS (EXP)(HORD)(LORD)
1544                                ;ARITHMETIC OPERATIONS ARE NUMBERED 0-7;
1545                                ;THE ADDRESSING MODES ARE NUMBERED 0-7;
1546
1547          007000                        EMT=007000
1548
1549          007000                        FPGET=EMT+00            ;OPERATIONS
1550          007010                        FPADD=EMT+10
1551          007020                        FPSUB=EMT+20
1552          007030                        FPDIV=EMT+30            ;DIVIDE BY ZERO ERROR
1553          007040                        FPMUL=EMT+40
1554          007050                        FPPOW=EMT+50            ;NEGATIVE POWER ERROR
1555          007060                        FPPUT=EMT+60            ;EXPONENT ERROR
1556
1557          007070                        FPNOR=EMT+70            ;FUNCTIONS
1558          007071                        FPINT=EMT+71
1559          007072                        FPSGN=EMT+72
1560          007073                        FPABS=EMT+73
1561          007074                        FPNEG=EMT+74
1562          007075                        FREAD=EMT+75
1563          007076                        FPRINT=EMT+76
1564          007077                        FZER=EMT+77             ;100 TO 177 UNUSED
1565
1566          007200                        FCODE=EMT+200           ;COMPUTED OPERATION IN AC
1567
1568
1569                                ;201 TO 377 UNUSED
1570
1571          000001                        IPTR=1          ;&PTR
1572          000002                        XPTR=2          ;AUTO INDEX PTR BY TWO WORDS,
1573          000003                        INTO=3          ;STACK
1574          000003                        FROM=3          ;STACK
1575          000004                        THROUGH=4       ;STACK
1576          000005                        IMMED=5         ;DATA FOLLOWS
1577          000006                        REL=6           ;RELATIVE (ADDR=,)
1578          000000                        STACK=0         ;ADDRESSING MODES
1579          000000                        DIRECT=0        ;ABSOLUTE (FPADD + DIRECT,ADDR) (NON=P,I,C,)
1580                                ;"DIRECT" AND "INDEX" ARE FOLLOWED BY COMMA AND ADDRESS,
1581
```

```
1583
1584                                    ;FOCAL-11 FLOATING POINT  HANDLER
1585
1586  005472  010546                    FEMT:   MOV     %5,     -(SP)           ;SAVE ALL REGS.
1587  005474  010446                            MOV     %4,     -(SP)           ;
1588  005476  010346                            MOV     %3,     -(SP)           ;
1589  005500  010246                            MOV     %2,     -(SP)           ;
1590  005502  010146                            MOV     %1,     -(SP)           ;
1591  005504  010046                            MOV     %0,     -(SP)           ;
1592  005506  016600  000014                    MOV     14(SP),TEMP             ;PICKUP ADDR OF LOC, AFTER CALL.
1593  005512  116005  177776                    MOVB    -2(TEMP),AL             ;PICKUP CODED BYTE.
1594  005516  100001                            BPL     FPURE                   ;TEST FOR CALCULATED CODES.
1595  005520  010105                            MOV     AC,     AL              ;YES, USE THOSE INSTEAD.
1596  005522  110546                    FPURE:  MOVB    AL,     -(SP)           ;SAVE A COPY OF IT.
1597  005524  122716  000070                    CMPB    #070,   @SP             ;TEST FOR FUNCTION CALL
1598  005530  003412                            BLE     FLTDO                   ;SKIP ADDRESS FORMATION
1599  005532  042705  177770                    BIC     #-10,   AL              ;LEAVE THE ADDRESS MODE BITS.
1600  005536  116505  005662                    MOVB    INADDR(AL),AL           ;MAKE INDEX FOR ADDRESS FORMATION ROUTINES.
1601  005542  060507                            ADD     AL,     PC              ;SETUP "PTR" AS DATA POINTER.
1602  005544  112201                    FLTDO1: MOVB    (PTR)+,AC               ;GET THE DATA (AE=AC)
1603  005546  112205                            MOVB    (PTR)+,AL               ;LOW ORDER 8 BITS
1604  005550  000305                            SWAB            AL              ;LEFT SHIFT 8
1605  005552  105005                            CLRB            AL              ;...
1606  005554  011204                            MOV     @PTR,   AH              ;HIGH ORDER BYTE.
1607  005556  111600                    FLTDO:  MOVB    @SP,    TEMP            ;LET'S TRY THOSE CODES
1608  005560  010246                            MOV     PTR,    -(SP)           ;SAVE POINTER FOR "PUT"
1609  005562  016702  174032                    MOV     HORD,   BH              ;SETUP THE FLOATING AC
1610  005566  016703  174030                    MOV     LORD,   BL              ;...
1611  005572  105003                            CLRB            BL              ;...
1612  005574  106200                            ASRB            TEMP            ;AGAIN,
1613  005576  106200                            ASRB            TEMP            ;MAKE AN EVEN TABLE ADDRESS
1614  005600  042700  177761                    BIC     #177761,TEMP            ;CLEAR LEFT HALF
1615  005604  004770  006126                    JSR     PC,     @OPADDR(TEMP)   ;DO THE OPERATION!
1616  005610  004767  000222                    JSR     PC,     NORF            ;NORMALIZE IT.
1617  005614  010367  174002                    MOV     BL,     LORD            ;SAVE RESULTS
1618  005620  010267  173774                    MOV     BH,     HORD            ;...
1619  005624  022626                    FLTX:   CMP     (SP)+,  (SP)+           ;DUMP THE POINTER AND THE CODES
1620  005626  000474                            BR      PWREGS                  ;RESTORE REGISTERS AND RETURN
1621
1622
```

```
1624
1625                                    ;"FPPUT"
1626
1627  005630  005726                    PUTF:   TST     (SP)+                   ;IGNORE RETURN FROM THIS ONE.
1628  005632  011600                            MOV     @SP,    TEMP            ;GET THE POINTER BACK
1629  005634  010210                            MOV     BH,     @TEMP           ;SAVE HIGH ORDER PART
1630  005636  016704  173754                   MOV     BE,     AH              ;COPY "(BE)" FOR SAVE AND TESTS.
1631  005642  000303                            SWAB    BL                      ;POSITION LOW ORDER BYTES.
1632  005644  110340                            MOVB    BL,     -(TEMP)         ;...
1633  005646  110440                            MOVB    AH,     -(TEMP)         ;SAVE HALF OF EXPONENT.
1634  005650  100001                            BPL     .+4                     ;TEST FOR ALL SIGN BITS.
1635  005652  005104                            COM     AH                      ;...
1636  005654  000304                            SWAB    AH                      ;CHECK HIGH ORDER EXP.
1637  005656  001762                            BEQ     FLTX                    ;OK=RETURN
1638  005660  104631                            ERROR+201+12.,+12.              ;NO=EXP OVER 38!
1639
1640
1641                                    ;ADDRESS MODES
1642
1643  005662     126                    INADDR: .BYTE   EMDIR -FLTDO1           ;ADDRESS FOLLOWS.
1644  005663     000                            .BYTE   FLTDO1-FLTDO1           ;"PTR" CONTAINS ADDRESS.
1645  005664     162                            .BYTE   EMIND -FLTDO1           ;((PTR)+4) IS  THE ADDRESS .
1646  005665     152                            .BYTE   EMTO  -FLTDO1           ;STACK HAS DATA
1647  005666     172                            .BYTE   EMTHR -FLTDO1           ;STACK HAS ADDRESS
1648  005667     140                            .BYTE   EMIME -FLTDO1           ;DATA FOLLOWS
1649  005670     200                            .BYTE   EMREL -FLTDO1           ;((TEMP)) IS ADDRESS - (TEMP)
1650  005671     234                            .BYTE   NOP   -FLTDO1           ;
                                                                               ;BYTES ABOVE MUST BE LESS THAN 177.
1651                                                                           ;
1652
1653  005672  011002                    EMDIR:  MOV     @TEMP,  PTR             ;TO THE PC1 AND PICKUP "X"
1654  005674  062766  000002  000016            ADD     #2,     16(SP)          ;MOVE PC1 POINTER PAST DATA.
1655  005702  000720                            BR      FLTDO1                  ;RETURN TO HANDLER.
1656
1657  005704  062766  000004  000016    EMIME:  ADD     #4,     16(SP)          ;THEN
1658  005712  010002                            MOV     TEMP,   PTR             ;COPY ADDRESS OF DATA,
1659  005714  000713                            BR      FLTDO1                  ;INDEX THROUGH THE STACK.
1660
1661  005716  010602                    EMTO:   MOV     SP,     PTR             ;INDEXES THROUGH THE STACK.
1662  005720  062702  000022                   ADD     #22,    PTR             ;MOVE POINTER TO JOB.
1663  005724  000707                            BR      FLTDO1                  ;GO USE IT.
1664
1665  005726  022222                    EMIND:  CMP     (PTR)+, (PTR)+          ;AUTO INDEX THE POINTER
1666  005730  010266  000006                   MOV     PTR,    6(SP)           ;COPY AND UPDATE PTR.
1667  005734  000703                            BR      FLTDO1                  ;CONTINUE.
1668
1669  005736  016602  000022            EMTHR:  MOV     22(SP), PTR             ;USE JOB AS THE ADDRESS.
1670  005742  000700                            BR      FLTDO1                  ;CONTINUE
1671
1672  005744  061000                    EMREL:  ADD     @TEMP,  TEMP            ;COMPUTE ABSOLUTE ADDRESS.
1673  005746  010002                            MOV     TEMP,   PTR             ;USE THAT RESULT.
1674  005750  000751                            BR      EMDIR+2                 ;GO MOVE PC.
```

```
1676
1677
1678                                    ;POWER-FAIL
1679
1680  005752  005767  [2662            PWRDWN: TST     WHOOPS                  ;CHECK FOR POWER-FAIL OR AUTO-RESTART,
1681  005756  001011                            BNE     PWRUP                   ;IF NON-ZERO, THEN IT IS POWER-UP,
1682  005760  01 [546                           MOV     %5,     -(SP)           ;
1683  005762  010446                            MOV     %4,     -(SP)           ;SAVE 0-5,
1684  005764  010346                            MOV     %3,     -(SP)           ;
1685  005766  010246                            MOV     %2,     -(SP)           ;
1686  005770  010146                            MOV     %1,     -(SP)           ;
1687  005772  010046                            MOV     %0,     -(SP)           ;
1688  005774  010667  173640                    MOV     SP,     WHOOPS          ;FINALLY SAVE THE STACK POINTER HIMSELF,**
1689  006000  000000                    NOP:    HALT                            ;STOP THE PROCESS; FALL-THROUGH FOR FAILSAFE,
1690
1691
1692                                    ;AUTO-RESTART
1693
1694  006002  016706  173632            PWRUP:  MOV     WHOOPS, SP              ;RELOAD THE STACK POINTER
1695  006006  005067  173626                    CLR     WHOOPS                  ;RESET THE SWITCH
1696  006012  012737  000101  177560            MOV     #101,   @#TKS           ;RESTORE INTERRUPT ENABLE AND SET READER RUN
1697  006020  012600                    PWREGS: MOV     (SP)+,  %0              ;
1698  006022  012601                            MOV     (SP)+,  %1              ;
1699  006024  012602                            MOV     (SP)+,  %2              ;
1700  006026  012603                            MOV     (SP)+,  %3              ;
1701  006030  012604                            MOV     (SP)+,  %4              ;
1702  006032  012605                            MOV     (SP)+,  %5              ;RESUME PROCESS AND RESTORE THE STATUS,
1703  006034  000002                    PWRON:  RTI                             ;...,
1704
1705                                    ;FOR DEBUGGING WITH SINGLE STEP, (NOT REENTRANT)
1706                                    ;       TST     (SP)+                   ;MOVE POINTER
1707                                    ;       MOV     (SP)+,  STATUS          ;RESTORE STATUS
1708                                    ;       MOV     -6(SP), PC              ;LET THIS INST BE EXECUTED, THEN TRAP,
```

```
1710
1711                                    ;NORMALIZE RESULT AFTER DOING AN OPERATION.
1712
1713  006036  005702                    NORF:   TST             BH              ;TEST FOR ZERO
1714  006040  001002                            BNE             NORM2           ;OBVIOUSLY THERE SO SHIFT
1715  006042  005703                            TST             BL              ;CHECK LOW PART.
1716  006044  001452                            BEQ             NORMZ           ;RESULT IS ZERO!
1717  006046  005367  173544            NORM2:  DEC             BE              ;ROTATE LEFT FOR THE TEST
1718  006052  006303                            ASL             BL              ;TAKE CARE OF 2-DONE CASES:
1719  006054  006102                            ROL             BH              ;0,1XXX ;V=1
1720  006056  102405                            BVS             NORMD           ;1,0XXX ;V=1
1721  006060  100372                            BPL             NORM2           ;0,0 TEST
1722  006062  032702  077777                    BIT     #77777, BH              ;1,100,XXX=?
1723  006066  001367                            BNE             NORM2           ;NO!
1724  006070  000261                            SEC                             ;YES=NEG, POWER OF 2
1725  006072  006002            NORMD:  ROR             BH              ;RESTORE TO PROPER PLACES,
1726  006074  006003                            ROR             BL              ;...,
1727  006076  005267  173514                    INC             BE              ;---
1728  006102  062703  000200                    ADD     #200,   BL              ;ROUND
1729  006106  005502                            ADC             BH
1730  006110  102770                            BVS     NORMD
1731  006112  000207                            RTS             PC              ;RETURN
1732
1733  006114  010167  173476            GETF:   MOV     AC,     BE              ;COPY THE OPERAND.
1734  006120  010402                            MOV     AH,     BH
1735  006122  010503                            MOV     AL,     BL
1736  006124  000207                            RTS             PC
```

```
1738
1739  006126  006114                OPADDR: GETF                    ;FLOATING POINT OPERATOR HANDLERS.
1740  006130  006230                        ADDFF
1741  006132  006222                        SUBFF
1742  006134  006566                        DIVFF
1743  006136  006426                        MULFF
1744  006140  006650                        POWF
1745  006142  005630                        PUTF
1746  006144  006204                        NORX
1747
1748  006146  006366                FNADDR: INTY                    ;FLOATING POINT FUNCTION HANDLERS
1749  006150  006330                        INTF
1750  006152  006532                        SGNF
1751  006154  006556                        ABSFF
1752  006156  006562                        NEGFF
1753  006160  007542                        READF
1754  006162  007012                        PRNTF
1755  006164  006356                        INTZ
1756
1757  006166  062706  000006        MULZ:   ADD     #6,     SP              ;DUMP 'SIGN+COUNT+OLD RETURN',
1758  006172  005067  173420        NORMZ:  CLR     BE
1759  006176  005002                        CLR     BH
1760  006200  005003                        CLR     BL
1761  006202  000207                        RTS     PC
1762
1763  006204  116601  000004        NORX:   MOVB    4(SP),  AC              ;LOOK WHO'S HERE!
1764  006210  042701  177770                BIC     #-10,   AC              ;USE FUNCTION CODES TO
1765  006214  006301                        ASL     AC                      ;FORM WORD ADDRESS IN TABLE AND
1766  006216  000171  006146                JMP     @FNADDR(AC)             ;JUMP TO THE FUNCTION PROCESSES.
```

```
1768
1769                                    ;FUNDAMENTAL FLOATING POINT ROUTINES
1770
1771  006222  005404                    SUBFF:  NEG     AH                      ;SUBTRACT LOW ORDER,
1772  006224  005405                            NEG     AL                      ;SUBTRACT LOW CARRY,
1773  006226  005604                            SBC     AH                      ;FINISH W/O OVERFLOW CHECK
1774
1775                                    ;ALIGN AND ADD ROUTINE
1776
1777                                    ;IF A=0 , USE B
1778                                    ;IF B=0 , USE A
1779                                    ;SET AC=X(A)-X(B)
1780                                    ;IF |X(A)-X(B)|>30,, USE LARGER OF X(A) OR X(B)
1781
1782                                    ;SET A=A/2
1783                                    ;SET B=B/2              <-,
1784                                    ;IF AC=0, GO ADD          ,
1785                                    ;IF AC>0, SET AC=AC-1;  -->,
1786                                    ;IF AC<0, SET AC=AC+1;  A=A/2;
1787
1788
1789  006230  005704                    ADDFF:  TST     AH                      ;TEST FOR A=0
1790  006232  001455                            BEQ     INTY                    ;USE B
1791  006234  005702                            TST     BH                      ;TEST FOR B=0
1792  006236  001726                            BEQ     GETF                    ;USE A
1793  006240  012700  001616                    MOV     #BE,    TEMP            ;CREATE EXP, POINTER,
1794  006244  161001                            SUB     @TEMP,  AC              ;COMPARE EXPONENTS
1795  006246  003410                            BLE             ALIGNA          ;AND SAVE COUNT,
1796  006250  022701  000036                    CMP     #30.,   AC              ;TEST FOR POSSIBILITY THAT
1797  006254  003423                            BLE             ALTAKA          ;A IS TOO LARGE =USE A,
1798  006256  006202                    ALIGNB: ASR             BH              ;SHIFT DOUBLE B TO
1799  006260  006003                            ROR             BL              ;THE RIGHT AND ADD TO
1800  006262  005210                            INC             @TEMP           ;THE EXPONENT
1801  006264  005301                            DEC             AC              ;COUNT THE SHIFTS
1802  006266  003373                            BGT             ALIGNB          ;TEST FOR REPEAT (AC>0)
1803  006270  022701  177743            ALIGNA: CMP     #-29.,  AC              ;TEST RELATIVE SIZE
1804  006274  002104                            BGE             ALGZA           ;B IS TOO LARGE = EFFECTIVE ZERO A,
1805  006276  006204                            ASR             AH              ;SHIFT OVER
1806  006300  006005                            ROR             AL              ;...
1807  006302  005201                            INC             AC              ;COUNT SHIFTS
1808  006304  003774                            BLE             .-6             ;REPEAT UNTIL EXTRA 1 DONE,
1809  006306  006202                            ASR             BH              ;GIVE AN EXTRA SHIFT
1810  006310  006003                            ROR             BL              ;TO THE RESULT TO
1811  006312  005210                            INC             @TEMP           ;AVOID OVERFLOW, BUMP THE EXP,
1812  006314  060503                            ADD     AL,     BL              ;LEAVE RESULT IN B,
1813  006316  005502                            ADC             BH              ;ADD THE LOW ORDER CARRY
1814  006320  060402                            ADD     AH,     BH              ;FINISH W/O OVERFLOW CHECK
1815  006322  000471                            BR      ALGZA                   ;RETURN FOR NORM, ETC, OF SMALL DIFFERENCES,
1816                                                                            ;
1817  006324  060110                    ALTAKA: ADD     AC,     @TEMP           ;CORRECT THE EXPONENT
1818  006326  000674                            BR      GETF+4                  ;GO COPY THE DATA
```

```
1820
1821
1822                                    ;FPINT (BE,BH,BL)=> (31.,BH)
1823
1824  006330  012701  000016          INTF:  MOV     #BE,    AC              ;CREATE EXP. POINTER
1825  006334  005711                         TST             @AC             ;MAKES B INTO A+
1826  006336  003407                         BLE             INTZ            ;INTEGER BUT LEAVES
1827  006340  022711  000037          INTG:  CMP     #31.,   @AC             ;THE EXPONENT CORRECT,
1828  006344  003406                         BLE             INTX            ;NO CHANGE POSSIBLE NOW,
1829  006346  006202                         ASR             BH              ;BRING IN SIGN BIT LEFT,
1830  006350  006003                         ROR             BL              ;ROTATE DP UNTIL
1831  006352  005211                         INC             @AC             ;THE EXPONENT CONTAINS
1832  006354  000771                         BR              INTG            ;31 DECIMAL.
1833                                                                          ;
1834  006356  004767  177610          INTZ:  JSR     PC,     NORMZ           ;LEAVE ZERO RESULT
1835  006362  010366  000010          INTX:  MOV     BL,     8.(SP)          ;COPY INTO THE COPY OF "AC"
1836  006366  000207                  INTY:  RTS             PC              ;RETURN ONE MORE LEVEL
1837                                  ;
1838  006370  062706  000012          ZERODM: ADD    #12,    SP              ;DUMP 'BH','R5','R5','PTR', AND 'CODES',
1839  006374  000611                         BR      PWREGS                  ;GO RESTORE REGISTERS.
1840
1841  006376  007073                  XABS:  FPABS                   ;SIMPLE CALLS FOR SIMPLE FUNCTIONS
1842  006400  000207                         RTS     PC
1843
1844  006402  007072                  XSGN:  FPSGN
1845  006404  000207                         RTS     PC
1846
1847  006406  016746  173206          XITR:  MOV     HORD,   -(SP)           ;SAVE SIGN
1848  006412  007073                         FPABS                           ;ABS.
1849  006414  007071                         FPINT
1850  006416  005726                         TST     (SP)+
1851  006420  100001                         BPL     .+4
1852  006422  007074                         FPNEG
1853  006424  000207                         RTS     PC
```

```
1855                                    ;THE FOLLOWING THREE PAGES CONTAIN
1856                                    ;"FPMUL","FPDIV","FPSGN","FPNEG","FPABS", AND "FPPOW",
1857
1858                                    ;(2 WORD)*(2 WORD) MULTIPLY ROUTINE
1859                                    ;(TEMP,AC,BH,BL)*(AH,AL)=>(0,0,BH,BL)!AH,AL!)
1860
1861
1862  006426  004767  000302          MULFF:  JSR     PC,     SIGN            ;COMPUTE SIGN OF RESULT AND SAVE,
1863  006432  001655                          BEQ             MULZ            ;RESULT IS ZERO IF  A IS ZERO,
1864  006434  006202                  MDP0:   ASR             BH              ;...
1865  006436  006003                          ROR             BL              ;...
1866  006440  103003                          BCC             .+10            ;NO CARRY=NO ADD
1867  006442  060501                          ADD     AL,     AC              ;UPDATE THE RESULT
1868  006444  005500                          ADC             TEMP            ;...
1869  006446  060400                          ADD     AH,     TEMP            ;...
1870  006450  006000                          ROR     TEMP                    ;PUT IT DOWN ONCE,
1871  006452  006001                          ROR     AC                      ;...
1872  006454  005316                          DEC             @SP             ;COUNT BITS,
1873  006456  001366                          BNE             MDP0            ;REPEAT,
1874  006460  010002                  SIGND:  MOV     TEMP,   BH              ;COPY THE RESULT,
1875  006462  010103                          MOV     AC,     BL              ;...
1876  006464  162626                          SUB     (SP)+,  (SP)+           ;RESTORE THE SIGN OF
1877  006466  100003                          BPL     BDIVX                   ;THE RESULT,
1878  006470  005402                  NEGX:   NEG             BH              ;LEAVE A NEGATIVE RESULT
1879  006472  005403                          NEG             BL              ;...
1880  006474  005602                          SBC             BH              ;...
1881  006476  062703  000001          BDIVX:  ADD     #1,     BL              ;ROUND
1882  006502  005502                          ADC             BH              ;...
1883  006504  005726                          TST     (SP)+                   ;DUMP OLD RETURN ADDRESS (OR DATA SAVED)
1884  006506  010305                  ALGZA:  MOV     BL,     AL              ;COPY RESULT
1885  006510  010204                          MOV     BH,     AH              ;TO TEST SIZE,
1886  006512  100003                          BPL     ALGZB                   ;TAKE ABSOLUTE VALUE,
1887  006514  005404                          NEG             AH              ;...
1888  006516  005405                          NEG             AL              ;...
1889  006520  005604                          SBC             AH              ;...
1890  006522  001002                  ALGZB:  BNE     .+6                     ;CHECK FOR HIGH ORDER WORD NOT ZERO,
1891  006524  000305                          SWAB    AL                      ;CHECK FOR LOW BYTE NOT ZERO,
1892  006526  001621                          BEQ     NORMZ                   ;MAKE ALL ZERO RESULT,
1893  006530  000207                          RTS             PC              ;AND GET BACK TO THIS LEVEL,
1894                                                                          ;
1895  006532  010246                  SGNF:   MOV     BH,     -(SP)           ;SAVE HIGH ORDER PART,
1896  006534  001763                          BEQ     ALGZA-2                 ;SIGN OF ZERO IS ZERO,
1897  006536  012767  000001  173052  SGN1:   MOV     #1,     BE              ;CREATE A FLOATING ONE,
1898  006544  012702  040000                  MOV     #40000, BH              ;...
1899  006550  005003                          CLR             BL              ;...
1900  006552  005716                          TST             @SP             ;TEST THE SIGN OF STACK ENTRY,
1901  006554  000744                          BR      NEGX-2                  ;GO TEST FOR RESULT,
1902                                                                          ;
1903  006556  005702                  ABSFF:  TST     BH                      ;TEST B AND PUT DUMMY ON STACK,
1904  006560  100363                          BPL     SGNF-2                  ;GO USE OTHER BRANCH,
1905  006562  004767  177702          NEGFF:  JSR     PC,     NEGX            ;DOES NOT RETURN HERE;DUMMY ON STACK,
1906                                                                          ;
```

```
1908
1909                                    ;(4 WORD)/(2 WORD) DIVIDE ROUTINE
1910                                    ;(0,0,BH,BL)/(AH,AL)=> BH,BL;|AH,AL|
1911
1912  006566  005401                    DIVFF:  NEG     AC                      ;TAKE DIFFERENCE OF EXPONENTS
1913  006570  005201                            INC     AC                      ;PLUS ONE.
1914  006572  004767  000136                    JSR     PC,     SIGN            ;MAKE SIGN OF RESULT ABS. VALS.
1915                                                                            ;ADD TEMP+AC. ADD WITH COUNTER
1916  006576  001421                            BEQ     DIVZER                  ;DIVIDE BY ZERO!
1917  006600  160503                    DIFL:   SUB     AL,     BL              ;USE A TEMPORARY RESULT
1918  006602  005602                            SBC     BH                      ;MAKE A TRIAL RUN
1919  006604  020204                            CMP     BH,     AH              ;CONTINUE THE ILLUSION
1920  006606  100403                            BMI     .+10                    ;COMPUTE INTERMEDIATE VALUE.
1921  006610  160402                            SUB     AH,     BH              ;BRANCH IF NO=ACTUAL CARRY.
1922  006612  000261                            SEC                             ;NO=GO=UPDATE HIGH
1923  006614  000403                            BR      .+10                    ;NO=GO=UPDATE LOW
1924  006616  060503                            ADD     AL,     BL              ;SET UP THE CARRY DATE FOR ROTATE
1925  006620  005502                            ADC     BH                      ;SKIP
1926  006622  000241                            CLC                             ;DUMP TRIAL RESULT+CLEAR CARRY
1927  006624  006101                            ROL             AC              ;ROTATE ALL
1928  006626  006100                            ROL             TEMP            ;ROTATE ALL
1929  006630  006303                            ASL             BL              ;ROTATE ALL
1930  006632  006102                            ROL             BH              ;ROTATE ALL
1931  006634  005316                            DEC             @SP             ;COUNT
1932  006636  001360                            BNE     DIFL                    ;REPEAT THE LOOP
1933  006640  000707                            BR              SIGND           ;GO DOCTOR THE SIGN
1934                                                                            ;
1935  006642  104635                    DIVZER: ERROR+20!+14,+14,               ;DIVISION BY ZERO NOT ALLOWED.
1936
1937
1938                                            .EOT
1939
1940                                    ;INTEGER EXPONENT CALCULATIONS
1941
1942  006644  006204                    POWF1:  ASR     AH                      ;REDUCE TO INTEGER
1943  006646  005201                            INC     AC                      ;COUNT SHIFTS
1944  006650  022701  000017            POWF:   CMP     #15.,   AC              ;TEST FOR RANGE OF POWER
1945  006654  103426                            BLO     FERRO                   ;OUT OF THE RANGE, FRACTIONAL
1946  006656  001372                            BNE     POWF1                   ;GO TAKE INTEGER PART OF A.
1947  006660  010446                            MOV     AH,     -(SP)           ;SAVE THE COUNT (COULD BE ZERO)
1948  006662  100423                            BMI     FERRO                   ;TEST FOR NEGATIVE POWER.
1949  006664  001724                            BEQ     SGN1                    ;INITIALIZE MULTIPLICAND TO ONE
1950  006666  010246                            MOV     BH,     -(SP)           ;COPY THE BASE
1951  006670  001637                    POWS:   BEQ     ZERODM                  ;DON'T LET FPMUL SEE ZERO!
1952  006672  016746  172720                    MOV     BE,     -(SP)           ;SAVE EXPONENT OF THE BASE
1953  006676  010346                            MOV     BL,     -(SP)           ;INTO ARGUMENT POSITION
1954  006700  012605                    POWDO:  MOV     (SP)+,  AL              ;RESTORE EXP. OF THE BASE
1955  006702  012601                            MOV     (SP)+,  AC              ;...
1956  006704  012604                            MOV     (SP)+,  AH              ;...
1957  006706  005316                            DEC             @SP             ;COUNT ON THE STACK.
1958  006710  001672                            BEQ             BDIVX           ;FINISHED!
1959  006712  010446                            MOV     AH,     -(SP)           ;SAVE EXP
1960  006714  010146                            MOV     AC,     -(SP)           ;...
1961  006716  010546                            MOV     AL,     -(SP)           ;...
```

```
  1962  006720  004767  177122                  JSR     PC,     NORM2   ;NORMALIZE FOR ACCURACY.
  1963  006724  004767  177476                  JSR     PC,     MULFF   ;MULTIPLY ONCE
  1964  006730  000763                          BR              POWD0   ;REPEAT
  1965                                                                    ;
  1966  006732  104637                  FERR01: ERROR+20!+15.,+15.      ;TOO LARGE OR NEGATIVE POWER!
```

```
1968
1969
1970
1971                                    ;COMPUTE SIGN OF RESULT FOR MUL/DIV,
1972
1973                                    ;ALSO SAVE THAT SIGN ON THE STACK,
1974
1975  006734  010246                    SIGN:   MOV     BH,     -(SP)   ;FIND SIGN OF RESULT
1976  006736  001754                            BEQ     POWS            ;RESULT IS ZERO IF B IS ZERO,
1977  006740  042716  077777                    BIC     #077777,@SP     ;AS THE XOR OF ARGUMENT'S
1978  006744  060416                            ADD     AH,     @SP     ;SIGNS!
1979  006746  060167  172644                    ADD     AC,     BE      ;COMPUTE EXPONENT
1980  006752  005000                            CLR     TEMP            ;CLEAR THE HIGH SIDE OF MUL-DIV
1981  006754  005001                            CLR     AC              ;...
1982  006756  012746  000037                    MOV     #31.,   -(SP)   ;CREATE BIT COUNTER
1983  006762  005702                            TST     BH              ;TAKE ABSOLUTE VALUES
1984  006764  002003                            BGE             .+10    ;TAKE ABSOLUTE VALUES
1985  006766  005402                            NEG             BH      ;NEGATE B
1986  006770  005403                            NEG             BL      ;NEGATE B
1987  006772  005602                            SBC             BH      ;NEGATE B
1988  006774  005704                            TST             AH      ;TEST SECOND ARGUMENT
1989  006776  002003                            BGE             .+10    ;TEST SECOND ARGUMENT
1990  007000  005404                            NEG             AH      ;NEGATE A
1991  007002  005405                            NEG             AL      ;NEGATE A
1992  007004  005604                            SBC             AH      ;NEGATE A
1993  007006  000176  000004                    JMP     @4(SP)          ;GO BACK
```

```
 1995
 1996                                   ;"FPRINT"
 1997
 1998                                   ;FLOATING POINT OUTPUT CONVERSION FUNCTION
 1999
 2000                                   ;[RE-ENTRANT!]
 2001                                   ;FORMAT DATA IS TAKEN FROM "PTR"
 2002                                   ;THIS PAGE PRODUCES A 7 DIGIT STRING PLUS LEADING ZEROS,
 2003
 2004  007012  005046                  PRNTF:  CLR     -(SP)                   ;OPEN PLACE FOR EXPONENT OF TEN,
 2005  007014  012704  000040                  MOV     #40,    CHAR            ;SETUP LEADING SPACE
 2006  007020  005702                          TST             BH              ;TAKE THE ABSOLUTE VALUE
 2007  007022  003004                          BGT     FLOSGN                  ;PRINT SPACE FOR POSITIVE NUMBER,
 2008  007024  001431                          BEQ     FOGO3                   ;PRINT ALL ZEROS
 2009  007026  007074                          FPNEG                           ;NEGATE THE NEGATIVE FLAC
 2010  007030  062704  000015                  ADD     #15,    CHAR            ;CREATE MINUS SIGN
 2011  007034  012701  001616          FLOSGN: MOV     #BE,    AC              ;CREATE POINTER TO EXPONENT
 2012  007040  005216                          INC     @SP
 2013  007042  021127  000004          FOGO1:  CMP     @AC,    #4
 2014  007046  002411                          BLT     FOGO2
 2015  007050  003004                          BGT     PTEN-2
 2016  007052  126727  172543  000120          CMPB    1+WORD, #120
 2017  007060  002413                          BLT     FOGO3
 2018  007062  007045                          FPMUL+IMMED
 2019  007064  064375  063146          PTEN:   064375,063146
 2020  007070  000763                          BR      FOGO1-2                 ;REPEAT
 2021  007072  005711                  FOGO2:  TST     @AC
 2022  007074  003005                          BGT     FOGO3
 2023  007076  005316                          DEC     @SP
 2024  007100  007045                          FPMUL+IMMED
 2025  007102  000004  050000          TEN:    4,50000
 2026  007106  000755                          BR      FOGO1
 2027                                                                          ;TEST RANGE FOR 0-4
 2028  007110  005046                  FOGO3:  CLR     -(SP)                   ;EXTRA HOLE FOR CARRY,
 2029  007112  104004                          PRINTC                          ;PRINT SIGN,
 2030  007114  012702  001612                  MOV     #FLARG, PTR             ;SET TEMPORARY POINTER
 2031  007120  012700  000007                  MOV     #7,,    TEMP            ;SETUP 7D COUNTER
 2032  007124  007061                  FOGO4:  FPPUT+IPTR                      ;ROUND LSB AND SAVE
 2033  007126  007071                          FPINT                           ;GET THE DIGIT
 2034  007130  010146                          MOV     AC,     -(SP)           ;STACK RESULTS
 2035  007132  007074                          FPNEG                           ;REMOVE THE DIGIT
 2036  007134  007011                          FPADD+IPTR                      ;...
 2037  007136  007040  007102                  FPMUL+DIRECT,TEN                        ;AND DIG OUT THE NEXT DIGIT,
 2038  007142  005300                          DEC     TEMP                    ;COUNT DIGITS
 2039  007144  003367                          BGT     FOGO4                   ;AND TRY AGAIN
 2040  007146  005046                          CLR     -(SP)                   ;OPEN SPACE FOR FORMAT DATA (VIA "PTR")
 2041  007150  116616  000036                  MOVB    36(SP), @SP             ;GET DECIMAL PART OF %X.0Y, (OLD "PTR")
 2042  007154  012746  177400                  MOV     #177400,-(SP)           ;ADD CORRECTION FACTOR,
 2043  007160  007003                          FPGET+FROM+STACK                        ;COMPUTE THE FRACTION
 2044  007162  007045                          FPMUL+IMMED                     ;SCALE STEP NOS, BY
 2045  007164  000016  062000                  16,062000                       ;100.X2**7
 2046  007170  007071                          FPINT                           ;...
 2047  007172  022626                          CLOSE+STACK                     ;RESULT IS IN "AC"
 2048                                  ;NO MORE CALLS TO FLOAT,
```

```
2050
2051                                    ;COMPUTE THE L,S,D, POSITION
2052
2053  007174  010605                            MOV     SP,     R5          ;START POINTER
2054  007176  116603  000035                   MOVB    35(SP), F           ;COPY TOTAL DIGITS REQUESTED,
2055  007202  016602  000020                   MOV     20(SP), E           ;DIG UP THE NUMBER OF INTEGER DIGITS COMPUTED,
2056  007206  020301                            CMP     F,      AC          ;CORRECT FOR ILLOGICAL REQUESTS
2057  007210  003002                            BGT     .+6                 ;OF DECIMALS > FIELD SIZE,
2058  007212  010301                            MOV     F,      AC          ;,,,
2059  007214  005301                            DEC     AC                  ;AC=F-1
2060  007216  010104                            MOV     AC,     CHAR        ;USING E FORMAT?
2061  007220  012700  000006                   MOV     #6,     P           ;INIT, SIX DIGIT OUTPUT,
2062  007224  020302                            CMP     F,      E           ;ENOUGH SPACE?
2063  007226  002001                            BGE     .+4                 ;SKIP TO YES,
2064  007230  005003                            CLR     F                   ;NO, USE E FORMAT,
2065  007232  005703                            TST     F                   ;ASK QUESTION
2066  007234  001411                            BEQ     TOF                 ;YES, MOVE AHEAD, AOK,
2067  007236  060201                            ADD     E,      AC          ;D+E
2068  007240  020301                            CMP     F,      AC          ;F < D+E?
2069  007242  002001                            BGE     TOG                 ;I,E, ENOUGH SPACE?
2070  007244  010301                            MOV     F,      AC          ;ANY DIGITS VISIBLE?
2071  007246  020100          TOG:    CMP     AC,     P           ;MUST BE LESS THAN SIX,
2072  007250  002003                            BGE     TOF                 ;KEEP 6 IN P,
2073  007252  010100                            MOV     AC,     P           ;YES, ROUND TO D+E
2074  007254  002001                            BGE     TOF                 ;YES, USE P
2075  007256  005000                            CLR     P                   ;NO, ROUND LEAD DIGIT,
2076  007260  010001          TOF:    MOV     P,      AC          ;SAVE NO, OF AVAILABLE DIGITS
2077  007262  005100                            COM     P                   ;P = -(P+1) TO ROUND
2078  007264  006300                            ASL     P                   ;COMPUTE WORD INDEX TO +5 ROUND,
2079  007266  012746  000016                   MOV     #16,    -(SP)       ;INDEX FOUND
2080  007272  060516                            ADD     R5,     @SP         ;COMPUTE END ADDRESS
2081  007274  061600                            ADD     @SP,    P           ;COMPUTE +5 ROUND ADDRESS,
2082
2083                                    ;ROUND THE B,C,D'S
2084
2085  007276  005215          TOT:    INC     @R5                         ;FUDGE FACTOR
2086  007300  020500                            CMP     R5,     P           ;SPECIAL ROUNDUP FACTOR
2087  007302  001002                            BNE     TOR                 ;TEST FOR THERE ALREADY
2088  007304  062715  000004                   ADD     #4,     @R5         ;ROUND BY +4 (DON'T CHANGE THSI TO 5!)
2089  007310  022715  000012          TOR:    CMP     #10.,   @R5         ;OVERFLOW CAUSE
2090  007314  003003                            BGT     TOS                 ;SKIP IF OK
2091  007316  162725  000012                   SUB     #10.,   (R5)+       ;CORRECT FOR OVERFLOW
2092  007322  000765                            BR      TOT                 ;AND CARRY
2093  007324  005725          TOS:    TST     (R5)+                       ;BUMP THE POINTER,
2094  007326  020500                            CMP     R5,     P           ;TEST FOR ROUNDOFF START
2095  007330  001765                            BEQ     TOR-4               ;GO DO IT,
2096  007332  021605                            CMP     @SP,    R5          ;TEST FOR END,
2097  007334  003373                            BGT     TOS                 ;REPEAT
2098  007336  001402                            BEQ     .+6                 ;SKIP IF NO CARRY ONE,
2099  007340  005202                            INC     E                   ;UPDATE "E"
2100  007342  005201                            INC     AC                  ;AND FIELD SIZE,
2101  007344  004767  000006                   JSR     PC,     FPRNT       ;CALL OUTPUT ROUTINE
2102  007350  062706  000024                   ADD     #24,    SP          ;CORRECT STACK
2103  007354  000547                            BR      FIG04               ;AND RETURN TO TOP LEVEL,
```

```
2105
2106                                    ;FIXED POINT OUTPUT FORMAT
2107
2108  007356  010300                    FPRNT:  MOV     F,      P               ;USE LARGER OF F-D OR E
2109  007360  001433                            BEQ     FLOUT                   ;TEST FOR FLOATING POINT FORMAT
2110  007362  160400                            SUB     CHAR,   P               ;COMPUTE INTEGER PLACES REQUESTED
2111  007364  020002                            CMP     P,      E               ;FORMAT OK?
2112  007366  002003                            BGE     FPRNTP                  ;YES, GO DO IT.
2113  007370  010200                            MOV     E,      P               ;SHIFT DECIMAL POINT RIGHT
2114  007372  020203                            CMP     E,      F               ;BUT NOT TOO FAR!
2115  007374  003025                            BGT     FLOUT                   ;GO USE "E" FORMAT ANYWAY.
2116  007376  005004                    FPRNTP: CLR     CHAR                    ;0
2117  007400  020002                            CMP     P,      F               ;TEST DISTANCE TO DOT.
2118  007402  003014                            BGT             FTRY            ;GO LINE UP A DIGIT
2119  007404  005302                            DEC     E                       ;COUNT DOWN INTEGER DIGITS
2120  007406  005301                            DEC     AC                      ;COUNT DOWN STRING DATA.
2121  007410  002401                            BLT     LTZERO                  ;PRINT TRAILING ZERO
2122  007412  014504                            MOV     -(R5),  CHAR            ;GET AN ACTUAL DIGIT
2123  007414  004767  000112           LTZERO: JSR     PC,     PRNT1           ;OR A REAL DIGIT.
2124  007420  005303                            DEC     F                       ;COUNT DOWN THE FIELD.
2125  007422  003446                            BLE     DXIT                    ;BRANCH IF END OF DATA FORMAT.
2126  007424  005300                            DEC     P                       ;COUNT DOWN DISTANCE TO DOT.
2127  007426  001363                            BNE     FPRNTP                  ;THERE YET?
2128  007430  104456                            PRINT+  !,                      ;YES
2129  007432  000761                            BR      FPRNTP                  ;CONTINUE
2130                                                                            ;
2131  007434  020027  000001           FTRY:   CMP     P,      #1              ;PRINT AT LEAST 0
2132  007440  003765                            BLE     LTZERO                  ;...
2133  007442  012704  177760                    MOV     #40-60, CHAR            ;OR A LEADING SPACE
2134  007446  000762                            BR      LTZERO                  ;...
2135
```

```
2137                                    ;FLOATING POINT OUTPUT FORMAT
2138
2139  007450  010246                    FLOUT:  MOV     E,      -(SP)           ;SAVE POWER OF TEN FOR LATER,
2140  007452  005002                            CLR     E                       ;0,XXXX
2141  007454  012700  000001                    MOV     #1,     P               ;...
2142  007460  012703  000007                    MOV     #7,     F               ;FIELD SIZE
2143  007464  004767  177706                    JSR     PC,     FPRNTP          ;PRINT FIRST DIGIT GROUP
2144  007470  104505                            PRINT+  'E                      ;E+-XXX
2145  007472  012602                            MOV     (SP)+,  PTR             ;LOOK AT EXPONENT AGAIN
2146  007474  002003                    PRNTS:  BGE     PRNTP-2                 ;GO PRINT SIGN,
2147  007476  005402                            NEG             PTR             ;CORRECT E TO ABSOLUTE VALUE
2148  007500  104455                            PRINT+  '-                      ;PRINT SIGN,
2149  007502  000401                            BR      PRNTP                   ;FALL INTO DIGIT PRINT ROUTINE,
2150  007504  104453                            PRINT+  '+                      ;...
2151  007506  104644  000144            PRNTP:  DIGTST, 100,                    ;PRINT 2 OR 3 DIGITS
2152  007512  120427  000060                    CMPB    CHAR,   #60             ;...
2153  007516  001401                            BEQ     PRNT2                   ;IGNORE FIRST ZERO,
2154  007520  104604                            PRINTC                          ;...
2155  007522  104644  000012            PRNT2:  DIGTST, 10,                     ;PRINT 2 DIGITS FOR SURE
2156  007526  104604                            PRINTC                          ;...
2157  007530  010204                            MOV     PTR,    CHAR            ;PRINT LAST DIGIT
2158  007532  062704  000060            PRNT1:  ADD     #60,    CHAR            ;...
2159  007536  104604                            PRINTC                          ;...
2160  007540  000207                    DXIT:   POPJ                            ;AND RETURN,
2161
2162
2163
2164                                    ;END OF OUTPUT CONVERSION ROUTINES
```

```
2166
2167
2168                                    ;"FREAD"
2169
2170                                    ;FLOATING POINT INPUT CONVERSION FUNCTION
2171                                    ;(RE-ENTRANT)
2172
2173  007542  016603  000014           READF:  MOV     14(SP), AXOUT           ;(THIS CODE IS NOT RE-ENTRANT!)
2174  007546  004767  000130                   JSR     PC,     DECONV          ;CONNECT FIRST DIGIT GROUP
2175  007552  005002                           CLR     PTR                     ;START FRACTION COUNTER.
2176  007554  120427  000056                   CMPB    CHAR,   #'.             ;DID IT END IN PERIOD?
2177  007560  001002                           BNE             FIGO1           ;NO, GO FINISH THE ANSWER
2178  007562  004767  000134                   JSR     PC,     DECON1          ;COUNT DIGITS AND APPEND RESULT.
2179  007566  010246                   FIGO1:  MOV     PTR,    -(SP)           ;OPEN SPACE ON THE STACK
2180                                                                           ;FOR NO. OF DEC. DIGITS.
2181  007570  132767  000020  172033           BITB    #NALPHA,SWITCH          ;FROM GETLN?
2182  007576  001417                           BEQ     FIGO2                   ;YES, LEAVE
2183  007600  120427  000105                   CMPB    CHAR,   #'E             ;DID IT END WITH "E"?
2184  007604  001014                           BNE             FIGO2           ;NO, GO EXIT
2185  007606  007060  001612                   FPPUT+DIRECT,   FLARG           ;YES, SAVE PRESENT RESULT
2186  007612  104614                           GETC                            ;READ ON PAST THE 'E',
2187  007614  004767  000062                   JSR     PC,     DECONV          ;READ EXPONENT
2188  007620  005701                           TST     AC                      ;CORRECT FOR SIGN OF EXPONENT.
2189  007622  100401                           BMI     .+4                     ;...
2190  007624  007074                           FPNEG                           ;...
2191  007626  007071                           FPINT                           ;MAKE AN INTEGER IN 'AC'
2192  007630  060116                           ADD     AC,     @SP             ;SAVE EXPONENT AND USE AS COUNTER
2193  007632  007000  001612                   FPGET+DIRECT,   FLARG           ;RECOVER VALUE OF DECIMAL FRACTION
2194  007636  012702  007064           FIGO2:  MOV     #PTEN,  PTR             ;INITIALIZE CORRECTION POINTER
2195  007642  012601                           MOV     (SP)+,  AC              ;TEST THE SIGN OF THE EXPONENT
2196  007644  100003                           BPL     FIGOE                   ;IF +,CORRECT BY *.10
2197  007646  005401                           NEG     AC                      ;TAKE ABS. VAL. OF THE EXP COUNT.
2198  007650  012702  007102                   MOV     #TEN,   PTR             ;IF -, CORRECT BY *10.
2199  007654  005301           FIGOE:  DEC     AC                      ;DO THE CORRECTION.
2200  007656  002402                           BLT     FIGO3                   ;LEAVE WHEN FINISHED.
2201  007660  007041                           FPMUL+IPTR                      ;CORRECT FLAC VIA POINTER ON THE STACK.
2202  007662  000774                           BR      FIGOE                   ;REPEAT UNTILL EXPONENT COMPLETED.
2203                                                                           ;
2204  007664  010466  000016           FIGO3:  MOV     CHAR,   16(SP)          ;INTERNAL DATA-YES,
2205                                                                           ;COPY TERMINATING CHARACTER,
2206  007670  010366  000014                   MOV     AXOUT,  14(SP)          ;COPY TEXT POINTER
2207  007674  005726                   FIGO4:  TST     (SP)+                   ;LEAVE VALUE
2208  007676  000167  175722                   JMP     FLTX                    ;TO BE RESTORED AT "PWREGS"
2209
2210
```

```
2212
2213                                    ;CONVERT A GROUP OF NUMBERS,
2214
2215  007702  007077                    DECONV: FZER                    ;CLEAR INITIAL VALUE AND 'AC'
2216  007704  120427  000201                    CMPB    CHAR,   #201    ;+?
2217  007710  001404                            BEQ             DECON1  ;YES
2218  007712  120427  000202                    CMPB    CHAR,   #202    ;-?
2219  007716  001002                            BNE             DECON1+2 ;NO
2220  007720  005101                            COM             AC      ;YES-REVERSE SIGN
2221  007722  104614                    DECON1: GETC                    ;GET NEXT CHARACTER
2222  007724  105704                            TSTB            CHAR    ;CHAR FOR 'TERMS'
2223  007726  100704                            BMI             DXIT    ;LEAVE ON THE TERMINATOR
2224  007730  120427  000056                    CMPB    CHAR,   #'.     ;TEST FOR GROUP TERMINATOR
2225  007734  001701                            BEQ             DXIT    ;AND LEAVE IF FOUND
2226  007736  104654  010006                    SKPNON, DETN            ;CHECK FOR NUMBER
2227  007742  132767  000020  171661            BITB    #NALPHA,SWITCH  ;TEST FOR ALPHA AS TERM,
2228  007750  001673                            BEQ     DXIT            ;CLEAR=YES,  IF =1 THEN CAN'T BE E-FORMAT EITHER
2229  007752  120427  000105                    CMPB    CHAR,   #'E     ;CHECK FOR SPECIAL FORMAT
2230  007756  001670                            BEQ     DXIT            ;...
2231  007760  042704  177740                    BIC     #-40,   CHAR    ;CLEAR HIGH ORDER BITS OF "A-Z" ASCII CODES,
2232  007764  010446                    DECOY:  MOV     CHAR,   -(SP)   ;SAVE THIS DIGIT ON STACK
2233  007766  012746  000017                    MOV     #15.,   -(SP)   ;LOAD INTEGER EXPONENT ONTO STACK
2234  007772  007040  007102                    FPMUL+DIRECT,   TEN     ;MULTIPLY  PRESENT VALUE BY 10
2235  007776  007013                            FPADD+FROM+STACK                ;ADD IN NEW DIGIT
2236  010000  022626                            CLOSE+STACK             ;REMOVE ARG, FROM STACK,
2237  010002  005202                    DECON2: INC             PTR     ;COUNT DIGITS
2238  010004  000746                            BR      DECON1          ;GO BACK FOR MORE,
2239                                            ;
2240  010006  042704  177760            DETN:   BIC     #-20,   CHAR    ;USE BCD PART OF ASCII NUMBER,
2241  010012  001364                            BNE             DECOY   ;GO USE THE DIGIT IF NON-ZERO,
2242  010014  007040  007102                    FPMUL+DIRECT,TEN                ;UPDATE PRESENT VALUE
2243  010020  000770                            BR      DECON2          ;REPEAT,
```

```
2245
2246                                    ;"INCH" AND "OUTCH"
2247
2248                                    ;USING 'INDEV' AND OUTDEV' FOR 8-BIT I/O.
2249
2250  010022  016704  171604           XI33:   MOV     INDEV,  CHAR            ;END BUFFER TO POINTER
2251  010026  022704  177560                   CMP     #TKS,   CHAR            ;TEST FOR DEVICE
2252  010032  001420                           BEQ     KITH                    ;LOW SPEED
2253  010034  105714                           TSTB    @CHAR                   ;WAIT FOR FLAG
2254  010036  100376                           BPL     .-2                     ;WAIT FOR KEYBOARD DONE FLAG
2255  010040  116404  000002                   MOVB    2(CHAR),CHAR            ;SAVE INPUT BUFFER
2256  010044  005277  171562           XI33X:  INC     @INDEV                  ;REQUEST NEXT
2257  010050  000205                           RTS     R5                      ;RETURN FROM I/O DEVICE ROUTINE.
2258
2259  010052  010446                   XOUT:   MOV     CHAR,   -(SP)           ;SAVE DATA
2260  010054  016704  171554                   MOV     OUTDEV, CHAR            ;LOAD POINTER
2261  010060  105714                           TSTB    @CHAR                   ;TEST FLAG
2262  010062  100376                           BPL     .-2                     ;WAIT
2263  010064  111664  000002                   MOVB    @SP,    2(CHAR)         ;OUTPUT DATA
2264  010070  012604                           MOV     (SP)+,  CHAR            ;RESET CHARACTER
2265  010072  000205                           RTS     R5                      ;RETURN
2266
2267
2268                                    ;THIS KEYBOARD CONTROL IS ASYNCHRONOUS
2269
2270  010074  016704  171536           KITH:   MOV     KIN,    CHAR            ;TEST FOR DATA READY.
2271  010100  100775                           BMI     KITH                    ;WAIT FOR INTERRUPT DONE
2272  010102  005167  171530                   COM     KIN                     ;RESET
2273  010106  000756                           BR      XI33X                   ;RETURN
2274
2275                                    ;KEYBOARD INTERRUPT HANDLER
2276
2277  010110  013746  177562           KINT:   MOV     @#TKS+2, -(SP)
2278  010114  121627  000203                   CMPB    @SP,    #203    ;TEST FOR CONTROL-C
2279  010120  001004                           BNE     KINT1           ;NO
2280  010122  105167  171453                   COMB    CCFLG           ;YES, CHANGE DATE
2281  010126  001003                           BNE     KINT2           ;TWO?
2282  010130  104401                           ERROR+201+0,+0,         ;YES
2283  010132  105067  171443           KINT1:  CLRB    CCFLG
2284  010136  005767  171474           KINT2:  TST     KIN                     ;ANY ROOM?
2285  010142  100401                           BMI     .+4                     ;YES
2286  010144  104645                           ERROR+201+18,+18',              ;INPUT BUFFER OVERFLOW,
2287  010146  012667  171464                   MOV     (SP)+,KIN               ;READ DATA; CLEAR INTERRUPT DONE
2288  010152  000002                           RTI                             ;CONTINUE
```

```
2290
2291                                    ;SQUARE ROOT FUNCTION <OPTIONAL>
2292
2293  010154  012705  001616            XSQT:   MOV     #BE,    R5              ;POINT TO FLOATING RESULT (3 WORDS)
2294  010160  012702  001[41]            MOV     #FLARG+2,PTR                    ;POINT TO TEMP STORAGE (2 WORDS)
2295  010164  011246                     MOV     @PTR,   -(SP)                   ;SAVE VALUE FOR NEWTON'S METHOD
2296  010166  100001                     BPL     .+4                             ;TEST FOR
2297  010170  104443                     ERROR+201+17,+17.                       ;IMAGINARY ROOTS,
2298  010172  001427                     BEQ     SQUEND                          ;ARG AND RESULT ARE ZERO,
2299  010174  014246                     MOV     -(PTR), -(SP)                   ;SAVE OTHER BYTE
2300  010176  106212                     ASRB    @PTR                            ;MAKE FIRST APPROX,
2301  010200  105512                     ADCB    @PTR                            ;...
2302  010202  012762  060320  000002     MOV     #60320, 2(PTR)                  ;...
2303  010210  007003                CLCU:   FPGET+FROM+STACK                        ;MAKE SUCCESSIVE
2304  010212  007031                       FPDIV+IPTR                            ;APPROXIMATIONS,
2305  010214  007011                       FPADD+IPTR                            ;...
2306  010216  005315                       DEC     @R5                           ;ARE THE EXPONENTS
2307  010220  121215                       CMPB    @PTR,   @R5                   ;EQUAL?
2308  010222  001010                       BNE     ROOTGO                        ;NO, TRY AGAIN
2309  010224  026267  000002  171366       CMP     2(PTR), WORD                  ;YES, TEST SOME MORE
2310  010232  001004                       BNE     ROOTGO                        ;NO
2311  010234  126267  000001  171361       CMPB    1(PTR), LORD+1                ;LAST TEST FOR FSQT(3)
2312  010242  103002                       BHIS    SQX                           ;...
2313  010244  007061                ROOTGO: FPPUT+IPTR                            ;SAVE NEW ESTIMATE
2314  010246  000760                       BR      CLCU                          ;TRY AGAIN
2315  010250  005726                SQX:    TST     (SP)+                         ;REPAIR STACK
2316  010252  005726                SQUEND: TST     (SP)+
2317  010254  000207                       POPJ                                  ;RETURN
2318
2319
2320
2321
2322
2323
2324                                    ;CLOCK CONTROLLER (STARTED BY FX(-1,177546,192))
2325                                    ;[OVERHEAD=30,3 USECS] <OPTIONAL>
2326
2327  010256  062767  000400  000010  CINI:   ADD     #400,   CLKT            ;COUNT 1/60 SEC,
2328  010264  005567  000006                  ADC             CLKT+2          ;MILLISEC
2329  010270  000002                       RTI                                   ;CONTINUE
2330
2331
2332
2333
2334                                    ;CLOCK TIME-OF-DAY FUNCTION IN 60THS (OR 50THS) OF A SECOND
2335                                    ;[OVERFLOW AFTER 12,94 DAYS]
2336
2337  010272  007005                XFCLK:  FPGET+IMMED                            ;LOAD TIME
2338  010274  000427  010276        CLKT:   427,    0+,                           ;TIME-OF-DAY [IMPURE-USER INDEPENDENT]
2339  010300  007070                       FPNOR                                 ;NORMALIZE
2340  010302  000207                       POPJ                                  ;RETURN
```

```
2342
2343                                    ; *OPERATE*=PROGRAMMABLE I/O COMMAND, <OPTIONAL>
2344
2345            020724                  Z=IOLIST+IOLIST
2346
2347  010304    006301                  IOFIX:  ASL     AC                              ;MAKE EVEN
2348  010306    016102  167454                  MOV     IOPATCH-Z(AC),  PTR             ;GET PATCH ADDRESS
2349  010312    016112  167436                  MOV     IOGO-Z(AC),     @PTR            ;STORE THE PATCH
2350  010316    022712  177550                  CMP     #PRS,           @PTR            ;H,S,?
2351  010322    001004                          BNE     IOQ                             ;NO
2352  010324    032732  004200                  BIT     #4200,          @(PTR)+         ;YES, 'BUSY' OR 'DONE'?
2353  010330    001001                          BNE     IOQ                             ;YES
2354  010332    005252                          INC     @-(PTR)                         ;NO, SET READER ENABLE,
2355  010334    104614                  IOQ:    GETC                                    ;LOOK AT NEXT TEXT CHAR,
2356  010336    104634                  PROGIO: SPNOR                                   ;GOTO NEXT LETTER
2357  010340    104602  010352  010304         SORTC,IOLIST,IOFIX                      ;TEST
2358  010346    000167  173102                  JMP     PROC                            ;CONTINUE THE LINE,
2359
2360
2361            010352                  IOLIST=.
2362  010352    045522  052120  051514  .ASCII  "RKPTLSH"
      010360      110
2363  010361      000                   .BYTE   0
2364                                    .EVEN
2365
2366
2367
2368            010362                  IOGO=.
2369  010362    177550                          PRS
2370  010364    177560                          TKS
2371  010366    177554                          PPS
2372  010370    177564                          TPS
2373  010372    177514                          LPS
2374  010374    177700                          177700          ;USE %R0 IF USER HAS NO GT40; WRITING IN IT WON'T HURT
2375  010376    175614                          175614          ;ADDRESS OF PDP-10 OUTPUT
2376
2377
2378            010400                  IOPATCH=.
2379  010400    001632                          INDEV
2380  010402    001632                          INDEV
2381  010404    001634                          OUTDEV
2382  010406    001634                          OUTDEV
2383  010410    001634                          OUTDEV
2384  010412    001634                          OUTDEV
2385  010414    001634                          OUTDEV
2386                                    ;***...
```

```
2388                                    ;FAST SINE,COSINE <OPTIONAL>
2389
2390                                    ;SMALL: 51(10) WORDS
2391                                    ;SIN(X)=COS(PI/2-X)
2392                                    ;COS(4X)=8*<COS(X)^4-COS(X)^2>+1
2393                                    ;COS(X)=1-X^2/2+X^4/24-X^6/720
2394                                    ;       =1-X^2*(1-X^2/12+X^4/360)/2
2395                                    ;       =1-X^2*(1-X^2*<1-X^2/30>/12)/2
2396                                    ;       =1+S*(1+S*<1+S/30>/12)/2
2397
2398  010416  007074                    FSIN:   FPNEG                           ;SIN(X)=COS(PI/2-X)
2399  010420  007015                            FPADD+IMMED                     ;...
2400  010422  166001  062207                            166001, 62207         ;PI/2=1.570796
2401  010426  024646                    FCOS:   OPEN+STACK                      ;USE STACK AS TEMPORARY STORAGE
2402  010430  005002                            CLR     PTR                     ;CLEAR COUNTER
2403  010432  007063                            FPPUT+INTO+STACK                        ;SAVE X
2404  010434  007043                            FPMUL+FROM+STACK                        ;TAKE X SQUARED,
2405  010436  005737  001616                    TST     @#BE                    ;X^2<125?
2406  010442  002411                            BLT     FCOS2                   ;YES
2407  010444  111602                            MOVB    @SP,    PTR             ;COPY EXPONENT OF TWO
2408  010446  112716  177777                    MOVB    #-1,    @SP             ;0 IF EVEN
2409  010452  006202                            ASR     PTR                     ;FIND RESULT IF DIVIDED BY FOUR  (EXP-2),
2410  010454  103401                            BCS     .+4                     ;=-1 IF ODD
2411  010456  106316                            ASLB    @SP                     ;=-2 IF EVEN
2412  010460  005202                            INC     PTR                     ;SET COUNTER
2413  010462  007003                            FPGET+FROM+STACK                        ;COPY RESULT
2414  010464  007043                            FPMUL+FROM+STACK                        ;S=-X^2
2415  010466  007074                    FCOS2:  FPNEG                           ;...
2416  010470  007063                            FPPUT+INTO+STACK                        ;PUT S ON STACK,
2417  010472  007035                            FPDIV+IMMED                     ;TAKE FIRST FRACTION
2418  010474  000005  074000                            5,74000               ;30
2419  010500  007010  010560                    FPADD+DIRECT,FLTONE             ;COMPUTE INNER EXPRESSION
2420  010504  007035                            FPDIV+IMMED                     ;...
2421  010506  000004  060000                            4,60000               ;12
2422  010512  007043                            FPMUL+FROM+STACK                        ;FIND ( )
2423  010514  007010  010560                    FPADD+DIRECT,FLTONE             ;...
2424  010520  005337  001616                    DEC     @#BE                    ;DIVIDE BY TWO (EXP-1)
2425  010524  007043                            FPMUL+FROM+STACK                        ;COMPUTE LAST OF SERIES
2426  010526  007010  010560          FCOS4:  FPADD+DIRECT,FLTONE             ;...
2427  010532  005302                            DEC     PTR                     ;COUNT "RECURSIONS"
2428  010534  002645                            BLT     SQX                     ;LEAVE WHEN DONE,
2429  010536  007063                            FPPUT+INTO+STACK                        ;FIND COS^2
2430  010540  007043                            FPMUL+FROM+STACK                        ;...
2431  010542  007063                            FPPUT+INTO+STACK                        ;SAVE COS^2
2432  010544  007043                            FPMUL+FROM+STACK                        ;COS^4
2433  010546  007023                            FPSUB+FROM+STACK                        ;COS^4-COS^2
2434  010550  062737  000003  001616            ADD     #3,     @#BE            ;MULTIPLY BY EIGHT,
2435  010556  000763                            BR      FCOS4                   ;REPEAT
2436                                                                            ;
2437  010560  000001  040000          FLTONE:                 1,040000          ;CONSTANT OF ONE,
```

```
2439
2440
2441
2442                                    ;SYMBOL TABLE TYPEOUT ROUTINE <OPTIONAL>
2443
2444                                    ;USED BY *TYPE*ASK*
2445                                    ;VIA 'ATLIST'
2446
2447  010564  016705  171056           TDUMP:  MOV     STARTV, R5              ;INIT POINTER
2448  010570  005715                    TDUMP1: TST             @R5             ;TEST FOR NULL ENTRY
2449  010572  001007                            BNE             TDUMP2          ;GO TYPE NAME, ETC,
2450  010574  062705  000010                   ADD     #10,    R5              ;MOVE POINTER
2451  010600  020567  171046           TDUMP3: CMP     R5,     BOTTOM          ;TEST LIMITS
2452  010604  103771                            BLO     TDUMP1                  ;TRY AGAIN
2453  010606  005726                            TST     (SP)+                   ;BUMP STACK ("TASK" RETURN)
2454  010610  000207                            POPJ                            ;LEAVE *TYPE*ASK*
2455                                                                            ;
2456  010612  104642  020123           TDUMP2: PRINT2, "S                      ;MAKE COMMANDS!
2457  010616  112504                            MOVB    (R5)+,  CHAR            ;READ FIRST LETTER OF NAME N
2458  010620  104604                            PRINTC                          ;AND PRINT SAME
2459  010622  112504                            MOVB    (R5)+,  CHAR            ;READ SECOND LETTER
2460  010624  100401                            BMI     .+4                     ;IGNORE SPACE
2461  010626  104604                            PRINTC                          ;AND PRINT
2462  010630  104450                            PRINT+  !(                      ;OPEN PARENTHESIS
2463  010632  005725                            TST     (R5)+                   ;ALL ZEROS?
2464  010634  001410                            BEQ     TDUMP4                  ;YES
2465  010636  005745                            TST     -(R5)
2466  010640  112502                            MOVB    (R5)+,  PTR             ;COPY SUBSCRIPT #1 AND TEST SIGN,
2467  010642  004767  176626                    JSR     PC,     PRNTS           ;PRINT SUBSCRIPT DIGITS,
2468  010646  104454                            PRINT+  !,                      ;COMMA
2469  010650  112502                            MOVB    (R5)+,  PTR             ;COPY LEFT-HAND BITS AND
2470  010652  004767  176616                    JSR     PC,     PRNTS           ;PRINT SAME
2471  010656  104642  036451           TDUMP4: PRINT2, ")=                     ;CLOSE PARE
2472  010662  010502                            MOV     R5,     PTR             ;COPY POINTER,
2473  010664  007001                            FPGET+IPTR                      ;LOAD THE VALUE OF THE VARIABLE
2474  010666  016702  170734                    MOV     FISW,   PTR             ;LOAD FORMAT DATA
2475  010672  007076                            FPRINT                          ;PRINT CONTENT OF FLAC
2476  010674  104642  005015                    PRINT2, CRLF                    ;PRINT CRLF AT END OF LINE
2477  010700  022525                            CMP     (R5)+,  (R5)+           ;MOVE POINTER TO NEXT ENTRY
2478  010702  000736                            BR              TDUMP3          ;CONTINUE THE SCAN
```

```
2480
2481                                    ;RANDOM NUMBER GENERATOR <OPTIONAL>
2482
2483                                    ;FRAN() = A STATISTICALLY RANDOM, PSEUDO-NOISE SHIFT REGISTER
2484
2485                                    ;WITH PERIODICITY = 32767(10),
2486
2487                                    ;AVERAGE = ,00060
2488
2489                                    ;RANGE = +1 TO -1
2490
2491  010704  012702  001602    XRAN:   MOV     #LSPR,  PTR
2492  010710  012705  000014            MOV     #14.,   R5
2493  010714  005767  170700            TST     HORD                    ;NON-ZERO ARGUMENT INITIALIZES
2494  010720  001402                    BEQ     XROL
2495  010722  012712  107654            MOV     #107654,@PTR
2496  010726  011201            XROL:   MOV     @PTR,   AC
2497  010730  006101                    ROL     AC
2498  010732  100401                    BMI     .+4
2499  010734  005101                    COM     AC                      ;XOR BITS 13+14
2500  010736  006101                    ROL     AC
2501  010740  006101                    ROL     AC
2502  010742  006112                    ROL     @PTR                    ;(THANKS JOHN LARKIN!)
2503  010744  005305                    DEC     R5
2504  010746  003367                    BGT     XROL
2505  010750  011267  170644            MOV     @PTR,   HORD            ;RANGE IS +1-1
2506  010754  000207                    POPJ
```

```
2508
2509                                    ;CHARACTER I/O FUNCTION <OPTIONAL>
2510
2511                                    ;FCHR(-1):INPUT ASCII
2512                                    ;FCHR(FOO):OUTPUT ASCII
2513                                    ;FCHR(213,64+0A,64+0B,-1): 3 OUT AND 1 IN,
2514
2515  010756  104660                    XCHMO:  EVAL,X
2516  010760  007071                    XCHR:   FPINT                           ;FORM  INT OF ARG, (ENTRY POINT)
2517  010762  010446                            MOV     CHAR,   -(SP)           ;SAVE NEXT CHARACTER,
2518  010764  010104                            MOV     AC,     CHAR            ;PREPARE TO PRINT
2519  010766  100406                            BMI     XCHR1                   ;BUT PERHAPS GO READ,
2520  010770  104610                            OUTCH                           ;OUTPUT
2521  010772  012604                    XCHARG: MOV     (SP)+,  CHAR            ;RESTORE NEXT CHARACTER
2522  010774  120427  000214                    CMPB    CHAR,   #214            ;ANY MORE? (I,E, COMMA?)
2523  011000  001766                            BEQ     XCHMO                   ;YES!
2524  011002  000207                            POPJ                            ;RETURN
2525                                                                            !
2526  011004  104612                    XCHR1:  INCH                            ;LOOK FOR INPUT
2527  011006  042704  177400                    BIC     #-400,  CHAR            ;8-BIT ASCII
2528  011012  010467  170602                    MOV     CHAR,   WORD            ;SAVE RESULT
2529  011016  012767  000017  170572            MOV     #15.,   BE              ;SET THE EXPONENT
2530  011024  000762                            BR      XCHARG                  !...
2531
2532
2533
2534
2535
2536                                    ;EXECUTE USER FUNCTIONS!!  <OPTIONAL>
2537
2538                                    ;SET Z=FSBR(GROUPNO,ARG)
2539
2540  011026  004567  171460            XFSBR:  JSR     R5,     GTESTW          ;LOAD 'LINENO'
2541  011032  005000                            CLR     TEMP                    ;ZERO SUBSCRIPT
2542  011034  004767  173730                    JSR     PC,     WHIPV           ;PREPARE AMPERSAND
2543  011040  010246                            MOV     PTR,    -(SP)           ;SAVE POINTER,
2544  011042  104660                            EVAL,X                          ;SAVE ARG,
2545  011044  007064                            FPPUT+THROUGH+STACK             ;LOAD ARGUMENT
2546  011046  010446                            MOV     CHAR,   -(SP)           ;SAVE LAST CHARACTER
2547  011050  112704  000216                    MOVB    #CR,    CHAR            ;LOAD NEW TERMINATOR (C,R,).
2548  011054  004767  173010                    JSR     PC,     DO+2            ;DO THE SUBROUTINE
2549  011060  012604                            MOV     (SP)+,  CHAR            ;RESTORE LAST R-PAR,
2550  011062  012602                            MOV     (SP)+,  PTR             ;DIG UP POINTER,
2551  011064  007001                            FPGET+IPTR                      ;RETRIEVE RESULT,
2552  011066  000207                            POPJ                            ;RETURN,
```

```
2554
2555                                    ;A/D CONVERTER FUNCTION FOR ADO1-11 <OPTIONAL>
2556
2557                                    ;SET Z=FADC(GAIN,CHANNEL)
2558
2559  011070  007071                    XADC:   FPINT                           ;READ GAIN AND
2560  011072  006301                            ASL     AC                      ;LOAD IT INTO THE AC POSITION,
2561  011074  006101                            ROL     AC                      ;...
2562  011076  010146                            MOV     AC,     -(SP)           ;SAVE RESULT
2563  011100  104660                            FVAL,X                          ;READ CHANNEL NUMBER,
2564  011102  007071                            FPINT                           ;MAKE IT AN INTEGER,
2565  011104  150166  000001                    BISB    AC,     1(SP)           ;COMBINE THE BITS
2566  011110  012667  165654                    MOV     (SP)+,  ADCS            ;LOAD INTO STATUS,
2567  011114  007077                            FZER                            ;CLEAR 'FLAC' AND DELAY
2568  011116  016767  165650  170474            MOV     ADDB,   HORD            ;READ RESULT,
2569  011124  007045  000011  050000            FPMUL+IMMED,17-6,50000          ;1/102.4
2570  011132  000207                            POPJ                            ;RESULT IN VOLTS,
2571
2572          176770                            ADCS=176770
2573          176772                            ADDB=176772
```

```
2575                                    ;EXPERIMENTAL FUNCTION <OPTIONAL>
2576
2577                                    ;FX( 1,BUSS, ADDR)                      ;READ BYTE
2578                                    ;FX( 0,BUSS, ADDR,DATA)                 ;"AND" WORD
2579                                    ;FX(-1,BUSS, ADDR,DATA)                 ;LOAD BYTE
2580
2581  011134  007071                    XEX:    FPINT
2582  011136  006301                            ASL     AC
2583  011140  010146                            MOV     AC,     -(SP)           ;SAVE FUNCTION CODE
2584  011142  005046                            CLR     -(SP)                   ;START BUS ADDRESS
2585  011144  122704  000214                    CMPB    #214,   CHAR
2586  011150  001033                            BNE     XEX4
2587  011152  104614                    XEX2:   GETC                            ;MOVE ON,
2588  011154  100424                            BMI     XEX3                    ;LEAVE IF TERMINATOR,
2589  011156  120427  000060                    CMPB    CHAR,   #60             ;TEST FOR ALPHA
2590  011162  103414                            BLO     XEX2A                   ;USE EXPRESSION
2591  011164  120427  000067                    CMPB    CHAR,   #67
2592  011170  101011                            BHI     XEX2A
2593  011172  042704  177770                    BIC     #-10.,  CHAR            ;CREATE OCTAL ADDRESS
2594  011176  006316                            ASL     @SP                     ;
2595  011200  006316                            ASL     @SP                     ;
2596  011202  006316                            ASL     @SP                     ;
2597  011204  050416                            BIS     CHAR,   @SP             ;
2598  011206  000761                            BR              XEX2            ;
2599                                                                            ;
2600  011210  110115                    XEM:    MOVB    AC,     @R5             ;LOAD
2601  011212  000207                            POPJ                            ;LEAVE
2602                                                                            ;
2603  011214  004767  173454            XEX2A:  JSR     PC,     GETARG          ;READ ADDR. VARIABLE.
2604  011220  007001                            FPGET+IPTR
2605  011222  007071                            FPINT                           ;CAUTION: BIT 15 IS SIGN BIT
2606  011224  010116                            MOV     AC,     @SP
2607  011226  122704  000214            XEX3:   CMPB    #214,   CHAR            ;COMMA?
2608  011232  001002                            BNE     XEX4
2609  011234  104660                            EVAL,X                          ;READ DATA
2610  011236  007071                            FPINT                           ;PUT INTEGER IN 'AC'
2611  011240  012605                    XEX4:   MOV     (SP)+,  R5              ;GET ADDRESS
2612  011242  012602                            MOV     (SP)+,  PTR             ;RESTORE CODE AND TEST.
2613  011244  100426                            BMI     XEL
2614  011246  003023                            BGT     XER
2615  011250  011505                    XET:    MOV     @R5,    R5              ;LOGICAL "AND"
2616  011252  005105                            COM     R5                      ;   "
2617  011254  040501                            BIC     R5,     AC              ;   "
2618  011256  020127  100000                    CMP     AC,     #100000         ;IS IT -32768?
2619  011262  001007                            BNE     XEXIT                   ;NO
2620  011264  012767  140000  170326            MOV     #140000,HORD            ;YES-IT IS A SPECIAL CASE
2621  011272  012767  000020  170316            MOV     #20,    BE
2622  011300  000207                            POPJ
2623  011302  010167  170312            XEXIT:  MOV     AC,     HORD            ;SAVE RESULT
2624  011306  012767  000017  170302            MOV     #17,    BE              ;AS INTEGER,
2625  011314  000207                            POPJ                            ;GO BACK TO FUNCTION CONTROL.
2626
2627  011316  111501                    XER:    MOVB    @R5,    AC              ;READ THE UNIBUS
2628  011320  000770                            BR      XEXIT                   ;
```

```
2629                                                                    ;
2630  011322  020705                XEL:    CMP     PC,     R5        ;PROTECT FOCAL
2631  011324  101731                        BLOS    XEM               ;PROM
2632  011326  104633                XERR:   ERROR+201+13,+13,         ;DISALLOWED BUS ADDRESSES,
```

```
2634
2635
2636            011330                  BUFBEG=.                                ;BEGINNING OF TEXT BUFFER.
2637
2638
2639                                                                            ;ONCE ONLY CODE.
2640   011330   042706   000277          INIT:   BIC     #277,   SP              ;SET IO XX7500
2641   011334   010667   170312                  MOV     SP,     BOTTOM          ;COPY
2642   011340   016706   170344                  MOV     STACK0+2,SP             ;START STACK
2643   011344   012701   000064                  MOV     #64.,   AC              ;ZERO VECTORS
2644   011350   005021                           CLR     (AC)+
2645   011352   020167   170332                  CMP     AC,     STACK0+2        ;AND STACK
2646   011356   103774                           BLO     .-6
2647   011360   012737   010256   000100        MOV     #CINT,  @#100           ;SETUP CLOCK POINTER
2648   011366   012737   000340   000102        MOV     #340,   @#102           ;BR6
2649   011374   000137   003070                  JMP     @#ENIT2                 ;GO ERASE VARIABLE AREA AND START
2650
2651
```

```
2653            001                     .IFDF   GT40            ;GT40 FUNCTIONS
2654
2655                                    ; **DNEW**
2656
2657
2658                                    ;DNEW WRITTEN BY R. FRIEDENTHAL
2659                                    ;LAST EDIT:     2/25/73
2660
2661
2662                                    ;COPYRIGHT 1972,1973 DIGITAL EQUIPMENT CORPORATION
2663                                    ;MAYNARD, MASS  01754
2664
2665                                    ; THE FUNCTIONS IN DNEW ENABLE THE FOCAL-11 USER TO DRAW ON A
2666                                    ; GT40.
2667
2668
2669
2670                                    ;CONDITIONAL ASSEMBLY PARAMETERS
2671
2672
2673                                    ;ASSEMBLING WITH NO PARAMETERS DEFINED IN THE ASSEMBLY STRING,
2674                                    ;PRODUCES FOCAL-11 WHICH WILL RUN IN AN 11/05
2675
2676                                    ;ASSEMBLING WITH GT40 DEFINED PRODUCES AN 8K FOCAL-GT WITH NO
2677                                    ;LPS FUNCTIONS
2678
2679                                    ;ASSEMBLING WITH GT40 AND RT DEFINED PRODUCES AN 8K FOCAL-RT
2680                                    ;WITH A BUFFER 1000 WORDS LONG
2681
2682                                    ;ASSEMBLING WITH GT40, RT, AND NOBUFF DEFINED PRODUCES AN 8K FOCAL-GT
2683                                    ;WITH LPS FUNCTIONS AND AN FSAM THAT DOES NOT USE A BUFFER
2684
2685                                    ;THE USER MAY DEFINE STANDARD PARAMETERS IN HIS ASSEMBLY STRING
2686                                    ;BY DEFINING 'PARAM' IN THAT STRING FOLLOWED BY THE NEW PARAMETERS.
2687                                    ;IN THIS CASE, HE MUST DEFINE:
2688
2689                                    ;       KSIZE           THE SIZE OF CORE IN WORDS
2690                                    ;       DSIZE           THE SIZE OF THE DISPLAY FILE IN WORDS
2691                                    ;       TSIZE           THE SIZE OF THE TEXT PORTION OF THE DISPLAY FILE
2692
2693                                    ;IF THE USER DEFINES GT40, RT, AND PARAM, HE MUST ALSO DEFINE
2694                                    ;(IN ADDITION TO THE 3 PARAMETERS ABOVE):
2695
2696                                    ;       LPBSIZ          THE SIZE OF THE LPS BUFFER (IN WORDS)
2697
2698
2699                                    ;ASSEMBLING ANY OF THE ABOVE WITH 'TERM' DEFINED PRODUCES A VERSION THAT
2700                                    ;INCLUDES TERMINAL CODE
2701
2702
2703
2704
2705                                            .ENABLE AMA
2706                                            .ENABLE ABS
```

```
                                ;
                                ;THE FOLLOWING PARAMETERS DETERMINE THE SIZE OF DISPLAY FILE
                                ;RELATIVE TO FOCAL BUFFER AREA, THE PORTION OF THE DISPLAY
                                ;FILE DEVOTED TO TEXT INITIALLY, AND THE TYPE OF FOCAL TO PRODUCE.
                                ;THE USER WHO DEFINES 'PARAM' IN THE ASSEMBLY STRING MUST DEFINE
                                ;ALSO EACH OF THE PARAMETERS DEFINED BELOW.


     002                                .IFNDF  PARAMS  ;8K DEFINITIONS, DEFAULT DEFINITIONS
020000                                  KSIZE=20000     ;CORE SIZE (20000=8K WORDS)
     003                                .IFDF   RT      ;IF LPS CODE
     004                                .IFNDF  NOBUFF  ;IF FULL SUPPORT
                                        DSIZE=4000      ;SET DISPLAY FILE SIZE
                                        LPBSIZ=1762     ;1000 WORD INPUT BUFFER
     003                                .ENDC
     004                                .IFDF   NOBUFF  ;IF NON-BUFFERED SUPPORT,
                                        DSIZE=5000      ;MORE ROOM FOR DISPLAY FILE
                                        LPBSIZ=0        ;NO BUFFER
     003                                .ENDC
     002                                .ENDC
     003                                .IFNDF  RT      ;IF NO LPS SUPPORT, PRODUCE FOCAL-GT
005300                                  DSIZE=5300      ;WITH LARGE DIPSLAY FILE
000000                                  LPBSIZ=0        ;AND NO BUFFER
     002                                .ENDC
003000                                  TSIZE=3000      ;IN ALL CASES PORTION OF DISPLAY FILE DEVOTED
                                                        ;TO TEXT INITIALLY
     001                                .ENDC


                                ;       .IFDF   K4      ;4K DEFININTIONS (DEFINE PARAM +THESE VALUES IN ASSEMBLY STRING)
                                ;       KSIZE=10000     ;CORE SIZE
                                ;       DSIZE=1000      ;DISPLAY FILE SIZE
                                ;       TSIZE=700       ;TEXT FILE SIZE
                                ;       LPBSIZ=0        ;LPS BUFFER (NONE)
                                ;       .ENDC



                                ;THE FOLLOWING RECALCULATE SIZES TO MAKE THEM DIVISIBLE BY LOCS
                                ;WHICH CONSIST OF 6 BYTES EACH
001625                                  DSIZE=DSIZE+DSIZE/6
012576                                  DDSIZE=DSIZE*6  ;FINAL DISPLAY FILE SIZE
001000                                  TSIZE=TSIZE+TSIZE/6
006000                                  TTSIZE=TSIZE*6  ;FINAL TEXT PORTION
037770                                  KKSIZE=KSIZE+KSIZE-10   ;FINAL CORE SIZE
003124                                  ENAB=3124               ;ENABLE BITS+INTENSITY 4
                                ; THE FOLLOWING DEFINE GT40 INSTRUCTIONS USED BY DNEW
164000                                  DNOP=164000             ;DISPLAY NOP
173400                                  DSTP=173400             ;DISPLAY STOP, WITH INTERRUPT
160000                                  DJMP=160000             ;DISPLAY JUMP(DESTINATION FOLLOWS IN FILE)
```

```
                        114000                          DPT=114000              ;POINT MODE
                        100000                          DCHR=100000             ;CHARACTER MODE
                        110000                          DVEC=110000             ;LONG VECTOR MODE

                        000211                          RTPAR=211               ;A RIGHT PARENTHESIS
                        000012                          LF=12                   ;LINE FEED CHARACTER
                        000015                          CR=15                   ;CARRIAGE RETURN
                        000007                          BELL=7                  ;^G RINGS THE BELL
                        000000                          LOC=TEMP                ;MAKE LOC A GENERAL REGISTER
                        104660                          EVALU=104400+260        ;TRAP TO EVALUATE NEXT ARG
                        104614                          GETCH=104400+214        ;GET NEXT CHARACTER FROM FUNCTION
                                                ; GT DEVICE REGISTERS:
                        172000                          GTPC=172000
                        172002                          GTSTAT=GTPC+2           ;GT STATUS
                        172004                          GTX=GTSTAT+2            ;GT X COORDINATE
                        172006                          GTY=GTX+2               ;GT Y COORD
                        170404                          CLKIN=170404            ;LPS CLOCK STATUS
                        170400                          ADIN=170400             ;LPS A/D STATUS
                        170402                          LEDOUT=ADIN+2           ;LPS BUFFER AND LED OUTPUT
                        170406                          PRESET=170406           ;LPS PRESET REGISTER
                        000134                          CLKADR=134              ;LPS CLOCK INTERRUPT VECTOR

                        000320                          STPADR=320              ;STOP INTERRUPT ADDRESS
                        000300                          TENADR=300              ;TEN INPUT VECTOR
                        175610                          TENIN=175610            ;TEN INPUT STATUS
                        175614                          TENOUT=175614           ;TEN OUTPUT STATUS
                        000324                          LPADR=STPADR+4                  ;LITE PEN IN, ADDRESS


                                                ;       .MACRO RESUME                   ;RESTART THE DISPLAY,
                                                ;       MOV     #1,@#GTPC               ;BY TRYING TO MOVE A 1 INTO GTPC
                                                ;       .ENDM


                        011330                          .=INIT          ;NEW INIT CODE
011330  013706  001710                          MOV     STACKO+2,SP             ;SET UP STACK POINTER
011334  012701  000064                          MOV     #64,AC                  ;GET START OF VECTOR AREA,
011340  005021                                  CLR     (AC)+                   ;AND CLEAR UNTIL,
011342  020137  001710                          CMP     AC,STACKO+2             ;BASE OF STACK
011346  103774                                  BLO     .-6
011350  012737  010256  000100                  MOV     #CINT,@#100             ;INSERT CLOCK VECTOR
011356  012737  000340  000102                  MOV     #340,@#102              ;AND STATUS
011364  012737  024634  000320                  MOV     #STPINT,@#STPADR                ;STOP VECTOR,
011372  012737  000340  000322                  MOV     #340,@#STPADR+2
011400  012737  024526  000324                  MOV     #LPINT,@#LPADR          ;LITE PEN VECTOR
011406  012737  000340  000326                  MOV     #340,@#LPADR+2
                           002                  .IFDF   TERM                    ;IN TERMINAL SETUP,
011414  012737  023116  000300                  MOV     #TENINT,@#TENADR        ;SET UP TEN INPUT INTERRUPT
011422  012737  000340  000302                  MOV     #340,@#TENADR+2
                           001                  .ENDC
                           002                  .IFDF   RT              ;IF LPS SUPPORT
                                                MOV     #CLKINT,@#CLKADR        ;SETUP CLOCK INTERRUPT
                                                MOV     #340,@#CLKADR+2
                           001                  .ENDC
```

```
2815  011430  004737  023166                  JSR     PC,INITD                ;GO SET UP DISPLAY FILE AND
2816  011434  000137  003070                  JMP     @#ENII2                 ;THEN GO START
2817                                                  ;HOWEVER, ERASEV ERASES THE JMP @#INII2
2818                                                  ;BEFORE IT GETS TO EXECUTE, FOCAL STARTS ANYWAY
2819                                                  ;BUT HOW?
2820
2821
2822          002102                          .=STARTX+46             ;CHANGE FOCAL'S HELLO,
2823  002102  104443                          104443                  ;TO # SIGN
2824
2825          010110                          .=KINT                  ;CHANGE KEYBOARD INPUT ROUTINE
2826  010110  013746  177562                  MOV     @#TKS+2,-(SP)   ;GET CHAR
2827  010114  121627  000203                  CMPB    @SP,#203        ;TEST FOR ^C
2828  010120  001006                          BNE     KINT3           ;NO
2829  010122  105137  001601                  COMB    CCFLG           ;YES, UPDATE FLAG
2830  010126  001005                          BNE     KINT4           ;TWO?
2831  010130  004737  023166                  JSR     PC,INITD        ;YES RESET SCREEN,
2832  010134  104601                          ERROR+201+0,+0,         ;AND FOCAL
2833  010136  105037  001601          KINT3:  CLRB    CCFLG
2834  010142  005737  001636          KINT4:  TST     KIN             ;ANY ROOM?
2835  010146  000137  024472                  JMP     TSTERR          ;GO DO EXCESS CODE
2836
2837
2838             002                          .IFDF TERM              ;IF TERMINAL HOOKUP,
2839          010110                          .=KINT                  ;1ST GO CHECK FOR ^F AND ^T
2840  010110  000137  023016                  JMP     K
2841             001                          .ENDC
2842
2843
2844          002016                          .=ERR2+44               ;ON ANY ERROR,
2845  002016  000137  024504                  JMP     RESETD          ;RESTART THE DISPLAY
2846
2847
2848
2849
2850
2851
2852          000054                          NUMF=54                 ;SIZE OF ADDITION TO FUNCTION LIST
2853
2854             002                          .IFDF   RT              ;FOR LPS ADDITION
2855                                          NUMF=NUMF+14            ;INCREASE THIS SIZE
2856             001                          .ENDC
2857
2858
2859
2860
2861
2862
2863          001710                          .=STACKO+2
2864  001710  001024                          1100-NUMF               ;SHORTEN STACK IN OVERFLOW ROUTINE
2865          001024                          .=1100-NUMF             ;INSERT NEW FUNCTIONS
2866  001024  000000                          0                       ;END DELIMETER
2867  001026  000000                          0
2868             002                          .IFDF   RT                      ;LPS FUNCS
```

```
2869
2870
2871
2872
2873
2874
2875                001
2876   001030   024130
2877   001032   013462
2878   001034   024064
2879   001036   002740
2880   001040   024004
2881   001042   014153
2882   001044   024000
2883   001046   014133
2884   001050   023700
2885   001052   013467
2886   001054   023364
2887   001056   014164
2888   001060   023514
2889   001062   014054
2890   001064   023274
2891   001066   014030
2892   001070   023264
2893   001072   002764
2894   001074   023254
2895   001076   013742
2896   001100   023244
2897   001102   014067
2898
2899            001774
2900            000000
2901            000000
2902               002
2903            000150
2904            002004
2905               001
2906               002
2907               003
2908
2909               002
2910               003
2911
2912               002
2913
2914
2915               001
2916            002154
2917
2918            023016
2919            001652
2920   001652   023014
2921
2922            025172
```

```
XTIC                    ;FTIC(ARG)      =RETURNS OR SETS TIME
14047
XSAM                    ;FSAM(CHAN)     =RETURNS A/D OF CHANNEL
14001
XLED                    ;FNUM(ARG)      =LOADS LEDS WITH ARG
13630
.ENDC
XCLR                    ;FCLR()         =CLEAR GRAPHICS FILE
13462
XLP                     ;FLP(LOC)       =RETURNS LAST LP HIT(DUMMY ARG)
2740
XYCO                    ;FYCO(LOC)      =RETURNS Y COORD OF LOC
14153
XXCO                    ;FXCO(LOC)      =RETURNS X COORD OF LOC
14133
XDIS                    ;FDIS(LINE,INT,BLINK,LP)
13467                   ;       =SET MODE OF NEXT ENTRY
XTXT                    ;FTXT(LOC,ABCD) =INSERT UP TO
14164                   ;4 CHAR,S OF TEXT
XSKP                    ;FSKP(LOC [,LOC2])      =INSERTS A JMP
14054                   ;TO LOC2 AT LOC OR TO TEXT IF NO LOC2
XSET                    ;FSET(LOC,X,Y)=INSERTS INVIS ABSO,, POINT
14030
XPT                     ;FPT(LOC,X,Y)   =INSERT VIS, ABS, PT,
2764
XMOV                    ;FMOV(LOC,X,Y)  =INSERTS INV, VECTOR
13742
XVEC                    ;FVEC(LOC,X,Y)  =INSERTS VIS, LONG VECTOR
14067
                        ;IF SO, OUTPUT LINE NUMBERS, ETC,
GSIZE=1774              ;SIZE OF NEW GT40 CODE
TERSIZ=0
LPSSIZ=0
.IFDF   TERM            ;IF TERMINAL CODE INCLUDED,
TERSIZ=150              ;DEFINE ITS SIZE,
GSIZE=GSIZE+10  ;AND ALLOW FOR TERMINAL CODE IN MAIN
.ENDC
.IFDF   RT              ;IF LPS FUNCTONS IN,
.IFNDF  NOBUFF          ;IF FULL SUPPORT,
LPSSIZ=450              ;DEFINE ITS SIZE,
.ENDC
.IFDF   NOBUFF          ;IF NO BUFFER SUPPORT,
LPSSIZ=250              ;LESS CODE USED
.ENDC
LPSSIZ=LPSSIZ+LPBSIZ+LPBSIZ     ;ADD IN THE BUFFER
GSIZE=GSIZE+10          ;EXTEND MAIN CODE SIZE
.ENDC
FSIZE=GSIZE+TERSIZ+LPSSIZ       ;SIZE OF DNEW

DBOT=KKSIZE=FSIZE=DDSIZE        ;END OF FOCAL=CORE SIZE=LOADER SIZE=FNE=
.=BOTTOM                ;THE GRAPHICS FILE STARTS WHERE FOCAL ENDS
DBOT=2

.=DBOT+FSIZE            ;DISPLAY FILE FOLLOWS FUNCTIONS
```

```
2923  025172  173660                  DFILE:  .WORD   173660          ;ORIGINAL STATUS: NOITALICS, GLOW
2924                                                                  ;AND STOP
2925  025174  117124                          .WORD   DPT+ENAB        ;POINT MODE+ENABLE BITS
2926  025176  000000                          .WORD   0               ;SET START OF GRAPHICS TO (0,0)
2927  025200  000000                          .WORD   0
2928  025202  114000                  LOC0:   .WORD   DPT             ;START OF USER FILE; STARTS WITH POINT
2929  025204  000000                          .WORD   0               ;USER MAY CHANGE IT
2930  025206  000000                          .WORD   0               ;LIGHT PEN HITS SHOW UP IN THESE CO-
2931                                                                  ;ORDINATES.
2932  025210  160000                          .WORD   DJMP            ;INITIALLY, JUMPS TO TEXT FILE FOLLOWS
2933  025212  031772                          .WORD   TEXT
2934          031772                          .=LOC0-6+DDSIZE-TTSIZE  ;LEAVE ROOM FOR GRAPHICS,
2935  031772  117124                  TEXT:   .WORD   DPT+ENAB        ;AND START TEXT WITH PT, AT (0,700)
2936  031774  000000                          .WORD   0
2937  031776  001360                          .WORD   1360
2938  032000  100000                          .WORD   DCHR            ;CHARACTER MODE
2939                                          ;FILE EXTENDS UP INTO TOP OF CORE, ENDS WITH A STOP
2940
2941                                  ; THE FUNCTIONS START HERE, WHERE FOCAL STORAGE AREA ENDS
2942          023166                          .=DBOT+TERSIZ+LPSSIZ    ;MAIN CODE STARTS AFTER LPS AND TERMINAL CODE, IF ANY
2943
2944
2945                                  ;INITD SETS UP THE DISPLAY FILE EACH RESTART
2946  023166  012737  025202  024656  INITD:  MOV     #LOC0,LPLOC     ;SET LPLOC TO LOC0
2947  023174  005037  024662                  CLR     MODE
2948             002                          .IFDF   TERM            ;IN TERMINAL HOOKUP,
2949  023200  005037  023156                  CLR     TOHOST          ;CLEAR FLAGS, START IN FOCAL
2950             001                          .ENDC
2951             002                          .IFDF   RT              ;FOR LPS,
2952                                          TST     DTIME           ;IF TIMER 0,
2953                                          BEQ     NOLPS           ;DON'T WRITE INTO DEVICE REGISTER
2954                                          CLR     @#CLKIN         ;ELSE, STOP CLOCK,
2955                                          CLR     DTIME           ;AND CLEAR TIMER
2956             001                          .ENDC
2957  023204  005037  024664          NOLPS:  CLR     STPFLG          ;LET DISPLAY STOP
2958  023210  012700  025202                  MOV     #LOC0,LOC       ;PTR TO DISPLAY FILE
2959  023214  012720  114000                  MOV     #DPT,(LOC)+     ;START WITH A PT INSTR,
2960  023220  005020                          CLR     (LOC)+          ;(0,0) COORDS
2961  023222  005020                          CLR     (LOC)+
2962  023224  012737  031772  024666          MOV     #LOC0-6+DDSIZE-TTSIZE,TXTPTR    ;LEAVE CELLS FOR GRAPHICS
2963  023232  004737  024144                  JSR     PC,TJUMPS               ;FILL GRAPHICS FILE
2964  023236  004737  024362                  JSR     PC,TCLEAR       ;SET UP TEXT FILE
2965                                                                  ;WITH JUMPS TO TEXT FILE
2966                                                                  ;AND START DISPLAY
2967  023242  000207                          RTS     PC              ;AND RETURN TO START CODE IN FOCAL
2968                                  ;THE 4 GRAPHICS FUNCTIONS: ALL USE THE SAME CODE TO PICK UP
2969                                  ; THE CORRECT LOC,S AND COORD,S AND TO
2970                                  ; INSERT THEM,
2971                                  ;FVEC: VISIBLE VECTOR
2972  023244  052737  110000  024662  XVEC:   BIS     #DVEC,MODE              ;MODE MAY CONTAIN MODES PUT
2973                                                                  ;THERE BY FDIS
2974  023252  000414                          BR      VIS             ;GO SET VISIBLE BIT
2975                                  ;FMOV, INVISIBLE VECTOR
2976  023254  052737  110000  024662  XMOV:   BIS     #DVEC,MODE              ;ALSO LONG VECTOR
```

```
2977  023262  000414                          BR      INV             ;BUT INVISIBLE, DON'T SET BIT
2978                                ;FPT, VISIBLE ABSOLUTE POINT
2979  023264  052737  114000  024662  XPT:    BIS     #DPT,MODE
2980  023272  000404                          BR      VIS
2981                                ;FSET, INVISIBLE ABSOLUTE POINT
2982  023274  052737  114000  024662  XSET:   BIS     #DPT,MODE
2983  023302  000404                          BR      INV
2984  023304  012737  040000  024660  VIS:    MOV     #40000,HIBITS
2985  023312  000402                          BR      GRALL
2986  023314  005037  024660          INV:    CLR     HIBITS          ;HIBITS ALREADY CLEAR, BUT MAKE SURE
2987  023320  004737  024176          GRALL:  JSR     PC,LOCGO        ;GET LOCATION OF 1ST COORD (NO STOP)
2988  023324  013720  024662                  MOV     MODE,(LOC)+     ;SET MODE
2989  023330  004737  023344                  JSR     PC,PLUGCO       ;PUT X COORDINATE INTO FILE
2990  023334  004737  023344                  JSR     PC,PLUGCO       ;PUT Y COORDINATE IN
2991  023340  000137  024312                  JMP     RETNEW          ;RETNEW STARTS DISPLAY AND EXITS
```

```
 2993
 2994                                   ;PLUGCO PICKS UP THE NEXT ARG, PUTS IT IN DISPLAYABLE
 2995                                   ; FORMAT, AND ENTERS IT IN THE CURRENT CELL
 2996                                   ;   OR CALL PLUGCO ONCE AND START IT WITH JSR PC,PLUGCO+4
 2997  023344  010046                   PLUGCO: MOV     LOC,-(SP)       ;SAVE LOC ON STACK
 2998  023346  104660                           EVALU                   ;GET THE ARG
 2999  023350  007071                           FPINT           ;INTO AC
 3000  023352  012600                           MOV     (SP)+,LOC       ;FETCH LOC
 3001  023354  004737  024264                   JSR     PC,PUTCO                ;PUTCO FORMATS IT INTO AC
 3002  023360  010120                           MOV     AC,(LOC)+               ;PLUG IT IN
 3003  023362  000207                           RTS     PC
 3004
 3005                                   ;FTXT WILL INSERT UPTO 4 CHAR,S INTO A LOC,,
 3006                                   ; FTXT(LOC,C1,C2,C3,...)
 3007                                   ;LESS THAN 4 ARGS INSERTS NULLS; MORE THAN 4 GOES ON TO NEXT LOC,,
 3008                                   ;FTXT WILL MAKE ALL VALUES >40 BY ADDING 100 TO THEM IF NECESSARY
 3009  023364  004737  024164           XTXT:   JSR     PC,LOCSET       ;GET LOC, INTO LOC
 3010  023370  020037  024666           MORTXT: CMP     LOC,TXTPTR      ;IF BEYOND GRAPHICS FILE,
 3011  023374  002402                           BLT     PUTTXT
 3012  023376  162700  000006                   SUB     #6,LOC          ;SET LOC BACK 1 LOC
 3013  023402  012710  100000           PUTTXT: MOV     #DCHR,(LOC)     ;INSERT CHARACTER MODE
 3014  023406  053720  024662                   BIS     MODE,(LOC)+     ;AND MODE BITS
 3015  023412  012737  177774  024660           MOV     #-4,HIBITS      ;HIBITS COUNTS OFF 4 CHAR,S
 3016  023420  120427  000211           ASCII:  CMPB    CHAR,#RTPAR     ;NO MORE ARGS?
 3017  023424  001002                           BNE     NOTYET
 3018  023426  005001                           CLR     AC              ;YES, INSERT ZEROES,
 3019  023430  000412                           BR      JUST0           ;AND STOP GETTING ARG,S
 3020
 3021  023432  010046                   NOTYET: MOV     LOC,-(SP)       ;SAVE LOC AROUND EVAL
 3022  023434  104660                           EVALU
 3023  023436  007071                           FPINT                   ;NEXT ARG INTO AC
 3024  023440  120127  000040           MAKSUR: CMPB    AC,#40          ;THEN ADD #100 TO VALUES <40
 3025  023444  002003                           BGE     DUNASC
 3026  023446  062701  000100                   ADD     #100,AC
 3027  023452  000772                           BR      MAKSUR
 3028  023454  012600                   DUNASC: MOV     (SP)+,LOC       ;RETRIEVE LOC
 3029  023456  110120                   JUST0:  MOVB    AC,(LOC)+       ;INSERT THE CHARACTER
 3030  023460  005237  024660                   INC     HIBITS
 3031  023464  002755                           BLT     ASCII           ;GO GET NEXT 4
 3032  023466  120427  000211                   CMPB    CHAR,#RTPAR     ;END OF ARG,S REACHED?
 3033  023472  001336                           BNE     MORTXT          ;NOPE DO ANOTHER LOC
 3034  023474  010001                           MOV     LOC,AC          ;YES, RETURN NEXT LOC
 3035  023476  062701  000004                   ADD     #4,AC           ;MAKE SURE AC POINTS IN NEXT LOC
 3036  023502  012737  000001  172000           MOV     #1,@#GTPC       ;RESUME DISPLAY
 3037  023510  000137  024110                   JMP     RETLOC          ;VIA XLP CODE
 3038
 3039
 3040                                   ;FSKP INSERTS A BLOCK CONTAINING A JMP TO SOMEHWERE IN THE GRAPHICS
 3041                                   ; FSKP(LOC) INSERT A JMP TO THE TEXT FILE AND CLEARS THE TEXT FILE;
 3042                                   ; FSKP(LOC1,LOC2) INSERTS A JUMP TO LOC2
 3043  023514  004737  024164           XSKP:   JSR     PC,LOCSET       ;GET 1ST LOC, INTO LOC
 3044  023520  052737  100000  024662           BIS     #100000,MODE    ;MAKE MODE INTO CHAR MODE WORD
 3045  023526  013720  024662                   MOV     MODE,(LOC)+     ;INSERT MODE OF JUMP
 3046  023532  012720  160000                   MOV     #DJMP,(LOC)+    ;INSERT JUMP
```

```
3047  023536  010046                          MOV     LOC,-(SP)       ;SAVE LOC,
3048  023540  013737  024666  024660          MOV     TXTPTR,HIBITS   ;SAVE OLD TEXT START
3049  023546  120427  000211                  CMPB    CHAR,#RTPAR     ;RIGHT PAR, MEANS NO 2ND ARG
3050  023552  001433                          BEQ     TXTJMP          ;WHICH MEANS JUMP TO TEXT
3051  023554  104660                          EVALU                   ;GET 2ND ARG
3052  023556  005737  001620                  TST     HORD            ;ON ARG <0,
3053  023562  002411                          BLT     NOTEXT          ;JUMP TO START OF DISPLAY FILE
3054  023564  123727  001616  000014          CMPB    BE,#14          ;ON POTENTIAL INTEGER OVERFLOW,
3055  023572  003020                          BGT     LEAST           ;USE END OF TEXT
3056  023574  007045                          FPMUL+IMMED             ;CALCULATE 2ND LOCATION
3057  023576  000003                  FP6:    3                       ;LOC2*3 WORDS
3058  023600  060000                          60000
3059  023602  007071                          FPINT                   ;MAKE AN INTEGER
3060  023604  000402                          BR      NOTBAK
3061  023606  012701  177770          NOTEXT: MOV     #-10,AC         ;JUMP TO START OF FILE
3062  023612  062701  025202          NOTBAK: ADD     #LOC0,AC        ;+START OF FILE(LOC0)
3063  023616  102406                          BVS     LEAST           ;ON OVERFLOW, LOC2 TOO HIGH
3064  023620  020137  024666                  CMP     AC,TXTPTR       ;IF 2ND LOC OVERLAPS TEXT,
3065  023624  003421                          BLE     INFILE
3066  023626  020127  037662                  CMP     AC,#LOC0-6+DDSIZE-110   ;TEST FOR BEYOND WHOLE FILE
3067  023632  002406                          BLT     NOTOUT
3068  023634  012737  037662  024666  LEAST:  MOV     #LOC0-6+DDSIZE-110,TXTPTR       ;TEXT ONLY ONE LINE
3069  023642  013701  024666          TXTJMP: MOV     TXTPTR,AC       ;IF OUT, MAKE ITA JUMP TO START OF TEXT
3070  023646  000402                          BR      WAYOUT
3071  023650  010137  024666          NOTOUT: MOV     AC,TXTPTR       ;CLEAR OUT TEXT FILE
3072  023654  013700  024660          WAYOUT: MOV     HIBITS,LOC      ;SET UP LOC FOR TJUMPS WITH OLD TEXT START
3073  023660  004737  024144                  JSR     PC,TJUMPS       ;ADD JUMPS FROM OLD START TO NEW START
3074  023664  004737  024362                  JSR     PC,TCLEAR       ;STARTING AT 2ND LOC, CLEAR TEXT
3075  023670  012600                  INFILE: MOV     (SP)+,LOC       ;GET LOC FROM STACK
3076  023672  010110                          MOV     AC,(LOC)        ;AND INSERT THE JUMP ADDR,(NO 3RD WORD)
3077  023674  000137  024312                  JMP     RETNEW          ;DONE, GO RETURN NEXT FREE LOC
3078
3079
3080
3081                                  ;FDIS SETS UP MODE WITH THE SPECIFIED MODES, THE NEXT ENTRY IN THE
3082                                  ;FDIS SETS UP MODE WITH THE SPECIFIED MODES,  THE NEXT ENTRY IN THE
3083                                  ;GRAPHICS FILE USES THIS MODE, THEN CLEARS IT,
3084                                  ;FDIS(LINETYPE[0-3],INTENSITY[0-7],BLINK[0-1],L,P,[0-1])
3085  023700  007071                  XDIS:   FPINT                   ;GET LINE TYPE INTO LOW BITS
3086  023702  010137  024662                  MOV     AC,MODE         ;ENTER THE LINE TYPE
3087  023706  042737  177770  024662          BIC     #177770,MODE    ;MAKE SURE NO OVERFLOW
3088  023714  104660                          EVALU                   ;GET INTENSITY
3089  023716  007071                          FPINT
3090  023720  000301                          SWAB    AC              ;PUT IT IN BITS 7,8,9
3091  023722  006201                          ASR     AC
3092  023724  042701  176177                  BIC     #176177,AC      ;NO OVERFLOW
3093  023730  050137  024662                  BIS     AC,MODE         ;ENTER IN MODE
3094  023734  104660                          EVALU                   ;GET BLINK
3095  023736  005737  001620                  TST     HORD            ;POS HORD MEANS BLINK
3096  023742  001403                          BEQ     NOBLI
3097  023744  052737  000010  024662          BIS     #10,MODE                ;SET BLINK BIT
3098  023752  104660                  NOBLI:  EVALU                   ;GET LITE PEN SENSITIVITY
3099  023754  005737  001620                  TST     HORD
3100  023760  001403                          BEQ     NOLP
```

```
3101  023762  052737  000040  024662          BIS     #40,MODE        ;SET LP SENSITIVE BIT
3102  023770  052737  002124  024662  NOLP:   BIS     #2124,MODE      ;SET ALL ENABLE BITS
3103  023776  000207                          RTS     PC              ;RETURN TO FOCAL
3104
3105
3106                                  ;FXCO AND FYCO RETURN X AND Y COORDINATES OF LOC, SPECIFIED IN CALL
3107
3108  024000  005046                  XXCO:   CLR     -(SP)           ;SAVE OFFSET OF COORD, IN STACK
3109  024002  000402                          BR      COCO
3110  024004  012746  000002          XYCO:   MOV     #2,-(SP)        ;Y COORD, ONE WORD DOWN
3111  024010  004737  024204          COCO:   JSR     PC,LOCGET       ;GET LOC; INTO LOC(+4 TO NOT STOP DISPLAY)
3112  024014  005001                          CLR     AC              ;SET UP WORD MANTISSA
3113  024016  022027  160000                  CMP     (LOC)+,#DJMP    ;ON DJMP IN THIS SLOT,
3114  024022  002015                          BGE     OUTCO           ;RETURN 0
3115  024024  062600                          ADD     (SP)+, LOC      ;WITH OFFSET,LOC POINTS TO COORD
3116  024026  011001                          MOV     (LOC),AC        ;GET COORD IN DISPLAY FORMAT
3117  024030  032701  020000                  BIT     #20000,AC       ;TEST NEGATIVE BIT
3118  024034  001405                          BEQ     POSOUT          ;IF POSITIVE, EVERYTHING OK
3119  024036  011001                          MOV     (LOC),AC        ;IF NEG, MAKE INTO A 16 BIT NEGATIVE
3120  024040  042701  060000                  BIC     #60000,AC       ;GET RID OF NEG BIT AND VIS-INV BIT,
3121  024044  005401                          NEG     AC              ;AND NEGATE
3122  024046  000403                          BR      OUTCO
3123  024050  051001                  POSOUT: BIS     (LOC),AC        ;AC=0 ON POS, COORD
3124  024052  042701  040000                  BIC     #40000,AC       ;CLEAR VIS-INV, BIT
3125  024056  004737  024340          OUTCO:  JSR     PC,FLOAT        ;FLOAT COORD INTO FLAC
3126  024062  000207                          RTS     PC              ;AND RETURN TO FOCAL
3127
3128
3129                                  ;XLP RETURNS LOC,, OF LAST LP HIT; THE INTERRUPT HAS PUT
3130                                  ; THIS LOCATION INTO LPLOC
3131  024064  007071                  XLP:    FPINT                   ;GET ARG
3132  024066  005701                          TST     AC
3133  024070  001002                          BNE     RETNOW          ;ON ,EQ, 0,
3134  024072  005037  024656                  CLR     LPLOC           ;WAIT FOR LP HIT
3135  024076  005737  024656          RETNOW: TST     LPLOC           ;BEFORE RETURNING
3136  024102  001775                          BEQ     RETNOW
3137  024104  013701  024656                  MOV     LPLOC,AC        ;GET ADDRESS
3138  024110  162701  025206          RETLOC: SUB     #LOC0+4,AC      ;MAKE INTO AN OFFSET FROM LOC0
3139  024114  004737  024340                  JSR     PC,FLOAT        ;FLOAT INTO FLAC
3140  024120  007030  023576                  FPDIV+DIRECT,FP6                ;/3 TO GET LOC FOR USER
3141  024124  007071                          FPINT
3142  024126  000207                          RTS     PC              ;BACK TO FOCAL
3143
3144
3145                                  ;XCLR CLEARS THE SCREEN OF GRAPHICS
3146
3147  024130  004737  024164          XCLR:   JSR     PC,LOCSET       ;GET ARG AND STOP DISPLAY
3148  024134  004737  024144                  JSR     PC,TJUMPS       ;CLEAR THE SCREEN UP TO TEXT POINTER
3149  024140  000137  024312                  JMP     RETNEW          ;RETURN ARG+1 AND START DISPLAY
3150
3151          024144                          GEND=.                  ;INSERT LPS CODE NOW, IF ANY
3152
3153
3154
```

```
3155
3156
3157          002                   .IFDF   RT              ;LPS FUNCTIONS
3158                                .=DBOT                  ;START THESE AT BOTTOM OF DNEA SO PATHCABLE OUT
3159
3160          003                   .IFDF   NOBUFF
3161
3162                        ;XSAM SAMPLES THE ANALOG CHANNEL CONTAINED IN THE FUNCTION ARG
3163
3164
3165                        XSAM:   FPINT                   ;GET CHANNEL INTO AC,
3166                                SWAB    AC              ;HIGH BYTE
3167                                INCB    AC      ;SET ENABLE BIT(****OR SET BIT 4 FOR SCHMITT)
3168                                MOV     AC,@#ADIN       ;INITIATE A/D
3169                        NOAD:   TSTB    @#ADIN          ;DONE?
3170                                BGT     NOAD            ;IF NOT, WAIT
3171                                MOV     @#ADIN+2,AC     ;GET THE VALUE
3172                                JSR     PC,FLOAT        ;INTO FLAC
3173                                RTS     PC
3174          002                   .ENDC
3175
3176          003                   .IFNDF  NOBUFF          ;IF BUFFERED FSAM,
3177
3178                                LPSBUF=.                ;START OF BUFFER
3179                                .=.+LPBSIZ+2+LPBSIZ     ;LEAVE ROOM FOR BUFFER
3180
3181                        ;XSAM STARTS AD CONVERSIONS DEPENDING UPON THE
3182                        ;TYPE AND NUMBER OF ARGNUMENTS.
3183                        ;ONE ARG POSITIVE DOES ONE SAMPLE AND RETURNS ITS VALUE,
3184                        ;ONE ARG NEGATIVE RETURNS -NTH SAMPLE IN BUFFER
3185                        ;TWO ARGS, 2ND POSITIVE SAMPLES 2ND NUMBER OF POINTS INTO BUFFER
3186                        ;TWO ARGS, 2ND NEGATIVE SAMPLES 2ND NUMBER OF POINTS BUT WAITS FOR
3187                        ;       SCHMITT TRIGGER BEFORE STARTING
3188                        ;THREE ARGS SAMPLES TWO CHANNELS ON EACH TIC,1ST ARG AND 3RD ARG
3189
3190                        XSAM:   FPINT                   ;GET 1ST ARG
3191                                TST     -(SP)           ;USE STACK AS TEMPORARY
3192                                NOP                     ;****DEBUG
3193                                MOV     AC,LOC          ;ON NEG. ARG
3194                                NEG     LOC             ;MAKE AN INDEX INTO BUFFER,
3195                                ASL     LOC
3196                                BGT     RETDAT          ;AND JUST RETURN DATA
3197                                BIC     #177770,AC      ;ELSE
3198                                MOV     AC,(SP)         ;SAVE CHANNEL
3199                                SWAB    AC              ;AND MAKE AD CONVERSION WORD,
3200                                INC     AC              ;WITH START BIT ON
3201                                MOV     AC,LPREG
3202                                MOV     #2,LOC          ;SET LOC TO 1 POINT
3203                                CMPB    CHAR,#RTPAR     ;AND IF NO MORE ARGS,
3204                                BEQ     SAMLOP          ;GO SAMPLE ONCE
3205                                MOVB    #40,LPREG       ;ELSE SET CLOCK OFLOW BIT
3206                                EVAL!!
3207                                FPINT                   ;AND GET NUMBER OF POINTS
3208                                MOV     AC,LOC          ;IF <=0 POINTS,
```

```
3209                                    BGT     NOSH
3210                                    NEG     LOC             ;MAKE POSITIVE,
3211                                    CLR     @#CLKIN         ;STOP CLOCK,
3212                                    MOV     #20506,@#CLKIN  ;AND MAKE CLOCK WAIT FOR SCHMITT
3213                            NOSH:   CMP     LOC,#LPBSIZ     ;ON MORE POINTS THAN SPACE
3214                                    BLE     SIZOK
3215                                    MOV     #LPBSIZ,LOC     ;DECREASE TO MAXIMUM
3216                            SIZOK:  ASL     LOC             ;MAKE AN INDEX
3217                                    BEQ     ABAD            ;ON 0 POINTS REQUESTED, JUST RETURN
3218                                    CMPB    CHAR,#RTPAR
3219                                    BEQ     SAMLOP
3220                                    MOV     LOC,-(SP)       ;SAVE LOC AROUND ARITHMETIC
3221                                    EVALU                   ;NOW GET 2ND CHANNEL, IF ANY
3222                                    FPINT
3223                                    MOV     (SP)+,LOC
3224                                    BIC     #177770,AC
3225                                    MOV     AC,(SP)         ;PUT IT OVER 1ST CHANNEL IN STACK
3226                            SAMLOP: MOV     LPREG,@#ADIN    ;START CONVERSIONS
3227                            NOAD:   TSTB    @#ADIN          ;WAIT FOR DONE FLAG
3228                                    BGE     NOAD
3229                                    MOV     @#ADIN+2,LPSBUF(LOC)    ;ENTER VALUE IN BUFFER
3230                                    DEC     LOC             ;DECR, COUNTER
3231                                    DEC     LOC
3232                                    BLE     DUNAD           ;TILL EXIT
3233                                    NOP                     ;****DEBUG
3234                                    CMPB    @#ADIN+1,(SP)   ;IF 2ND CHANNEL DIFFERENT THAN 1ST,
3235                                    BEQ     SAMLOP
3236                                    MOVB    (SP),@#ADIN+1   ;ASK FOR AD RIGHT NOW
3237                                    INC     @#ADIN          ;BY SETTING START BIT
3238                                    BR      NOAD            ;THEN GO GET THE VALUE
3239                            DUNAD:  CLR     @#ADIN          ;STOP A/D'S
3240                                    NOP                     ;****DEBUG
3241                                    TST     (LOC)+          ;POINT INDEX TO LAST A/D
3242                            RETDAT: MOV     LPSBUF(LOC),AC  ;GET THE VALUE,
3243                                    JSR     PC,FLOAT        ;INTO THE FLAC
3244                            ABAD:   TST     (SP)+           ;AND RESET THE STACK
3245                                    RTS     PC              ;A BEAUTIFUL JOB,
3246
3247                            LPREG:  .WORD   0               ;CONTAINS SAMPLING WORD FOR DEVICE REG
3248
3249                    002             .ENDC
3250
3251
3252
3253
3254                            ;XLED TAKES THE VALUE IN FLAC AND PUTS IT IN THE LEDS
3255
3256                            XLED:   CLR     HIBITS          ;CLEAR A COUNTER
3257                                    TST     HORD            ;ON NEGATIVE VALUE,
3258                                    BLT     NEGNUM
3259                                    MOV     #2413,@#LEDOUT
3260                                    BR      NUMLOP
3261                            NEGNUM: FPABS                   ;MAKE IT POSITIVE,
3262                                    MOV     #2415,@#LEDOUT  ;SET TOP LED TO '-'
```

```
3263                    NUMLOP: FPINT           ;GET VALUE
3264                            MOV     AC,-(SP)        ;ONTO STACK
3265                            FPDIV+IMMED             ;DIVIDE VALUE BY 10
3266                    FP10:   4                       ;FP 10
3267                            050000
3268                            JSR     PC,XITR         ;GET INTEGER PART
3269                            FPMUL+DIRECT,FP10       ;AND RESTORE LESS REMAINDER
3270                            FPINT                   ;GET THIS VALUE
3271                            SUB     AC,(SP)         ;AND FIND DIFFERENCE
3272                            ADD     HIBITS,(SP)     ;ADD DIGIT NOW SHOWN
3273                            MOV     (SP)+,@#LEDOUT  ;AND DISPLAY NUMERAL
3274                            FPDIV+DIRECT,FP10       ;DIVIDE TO GET NEW VALUE
3275                            INCB    HIBITS+1        ;BOP COUNTER
3276                            CMPB    HIBITS+1,#4     ;AND DO ,
3277                            BLE     NUMLOP          ;5 DIGITS
3278                            RTS     PC
3279
3280
3281
3282                    ;XTIC SETS THE CLOCK,  ON POSITIVE ARG, IT MAKES THAT MANY
3283                    ;1/100'S THE INTERVAL, AND SETS THE TIME TO 0,
3284                    ;ON ZERO ARG, IT RETURNS THE TIME,
3285                    ;ON NEGATIVE ARG, IT WAITS FOR THE NEXT TIC, THEN RETURNS THE TIME
3286
3287                    XTIC:   FPINT                   ;GET THE ARG
3288                            NEG     AC              ;NEGATE THE ARG
3289                            TST     AC
3290                            BGE     NOSET           ;ON ARG ,LE, 0, RETURN TIME
3291                            MOV     AC,@#PRESET     ;ELSE SET CLOCK INTERVAL IN PRESET,
3292                            MOV     #507,@#CLKIN    ;AND START CLOCK AT 1/10000
3293                            CLR     DTIME           ;AND SET TIME TO 0, AND RETURN
3294                    NOSET:  BEQ     CLKLOP          ;ON ARG ,LT, 0,
3295                            MOV     AC,DTIME        ;SET TIMER POSITIVE,
3296                    CLKLOP: MOV     DTIME,AC                ;AND WAIT FOR CLOCK INTERRUPTS,
3297                            BGT     CLKLOP          ;TO SET IT ,LE, 0
3298                            NEG     AC              ;(MAKE POSITIVE)
3299                            JSR     PC,FLOAT        ;INTO FLAC,
3300                            RTS     PC              ;AND RETURN IT
3301
3302
3303                    ;CLKINT HANDLES LPS CLOCK INTERRUPTS,  IT INCREMENTS THE TIME BY
3304                    ;THE INERVAL IN 1/100'S,
3305
3306                    CLKINT: DEC     DTIME           ;DECREMENT TIMER
3307                            BVC     NOAROV          ;IF ARITHMETIC OVERFLOW,
3308                            CLR     DTIME           ;CLEAR IT
3309                    NOAROV: RTI                     ;AND RETURN
3310
3311                    DTIME:  .WORD   0               ;TIMER
3312
3313
3314            001             .ENDC
3315
3316
```

```
3317
3318
3319          024144                          .=GEND                  ;START GRAPHICS CODE WHERE IT LEFT OFF
3320
3321                                          ;TJUMPS ADDS DJMP,TEXI,0 CELLS STARTING AT THE ADDRESS IN LOC UNTIL
3322                                          ; THE ADDRESS IN TXTPTR
3323
3324  024144  013701  024666                 TJUMPS: MOV     TXTPTR,AC       ;PUT LAST ADDRESS IN GEN.REG
3325  024150  020001                         NOPLOP: CMP     LOC,AC
3326  024152  002401                                 BLT     MORNOP          ;FILLED TO TEXTPTR?
3327  024154  000207                                 RTS     PC              ;YES, RETURN
3328  024156  012720  164000                 MORNOP: MOV     #DNOP,(LOC)+    ;DNOP A WORD
3329  024162  000772                                 BR      NOPLOP
3330                                          ;LOCSET GETS LOC ARG INTO NEWLOC, TEST FOR IN RANGE, MULTIPLIES BY
3331                                          ; 3 AND ADDS START OF FILE, LOC ENDS UP CONTAINING ADDRESS OF CELL
3332  024164  005037  024664                 LOCSET: CLR     STPFLG  ;CLEAR FLAG SO DISPLAY WILL STOP
3333  024170  005737  024664                 STILON: TST     STPFLG  ;WAIT TILL STOP INT, RESETS IT
3334  024174  001775                                 BEQ     STILON          ;NOT DONE YET
3335  024176  007065                         LOCGO:  FPPUT+IMMED             ;LOC, FROM FLAC INTO NEWLOC FOR RETURN LATER
3336  024200  000000                         NEWLOC: 0
3337  024202  000000                                 0
3338  024204  007071                         LOCGET: FPINT                   ;INTEGER PART OF 1ST ARG
3339  024206  123727  001616  000014                 CMPB    BE,#14          ;ON POTENTIAL OVERFLOW,
3340  024214  001012                                 BGT     TOOBIG          ;USE LAST LOC,
3341  024216  007040  023576                         FPMUL+DIRECT,FP6                ;NOW CALCULATE ADDR,
3342  024222  007073                                 FPABS                   ;GET RID OF POSSIBLE NEGATIVE
3343  024224  007071                                 FPINT                   ;INTEGERIZE IT
3344  024226  010100                                 MOV     AC,LOC          ;LOC HAS ADDRESS OFFSET
3345  024230  062700  025202                         ADD     #LOC0,LOC       ;NOW IT HAS THE ADDRESS OF THE BLOCK
3346  024234  020037  024666                         CMP     LOC,TXTPTR
3347  024240  002410                                 BLT     BELOW
3348  024242  013700  024666                 TOOBIG: MOV     TXTPTR,LOC      ;ON ABOVE, MAKE LOC=LAST LOC
3349  024246  162700  000006                         SUB     #6,LOC
3350  024252  005437  024202                         NEG     NEWLOC+2                ;MAKE RETURNED LOC NEG, ON ERR
3351  024256  105437  024201                         NEGB    NEWLOC+1
3352  024262  000207                         BELOW:  RTS     PC
3353
3354
3355                                          ;PUTCO TURNS AN INTEGER VALUE IN COARG INTO A COORDINATE SUITABLE
3356                                          ; FOR DISPLAY IN AC, HIBITS ALREADY CONTAINS VIS=INV BIT
3357                                          ; CALL BY:      MOV     COORD,AC        ;GET COORDINATE
3358                                          ;               JSR     PC,PUTCO
3359
3360  024264  005701                         PUTCO:  TST     AC              ;ON NEGATIVE COORDINATE,
3361  024266  002004                                 BGE     POSCO
3362  024270  052737  020000  024660                 BIS     #20000,HIBITS   ;SET NEGATIVE BIT
3363  024276  005401                                 NEG     AC              ;MAKE POSITIVE
3364  024300  053701  024660                 POSCO:  BIS     HIBITS,AC       ;PUT IN VIS=INV AND NEG=POS BITS
3365  024304  005037  024660                         CLR     HIBITS          ;AND CLEAR HIBITS FOR NEXT TIME
3366  024310  000207                                 RTS     PC
3367
3368
3369                                          ;RETNEW STARTS THE DISPLAY AND RETURN THE CALLER HIS LOC ARG PLUS 1
3370  024312  012737  000001  172000         RETNEW: MOV     #1,@#GTPC       ;RE-START DISPLAY
```

```
3371  024320  005037  024662                  CLR     MODE            ;ONLY USE THE SETTING ONCE
3372  024324  007000  024200                  FPGET+DIRECT,NEWLOC     ;ORIGINAL LOC, INTO FLAC
3373  024330  007015                          FPADD+IMMED             ;ADD 1
3374  024332  000001                  FP1:    1
3375  024334  040000                          40000
3376  024336  000207                          RTS     PC              ;RETURN TO FOCAL
3377
3378
3379                                  ;FLOAT FLOATS THE INTEGER IN AC INTO THE FLAC
3380  024340  012737  000017  001616  FLOAT:  MOV     #17,BE          ;TO POWER 17
3381  024346  010137  001620                  MOV     AC,HORD         ;AND HORD MANTISSA,
3382  024352  005037  001622                  CLR     LORD            ;BUT NO LORD
3383  024356  007070                          FPNOR                   ;NORMALIZE RESULTS
3384  024360  000207                          RTS     PC              ;RETURN TO CALLER
3385
3386
3387
3388                                  ;TCLEAR RESET THE TEXT FILE TO START AT TXTPTR AND CLEARS IT
3389
3390  024362  010046                  TCLEAR: MOV     LOC,-(SP)       ;SAVE REGISTER
3391  024364  005037  024674                  CLR     LFS
3392  024370  005037  024670                  CLR     ENDFLG          ;CLEAR COUNTER AND FLAG
3393  024374  013700  024666                  MOV     TXTPTR,LOC      ;SET UP THE POINTER
3394  024400  012720  170040                  MOV     #170040,(LOC)+  ;RESET STATUS FOR NO ITALICS
3395  024404  012720  117124                  MOV     #DPT+ENAB,(LOC)+        ;PUT IN ABS PT,, DEFAULTS ENABLED
3396  024410  005020                          CLR     (LOC)+          ;X=0
3397  024412  012720  001360                  MOV     #1360,(LOC)+    ;Y=1400
3398  024416  012720  100000                  MOV     #DCHR,(LOC)+    ;CHARACTER MODE
3399  024422  010037  024672                  MOV     LOC,INPTR       ;INPTR POINTS TO 1ST BYTE
3400  024426  005020                  CLRLOP: CLR     (LOC)+          ;CLEAR FILE
3401  024430  020027  037772                  CMP     LOC,#LOC0-6+DDSIZE      ;REACHED END?
3402  024434  002774                          BLT     CLRLOP          ;NO
3403  024436  012720  160000                  MOV     #DJMP,(LOC)+    ;INSERT JUMP,
3404  024442  012720  025172                  MOV     #DFILE,(LOC)+   ;TO START OF TEXT
3405  024446  012777  000177  000216          MOV     #177,@INPTR     ;INSERT CURSOR AT START OF FILE
3406  024454  010637  024664                  MOV     SP,STPFLG       ;AND SET STOP FLAG TO START DISPLAY
3407  024460  012737  025172  172000          MOV     #DFILE,@#GTPC   ;START DISPLAY
3408  024466  012600                          MOV     (SP)+,LOC
3409  024470  000207                          RTS     PC
3410
3411                                  ;TSTERR COMPLETES KEYBOARD INTERRUPT ROUTINE.  IT WILL RESUME
3412                                  ;THE DISPLAY ON AN ERROR SO THAT USER CAN STOP HIS PROGRAM BUT
3413                                  ;SAVE HIS DISPLAY BY OVERFLOWING THE INPUT BUFFER(2 CHARS)
3414
3415  024472  100401                  TSTERR: BMI     NO18            ;YES, NO OVERFLOW
3416  024474  104645                          ERROR+201+18.,+18.      ;AND GIVE ERROR
3417  024476  012637  001636          NO18:   MOV     (SP)+,KIN       ;GET CHAR
3418  024502  000002                          RTI
3419
3420
3421                                  ;RESETD WILL CLEAR THE STOP FLAG AND RESTART THE DISPLAY ON AN ERROR
3422
3423  024504  013701  001604          RESETD: MOV     PCF,AC          ;EXECUTE OVERLAYED INSTR,
3424  024510  010637  024664                  MOV     SP,STPFLG       ;SET THE STOP FLAG
```

```
3425  024514  012737  025172  172000          MOV     #LOC0-10,@#GTPC ;START DISPLAY
3426  024522  000137  002022                  JMP     ERR2+50
3427
3428
3429                                  ;LPINT HANDLES LIGHT PEN INTERRUPTS,  THESE OCCUR WHENEVER A LINE
3430                                  ; USER DIFINED AS LP SENSITIVE GETS HIT,
3431                                  ; THE LP INTERRUPT ROUTINE SIMPLY TAKES THE COORDS OF THE HIT AND
3432                                  ; PUTS THEM IN LOC 0 AND TAKES THE ADDRESS OF THE HIT AND PUTS IT
3433                                  ; INTO LPLOC.
3434  024526  010146                  LPINT:  MOV     AC,-(SP)        ;SAVE REGISTER
3435  024530  013701  025202                  MOV     LOC0,AC         ;ON CHAR MODE OR JUMP IN LOC0,
3436  024534  042701  163777                  BIC     #163777,AC      ;(BITS 11 AND 12 NOT SET)
3437  024540  001425                          BEQ     NOLPC0          ;DON'T INSERT
3438  024542  013701  172004                  MOV     @#GTX,AC        ;USE AC AS TEMP
3439  024546  032737  040000  025204          BIT     #40000,LOC0+2   ;TEST WHETHER COORDS IN LOC 0 NOW
3440                                                                  ;ARE VISIBLE
3441  024554  001402                          BEQ     INV0
3442  024556  052701  040000                  BIS     #40000,AC       ;IF NOT, SET INV BIT IN
3443                                                                  ;NEW X COORDINATE
3444  024562  010137  025204          INV0:   MOV     AC,LOC0+2       ;INSERT THE LP HIT COORDINATES
3445  024566  013701  172006                  MOV     @#GTY,AC        ;GET GTY ,
3446  024572  042701  176000                  BIC     #176000,AC      ;RID OF CHARACTER REGISTER BITS
3447  024576  020127  001400                  CMP     AC,#1400        ;AND MAKE SURE NOT ABOVE SCREEN
3448  024602  002402                          BLT     ONPIC
3449  024604  012701  001377                  MOV     #1377,AC
3450  024610  010137  025206          ONPIC:  MOV     AC,LOC0+4       ;MOVE Y COORD IN NOW
3451  024614  013737  172000  024656  NOLPC0: MOV     @#GTPC,LPLOC    ;NOW SAVE PC AT TIME OF HIT
3452  024622  012737  000001  172000          MOV     #1,@#GTPC       ;RESTART THE DISPLAY WHERE IT LEFT OFF
3453  024630  012601                          MOV     (SP)+,AC        ;RESTORE REGISTER
3454  024632  000002                          RTI                     ;AND RETURN
3455
3456
3457                                  ;STPINT CHECKS STOP FLAG WHENEVER THE GT40 REACHES THE END OF THE
3458                                  ;DISPLAY FILE,  IF SET, IT RESTARTS THE DISPLAY, IF ZERO, IT SETS
3459                                  ;IT BUT DOES NOT RESTART THE DISPLAY
3460
3461  024634  005737  024664          STPINT: TST     STPFLG
3462  024640  001403                          BEQ     STOPIT                  ;IF=0, DON'T START DISPLAY
3463  024642  012737  025174  172000          MOV     #LOC0-6,@#GTPC          ;ELSE, START IT AT LOC0
3464  024650  010637  024664          STOPIT: MOV     SP,STPFLG               ;IN ANY CASE, SET FLAG
3465  024654  000002                          RTI                             ;AND RETURN
3466
3467
```

```
3469                                        ; STORAGE AREA
3470  024656  025202                        LPLOC:  .WORD   LOC0             ;ADDRESS OF LAST LITE PEN HIT
3471  024660  000000                        HIBITS: .WORD   0                ;VISIBLE AND NEGATIVE BITS OF MODE
3472  024662  000000                        MODE:   .WORD   0                ;MODE OF NEXT DATUM (SET BY EDITS)
3473  024664  000001                        STPFLG: .WORD   1        ;>0 TELLS STOP INTERRUPT TO RESTART
3474  024666  000000                        TXTPTR: .WORD   0                ;POINTER TO START OF TEXT FILE
3475  024670  000000                        ENDFLG: .WORD   0        ;FLAG WHEN TEXT FILE=40 REACHED
3476  024672  000000                        INPTR:  .WORD   0        ;POINTER TO NEXT BYTE IN TEXT FILE
3477  024674  000000                        LFS:    .WORD   0        ;COUNTER OF LINES USED ON SCREEN
3478          024676                        SUBEND=.
3479                                                                 ;MAKES CURRENT LINE THE LAST
3480
3481
3482
3483
3484          010052                            .=XOUT
3485  010052  010446                        MOV     CHAR,-(SP)       ;SAVE OUTPUT CHARACTER
3486  010054  023727  001634  175614        CMP     OUTDEV,#TENOUT   ;ON ALL BUT TEN OUTPUT,
3487  010062  001402                        BEQ     NOS
3488  010064  000137  024720               JMP     TXTDIS           ;NOW DO MORE
3489  010070  000137  024734        NOS:   JMP     NOSHOW
3490
3491
3492          003042                            .=OUTY
3493  003042  000137  024676               JMP     TSTTXT           ;GO TEXT FOR GRAPHICS OUTPUT, TOO
3494
3495          001634                            .=OUTDEV
3496  001634  177700                        .WORD   177700           ;MAKE GRAPHICS DEFAULT OUTPUT MODE
3497
3498          001762                            .=ERR2+10
3499  001762  177700                        .WORD   177700           ;ALSO DEFAULT AFTER ERROR(OR ^C)
3500
3501
3502          024676                            .=SUBEND
3503
3504
3505                                        ;TSTTXT RETURNS WITH Z BIT SET IF TTY OR G IN USE
3506  024676  023727  001634  177564  TSTTXT: CMP     OUTDEV,#TPS
3507  024704  001403                        BEQ     EDITIT
3508  024706  023727  001634  177700        CMP     OUTDEV,#177700   ;OUTDEV=#R0 MEANS GRAPHICS
3509  024714  000137  003050        EDITIT: JMP     OUTY+6
3510
3511
3512                                        ;TEXTDIS DISPATCHES TO TXTHAN AND RETURNS TO XOUT ROUTINE
3513  024720  004737  025000        TXTDIS: JSR     PC,TXTHAN        ;DISPLAY ALL BUT 10 OUTPUT
3514  024724  023727  001634  177700        CMP     OUTDEV,#177700   ;IF SCREEN OUTPUT (O S)
3515  024732  001420                        BEQ     NOSEND           ;DON'T OUTPUT THIS CHAR,
3516  024734  013704  001634        NOSHOW: MOV     OUTDEV,CHAR
3517  024740  105714                        TSTB    @CHAR            ;WAIT FOR XMIT READY FLAG
3518  024742  100376                        BPL     .-2
3519  024744  111664  000002               MOVB    @SP,2(CHAR)      ;NOW SEND THE CHAR,
3520              002                       .IFDF   TERM
3521  024750  023727  001634  175614        CMP     OUTDEV,#TENOUT   ;THEN IF PDP-10 OUTPUT,
3522  024756  001006                        BNE     NOSEND
```

```
3523  024760  005737  023156                  TST     TOHOST          ;AND IN FOCAL,
3524  024764  001003                          BNE     NOSEND
3525  024766  000001                          WAIT
3526  024770  000001                          WAIT                    ;WAIT FOR A COUPLE OF INTERUPTS
3527  024772  000001                          WAIT
3528             001                          .ENDC
3529  024774  012604                  NOSEND: MOV     (SP)+,CHAR      ;NOW RESTORE CHAR,
3530  024776  000205                          RTS     R5
3531
3532
3533                                  ; TXTHAN, THE TEXT HANDLER, INSERTS THE CHARACTER IN @SP (PUT
3534                                  ; THERE BY THE FOCAL OUTPUT ROUTINE) INTO THE TEXT FILE.
3535                                  ; WHEN IT RECEIVES A LF LATER THAN END OF FILE-100 BYTES, OR
3536                                  ; WHEN IT HAS FILLED THE SCREEN, IT CALLS TCLEAR
3537                                  ; TO CLEAR THE TEXT FILE.
3538
3539  025000  010446                  TXTHAN: MOV     CHAR,-(SP)      ;SAVE REGISTER
3540  025002  016604  000004                  MOV     4(SP),CHAR      ;GET INPUT CHAR
3541  025006  042704  000200                  BIC     #200,CHAR       ;CLEAR BIT FOCAL SETS
3542  025012  120427  000007                  CMPB    CHAR,#BELL      ;ON BELL
3543  025016  001003                          BNE     SILENT
3544  025020  005237  172002                  INC     @#GTSTAT        ;RING THE BELL
3545  025024  000454                          BR      TXTDUN
3546  025026  120427  000033          SILENT: CMPB    CHAR,#33        ;ON ALTMODE,
3547  025032  001002                          BNE     .+6
3548  025034  112704  000044                  MOVB    #44,CHAR                ;ECHO $ SIGN
3549  025040  120427  000015                  CMPB    CHAR,#CR        ;CR?
3550  025044  001423                          BEQ     ADDCHR          ;YES, ADD IT TO FILE,
3551  025046  120427  000012                  CMPB    CHAR,#LF        ;LF?
3552  025052  001404                          BEQ     LFCHAR          ;YES, GO CHECK LINES ON SCREEN
3553  025054  120427  000040                  CMPB    CHAR,#40        ;ANY OTHER CONTROL CHAR?
3554  025060  002436                          BLT     TXTDUN          ;YES, IGNORE IT
3555  025062  000414                          BR      ADDCHR          ;INSERT ALL OTHER CHAR,S
3556  025064  005237  024674          LFCHAR: INC     LFS             ;BUMP LINE COUNT
3557  025070  005737  024670                  TST     ENDFLG          ;ENDFLG=1 MEANS LAST LINE
3558  025074  001004                          BNE     LASTL           ;YES, LAST LINE
3559  025076  023727  024674  000040          CMP     LFS,#40         ;SCREEN FILLED?
3560  025104  002403                          BLT     ADDCHR          ;NO, TREAT LF LIKE ANY OTHER CHAR,
3561  025106  004737  024362          LASTL:  JSR     PC,TCLEAR       ;CLEAR TEXT BUFFER STARTING AT TXTPTR
3562  025112  000421                          BR      TXTDUN          ;AND WAIT FOR NEXT CHAR,
3563  025114  110477  177552          ADDCHR: MOVB    CHAR,@INPTR     ;MOVE NEW CHAR, IN
3564  025120  023727  024672  037770          CMP     INPTR,#LOC0-6+DDSIZE-2  ;REACHED END OF FILE?
3565  025126  002367                          BGE     LASTL           ;THEN CLEAR THE FILE
3566  025130  023727  024672  037662          CMP     INPTR,#LOC0-6+DDSIZE-110        ;IF PTR PAST END-72 CHAR,S,
3567  025136  002402                          BLT     NOTFUL
3568  025140  005237  024670                  INC     ENDFLG          ;MAKE THIS THE LAST LINE,
3569  025144  005237  024672          NOTFUL: INC     INPTR           ;INPTR PT,S TO NEXT BYTE
3570  025150  112777  000177  177514          MOVB    #177,@INPTR     ;INSERT CURSOR THERE
3571  025156  012604                  TXTDUN: MOV     (SP)+,CHAR              ;RESTORE REGISTER
3572  025160  000207                          RTS     PC
3573
3574
3575
3576             002                          .IFDF TERM              ;CODE FOR TERMINAL HOOKUP FOLLOWS
```

```
3577          023016                        .=DBOT+LPSSIZ           ;TERM CODE FOLLOWS LPS CODE AT START OF DNEW
3578
3579                                        ;K CHECKS FOR ^F AND ^T, ON ^F IT THROWS THE TOHOST FLAG
3580                                        ;WHICH TELLS KEYBOARD INPUT TO GO TO EITHER THE PDP-10 OR TO FOCAL,
3581                                        ;ON ^T IT THROWS THE TOFOCL FLAG WHICH TELLS PDP-10 INPUT WHETHER TO
3582                                        ;GO DIRECTLY TO THE SCREEN OR TO FOCAL.
3583  023016  013[illegible]  17[illegible]  K:      MOV     @#TKS+2,-(SP)   ;GET CHARACTER FROM KEYBOARD
3584  023022  121[illegible]  [illegible]      CMPB    @SP,#206        ;^F?
3585  023026  00[illegible]                    BNE     NOTF
3586  023030  105137  023156                   COMB    TOHOST          ;YES , FLOP FLAG
3587  023034  012737  000101  175610           MOV     #101,@#TENIN    ;AND ALLOW INTERRUPTS IF NOT ON YET
3588  023042  000421                           BR      NOEKO           ;AND IGNORE THIS CHARACTER
3589  023044  121627  000224           NOTF:   CMPB    @SP,#224        ;^T?
3590  023050  001006                           BNE     NOTT
3591  023052  105137  023157                   COMB    TOFOCL          ;YES, FLOP FLAG
3592  023056  001013                           BNE     NOEKO           ;AND IF FOCAL LISTENING TO 10,IGNORE CHAR
3593  023060  012716  000337                   MOV     #337,@SP        ;ELSE, REPLACE CHAR WITH '_',
3594  023064  000412                           BR      GOFOC           ;AND GIVE IT TO FOCAL
3595  023066  105737  023156           NOTT:   TSTB    TOHOST          ;SENDINGG TO FOCAL?
3596  023072  001407                           BEQ     GOFOC           ;IF SO, GO DO EXECUTE USUAL CODE
3597  023074  105737  175614           TEST10: TSTB    @#TENOUT        ;ELSE, WHEN READY,
3598  023100  002375                           BGE     TEST10
3599  023102  111637  175616                   MOVB    @SP,@#TENOUT+2  ;SEND CHAR TO THE 10
3600  023106  005726                   NOEKO:  TST     (SP)+           ;RESET STACK,
3601  023110  000002                           RTI                     ;AND EXIT
3602  023112  000137  010114           GOFOC:  JMP     KINT+4          ;GO RE-ENTER KEYBOARD INTERRUPT ROUTINE
3603
3604                                        ;TENINT HANDLES PDP-10 INPUT INTERRUPTS,  IT SENDS THE CHARACTER EITHER
3605                                        ;TO BOTH FOCAL AND THE SCREEN OR JUST TO THE SCREEN DEPENDING ON THE
3606                                        ;STATE OF THE FLAG LABELLED 'TOFOCL'
3607
3608  023116  013746  175612           TENINT: MOV     @#TENIN+2,-(SP) ;GET INPUT
3609  023122  021627  000377                   CMP     (SP),#377       ;IGNORE RUBOUTS
3610  023126  001411                           BEQ     IGNORE
3611  023130  105737  023157                   TSTB    TOFOCL          ;FOCAL LISTENING TO 10?
3612  023134  001404                           BEQ     NORESP
3613  023136  005737  001636                   TST     KIN             ;OVERFLOW?
3614  023142  100763                           BMI     GOFOC
3615  023144  000402                           BR      IGNORE
3616  023146  004737  025000           NORESP: JSR     PC,TITHAN       ;IF FOCAL DEAF TO 10, JUST ECHO CHAR,,
3617  023152  005726                   IGNORE: TST     (SP)+           ;RESET STACK,
3618  023154  000002                           RTI                     ;AND RETURN
3619  023156     000           TOHOST: .BYTE   0               ;=1 MEANS TERMINAL, 0 MEANS FOCAL
3620  023157     000           TOFOCL: .BYTE   0               ;=1 MEANS LISTEN TO 10; 0 MEANS DON'T
3621                                   .ENDC                   ;END OF TERMINAL HOOKUP CODE
3622                                   .ENDC                   ;END OF GT40 CODE
3623          01[illegible]330                 .END    INIT
```

```
ABSFF     006556          1751    1903#
AC      =%000001            42#     53#     62#    485*    486*    552*    553*    554*    555     556*    557     614*    615
                           617     622*    623*    624*    625     634*    635     637     650*    662     665     779*    780
                           817     908*    909     911*    914     919     978*    987*    988     1036    1050*   1092    1221*
                          1225*   1226*   1227*   1228*   1229*   1286*   1287    1289    1291    1325    1330    1344*   1346*
                          1351    1365*   1376    1401*   1402*   1403    1405    1407*   1408    1521    1595    1602*   1733
                          1763*   1764*   1765*   1766    1794*   1796    1801*   1803    1807*   1817    1824*   1825    1827
                          1831*   1867*   1871*   1875    1912*   1913*   1927*   1943*   1944    1955*   1960    1979    1981*
                          2011*   2013    2021    2034    2056    2058*   2059*   2060    2067*   2068    2070*   2071    2073
                          2076*   2100*   2120*   2188    2192    2195*   2197*   2199*   2220*   2347*   2348    2349    2496*
                          2497*   2499*   2500*   2501*   2518    2560*   2561*   2562    2565    2582*   2583    2600    2606
                          2617*   2618    2623    2627*   2643*   2644*   2645    2797*   2798*   2799    3002    3018*   3024
                          3026*   3029    3034*   3035*   3061*   3062*   3064    3066    3069*   3071    3076    3086    3090*
                          3091*   3092*   3093    3112*   3116*   3117    3119*   3120*   3121*   3123*   3124*   3132    3137*
                          3138*   3324*   3325    3344    3360    3363*   3364*   3381    3423*   3434    3435*   3436*   3438*
                          3442*   3444    3445*   3446*   3447    3449*   3450    3453*
ADCS    = 176770          2566*   2572#
ADDB    = 176772          2568    2573#
ADDCHR    025114          3550    3555    3560    3563#
ADDFF     006230          1740    1789#
ADIN    = 170400          2778#   2779
AFIX      005344          1480    1492#
AGO       005372           318    1500#
AH      =%000004            65#   1606*   1630*   1633    1635*   1636*   1734    1771*   1773*   1789    1805*   1814    1869
                          1885*   1887*   1889*   1919    1921    1942*   1947    1956*   1959    1978    1988    1990*   1992*
AL      =%000005            66#   1593*   1595*   1596    1599*   1600*   1601    1603*   1604*   1605*   1735    1772*   1806*
                          1812    1867    1884*   1888*   1891*   1917    1924    1954*   1961    1991*
ALGZA     006506          1804    1815    1884#   1896
ALGZB     006522          1886    1890#
ALIGNA    006270          1795    1803#
ALIGNB    006256          1798#   1802
ALIST   = 001433           352#   1514
ALL     = 000001            70#   1067    1151
ALTAKA    006324          1797    1817#
ARGNXT    004476          1242#   1264
ARO       005330           321    1485#
ASCII     023420          3016#   3031
ASK       005244           262    1467#   1503
ASPACE    005340           319    1490#
ATAKE     005304           320     322    1479#   1486    1488    1491
ATLIST  = 001344           304#   1514
AXIN      001642           416#    504*    509     841*    558     842     859*    895*   1038*   1405    1477*
AXOUT   =%000003            44#     55      64     513*    519*    520*    522     523     535     695*    696     697     698
                           701     702*    705     754     756     875     891     892     898     901*    902*    906     912
                           914     919    1015    1048    1050    1053    1057*   1058*   1097    1112    1129    1140    1147*
                          1148*   1150    1153    1156*   1190    1194*   1470    1472*   1473    1481*   1495*   1496    1502*
                          1524    2173*   2206
BDIVX     006476          1877    1881#   1958
BE        001616           484#    660*   1630    1717*   1727*   1733*   1758*   1793    1824    1897*   1952    1979*   2011
                          2293    2405    2424*   2434*   2529*   2621*   2624*   3054    3339    3380*
BEGIN   = 001100           134#    439
BELL    = 000007          2768#   3542
BELOW     024262          3347    3352#
BH      =%000002            63#   1609*   1618    1629    1713    1719*   1722    1725*   1729*   1734*   1759*   1791    1798*
```

```
                        1809*   1813*   1814*   1829*   1864*   1874*   1878*   1880*   1882*   1885    1895    1898*   1903
                        1918*   1919    1921*   1925*   1930*   1950    1975    1983    1985*   1987*   2006
BL      =%000003          64#   1610*   1611*   1617    1631*   1632    1715    1718*   1726*   1728*   1735*   1760*   1799*
                        1810    1812*   1830*   1835    1865*   1875*   1879*   1881*   1884    1899*   1917*   1924*   1929*
                        1[illegible]    1986*
BOTTOM    001652        420#    850     1343    1381    1401    2451    2641*   2919
BUFBEG  = 011330        416     417     418     419     2636#
BUFFUL    002202        510     526#
BUFR      001644        417#    554     555*    558*    894*    1034    1118*   1403
CCFLG     001601        396#    2280*   2283*   2829*   2833#
CFRS      001654        424#    695     908     1119*
CHAR    =%000004         45#     56#     65      506     511     535*    537     569     570*    589     592     602     604
                        615     635     653     722*    724*    733     735     743*    747*    768*    770     773     775*
                        780*    785     787*    788*    790     793     801*    833     843     845     847     849     860*
                        875*    876     891*    898*    899     903*    904     907     950     956     960     976     978
                        980     982     1014    1015*   1047    1053*   1055    1080    1086    1128    1157*   1170    1214
                       1215    1246    1262    1265    1279    1282    1314    1318    1322    1326    1449    1471    1492*
                       1501*   1515    1524*   1525    1527    2005*   2010*   2060*   2110    2116*   2122*   2133*   2152
                       2157*   2158*   2176    2183    2204    2216    2218    2222    2224    2229    2231*   2232    2240*
                       2250*   2251    2253    2255*   2259    2260*   2261    2263*   2264*   2270*   2457*   2459*   2517
                       2518*   2521*   2522    2527*   2528    2546    2547*   2549*   2585    2589    2591    2593*   2597
                       2607    3016    3032    3049    3485    3516*   3517    3519*   3529*   3539    3540*   3541*   3542
                       3546    3548*   3549    3551    3553    3563    3571*
CHIN      002710        184     767#    769
CHINX     002756        776     779#
CINT      010256       2327#   2647    2801
CLCU      010210       2303#   2314
CLKADR  = 000134       2781#
CLKIN   = 170404       2777#
CLKT      010274       2327*   2328*   2338#
CLOSE   = 022626        251#    821     1224    2047    2236
CLRLOP    024426       3400#   3402
COCO      024010       3109    3111#
COMBUF  = 001461        393#    503     509
COMLST  = 001250        258#    988
COPYLN  = 104630        205#   1063    1100    1139
COPYLX    004040        206    1112#
CR      = 000015         74#    511     537     899     976    1055    1525    2547    2767#   3549
CRLF    = 005015         75#    488     800    1512    2476
DBOT    = 023016       2918#   2920    2922    2942    3577
DCHR    = 100000       2762#   2938    3013    3398
DDSIZE  = 012576       2752#   2918    2934    2962    3066    3068    3401    3564    3566
DEBG      001610        401#    496*    529*    872     874*    878    1494*   1499*
DECONV    007702       2174    2187    2215#
DECON1    007722       2178    2217    2219    2221#   2238
DECON2    010002       2237#   2243
DECOY     007764       2232#   2241
DELETE  = 000004         98#    545    1094
DETN      010006       2226    2240#
DFILE     025172       2923#   3404    3407
DIFL      006600       1917#   1932
DIGTSA    002654        218     743#    748
DIGTST  = 104644        217#   2151    2155
DIRECT  = 000000       1579#   2037    2185    2193    2234    2242    2419    2423    2426    3140    3341    3372
```

```
DIVFF     006566          1742    1912#
DIVZER    006642          1916    1935#
DJMP    = 160000          2760#   2932    3046    3113    3403
DNOP    = 164000          2758#   3328
DO        004066           265    1127#   2548
DOCONT    004204          1146    1149    1155#
DOER      004220          1135    1138    1160#
DOGR      004106          1131    1134#
DOGRP1    004124          1132    1141#
DOGRP2    004120          1139#   1152    1154
DOXIT     004210          1069    1156#
DPT     = 114000          2761#   2925    2928    2935    2959    2979    2982    3395
DSIZE   = 001625          2731#   2751#   2752
DSTP    = 173400          2759#
DUNASC    023454          3025    3028#
DVEC    = 110000          2763#   2972    2976
DXIT      007540          2125    2160#   2223    2225    2228    2230
E       =%000002            54#   2055#   2062    2067    2099#   2111    2113    2114    2117    2119#   2139    2140#
ECHOLS  = 001450           371#    772
ECOLOG    003252           891#    893     910
EDITIT    024714          3507    3509#
EFUN      004614          1253    1276#   1281
EFUN3     004656          1247    1292#
ELPAR     004504          1244#   1266
EMDIR     005672          1643    1653#   1674
EMIME     005704          1648    1657#
EMIND     005726          1645    1665#
EMREL     005744          1649    1672#
EMT     = 007000          1547#   1549    1550    1551    1552    1553    1554    1555    1557    1558    1559    1560    1561
                          1562    1563    1564    1566
EMTHR     005736          1647    1669#
EMTO      005716          1646    1661#
ENAB    = 003124          2756#   2928    2935    3395
ENDFLG    024670          3392#   3475#   3557    3568#
ENIT2     003070           806#   2649    2816
ENUM      004574          1252    1270#
ENUM2     004604          1272#   1294
EPAR      004460          1219    1237#
EPAR2     004510          1238    1246#
EPURE     004436          1222    1229#
ERASE     003744           266    1076#
ERASET  = 104640           213#   1082    1088
ERASEV  = 104636           211#    544     806    1083    1103
ERG       004010          1094#   1101
ERL       004002          1078    1087    1091#
ERROR   = 104400           243#    440     526     561     808     852     984    1026    1160    1176    1268    1307    1308
                          1388    1483    1638    1935    1966    2282    2286    2297    2632    2832    3416
ERRORC    003502           263     269     271     273     275     277     282     283     981     984#
ERR2      001752           468     476#   2844    3426    3498
ERT       003770          1081    1086#
ERTX      004050           214    1118#
ERV       003764          1083#   1096    1098    1099
ERVC      004034          1077    1103#
ERVX      005100           212    1401#
```

```
ETERM    004374          1214#   1263    1267
ETERMN   004566          1256    1267#
ETERM1   004544          1251    1261#
ETERM2   004406           116    1218#   1234
EVAL     004520                  1243    1249#   1297    1460
EVALU  = 104660          2770#   2998    3022    3051    3088    3094    3098
EVALUX   004664           230    1296#
EVAL.X = 104660           229#    951    1173    1177    1187    1284    1323    1328    1497    2515    2544    2563    2609
F      =%000003            55#   2054#   2056    2058    2062    2064*   2065    2068    2070    2108    2114    2124#   2142#
FCODE  = 007200          1230    1566#
FCONT    004306          1183    1188#
FCONT2   004312          1190#   1205
FCOS     010426           160    2401#
FCOS2    010466          2406    2415#
FCOS4    010526          2426#   2435
FEMT     005472            92    1586#
FERRO    006732          1945    1948    1966#
FIGOE    007654          2196    2199#   2202
FIGO1    007566          2177    2179#
FIGO2    007636          2182    2184    2194#
FIGO3    007664          2200    2204#
FIGO4    007674          2103    2207#
FINCR    004254           300    1177#
FINDLN = 104624           201#    897     973    1011    1049    1131
FINDN    002574           697#    704
FINDO    002622           700     705#
FINDX    002566           202     695#
FINERR   004252           298    1171    1176#   1179    1310
FINFIN   004266           297    1181#
FISW     001626           409#   1462    1521*   2474
FLARG    001612           402#   1231    1272    2030    2185    2193    2294
FLIMIT   004300           296    1185#
FLIST1   001336           300#   1175
FLIST2   001330           296#   1178
FLOAT    024340          3125    3139    3380#
FLOSGN   007034          2007    2011#
FLOUT    007450          2109    2115    2139#
FLTDO    005556          1598    1607#
FLTDO1   005544          1602#   1643    1644    1645    1646    1647    1648    1649    1650    1655    1659    1663    1667
                         1670
FLTONE   010560          2419    2423    2426    2437#
FLTX     005624          1619#   1637    2208
FLTZER   003830           399     993#   1261
FNADDR   006146          1748#   1766
FNTABL = 001166           168#   1286
FOGO1    007042          2013#   2020    2026
FOGO2    007072          2014    2021#
FOGO3    007110          2008    2017    2022    2028#
FOGO4    007124          2032#   2039
FOR      004222           267    1168#
FPABS  = 007073          1560#   1841    1848    3342
FPADD  = 007010          1200    1550#   2036    2235    2305    2399    2419    2423    2426    3373
FPDIV  = 007030          1552#   2304    2417    2420    3140
FPGET  = 007000           658     819    1199    1223    1229    1549#   2043    2193    2303    2337    2413    2473    2551
```

```
                    2604    3372
FPINT   = 007071     661    1324    1329    1558#   1849    2033    2046    2191    2516    2559    2564    2581    2605
                    2610    2999    3023    3059    3085    3089    3131    3141    3338    3343
FPMUL   = 007040    1553#   2018    2024    2037    2044    2201    2234    2242    2404    2414    2422    2425    2430
                    2432    2569    3056    3341
FPNEG   = 007074    1561#   1852    2009    2035    2190    2398    2415
FPNOR   = 007070    1293    1557#   2339    3383
FPPOW   = 007050    1554#
FPPUT   = 007060    1174    1186    1189    1201    1232    1273    1498    1555#   2032    2185    2313    2403    2416
                    2429    2431    2545    3335
FPRINT  = 007076     820    1463    1563#   2475
FPRNT     007356    2101    2108#
FPRNTP    007376    2112    2116#   2127    2129    2143
FPSGN   = 007072    1559#   1844
FPSUB   = 007020    1203    1551#   2433
FPURE     005522    1594    1596#
FP1       024332    3374#
FP6       023576    3057#   3140    3341
FREAD   = 007075     676    1271    1562#
FROM    = 000003     819    1203    1223    1574#   2043    2235    2303    2404    2413    2414    2422    2425    2430
                    2432    2433
FSIN      010416     158    2398#
FSIZE   = 002154    2916#   2918    2922
FTRY      007434    2118    2131#
FZER    = 007077    1564#   2215    2567
GALL      002554     663     672#
GALL1     002522     652     654     662#
GEND    = 024144    3151#   3319
GETARG    004674     657    1168    1309#   1468    2603
GETC    = 104614     189#    514     832     944     959     962     975     985    1248    1280    1316    1319    1451
                    1513    1519    2186    2221    2355    2587
GETCH   = 104614    2771#
GETF      006114    1733#   1739    1792    1818
GETLN   = 104622     199#    530     972    1010    1046    1091    1127    1520
GETLNX    002454     200     648#
GETVAR    004704    1254    1313#
GETX      003232     190     875#
GGROUP    002560     666     673#
GOFOC     023112    3594    3596    3602#   3614
GONE      002146     514#    524
GOTO      003436     268     954     972#
GRALL     023320    2985    2987#
GROOVY  = 104650     221#    1062    1099    1137
GROVX     002672     222     754#
GROVZ     002650     605     736#
GSERCH    004730    1317    1321#
GSIZE   = 002004    2899#   2904#   2916
GSWIP     005060    1383#   1387
GS0       004764    1327    1331#
GS1       004776    1321    1332    1342#
GTAKE     005170    1384    1426    1435#
GTEST     005160    1424    1432#
GTESTN    002562     655     676#
GTESTW    002512     660#    677    2540
```

```
GTPC    = 172000          2773#   2774    3036#   3370#   3407#   3425#   3451    3452#   3463#
GTRY      005134          1374    1377    1429#   1429
GTRY2     005144          1426#   1433
GTSTAT  = 172002          2774#   2775    3544#
GTX     = 172004          2775#   2776    3438
GTY     = 172006          2776#   3445
GT40    = 000000          2653
HIBITS    024660          2984#   2986#   3015#   3030#   3048#   3072    3362#   3364    3365#   3471#
HORD      001620           405#    953    1204    1609    1618#   1847    2016    2309    2493    2505#   2528#   2568#   2620#
                          2623#   3052    3095    3099    3381#
IF        003374           270     949#
IGNOR     002114           505#    507     512
IGNORE    023152          3610    3615    3617#
IMMED   = 000005          1576#   2018    2024    2044    2337    2399    2417    2420    2569    3056    3335    3373
INADDR    005662          1600    1643#
INCH    = 104612           187#    767    2526
INDEV     001632           412#    476#    500     783    2250    2256#   2379    2380
INFILE    023670          3065    3075#
INIT      011330          2640#   2795    3623
INITD     023166          2815    2831    2946#
INIT2     003072            87     771     807#
INLIST  = 001366           317#   1484
INPTR     024672          3399#   3405#   3476#   3563#   3564    3566    3569#   3570#
INPUTN    002204           517     529#
INTF      006330          1749    1824#
INTG      006340          1827#   1832
INTO    = 000003          1186    1189    1573#   2403    2416    2429    2431
INTX      006362          1828    1835#
INTY      006366          1748    1790    1836#
INTZ      006356          1755    1826    1834#
INV       023314          2977    2983    2986#
INV0      024562          3441    3444#
IOFIX     010304          2347#   2357
IOGO    = 010362          2349    2368#
IOLIST  = 010352          2345    2357    2361#
IOPATC  = 010400          2348    2378#
IOQ       010334          2351    2353    2355#
IPTR    = 000001           658    1199    1201    1229    1232    1273    1571#   2032    2036    2201    2304    2305    2313
                          2473    2551    2604
JMS     = 103414           175#    177     179     181     183     185     187     189     191     193     199     201     203
                           205     207     209     211     213     215     217     219     221     223     225     227     229
                           234     470
JUST0     023456          3019    3029#
K         023016          2840    3583#
KIN       001636           414#    478#   2270    2272#   2284    2287#   2834    3417#   3613
KINT      010110           113    2277#   2825    2839    3602
KINT1     010132          2279    2283#
KINT2     010136          2281    2284#
KINT3     010136          2828    2833#
KINT4     010142          2830    2834#
KITH      010074          2252    2270#   2271
KKSIZE  = 037770          2755#   2918
KSIZE   = 020000          2719#   2755
LAST      025106          3558    3561#   3565
```

```
LEAST     023634          3055    3063    3068#
LEDOUT  = 170402          2779#
LF      = 000012          2766#   3551
LFCHAR    025064          3552    3556#
LFS       024674          3391*   3477#   3556*   3559
LINCNT    001630           411#    792*    798*
LINENO    001624           408#    662*    667     698     754     1112*
LINERR    002314           531     561#    664     668     1079
LISTGO  = 001406           329#    1017
LIST9   = 001456           386#    1014*   1017
LIST6   = 001453           378#    1022
LOC     =%000000          2769#   2958*   2959*   2960*   2961*   2988*   2997    3000*   3002*   3010    3012*   3013*   3014*
                          3021    3028*   3029*   3034    3045*   3046*   3047    3072*   3075*   3076*   3113    3115*   3116
                          3119    3123    3325    3328*   3344*   3345*   3346    3348*   3349*   3390    3393*   3394*   3395*
                          3396*   3397*   3398*   3399    3400*   3401    3403*   3404*   3408*
LOCGET    024204          3111    3338#
LOCGO     024176          2987    3335#
LOCSET    024164          3009    3043    3147    3332#
LOC0      025202          2928#   2934    2946    2958    2962    3062    3066    3068    3138    3345    3401    3425    3435
                          3439    3444*   3450*   3463    3470    3564    3566
LORD      001622           406#   1610    1617*   2311    3382*
LOSE0     003422           958#    961
LOSE1     003426           955     960#
LOSE2     003414           945     956#
LOST      003370           944#    957
LPADR   = 000324          2787#   2805*   2806*
LPBSIZ  = 000000          2732#
LPINT     024526          2805    3434#
LPLOC     024656          2946*   3134*   3135    3137    3451*   3470#
LPS     = 177514            79#   2373
LPSSIZ  = 000000          2901#   2916    2942    3577
LSPR      001602           398#   2491
LTZERO    007414          2121    2123#   2132    2134
MAKSUR    023440          3024#   3027
MDP0      006434          1864#   1873
MODE      024662          2947*   2972*   2976*   2979*   2982*   2988    3014    3044*   3045    3086*   3087*   3093*   3097*
                          3101*   3102*   3371*   3472#
MODIFY    003552           274    1010#
MORNOP    024156          3336    3328#
MORTXT    023370          3010#   3033
MULFF     006426          1743    1862#   1963
MULZ      006166          1757#   1863
NALPHA  = 000020            71#   1270    2181    2227
NEGFF     006562          1752    1905#
NEGX      006470          1878#   1901    1905
NEWLOC    024200          3336#   3350*   3351*   3372
NOBLI     023752          3096    3098#
NOEKO     023106          3588    3592    3600#
NOLP      023770          3100    3102#
NOLPCO    024614          3437    3451#
NOLPS     023204          2957#
NOP       006000          1650    1689#
NOPLOP    024150          3325#   3329
NORESP    023146          3612    3616#
```

CROSS REFERENCE TABLE

```
NORF      006036          1616    1713#
NORMD     006072          1720    1725#   1730
NORMZ     006112          1716    1758#   1834    1892
NORM2     006046          1711    1717#   1721    1723    1962
NORX      006204          1760    1763#
NOS       010070          3487    3489#
NOSEND    024774          3515    3522    3524    3529#
NOSHOW    024734          3489    3516#
NOTBAK    023612          3060    3062#
NOTEXT    023606          3053    3061#
NOTF      023044          3585    3589#
NOTFUL    025144          3567    3569#
NOTOUT    023650          3067    3071#
NOTT      023066          3590    3595#
NOTYET    023432          3017    3021#
NO18      024476          3415    3417#
NUMF    = 000054          2852#   2864    2865
ONE     = 000200            69#    669
ONPIC     024610          3448    3450#
OPADDR    006126          1615    1739#
OPEN    = 024646           249#   1185    1188    2401
OPERR     004572          1244    1257    1260#
OPNEXT    004536          1255#   1274
OUT       003002           182    725     785#    880
OUTCH   = 104610           185#    789    2520
OUTCO     024056          3114    3122    3125#
OUTDEV    001634           413#    477#    796    2260    2381    2382    2383    2384    2385    3486    3495    3506    350
                          3514    3516    3521
OUTW      002764           777    781#
OUTX      003064           797    799     801#
OUTY      003042           791    794     796#    3492    3509
OUTZ      003012           572    786     788#
P       =%000000            52#   2061#   2071    2073#   2075#   2076    2077#   2078#   2081#   2086    2094    2108#   2110#
                          2111    2113#   2117    2126#   2131    2141#
PACKC   = 104616           191#    508     536    1018    1023    1485    1493
PACKX     003136           192    842#
PARAMS  = ****** U         2717
PARTBA    005206           220   1447#
PARTST  = 104646           219#    982    1292    1331
PBAR      003176           848    854#
PC      =%000007            49#    471#    478     482#    518#    529     533#    657#   1002#   1012#   1039#   1143#   1168#
                          1192#   1254#   1291#   1460#   1468#   1487    1494    1601#   1615#   1616#   1731#   1736#   1761#
                          1834#   1836#   1842#   1845#   1853#   1862#   1893#   1905#   1914#   1962#   1963#   2101#   2123#
                          2143#   2174#   2178#   2187#   2467#   2470#   2542#   2548#   2603#   2630    2815#   2831#   2963#
                          2964#   2967#   2987#   2989#   2990#   3001#   3003#   3009#   3043#   3073#   3074#   3103#   3111#
                          3125#   3126#   3139#   3142#   3147#   3148#   3327#   3352#   3366#   3376#   3384#   3409#   3513#
                          3561#   3572#   3616#
PCF       001604           399#    485     495#    519     522#    974#    990#   1130    1147    1155#   1191    1193#   3423
PC1       003524           264     495     977     990     991#
PC2       003504           983     985#    986
PERR      004672          1308#   1450
PLUGCO    023344          2989    2990    2997#
POPJ    = 000207           255#    991    1000    1207    1235    1439    2160    2317    2340    2454    2506    2524    2552
                          2570    2601    2622    2625
```

```
POSCO     024300          3361    3364#
POSOUT    024050          3118    3123#
POWDO     006700          1954#   1964
POWF      006650          1744    1944#
POWF1     006644          1942#   1946
POWS      006670          1951#   1976
PPS     = 177554            81#   2371
PRESET  = 170406          2780#
PRINT   = 104400           253#    484     502     795     822     858    1461    1469    2128    2144    2148    2150    2462
                          2468
PRINTA    002316           467     568#
PRINTC  = 104604           181#    723    1016    1054    1066    1529    2029    2154    2156    2159    2458    2461
PRINT2  = 104642           215#    479     483     488     800    1512    1532    2456    2471    2476
PRIN2A    002626           216     722#
PRNTF     007012          1754    2004#
PRNTLN  = 104626           203#    487    1052
PRNTP     007506          2146    2149    2151#
PRNTS     007474          2146#   2467    2470
PRNT1     007532          2123    2158#
PRNT2     007522           482    2153    2155#
PROC      003454           518     958     976#   1003    1104    1158    2358
PROCES    003452           975#    979     997    1192
PROGIO    010336           276    2356#
PRS     = 177550            80#    500     783    2350    2369
PSCAN     003444           974#   1143
PTEN      007064          2015    2019#   2194
PTR     =%000002            43#     54      63     481#    552     557#    696#    701#    744     746#    816#    904#    905#
                           909#    912     916#    921#   1172    1195#   1231#   1239    1240    1261#   1272#   1347#   1348#
                          1349#   1351#   1352#   1364#   1365    1375#   1382#   1383    1385#   1386    1423    1425    1427#
                          1428    1432    1435#   1436#   1462#   1475    1478#   1487#   1490    1602    1603    1606    1608
                          1653#   1658#   1661#   1662#   1665    1666    1669#   1673#   2030#   2145#   2147#   2157    2175#
                          2179    2194#   2198#   2237#   2294#   2295    2299    2300#   2301#   2302#   2307    2309    2311
                          2348#   2349#   2350    2352    2354#   2402#   2407#   2409#   2412#   2427#   2466#   2469#   2472#
                          2474#   2491#   2495#   2496    2502#   2505    2543    2550#   2612#
PUTCO     024264          3001    3360#
PUTF      005630          1627#   1745
PUTTXT    023402          3011    3019#
PWRDWN    005752           102    1680#
PWREGS    006020          1620    1697#   1839
PWRON     006034           897    1703#
PWRUP     006002          1681    1694#
READC   = 104606           183#    505    1013    1021    1479
READF     007542          1753    2173#
REL     = 000006          1577#
RESETD    024504          2845    3423#
RETLOC    024110          3037    3138#
RETNEW    024312          2991    3077    3149    3370#
RETNOW    024076          3133    3135#   3136
RETURN    003516           279     990#
ROOTGO    010244          2308    2310    2313#
RT      = ****** U        2720    2730    2811    2854    2868    2906    2951    3157
RTPAR   = 000211          2765#   3016    3032    3049
RUBIT     003200           844     855#
RUBX      003210           846     851     856     859#
```

```
RUBX2     003214          772     782     784     835     860#    873     879
R5      =%000005           46#     57#     66     452     453*    454     503*    504     513     585*    587     591     594*
                          595*    614     619     620*    622     624     625*    626*    634     656     659*    670*    673*
                          722     724     744     746     781     802*    823*    861*   1034*   1035*   1036*   1038    1113*
                         1115*   1214*   1217*   1218    1233    1237    1241    1242*   1276*   1277*   1278*   1279*   1283
                         1285*   1289    1296    1343*   1345    1346    1374*   1376*   1377*   1409*   1430*   1452    1474*
                         1476*   1477    1495    1517*   2053*   2080    2085*   2086    2088*   2089    2091*   2093    2094
                         2096    2122    2257*   2265*   2293*   2306*   2307    2447*   2448    2450*   2451    2457    2459
                         2463    2465    2466    2469    2472    2477    2492*   2503*   2540*   2600*   2611*   2615*   2616*
                         2617    2627    2630    3530*
SCHAR     003572          325    1015#   1019
SCONL     003566          327     781    1014#
SCONT     003564          326    1013#
SERR      003624          973    1011    1026#   1037    1093
SET     = 004222          280    1167#
SFIND     003610         1021#   1024
SFOUND    003620          331    1023#
SGNF      006532         1750    1895#   1904
SGN1      006536         1897#   1949
SIGN      006734         1862    1914    1975#
SIGND     006460         1874#   1933
SILENT    025026         3543    3546#
SKPLPR  = 104652          223#   1244    1267    1321
SKPNON  = 104654          225#    516     588     655    2226
SKPNOX    002372          226     602#
SORTB     002406          178     614#    618
SORTC   = 104602          179#    772     776    1480    2357
SORTD     002440          180     634#    638
SORTJ   = 104600          177#   1017    1022    1175    1178    1484    1514
SOUND     002424          616     622#
SOX       002420          593     603     619#    639     706     737     745     758
SP      =%000006           48#     439*    451     452*    453     454*    466*    469*    470*    471     480*    481     494*
                          568*    569*    570     571*    656*    659     785*    793     801     807*    817*    818*    854
                          855     950*    996     999    1047*   1048*   1128*   1129*   1130*   1141*   1142*   1144    1145
                         1155    1156    1157    1172*   1181*   1182*   1190*   1191*   1193    1194    1195    1196    1197*
                         1198*   1202    1206*   1218    1220    1239*   1240*   1241*   1246*   1247*   1249*   1282*   1283*
                         1285    1296*   1313*   1314*   1315*   1318*   1322*   1325*   1330*   1335*   1336*   1347    1348
                         1381*   1386    1423    1428    1432    1435    1436    1438*   1447    1448*   1449    1470*   1471*
                         1473*   1474    1475*   1500*   1501    1502    1586*   1587*   1588*   1589*   1590*   1591*   1592
                         1596*   1597    1607    1608*   1619    1627    1628    1654*   1657*   1661    1666*   1669    1682*
                         1683*   1684*   1685*   1686*   1687*   1688    1694*   1697    1698    1699    1700    1701    1702
                         1757*   1763    1835*   1838*   1847*   1850    1872*   1876*   1883    1895*   1900    1931*   1947*
                         1950*   1982*   1983*   1984    1985    1986    1987*   1989*   1990*   1991*   1975*   1977*   1978*
                         1982*   1993    2004*   2012*   2023*   2028*   2034*   2040*   2041*   2042*   2053    2054    2055
                         2079*   2080*   2081    2096    2102*   2139*   2145    2173    2179*   2192*   2195    2204*   2206*
                         2207    2232*   2233*   2259*   2263    2264    2277*   2278    2287    2295*   2299*   2315    2316
                         2407    2408*   2411*   2453    2517*   2521    2543*   2546*   2549    2550    2562*   2565*   2566
                         2583*   2584*   2594*   2595*   2596*   2597*   2606*   2611    2612    2640*   2641    2642*   2796*
                         2826*   2827    2997*   3000    3021*   3028    3047*   3075    3108*   3110*   3115    3390*   3406
                         3408    3417    3424    3434*   3453    3464    3485*   3519    3529    3539*   3540    3571    3583*
SPFC1   = 001445          366#   1484
SPTOL   = 104634          209#    515     532     651     949    1169    1255    2356
SPNORX    002126          210     585     839#
```

```
SPNXT    003124          832#   1242
SQUEND   010252         2298    2316#
SQX      010250         2312    2315#   2428
SRETN    002226          330     534     536#    538
SRNLST = 001400          324#   1022
STACK  = 000000          819     821    1174    1185    1186    1188    1189    1200    1203    1223    1224    1498    1578#
                        2043    2047    2235    2236    2303    2401    2403    2404    2413    2414    2416    2422    2425
                        2429    2430    2431    2432    2433    2545
STACKO   001706           89      95     105     439#    494     807    2642    2645    2796    2799    2863
START  = 104632          207#   1084    1089
STARTV   001646          418#    542     892    1342    1408*   2447
STARTX   002034          208     278     494#    521     559    2822
STATUS = 177776           82#    451*    498*
STILON   024170         3333#   3334
STLIN    003626          533    1012    1034#
STOP     001704          272     434#
STOPIT   024650         3462    3464#
STPADR = 000320         2783#   2787    2803*   2804#
STPFLG   024664         2957*   3332*   3333    3406*   3424*   3461    3464*   3473#
STPINT   024634         2803    3461#
STRING = 000030         1466#   1472    1500
SUBEND = 024676         3478#   3502
SUBFF    006222         1741    1771#
SWITCH = 001631          410#    648*    669*    672*   1060    1067    1095    1134    1141    1145*   1151    1270*   2181
                        2227
TASK   = 104656          227#   1459    1467
TASKX    005412          228    1514#   1522    1526
TASK4    005410          310     311    1513#   1516    1528    1533
TCLEAR   024362         2964    3074    3390#   3561
TCRLF    005404          307    1512#
TCRLF2   005464          308    1532#
TDUMP    010564          309    2447#
TDUMP1   010570         2448#   2452
TDUMP2   010612         2449    2456#
TDUMP3   010600         2451#   2478
TDUMP4   010656         2464    2471#
TEMP   =%000000           41#     52      61#    539*    540*    541     542     842*    849*    850     854*    855     857*
                         859     894     895    906*    907*    916     921    1144*   1153    1220*   1221    1233*   1336
                        1342*   1344    1364    1375    1382    1592*   1593    1607*   1612*   1613*   1614*   1615    1628*
                        1629*   1632*   1633*   1653    1658    1672*   1673    1793*   1794    1800*   1811*   1817*   1868*
                        1869*   1870*   1874    1928*   1980*   2031*   2038*   2541*   2769
TEN      007102         2025#   2037    2198    2234    2242
TENADR = 000300         2784#   2808*   2809*
TENIN  = 175610         2785#   3587*   3608
TENINT   023116         2808    3608#
TENOUT = 175614         2786#   3486    3521    3597    3599*
TERM   = 000000         2807    2838    2902    2948    3520    3576
TERM8  = 001412          333#    776     779     787
TERSIZ = 000150         2900#   2903#   2916    2942
TESTC  = 104620          193#   1076    1250    1309
TESTX    002334          198     585#
TEST10   023074         3597#   3598
TEX      002366          586     588     590     594#    636     699     703     734     736     755     757
TEXT     031772         2933    2935#
```

```
THISLN   001606          400#    697#    903     974    1057    1142    1150#
THROUG = 000004         1174    1200    1498    1575#   2545
TINTR    005430          305    1519#
TJUMPS   024144          [illegible]  3073    3148    3324#
TKS    = 177560          [illegible]7#    412     476     499#   1696#   2251    2277    2370    2826    3583
TLIST  = 001441          360#   1175    1178    1480
TOF      007260          [illegible]56    2072    2074    2076#
TOFOCL   023157         3591#   3611    3620#
TOG      007246         2069    2071#
TOHOST   023156         2949#   3523    3586#   3595    3619#
TOOBIG   024242         3340    3348#
TOP      001650          419#   1118
TOR      007310         2087    2089#   2095
TOS      007324         2090    2093#   2097
TOT      007276         2085#   2092
TPR      003534          301     312     996#
TPR1     003540          302     313     999#
TPS    = 177564           78#    413     477     796    2372    3506
TQUOT    005442          306    1524#   1530
TRAPH    001714          108     451#
TSIZE  = 001000         2734#   2753#   2754
TSTERR   024472         2835    3415#
TSTTXT   024676         3493    3506#
TTSIZE = 006000         2754#   2934    2962
TXTDTS   024720         3488    3513#
TXTDUN   025156         3545    3554    3562    3571#
TXTHAN   025000         3513    3539#   3616
TXTJMP   023642         3080    3069#
TXTPTR   024666         2962#   3010    3048    3064    3068#   3069    3071#   3324    3346    3348    3393    3474#
TYPE     005224          281    1459#   1464
UTX      003220          872#    877
VERR     004670         1288    1307#   1311    1312
VIS      023304         2974    2980    2984#
WAYOUT   023654         3070    3072#
WGO      003742         1059    1061    1069#
WHIPV    004770         1335#   2542
WHOOPS   001640          415#    497#   1680    1688#   1694    1695#
WRED     003730         1062    1066#
WRIG     003724         1063#   1068
WRITE    003646          284    1046#
WRITE1   003670         1051    1053#   1056
WRITE2   003654         1049#   1064
WTESTG   003712         1049    1060#
XABS     006376          142    1841#
XADC     011070          154    2559#
XCHARG   010772         2521#   2530
XCHMO    010756         2515#   2523
XCHR     010760          148    2516#
XCHR1    011004         2519    2526#
XCLR     024130         2876    3147#
XDELET   003272           99     897#
XDIS     023700         2884    3085#
XDOX     003336          909#    915     917     920     922
XDTHIS   003360          913     919#
```

FOCGT,002 CROSS REFERENCE TABLE

```
XD3       003276            898#    900
XECUTE    003544            285    1002#
XEL       011322           2611    2630#
XEM       011210           2600#   2631
XER       011316           2614    2627#
XERR      011326           2632#
XET       011250           2615#
XEX       011134            152    2581#
XEXIT     011302           2619    2623#   2628
XEX2      011152           2587#   2598
XEX2A     011214           2590    2592    2603#
XEX3      011226           2588    2607#
XEX4      011240           2586    2608    2611#
XFCLK     010272            164    2337#
XF8BR     011026            146    2540#
XITR      006406            162    1847#
XI33      010022            188    2250#
XI33X     010044           2256#   2273
XLP       024064           2878    3131#
XMOV      023254           2894    2976#
XOUT      010052            186    2259#   3484
XPRNTL    003100            204     816#
XPT       023264           2892    2979#
XPTR    = 000002           1572#
XRAN      010704            150    2491#
XROL      010726           2494    2496#   2504
XSET      023274           2890    2982#
XSGN      006402            144    1844#
XSKP      023514           2888    3043#
XSQT      010154            156    2293#
XTSTLP    002636            224     733#
XTXT      023364           2886    3009#
XVEC      023244           2896    2972#
XXCO      024000           2882    3108#
XYCO      024004           2880    3110#
Z       = 020724           2345#   2348    2349
ZERODM    006370           1838#   1951
.       = 023160             86#    112#    132#    134     168     175     177     179     181     183     185     187     189
                            191     193     199     201     203     205     207     209     211     213     215     217     219
                            221     223     225     227     229     234     258     304     317     324     329     333     352
                            360     366     371     378     386     393     395#    400     402     404     405     406     408
                            410     414     424     428#    501     543     774     834     992    1167    1290    1320    1404
                           1406    1482    1634    1808    1851    1866    1890    1920    1923    1984    1989    2057    2063
                           2098    2189    2254    2262    2285    2296    2338    2361    2368    2378    2410    2460    2498
                           2636    2646    2795#   2800    2822#   2825#   2839#   2844#   2863#   2865#   2919#   2922#   2934#
                           2942#   3151    3319#   3478    3484#   3492#   3495#   3498#   3502#   3518    3547    3577#
```

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE