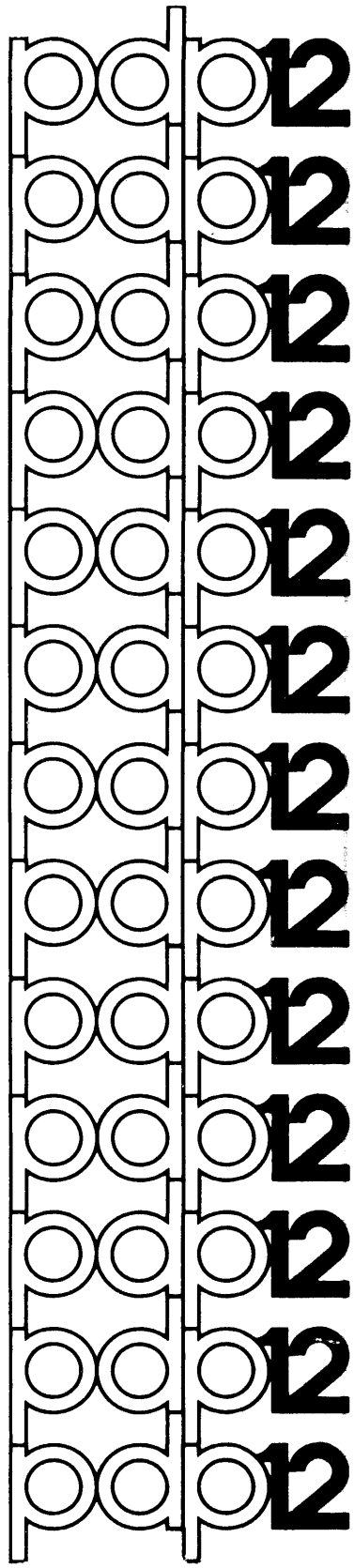
digital



FPP ASSEMBLER

USER'S GUIDE

DEC-12-AQZA-D

FPP ASSEMBLER

USER'S GUIDE

For additional copies, order DEC-12-AQZA-D from Program Library,

Digital Equipment Corporation, Maynard, Mass. 01754 Price: $3.00

First Printing, April, 1971

The material in this manual is for information purposes only and is subject to change without notice.

CONTENTS

1.0 INTRODUCTION

The FPP Assembler, designed for the PDP-12 Floating Point Processor, translates PDP-8 and floating point mnemonic operation codes in a source program into binary codes in two passes, the first to assign numeric values to symbols and place them in the symbol table, and the second to generate the binary coding and program listing. Using the FPP-12 hardware, an entire set of floating point instructions can be implemented for quicker calculations and expanded capabilities. Numeric values can be calculated as 2 word integers, 2 word double precision fractions, or 3 word floating point values. Assembler fundamentals are discussed only briefly in this document. Refer to the FPP User's Guide, DEC-12-GQZA-D, for detailed information on the floating point processor and its instruction set; refer to the LAP6-DIAL Programmer's Reference Manual, DEC-12-SE2D-D, for more information on assemblers.

2.0 HARDWARE REQUIREMENTS

The minimum hardware configuration is:

PDP-12/20 with 8K of core memory
RK8, DF32 (two are needed) and RSØ8 disks are among the supported options.

An FPP-12 is strongly recommended.

3.0 STATEMENT SYNTAX

A source program is a sequence of coding statements generally entered via the Teletype[1] in the general format:

tag, instruction expression/comment

A physical line of coding may be up to 95 characters long and is terminated by a carriage return; however, on the program listing, lines will be truncated. A semicolon can be used in a line of code (except in the comment field) to terminate a logical statement, thus permitting several statements to be typed on a single Teletype line. However, a set of logical statements separated by semicolons must not exceed the 95 character limit.

[1]Teletype is a registered trademark of Teletype Corporation.

A space is required in a statement:

1. after an instruction mnemonic
2. after an indirect I
3. before a slash (/) used to indicate a comment
4. as an OR operator.

Multiple spaces or tabs are equivalent to a single space. These characters are optional after a comma in a tag and before a statement.

3.1 Tags

A statement tag is indicated by preceding the statement to be labeled with a user-defined symbol followed by a comma. (Refer to Section 3.2). This format assigns the current value of the location counter to the tag. A mnemonic op code or defined pseudo-op cannot be a statement tag.

3.2 Expressions

The FPP Assembler interprets three expression formats:

1. Integer (used as addresses only)
2. Double precision (floating point with no exponent)
3. Floating point.

An expression can contain:

1. A user-defined symbol (equate statement or tag)
2. A period (.) to indicate the current location
3. PDP-8 operator and IOT instructions
4. An expression and a symbol or number combined by an operator.

FPP instructions and PDP-8 memory reference instructions are illegal symbols in expressions.

User symbols can be 1 to 6 alphanumeric characters in length and must have an initial alphabetic character. Any additional characters are ignored. Thus, the symbols

```
A
A1234
```

are acceptable, but in the symbol

```
ASYMBOLMAYBEMORETHAN6CHARACTERS
```

only the first six characters are stored as the symbol name. In this case, all characters after ASYMBO are ignored.

All integer expressions are computed in 24 bit 2's complement arithmetic and then truncated as necessary (e.g., to 15 bits for 2-word FPP memory reference instructions and to 12 bits for expressions).

The following are examples of legal integer (address) expressions:

```
START+1
123
BUFSIZ*2+BUFFER
-3
POINTER&76ØØ+3ØØ
(ADDRES+2
```

The radix pseudo-ops, OCTAL and DECIMAL, are required for integer expressions; floating point and double precision expressions always have a radix of 1Ø. All the symbols and numbers used in an expression must be of the same type. Note that a decimal number of more than eight digits may cause improper conversion.

Double precision numbers are treated as follows:

a. Double precision numbers equal to or greater than 1.0 in magnitude are converted to 2 word integers with the fractional part ignored. Thus,

   100.D is converted to Ø;Ø144 (octal)

   4Ø97.D is converted to 1;1

b. Double precision numbers less than 1.0 in magnitude are converted to 2 word fractions (floating point numbers unnormalized to have zero exponent). Thus,

   .5D is converted to 2ØØØ;Ø (octal)

   .125D is converted to Ø4ØØ;Ø

Note that numbers greater than $2^{23}-1$ should not be used in double precision format.

Examples of acceptable double precision numbers are:

1ØØ.D	1Ø.2D1	1Ø.2D+1	1ØD
1Ø.1D	1Ø2.ØD-1	1Ø.D	1ØD-2
.1Ø1D	.2D3	.Ø2D4	1D3

Some acceptable floating point numbers are:

1ØØ.	1Ø.2E1	1Ø.2E+1	1ØE
1Ø.1	1Ø2.ØE-1	1Ø.E	1ØE-2
.1Ø1	.2E3	.Ø2E4	1E3

3.3 Comments

A comment is a note added by the programmer at the end of a line of code, usually to indicate the logical sequence of the program. A slash (/), preceded by a space, is typed to specify the start of a comment.

4.0 ARITHMETIC AND LOGICAL OPERATORS

The operators used by the FPP Assembler and their functions in combining numbers or symbols to form expressions are:

OPERATOR	FUNCTION
+	2's complement addition
-	2's complement subtraction
*	multiplication
/	division (double precision and floating point only)
&	logical AND (integers only)
(tab)	inclusive OR (integers only)
(space)	inclusive OR used to separate two symbolic (integer) operators
!	inclusive OR (integer only)

5.0 INSTRUCTIONS

PDP-8 memory reference instructions and FPP codes are the legal defined mnemonics for use with the FPP Assembler. The following table lists these codes and their formats in an object program. Angle brackets (<>) indicate a required value; curved brackets ({}) indicate optional fields; the notation ␣ indicates typing a space. (The instructions are described in Appendix C.)

5.1 PDP-8 Memory Reference - 1 Word

<mnemonic> {␣I} ␣ <address expression[1]>

AND	JMP
DCA	JMS
ISZ	TAD

5.2 PDP-8 Operate and IOT - 1 Word

<expression of mnemonics combined by operators>

CIA	OSR	SPA	RDF	FPINT
CLA	RAL	SZA	RIB	FPICL
CLL	RAR	SZL	RIF	FPCOM
CMA	RTL	CDF	RMF	FPHLT
CML	RTR	KCC	TCF	FPST
HLT	SKP	KRB	TLS	FPRST
IAC	SMA	KRS	TPC	FPIST
LAS	SNA	KSF	TSF	
NOP	SNL			

5.3 FPP Memory Reference Format 1 - 1 or 2 Words

<mnemonic>{L}{␣I}␣<address expression>{,<index expression>{+}}

For this group of FPP instructions, the index expression is an integer expression truncated to 3 bits, so that it designates an index register. A plus sign can be placed after the index expression to indicate pre-incrementation of the index register. The L suffix with the mnemonic indicates 2 word format. If an L is not given, the Assembler will use 1 word format whenever possible. (Refer to the BASE pseudo-op, Section 9.5 and Referencing Memory, Section 10.0.)

FADD	FMUL	TRAP3
FADDM	FMULM	TRAP4
FDIV	FSTA	TRAP5
FLDA	FSUB	TRAP6
	JXN	TRAP7

5.4 FPP Memory Reference Format 2 - 2 Words

<mnemonic>␣<address expression>

---

[1] An address expression is a truncated integer expression.

For these instructions, the address is truncated to 15 bits.

JA	JGE
JAL	JNE
JEQ	JSA
JGT	JSR
JLT	SETB
JLE	SETX

5.5 FPP Index Register Format 1 - 1 Word

<mnemonic>␣<index expression>

The index expression is truncated to 3 bits and is the index register designator.

ALN
XTA
ATX

5.6 FPP Index Register Format 2 - 2 Words

<mnemonic> <value expression>,<index expression>

The value expression is an integer expression truncated to 12 bits.

LDX
ADDX

5.7 FPP Operates

<mnemonic>

FCLA	FNOP	JAC
FEXIT	FNORM	STARTD
FNEG	FPAUSE	STARTF

6.0 LITERALS

By starting an expression with a left parenthesis or square bracket (as explained below), the value after it is taken "literally" by the FPP assembler. There is then no need to specify an address or tag that contains the value. Internally the value of the literal expression is the address of the word(s) generated by the Assembler that contains the evaluated expression. Literals are used with memory reference instructions as follows:

1. PDP-8 memory reference instructions can use a literal for the address. If the expression starts with a left parenthesis (( ), then the literal is placed at the end of the current page; if it starts with a left bracket ([), the literal is placed at the end of page Ø. Integer literals are 1 word long, double precision are 2, and floating point are 3. Literal tables are built backwards from the end of the page so that the most recently encountered literal has the lowest core address. Note that if the origin is set back into a previously used page, the literal block is reset and any literal defined subsequently will destroy any previous literals. Similarly, if page Ø variables and constants are defined in the program, sufficient space must be left at the end of page Ø for all literals that will subsequently be defined using left brackets. Because locations Ø-17 are generally used for interrupts and auto-index registers, there may be only 112_{10} (160_8) literals in page Ø.

   The following examples illustrate the use of literal expressions with some PDP-8 instructions.

TAD (POINTER	generates 1 word literal with the lower 12 bits of the address of POINTER at end of a current page.
JMS SUB (.333D	generates 2 word double precision literal at end of current page and a pointer to the 2 word literal.
TAD [1.]	generates a 3 word floating point literal at end of page Ø. Only exponent of floating point number will be "TADed" at execution time.

2. Any literals defined by an FPP memory reference instruction are placed at the location specified by the LITORG pseudo-op. The enclosures left parenthesis and left bracket are equivalent with FPP literals. Literals that are floating point or integer occupy 3 words, double precision occupy 2 words. Be sure to leave enough room after the LITORG for all FPP literals.

   NOTE: FPP literals are not dumped in the order in which they occur, but in order of magnitude.

The following examples illustrate the use of literal expressions with FPP memory reference instructions.

FLDA (POINTER	generates 2 word address literal at LITORG. 15 bits of address literal are put into lower 15 bits of FAC (for JAC instruction), assuming instruction is executed in floating point mode.
FADD (2.D	generates 2 word double precision literal at LITORG, assuming instruction is executed in double precision mode.

```
FLDA  (1.Ø          generates 3 word floating point literal
                    at LITORG.
```

7.0 LINKS

If reference is made to an address that is not on the same page as the instruction, the FPP Assembler creates an indirect address linkage on the current page. The address can, therefore, be accessed during the second pass of the Assembler. For example, the coding:

```
ORG 2ØØ
TAD A
 .
 .
 .
PAGE
A,1Ø25
```

is equivalent to

```
ORG 2ØØ
TAD I X
X,A
 .
 .
 .
PAGE
A,1Ø25
```

All instructions generating links are flagged with the *LG* message. Each such message will be included in the error count.

8.0 DATA SPECIFICATION

A logical line of code may consist of just an expression, in one of the three acceptable formats. Such expressions can function as flags, pointers, constants, or symbols.

9.0 PSEUDO-OPERATIONS

A pseudo-operation is a defined mnemonic code that is included in the object program as a logical line to control some functions of the Assembler. No binary code is generated by a pseudo-op (except TEXT). The FPP Assembler pseudo-ops and their functions are listed below.

9.1 = (equate) The symbol to the left of the = is assigned the value and format to the right of it.

9.2 OCTAL — The radix is set for integer conversion to octal numbers; digits 8 and 9 are not flagged in octal numbers.

9.3 DECIMAL — The radix is set for integer conversion to decimal numbers.

9.4 PAGE — The Assembler's output is set to the start of the next core page.

9.5 BASE expr — The location of the FPP base register is assigned so that displacements for 1-word FPP instructions can be computed. (The program must actually set the base register.) To denote the location of the base register, the pseudo-op BASE is followed by the address (expr), e.g., BASE A+3. The expr used with BASE may not contain any symbols that are defined after the BASE pseudo-op. The correct sequence is illustrated by:

```
A,Ø.Ø
B,Ø.Ø
BASE A
FLDA A
```

If no BASE pseudo-op is included, all FPP memory reference instructions will be 2 words. Refer to section 10.0 for additional information on referencing memory.

9.6 TEXT — A string of text may be entered by using the pseudo-op TEXT followed by a space or tab, a delimiting character, a string of text, and the same delimiting character, issued in that order. The first printing character after TEXT is the delimiter and the text string is all the characters that follow it until the next occurrence of the delimiter or a carriage return. The characters space, tab, ;, and / can not be delimiters. For example

```
TEXT %DATA%
```

causes the word DATA to be printed with the code at assembly time as:

```
221 Ø4Ø1  TEXT %DATA%
222 24Ø1
```

9.7 END — Input is terminated. (This pseudo-op is optional, and is never printed on the listing.)

9.8 CHAIN "name" unit

Terminate assembly of current program and initiate assembly of the file name enclosed in quotes located on unit Ø-7. During pass 1 (and pass 2 if a LISTOF pseudo-op has been issued) of the Assembler, a successful chaining is indicated by a message in the form

CHAINING TO name

on the Teletype. Unit is always an expression, that has a value Ø-7, indicating the device. If no unit is specified, Ø is assumed. The programs that are to be chained together must all be on LINCtapes, or all on disks. Disk units 1Ø-17 are also referenced by the corresponding value in the range Ø-7

9.9 ORG expr — The starting location in memory for the Assembler's output is assigned the value of the lower 15 bits of the address expression expr to set the location counter. For example, to set the origin at location 4ØØ of field 1, the pseudo-op used is ORG 1Ø4ØØ.

If the ORG pseudo-op is omitted, an origin of 2ØØ in field Ø is assumed.

9.10 LITORG expr — Any literals generated by floating point instructions are to be placed at location expr. All addresses and floating point literals are stored as 3 word quantities; double precision literals are stored as 2 words. It is the user's responsibility to leave sufficient space after the literal origin to accommodate all floating point literals. If no LITORG pseudo-op is included in the program, all literals are placed immediately after the END statement. Only the last LITORG in a program is used.

9.11 LISTON — Turn program listing on. (There is no effect if listing is already on.)

9.12 LISTOF — Turn program listing off. (There is no effect if listing is already off.) This mnemonic is never listed.

9.13 EJECT — The mnemonic EJECT is printed and the next line of the program listing is started at the top of next page of line printer paper. (There is no effect if output is to the Teletype or after a LISTOF pseudo-op.)

9.14 IFnnn (conditional assembly)

The FPP Assembler has five conditional pseudo-ops. Four of them require a numeric parameter:

pseudo-op	function
IFZERO n	assemble if zero
IFNZRO n	assemble if not zero
IFPOS n	assemble if positive
IFNEG n	assemble if negative

where n is an integer expression. For each of the above conditional pseudo-ops, the expression n is evaluated and, if it fulfills the conditions of the pseudo-op (e.g., n equals zero for IFZERO), the subsequent coding is assembled. If the condition is not met, the subsequent coding is ignored until a $ is encountered as an expression. Assembly is continued after the $. Extra $ characters are treated as null input.

The fifth pseudo-op is used in the format:

IFREF symbol — assemble if symbol was previously defined or referenced

where symbol may be defined or undefined. When an IFREF statement is encountered, subsequent coding is assembled if the symbol after the pseudo-op has been defined or referenced in a previous statement. Note that use of a symbol with an IFREF pseudo-op or in a statement that was skipped during assembly because the condition required by a preceding conditional pseudo-op was not met does not constitute a reference to the symbol.

If the symbol has not been previously defined or referenced assembly is continued as above with the other conditional pseudo-ops.

Use of some of the conditional assembly pseudo-ops is illustrated in the next example.

```
0001                          IFPOS   -1
0002                    A,    0,0
0003                          $
0004                          IFNEG   -1
0005 00200 0000 B,    0,0
     00201 0000
     00202 0000
0006                          $
0007                          IFREF   A
0010                          TAD     A
0011                          $
0012                          IFREF   B
0013 00203 1200               TAD     B
0014                          $
0015                          IFREF   C
0016                          TAD     C
0017                          $
```

```
B       00200
```

10.0 REFERENCING MEMORY

A PDP-12 computer with an FPP-12 is basically a 32K machine. All of this memory is referenced via the 15 bit address field provided by the 2 word memory reference instructions. When it is necessary to conserve memory, the short form (1 word) of the memory reference instructions and the base page can be used. Those instructions that have a floating point operand can use this short form:

FADD	FDIV	FMUL	FSTA
FADDM	FLDA	FMULM	FSUB

The base page is a movable page Ø assigned by the user. To determine the location of the operand, the displacement field (address expression) is multiplied by 3 and added to the contents of the base register. Thus, using the single word form of the instruction, any location within 128*3 locations of the base register can be referenced. (Only 128*3 locations can be accessed because the displacement field has only 7 bits.) The location of the base page (via BASE) and the operands (via ORG,=,etc.) must be defined in the coding before the FPP instruction. Then the short form of the instruction will be executed unless the suffix L is added, forcing the long (2 word) form.

Consider the following example of the BASE pseudo-op:

```
     ORG 2ØØ
      A,Ø.Ø
      B,Ø.Ø
      C,Ø.Ø
      D,Ø.Ø
     BASE 2ØØ
      .
      .
      .
     SETB 2ØØ
     FLDA A
     FADD B
     FMUL C
     FSTA D        /D=(A+B)*C
```

This same program can be written with a subroutine:

```
     ORG 2ØØ
   A,Ø.
   B,Ø.
   C,Ø.
   D,Ø.
     .
     .
     .
     .
```

```
      SETB 2ØØ
      JSA SUBR
      .
      .
      .
      BASE Ø
SUBR,   Ø;Ø          /leave 3 words for JSA
      FLDA Ø         /A
      FADD 3         /B
      FMUL 6         /C
      FSTA 11        /D
      JA SUBR        /return
```

This routine performs the same operation as the first one. The values Ø, 3, 6, and 11 are used with BASE Ø so that the Assembler generates the correct 1 word instructions.

11.0 USING THE ASSEMBLER

The FPP Assembler runs as a DIAL-MS[1] binary program. Therefore, after entering the source program into the source working area via DIAL-MS, type the command

```
→LO FPPASM,Ø )      (LINCtape systems)
```

or

```
→LO FPPASM,1Ø )     (disk systems)
```

to load the Assembler (it must be on the systems unit). The starting address is 112ØØ. If a CHAINed source file is being assembled, the source working area must contain the first file to be chained or a single line that chains to the first file. The resulting binary will be placed in the binary working area.

All error messages are printed in the listing above the line in which the error occurred. If no listing is being generated, the error messages are printed with the line number of the incorrect statement on the on-line output device. The FPP Assembler error messages are listed in Appendix A.

12.0 EXITING FROM THE ASSEMBLER

Two methods are available for exiting from the FPP Assembler:

CTRL/D - stops assembly of program and control returns to DIAL

CTRL/L - the listing is terminated but the assembly is continued.

[1] LAP6-DIAL-MS is referred to as DIAL-MS in this manual.

13.0 EXAMPLE

The following example illustrates some of the FPP instructions and their use with the FPP Assembler.

```
0001                          ORG     1000
0002                          PI=3.14159
0003 01000 1030 SINE,         JA      0               /ENTRY POINT FOR SINE(X)
     01001 0000
0004                                                  /CALLED BY      JSA     SINE
0005                                                  /WITH X IN FAC
0006                                                  /RETURNS SINE(X) IN FAC
0007 01002 3400               FDIV    (PI*2.          /DIVIDE ARG BY TWO PI
     01003 1100
0010 01004 6400               FSTA    X               /SAVE ARG
     01005 1072
0011 01006 0010               ALN     0               /INT(X)
0012 01007 0004               FNORM
0013 01010 2400               FSUB    X               /INT(X)-X
     01011 1072
0014 01012 4400               FMUL    (4.             /4*(INT(X)-X)
     01013 1103
0015 01014 0003 SINE2,        FNEG
0016 01015 6400               FSTA    X               /SAVE NEW VALUE OF X
     01016 1072
0017                                                  /THIS IS THE LOOP THAT
0020                                                  /NORMALIZES THE RANGE OF X
0021 01017 1060               JGT     SINE3           /ABS(X)
     01020 1022
0022 01021 0003               FNEG
0023 01022 2400 SINE3,        FSUB    (1.0            /IF(ABS(X).LE.0)GO TO SINE5
     01023 1106
0024 01024 0400               FLDA    X               /2*SGN(X)-X
     01025 1072
0025 01026 1050               JLT     SINE4
     01027 1034
0026 01030 2400               FSUB    (2.0
     01031 1111
0027 01032 1030               JA      SINE2
     01033 1014
0030 01034 1400 SINE4,        FADD    (2.0
     01035 1111
0031 01036 1030               JA      SINE2
     01037 1014
0032 01040 0400 SINE5,        FLDA    X               /X2=X*X
     01041 1072
0033 01042 4400               FMUL    X
     01043 1072
0034 01044 6400               FSTA    X2
     01045 1075
0035 01046 4400               FMUL    (.15148418E-3   /X2*C9
     01047 1114
0036 01050 1400               FADD    (-.46737656E-2  /C7+X2*C9
     01051 1117
0037 01052 4400               FMUL    X2              /X2*(C7+X2*C9)
     01053 1075
0040 01054 1400               FADD    (.7968968E-1    /C5+X2*(C7+X2*C9)
     01055 1122
0041 01056 4400               FMUL    X2              /X2*(C5+X2*(C7+X2*C9))
     01057 1075
```

```
0042 01060 1400          FADD     (=.64596371      /C3+X2*(C5+X2*(C7+X2*C9))
     01061 1125
0043 01062 4400          FMUL     X2               /X2*(C3+X2*(C5+X2*(C7+X2*C9)))
     01063 1075
0044 01064 1400          FADD     (PI/2.           /C1+X2*(C3+X2*(C5+X2*(C7+X2*C9)))
     01065 1130
0045 01066 4400          FMUL     X                /X*(C1+X2*(C3+X2*(C5+X2*(C7+X2*C9))))
     01067 1072
0046 01070 1030          JA       SINE             /RETURN SIN(X)
     01071 1000
0047 01072 0000 X,       0.0
     01073 0000
     01074 0000
0050 01075 0000 X2,      0.0
     01076 0000
     01077 0000



     01125 0000
     01126 5325
     01127 0420
     01106 0001
     01107 2000
     01110 0000
     01130 0001
     01131 3110
     01132 3747
     01111 0002
     01112 2000
     01113 0000
     01103 0003
     01104 2000
     01105 0000
     01100 0003
     01101 3110
     01102 3747
     01114 7764
     01115 2366
     01116 5735
     01117 7771
     01120 5466
     01121 6317
     01122 7775
     01123 2431
     01124 5053



0000 ERRORS
PI       0002 3110 3747
SINE     01000
SINE2    01014
SINE3    01022
SINE4    01034
SINE5    01040
X        01072
X2       01075
```

14.0 INTERNAL DESCRIPTION

The FPP Assembler is designed to handle standard PDP-8 assembly language, with simple extensions to include code for the FPP-12 floating point option. The Assembler runs under control of DIAL-MS on an 8K PDP-12 (except for single DF32 systems). Because the FPP-12 option effectively has a 32K directly addressable memory, all address computations are done in 24 bits. Double precision and floating point symbols may also be defined using the equate (=) pseudo-op.

14.1 Program Labels

START: read a new line from input file, unpacking the characters and putting them into LINE in ASCII: set CHRPTR and NCHARS.

ASMBL: check for the special cases:

1) line starting with slash - ignore it
2) $ sign - end of conditional assembly section
3) leading blanks or tabs - ignore them and go back to 1
4) if assembly off - ignore line

LUNAME:

1) save character pointer
2) look for an identifier followed by a comma and define tag; then go back to ASMBL
3) if no comma, look for equal sign and use EXPR to evaluate expression and set symbol to that value and type; then go to NEXTST.
4) if no equal sign, look for a blank; if a blank is found, jump to one of the instruction type handlers if the identifier was an op-code or pseudo-op, or go to GETEXP if an IOT, OPERATE, or user symbol
5) if no blank, go to GETEXP

GETEXP:

1) reset character pointer
2) evaluate line as an expression and output 1, 2 or 3 words, as specified
3) go to NEXTST

14.2 Op-Code Handlers

FPPMR:	handle FPP memory reference instructions using 1 word format if possible
FPPMRL:	handle FPP memory reference instructions using 2 word format
FPPS1:	handle FPP special format 1
PDP8MR:	handle PDP-8 memory reference instructions
FPPS2:	handle FPP special format 2
FPPS3:	handle FPP special format 3
PSEUDO:	jump to PSEUDO-OP handler
FPPS4:	handle FPP special format 4
FPPS5:	handle FPP special format 5

All op-code handlers jump back to NEXTST to get the next statement.

14.3 Major Subroutines[1]

PRINTC:	enter with the ASCII character in the AC to be printed on the output device (line printer or Teletype)
GETCHR:	get the next character from the input line, and skip one instruction upon return; if end of line encountered (CR, semicolon, slash) do not skip; multiple blanks and tabs return as a single blank
BACK1:	replace last character looked at; can be entered with AC non-zero, but can not be used to go back more than 1 character
GETNAM:	look for an identifer in the input line, starting at the current character; if no identifier present, return without skipping; if identifier is present, put first character into BUCKET (6 bits), next five into NAME1, NAME2, NAME3, with lower 6 bits of NAME3 set to zero; any characters of the identifier beyond 6 are scanned off and ignored; skip one instruction upon return if successful.
EXPR:	look for a legal expression in the input statement starting at the current character; return without skipping if no legal expression present; if a legal expression is found, the value is returned in EXPVAL, EXPVAL+1, EXPVAL+2, if floating point or double precision or integer (address); the type is returned in EXPTYP (0-undefined, 1-integer, 2-double precision 3-floating point);

[1] Enter and leave with AC=Ø unless otherwise noted.

	the last operator encountered in the expression is returned in LASTOP; skips 1 instruction if expression found.
ADRGET:	uses EXPR to get an expression and checks the type, if type was not integer (or undefined) it prints the *BX* message
ERMSG1:	writes out the next word as a 2 character error message on pass 1 or 2
ERMSG:	same as ERMSG1, except only prints on pass 2
LOOKUP:	takes the identifier in BUCKET, NAME1, NAME2, NAME3 and searches for it in the symbol table; if the name is not found, a new entry is started by setting up the link word and entering the 3 word name, the routine then returns with NEXT pointing to the place where the value will be stored and LTEMP pointing to the word into which the type will be stored; the return is then made without skipping. If the identifier is found, the return is made skipping one instruction with LTEMP pointing to the type word, NEXT pointing to the first word of the value and the type in the AC. LOOKUP always returns with data field set to 1 (this is where symbol table resides).
OUTWRD:	is entered with a word to be output in the AC; if pass 1 it returns immediately on pass 2, the word is placed into the binary working area at the location specified by LOCTR1, LOCTR2 if listing is on, the word is printed with the location on the listing if this is the first word generated by the current input line, the text of the line is also printed.
GETADR:	handles the general FPP instruction address, placing the value of the address field into FPPADR, FPPADR+1; setting INDEX to 1 if an index register field was present, otherwise to Ø, and leaving the index register number in the low three bits of EXPVAL+2.
FIXOPC:	puts the index register, increment bits and opcode bits together leaving them in OPCODE (used only by FPP code)
OCTOUT:	prints the AC as 4 octal digits followed by a blank
CHKKIL:	checks the Teletype for control-L and control-D (to abort listing and abort assembly respectively)

14.4 Important Switches and Variables

NEXT:	pointer to next available word in symbol table space
CHRPTR:	pointer to next character in input line
NCHARS:	2's complement of number of remaining characters minus 1
LINSIZ:	2's complement of number of characters in the input line
CHRCNT:	2's complement of number of characters in input buffer minus 1
LOCTR1:	high order word of 2 word location counter
LOCTR2:	low order word of 2 word location counter
LITRG1:	high order value of FPP literal origin
LITRG2:	low order value of FPP literal origin
BASER:	base register value assigned by BASE (2 words)
EXPVAL:	expression value returned by EXPR (3 words)
EXPTYP:	type of expression (0-undefined, 1-address, 2-double precision, 3-floating point)
EXPSW:	used internally by EXPR
WORD1: WORD2:	2 word address type operand
FPPADR:	holds address field for FPP memory reference instructions that use index field (2 words)
OPCODE:	contains opcode skeleton
INDEX:	set 1 if FPP instruction has index field
INDRCT:	set 1 if FPP instruction has no indirect
BUCKET:	contains first character of identifier after call to GETNAM
NAME1:	characters 2 and 3 of identifier
NAME2:	characters 4 and 5 of identifier
NAME3:	character 6 of identifier
LASTOP:	last operator encountered by EXPR
PASSNO:	pass number
ASMOF:	set 1 when assembly off in a conditional assembly section
LINENO:	dial line number
LISTSW:	set 1 if listing on

LSTON:	set 1 if listing controlled by LISTSW, Ø is no listing
OUTSWT:	set 1 if current input line has already been listed
SCSWT:	set 1 if current input line ended with a semicolon
RADIX:	set 1 for decimal, Ø for octal
LTEMP: EXTMP: EXTMP2:	general temporaries
EQUN:	holds identifier name while evaluating expression on right side of =
FPPSWT:	set 1 when EXPR being used to find address for FPPMRI
FPP2WD:	set 1 by EXPR to force 2 word FPPMR format
FPPWD2:	set 1 by FPP2WD=1 or EXPTYP=Ø for FPPMRI address
LITRL:	set 1 if expression is preceded by (or (literal
DIALRD:	address of DIAL-MS read routine
DIALWR:	address of DIAL-MS write routine
BBLOCK:	relative block number for binary working area block currently in core
BFUDGE:	absolute block number of block Ø of binary working area
SBLOCK:	relative block number for next block of source to be read
SFUDGE:	absolute block number of block Ø of source working area
STAR2Ø	set to Ø if first read from a source file (ignore *2Ø)
OTEMP: OCNT:	temporaries
PØLIT:	page Ø literal boundary address
CPLIT:	current page literal boundary address (modulo $2ØØ_8$)

14.5 Assembly of FPPASM

To assemble FPPASM, copy the files: FPPASM1, FPPASM2, and FPPASM3 onto the system device, enter a single line into the source working area:

"CHAIN FPPASM1"

then type "LO FPPASM,n"
where n is the unit containing the binary for FPPASM.

APPENDIX A

ERROR MESSAGES

The following error messages can appear before the line in error when attempting to assemble a program with the FPP Assembler, except the EG message which appears after the line. If a line of code includes statements terminated by a semicolon, then the error message for a statement precedes the printing of its octal value on the next line. Note that a fatal error causes an immediate return to DIAL after the message is printed.

BE - illegal equate
BX - bad address or index register expression
CH - chain error (fatal error)
EG - extraneous input
II - illegal indirect (FPP instructions only)
IR - illegal reference
LT - input line too long
MD - multiply defined symbol (tag or -)
MT - mixed types in expression
PO - page overflow
ST - symbol table full (fatal error)
UO - undefined origin or base
US - undefined symbol
LG - link generated

APPENDIX B

DETERMINING THE NUMBER OF SYMBOLS

The following inequality can be used to determine the number of symbols that can be defined by the user without overflowing the symbol table.

(Number Address Symbols + Number Double Precision Symbols)*6
+ (Number Floating Point Symbols)*7
+ (Number Floating Point Literals)*4
+ (Number Double Precision Literals)*3$\leq$3477

There can be 579 integer or double precision symbols.

APPENDIX C

SUMMARIES

C.1 Character Set

Keyboard	External (ASCII)	Internal
A	301	1
B	302	2
C	303	3
D	304	4
E	305	5
F	306	6
G	307	7
H	310	10
I	311	11
J	312	12
K	313	13
L	314	14
M	315	15
N	316	16
O	317	17
P	320	20
Q	321	21
R	322	22
S	323	23
T	324	24
U	325	25
V	326	26
W	327	27
X	330	30
Y	331	31
Z	332	32
[(SHIFT/K)	333	33
\ (SHIFT/L	334	34
] (SHIFT/M)	335	35
↑	336	36
←	337	Illegal (not displayed)
SPACE	240	40
!	241	41
"	242	42
#	243	Illegal (not displayed)
$	244	44
%	245	45
&	246	46
'	247	Illegal (not displayed)
(	250	50
)	251	51
*	252	52
+	253	53
,	254	54
-	255	55
.	256	56
/	257	57
0	260	60
1	261	61

Keyboard	External (ASCII)	Internal
2	262	62
3	263	63
4	264	64
5	265	65
6	266	66
7	267	67
8	270	70
9	271	71
:	272	72
;	273	73
<	274	74
=	275	75
>	276	76
?	277	77
@	300	Illegal (not displayed)
LINE FEED	212	37
RETURN	215	43 (not displayed)
ALTMODE	375	None (not displayed)
RUBOUT	377	None (not displayed)
CONTROL/I (TAB)	211	47 (not displayed)

C.2 Operators

Char	Operation
=	Define parameters
+	Combine symbols or numbers
-	Combine symbols or numbers
*	Multiply
/	Divide; comment (preceded by space)
.	Value of current location counter
;	Terminate coding line
&	Logical AND
!	Logical IOR (integer)
tab	Logical IOR (integer)
space	Logical IOR (integer)
return	Terminate statement

C.3 Pseudo-operations

BASE expr	Assign base register for 1-word instructions.
CHAIN "name" unit	Continue assembly at program "name" on unit (Ø-7).
DECIMAL	Set radix for integer conversion to decimal numbers.
EJECT	Continue listing at top of next line printer paper page.
END	Terminate input.
IFNEG n	Assemble if n is negative.
IFNZRO n	Assemble if n is not zero.
IFPOS n	Assemble if n is positive.
IFREF symbol	Assemble if symbol has already been defined or referenced.
IFZERO n	Assemble if n is zero.
LISTOF	Turn program listing off.
LISTON	Turn program listing on.
LITORG expr	Start literal list at location expr.
OCTAL	Set radix for integer conversion to octal numbers.
ORG expr	Set starting location for output to lower 15 bits of expr.
PAGE	Continue output at start of next core page.
TEXT	Print the line of text between delimiters.
=	Equate symbol on left of = to that on the right.

IOT MICROINSTRUCTIONS

Mnemonic	Octal	Operation
Keyboard/Reader		
KSF	6Ø31	Skip if keyboard/reader flag = 1
KCC	6Ø32	Clear AC and keyboard/reader flag
KRS	6Ø34	Read keyboard/reader buffer
KRB	6Ø36	Clear AC and read keyboard buffer, and clear keyboard flag
Teleprinter/Punch		
TSF	6Ø41	Skip if teleprinter/punch flag = 1
TCF	6Ø42	Clear teleprinter/punch flag
TPC	6Ø44	Load teleprinter/punch buffer, select and print
TLS	6Ø46	Load teleprinter/punch buffer, select and print, and clear teleprinter/punch flag

Mnemonic	Octal	Operation
Extended Memory (Type MC8/I)		
CDF	62n1	Change to data field n
CIF	62n2	Change to instruction field n
RDF	62n4	Read data field into AC
RIF	6224	Read instruction field into AC
RMF	6244	Restore memory field
RIB	6234	Read interrupt buffer

C.4 FPP Symbols

DATA REFERENCE INSTRUCTIONS

Mnemonic	Data Function
FLDA	C(Y)→FAC
FADD	C(Y)+C(FAC)→FAC
FADDM	C(Y)+C(FAC)→Y
FSUB	C(FAC)-C(Y)→FAC
FDIV	C(FAC)/C(Y)→FAC
FMUL	C(FAC)*C(Y)→FAC
FMULM	C(FAC)*C(Y)→Y
FSTA	C(FAC)→Y

INDEX REGISTER MODIFIERS

Mnemonic	Operation
JXN	The index register is incremented if bit 5=1 and a jump is executed to the address contained in bits 9-23 if index register X is nonzero.
TRAP3 thru TRAP7	The instruction trap status bit is set and the FPP-12 exits, causing a PDP interrupt. The unindexed operand address is dumped into the APT.
LDX	The contents of the index register specified to bits 9-11 are replaced by the contents of bits 12-23.
ADDX	The contents of bits 12-23 are added to the index register specified by bits 9-11.
ALN	The mantissa of the FAC is shifted until the FAC exponent equals the contents of the index register specified by bits 9-11. If bits 9-11 are zero, the FAC is aligned such that the exponent[1] = $(23)_{10}$. In double precision mode, an arithmetic shift is performed on the FAC fraction. The number of shifts is equal to the absolute value of the contents of the specified index register. The direction of shifting depends on the sign of the index register contents. A positive sign indicates a shift towards the least significant bit while a negative sign indicates a shift towards the most significant bit. The FAC exponent is not altered by the ALN instruction in double precision mode.

[1] Setting the exponent = $(23)_{10}$ integerizes or fixes the floating point number. The JAL instruction tests to see if fixing is possible.

Mnemonic	Operation
ATX	The contents of the FAC is fixed and the least significant 12 bits of the mantissa are loaded into the index register specified by bits 9-11. In double precision mode the least significant 12 bits of the FAC are loaded into the specified index register. The FAC itself is not altered by the ATX instruction.
XTA	The contents of the index register specified by bits 9-11 are loaded right justified into the FAC mantissa. The FAC exponent is loaded with $(23)_{10}$ and then the FAC is normalized. This operation is typically termed floating a 12-bit number. In double precision mode the FAC is not normalized.
OPERATES	
FEXIT	Dump active registers into the active parameter table, reset the FPP-12 run flip-flop to the Ø state, and interrupt the PDP processor.
FPAUSE	Wait for external synchronizing signal. This instruction is designed to cooperate with the AIP-12 option. IOT FPST (6555) will restart the FPP-12 executing the instruction following FPAUSE.
FCLA	Zero the FAC mantissa and exponent.
FNEG	Complement the FAC mantissa.
FNORM	Normalize the FAC. In double precision mode FNORM is a NOP.
STARTF	Start floating-point mode.
STARTD	Start double-precision mode.
FNOP	These single-word instructions perform no operation.
JAC	Jump to the location specified by the lst least significant 15 bits of the FAC mantissa.
CONDITIONAL JUMPS	
JEQ	Jump if the FAC = Ø
JGE	Jump if the FAC $\geq$ Ø
JLE	Jump if the FAC $\leq$ Ø
JA	Jump always
JNE	Jump if the FAC $\neq$ Ø
JLT	Jump if the FAC < Ø
JGT	Jump if the FAC > Ø
JAL	Jump if impossible to fix the floating point number contained in the FAC; i.e., if the exponent is greater than $(23)_{10}$.

POINTER MOVES

Mnemonic	Octal	Operation
SETX		Set XØ to the address contained in bits 9-23 of the instruction.
SETB		Set the base register to the address contained in bits 9-23.
JSR		Jump and save return. The jump is to the location specified in bits 9-23 and the return is saved in bits 21-35 of the 1st entry of the data block.
JSA		An unconditional jump is deposited in the address and address+1 where address is specified by bits 9-23. The FPC is set to address+2.

FLOATING POINT

FPINT	6551	Skip on FPP interrupt.
FPICL	6552	Clear the FPP interrupt flag and reset all important registers. FPICL is equivalent to an I/O present for the FPP.
FPCOM	6553	Load FPP command register and field bits of the APT pointer if: a) The FPP is not running and b) The FPP interrupt flag is reset.
FPHLT	6554	Force an FPP exit at the end of the current instruction.
FPST	6555	Load the 12 least significant 12-bits of the APT pointer and start if the FPP is a) not running and b) if the FPP interrupt flag is reset, FPST will restart the FPP following an FPAUSE instruction without re-initialization. If the FPP is started or restarted, FPST will skip.
FPRST	6556	Read the FPP status register into the AC.
FPIST	6557	Skip on the FPP interrupt flag. If the skip is granted, clear the flag, and read the FPP status into the AC.

FUNCTION OF PDP AC BITS WITH FPCOM (6553) IOT

ACØ	Select double precision mode.
AC1	Exit on exponent underflow error.

Mnemonic	Operation
AC2	Enable memory protection.
AC3	Enable interrupt.
AC4	Do not store op address on exits.
AC5	Do not store address of index registers on exits.
AC6	Do not store address of indirect pointer lists on exits.
AC7	Do not store FAC on exits.
AC8	Lockout CPU when FPP is active.
AC9, AC10, AC11	4K field select bits of "Active Parameter Table" pointer.

PDP AC AFTER READ STATUS IOT'S FPIST (6557) OR FPRST (6556)

AC0	Double precision mode.
AC1	Instruction trap.
AC2	CPU force trap
AC3	Divide by zero
AC4	Fraction overflow (double precision mode only)
AC5	Exponent overflow
AC6	Exponent underflow
AC7, AC8, AC9	Unused
AC10	Paused
AC11	Run

ACTIVE PARAMETER TABLE FORMAT

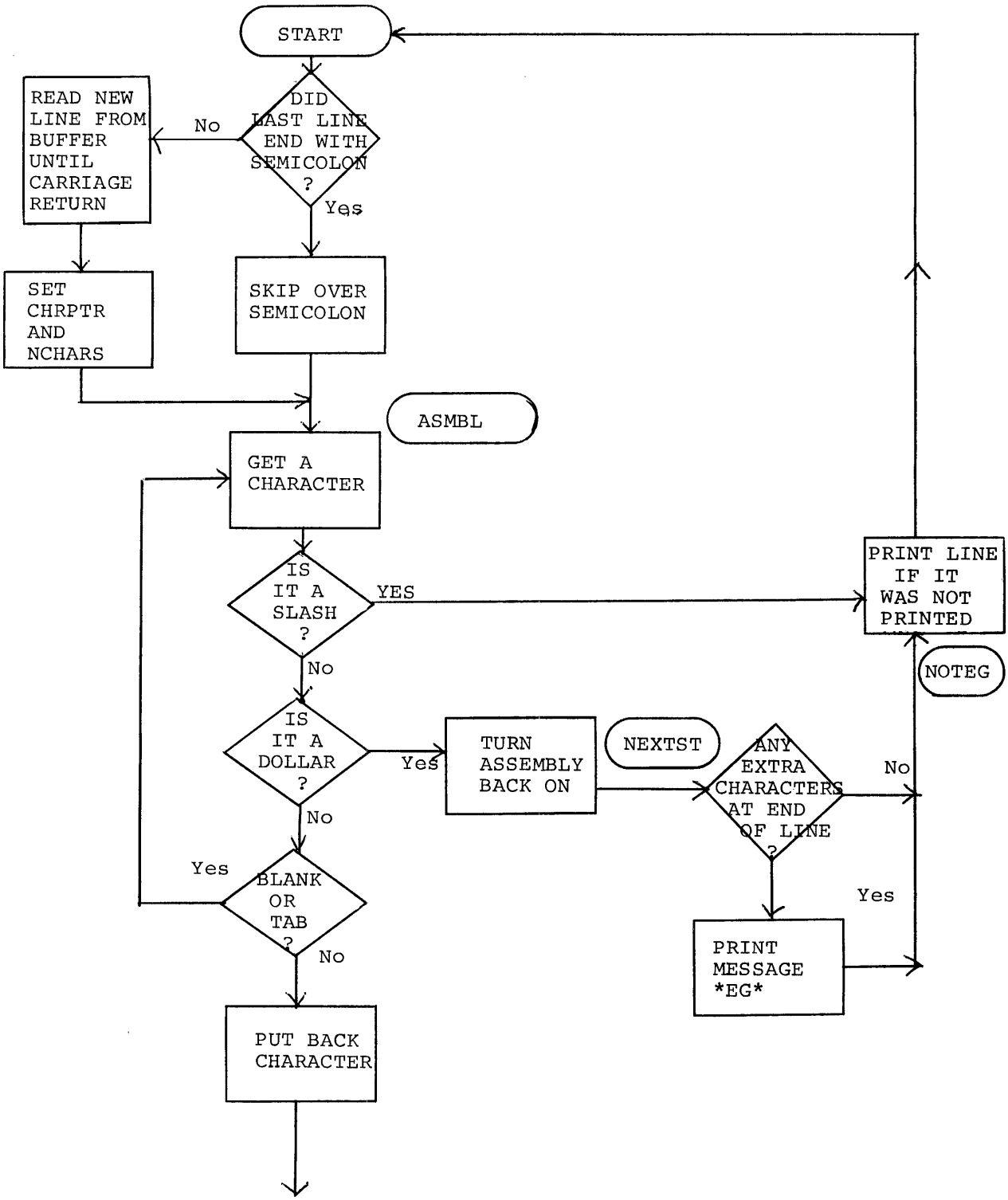
Location	Contents			
P	Field bits of Operand Address	Field bits of Base Register	Field bits of Index Register Loc'n	Field bits of FPC
P + 1	Lower 12 bits of FPC			
P + 2	Lower bits of index register 0 location X 0			
P + 3	Lower bits of Base Register			
P + 4	Lower bits of operand address			
P + 5	Exponent of FAC			
P + 6	MSW of FAC			
P + 7	LSW of FAC			

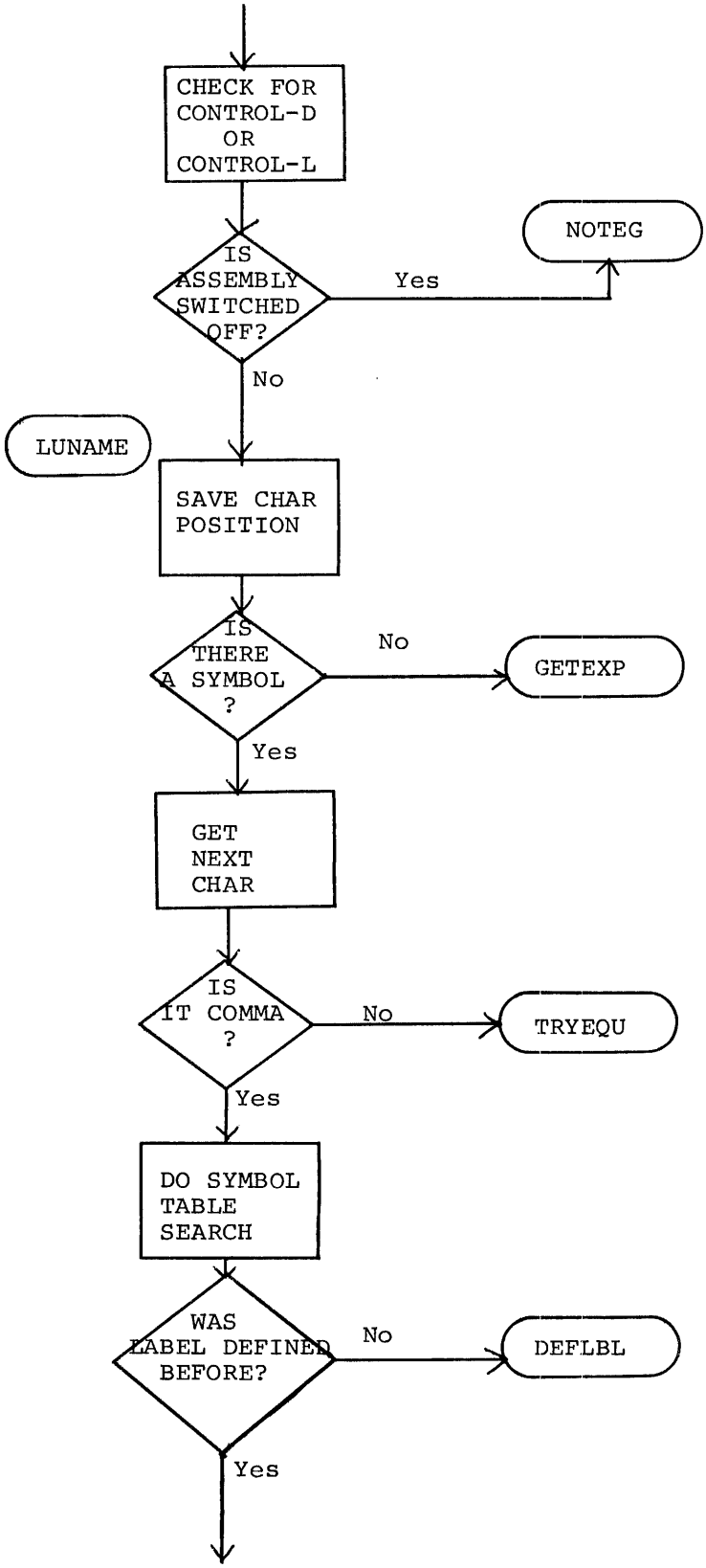
NOTE: APT address points to location P.

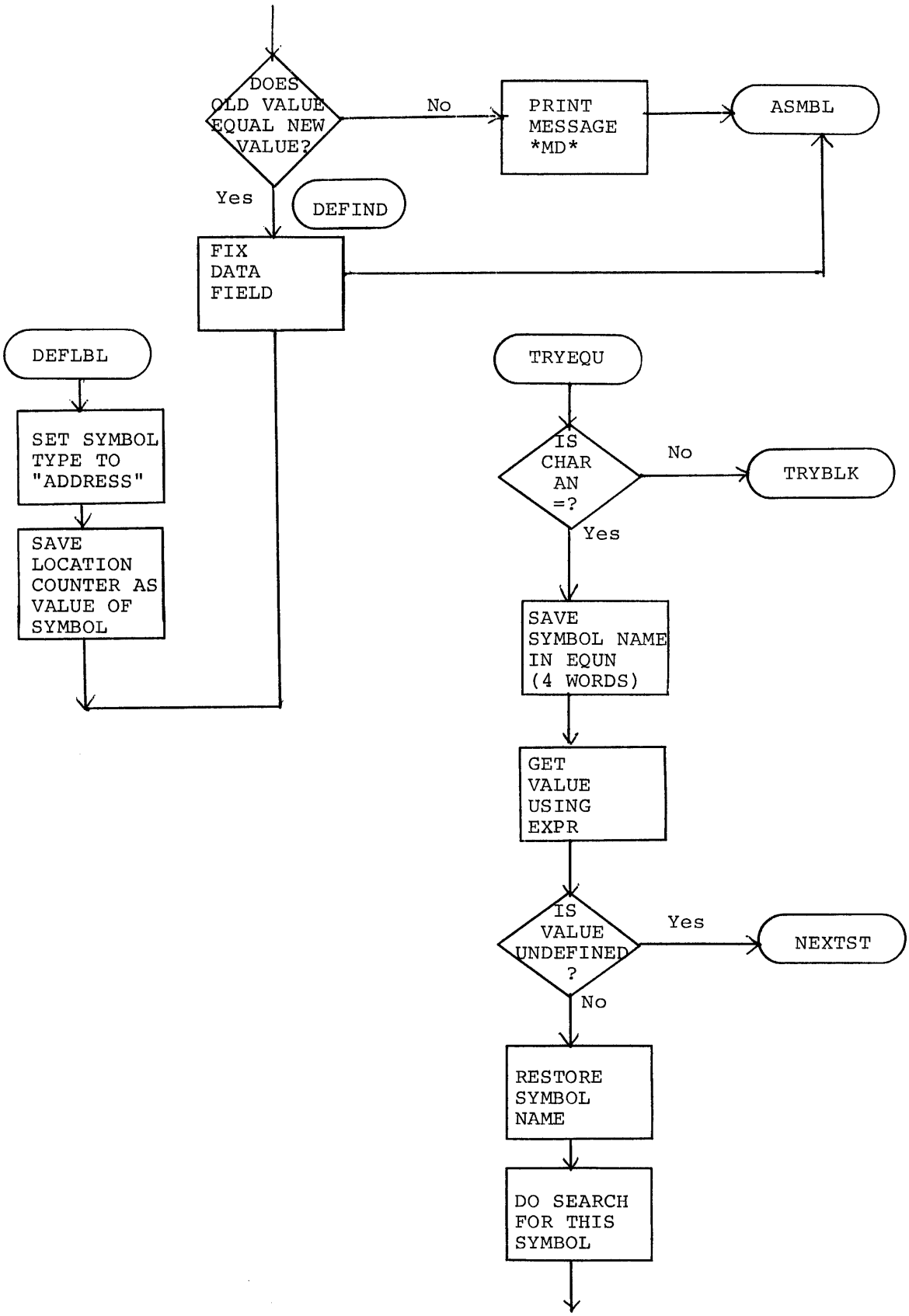
NOTE: APT address points to location P.

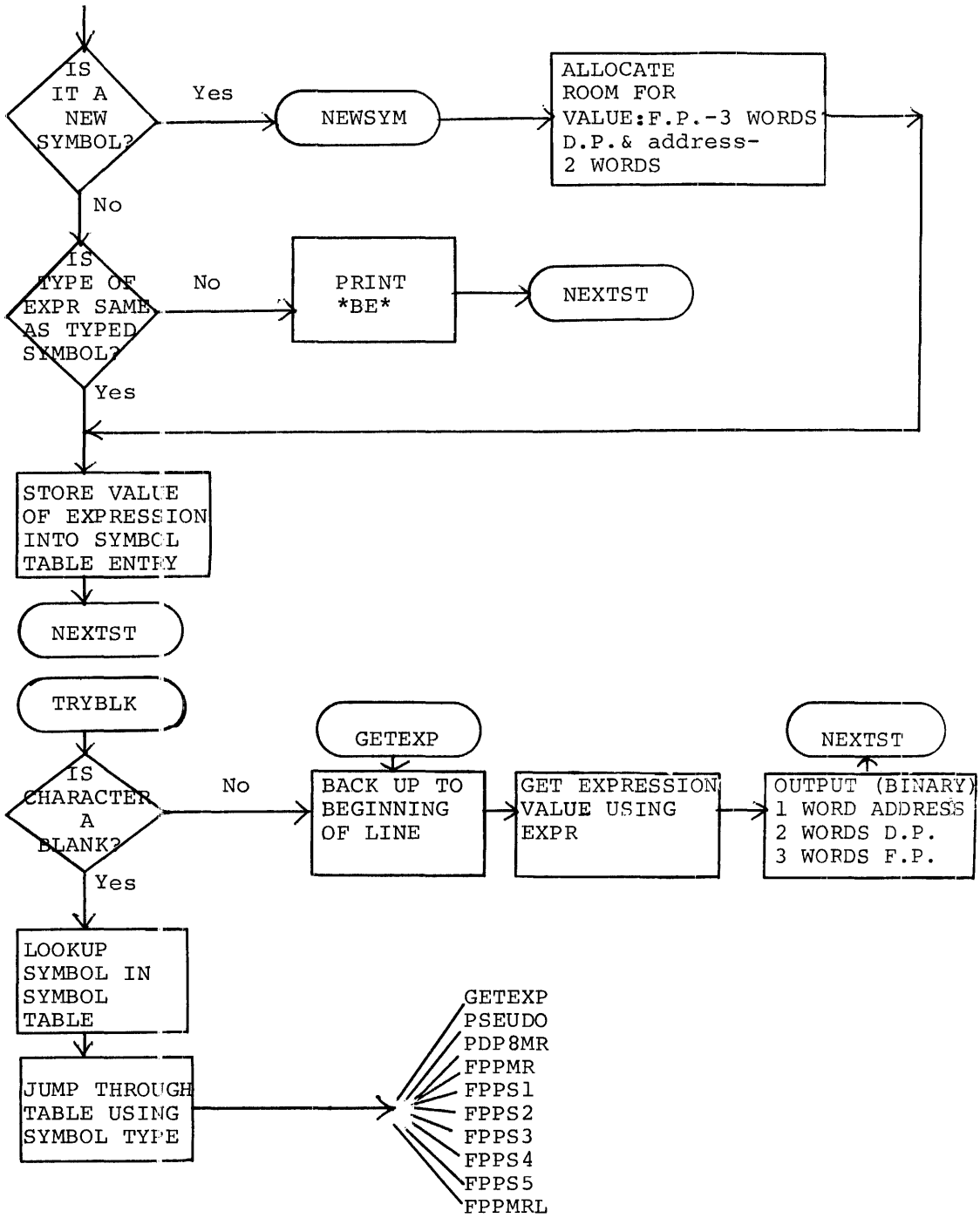
APPENDIX D

SYSTEM FLOW CHARTS




CHECK FOR CONTROL-D OR CONTROL-L
IS ASSEMBLY SWITCHED OFF?
Yes
NOTEG
No
LUNAME
SAVE CHAR POSITION
IS THERE A SYMBOL ?
No
GETEXP
Yes
GET NEXT CHAR
IS IT COMMA ?
No
TRYEQU
Yes
DO SYMBOL TABLE SEARCH
WAS LABEL DEFINED BEFORE?
No
DEFLBL
Yes


DOES OLD VALUE EQUAL NEW VALUE?
No
PRINT MESSAGE *MD*
ASMBL
Yes
DEFIND
FIX DATA FIELD
DEFLBL
SET SYMBOL TYPE TO "ADDRESS"
SAVE LOCATION COUNTER AS VALUE OF SYMBOL
TRYEQU
IS CHAR AN =?
No
TRYBLK
Yes
SAVE SYMBOL NAME IN EQUN (4 WORDS)
GET VALUE USING EXPR
IS VALUE UNDEFINED ?
Yes
NEXTST
No
RESTORE SYMBOL NAME
DO SEARCH FOR THIS SYMBOL


IS IT A NEW SYMBOL?
Yes
NEWSYM
ALLOCATE ROOM FOR VALUE:F.P.-3 WORDS D.P.& address- 2 WORDS
No
IS TYPE OF EXPR SAME AS TYPED SYMBOL?
No
PRINT *BE*
NEXTST
Yes
STORE VALUE OF EXPRESSION INTO SYMBOL TABLE ENTRY
NEXTST
TRYBLK
GETEXP
NEXTST
IS CHARACTER A BLANK?
No
BACK UP TO BEGINNING OF LINE
GET EXPRESSION VALUE USING EXPR
OUTPUT (BINARY) 1 WORD ADDRESS 2 WORDS D.P. 3 WORDS F.P.
Yes
LOOKUP SYMBOL IN SYMBOL TABLE
JUMP THROUGH TABLE USING SYMBOL TYPE
GETEXP
PSEUDO
PDP8MR
FPPMR
FPPS1
FPPS2
FPPS3
FPPS4
FPPS5
FPPMRL

```
CHAINING TO FPPASM2
CHAINING TO FPPASM3
```

```
0001                                        /FPP ASSEMBLER  04/01/71
0002                                        /COPYRIGHT
0003                                        /DIGITAL EQUIPMENT CORP.
0004                                        /MAYNARD, MASS.
0005                        FLD0=0
0006                        FLD1=10
0007                        ORG     3
0010 00003 0014 VERS,   14                  /VERSION NUMBER
0011 00004 0000 OLDN3,  0                   /TEMP FOR LOOKUP
0012 00005 0000 OTEMP,  0                   /A COUPLE OF TEMPS THAT
0013 00006 0000 OCNT,   0                   /DIDNT FIT INTO THEIR PAGE
0014 00007 7702 PAGSIZ, -76                 /LINES PER PAGE COUNTER
0015 00010 0000 X10,    0
0016 00011 0000 X11,    0
0017 00012 0000 X12,    0
0020 00013 0000 X13,    0
0021 00014 0000 X14,    0
0022 00015 0000 X15,    0
0023 00016 1140 NEXT,   FREE-1
0024 00017 0000 CHRPTR, 0
0025 00020 7777 NCHARS, -1                  /CHARACTER INPUT STUFF
0026 00021 0000 CPTMP,  0
0027 00022 0000 NCTMP,  0                   /USED TO SAVE CHAR POSITION
0030 00023 0000 LINSIZ, 0                   /SIZE OF LINE FOR PRINTING
0031 00024 7777 CHRCNT, -1                  /CHAR COUNT FOR INPUT FILE
0032 00025 7702 SIZPAG, -76                 /NUMBER OF LINES PER PAGE
0033 00026 0000 LOCTR1, 0                   /24 BIT LOCATION COUNTER
0034 00027 0200 LOCTR2, 200
0035 00030 4000 LITRG1, 4000                /FIRST WORD OF LITERAL (FP) ORG
0036 00031 0000 LITRG2, 0
0037 00032 4000 BASER,  4000                /BASE REGISTER SETTING
0040 00033 0000         0
0041 00034 0000 EXPVAL, 0                   /EXPRESSION VALUE
0042 00035 0000         0
0043 00036 0000         0
0044 00037 0000 EXPTYP, 0                   /EXPRESSION TYPE
0045 00040 0000 EXPSW,  0                   /FLAG=1 IF NO EXPR
0046 00041 0000 WORD1,  0                   /TEMPORARY 2 WORD OPERAND
0047 00042 0000 WORD2,  0
0050 00043 0000 FPPADR, 0                   /ADDRESS FIELD FOR FPP INDEX INS
0051 00044 0000         0
0052 00045 0000 OPCODE, 0                   /OPCODE OR PSEUDO-OP POINTER
0053 00046 0000 INDEX,  0                   /INDEX FLAG = 1 IF INDEX PRESENT
0054 00047 0000 INDRCT, 0                   /INDIRECT FLAG = 0 IF IND IN FPP
0055 00050 0001 XINCR,  1                   /FLAG = 0 IF + LEGAL IN INDEX EX
0056 00051 0000 BUCKET, 0                   /FIRST CHAR OF NAME
0057 00052 0000 NAME1,  0                   /CHARS 2 AND 3 OF NAME
0060 00053 0000 NAME2,  0                   /CHARS 4 AND 5 OF NAME
0061 00054 0000 NAME3,  0                   /CHAR 6 OF NAME AND TYPE
0062 00055 0000 LASTOP, 0                   /LAST OPERATOR ENCOUNTERRED IN E
0063 00056 0001 PASSNO, 1                   /PASS NUMBER
0064 00057 0000 ASMOF,  0                   /SET TO 1 WHEN ASSEMBLY OFF
0065 00060 0000 LINENO, 0                   /LINE NUMBER
```

```
0066 00061 0001 LISTSW, 1                       /LIST SWITCH
0067 00062 0001 LSTON,  1                       /NO LIST OPTION SWITCH
0070 00063 0000 OUTSWT, 0                       /OUT SWITCH
0071 00064 0000 SCSWT,  0                       /SEMICOLON SWITCH
0072 00065 0000 RADIX,  0                       /RADIX FOR INTEGERS (0 IS OCTAL)
0073 00066 0000 LTEMP,  0                       /TEMP USED BY LOOKUP
0074 00067 0000 EXTMP,  0                       /TEMPS USED BY EXPR AND OTHERS
0075 00070 0000 EXTMP2, 0
0076 00071 0000 EGUN,   0;0;0;0
     00072 0000
     00073 0000
     00074 0000
0077 00075 0000 FPPSWT, 0                       /1 WHHEN FINDING FPP ADR EXPR
0100 00076 0000 FPP2WD, 0                       /SET BY EXPR TO FORCE 2 WD FMT
0101 00077 0000 FPPWD2, 0                       /SET BY FPP2WD.OR.EXPTYP.EQ.0
0102 00100 0000 LITRL,  0                       /SET = 1 FOR LITERAL
0103 00101 7630 DIALRD, 7630                    /DIAL INPUT ROUTINE
0104 00102 7632 DIALWR, 7632                    /DIAL OUTPUT ROUTINE
0105 00103 0000 BBLOCK, 0                       /CURRENT BINARY OUT BLOCK (REL)
0106 00104 0370 BFUDGE, 370                     /BINARY FUDGE FACTOR
0107 00105 0000 SBLOCK, 0                       /CURRENT SOURCE INPUT BLOCK
0110 00106 0370 SFUDGE, 370                     /SOURCE FUDGE FACTOR
0111 00107 0000 STAR20, 0                       /IGNORE *20 AT BEGINNING OF FILE
0112 00110 0177 PBLIT,  177
0113 00111 0177 CPLIT,  177
0114 00112 0000 ACE,    0
0115 00113 0000 OPE,    0
0116 00114 0000 TMP,    0
0117 00115 0000 ACO,    0
0120 00116 0000 ACL,    0
0121 00117 0000 ACH,    0
0122 00120 0000 OPO,    0
0123 00121 0000 OPL,    0
0124 00122 0000 OPH,    0
```

```
0125                        EJECT
0126                        ORG 177          / ENTER INTERPRETER BY A JMS 177
0127 00177 0000 FPT,        0
0130               F7600=.                   /WRONG! WRONG! WRONG!
0131 00200 7400 F7400,      7400
0132 00201 3115             DCA ACO          /CLEAR AC OVFLO WD
0133 00202 1577             TAD I FPT
0134 00203 0210             AND F177
0135 00204 3363             DCA FADR         /GET LOW ORDER ADDRESS BITS
0136 00205 1577             TAD I FPT
0137 00206 0215             AND F200
0140 00207 7041             CIA
0141 00210 0177 F177,       AND FPT          /ADD BEGINNING OF PAGE IF NECESSARY
0142 00211 1363             TAD FADR
0143 00212 3363             DCA FADR
0144 00213 1577             TAD I FPT
0145 00214 2177             ISZ FPT          /BUMP FP LOC CTR
0146 00215 0200 F200,       AND F7400
0147 00216 7166             CLL CML CMA RTL
0150 00217 7006             RTL
0151 00220 1360             TAD FPJUMP
0152 00221 3241             DCA FPGOTO       /FORM JUMP FROM OPCODE
0153 00222 7620 FPSNL,      SNL CLA          /USED AS BASE FOR OPRS
0154 00223 5226             JMP .+3          /NO INDIRECT ADDRESSING
0155 00224 1763             TAD I FADR
0156 00225 3363             DCA FADR
0157 00226 1363             TAD FADR
0160 00227 3114             DCA TMP          /SAVE ADDRESS FOR FUTURE USE
0161 00230 1763             TAD I FADR
0162 00231 3113             DCA OPE
0163 00232 2363             ISZ FADR
0164 00233 1763             TAD I FADR
0165 00234 3122             DCA OPH
0166 00235 2363             ISZ FADR
0167 00236 1763             TAD I FADR
0170 00237 3121             DCA OPL          /LOAD OPERAND
0171 00240 3120             DCA OPO          /AND ZERO OVFLO WD
0172               FPGOTO=.
0173 00241 1177 AL1,        TAD [0           /RESERVE LOC 177
0174                                         /SHIFT AC LEFT 1
0175 00242 1115             TAD ACO
0176 00243 7104             CLL RAL
0177 00244 3115             DCA ACO
0200 00245 1116             TAD ACL
0201 00246 7004             RAL
0202 00247 3116             DCA ACL
0203 00250 1117             TAD ACH
0204 00251 7004             RAL
0205 00252 3117             DCA ACH
0206 00253 5641             JMP I AL1
0207 00254 1114 FPOPER,     TAD TMP          /OPR 0 =EXIT F.P. PACKAGE
0210 00255 7450             SNA
0211 00256 5577             JMP I FPT
```

```
0212 00257 1222          TAD FPSNL
0213 00260 3262          DCA FPSKIP      /OTHERWISE SAME AS PDP-8 CODE
0214 00261 1117          TAD ACH
0215 00262 7402 FPSKIP,  HLT
0216 00263 2177          ISZ FPT         /EXCEPT SKIP CONDITION REVERSED
0217 00264 5200          JMP F7600
0220 00265 1114 FPJMP,   TAD TMP
0221 00266 3177          DCA FPT
0222 00267 5200          JMP F7600       /FLOATING JUMP
0223 00270 1112 FPSTO,   TAD ACE
0224 00271 3514          DCA I TMP
0225 00272 1117          TAD ACH
0226 00273 2114          ISZ TMP
0227 00274 3514          DCA I TMP
0230 00275 1116          TAD ACL
0231 00276 3763          DCA I FADR      /REMEMBER FADR??
0232 00277 5200          JMP F7600
0233 00300 1121 FPLAC,   TAD OPL
0234 00301 3116          DCA ACL         /THIS LOC JUMPED TO FROM PAGE 2
0235 00302 1122          TAD OPH
0236 00303 3117          DCA ACH
0237 00304 1113 FPLACE,  TAD OPE
0240 00305 3112          DCA ACE         /BECAUSE FPLAC IS JUMPED TO, FALL THROUG
0241 00306 1117 ANORM,   TAD ACH
0242 00307 7450          SNA
0243 00310 1116          TAD ACL
0244 00311 7450          SNA
0245 00312 1115          TAD ACO
0246 00313 7650          SNA CLA         /IF FRACION IS ZERO,
0247 00314 5361          JMP ADCEXP      /ZERO EXPONENT
0250 00315 7332 NORMLP,  CLA CLL CML RTR /2000
0251 00316 1117          TAD ACH
0252 00317 7440          SZA             /IF ACH=6000,
0253 00320 5323          JMP .+3
0254 00321 1116          TAD ACL         /AND ACL=0 WERE DONE
0255 00322 7640          SZA CLA
0256 00323 7710          SPA CLA         /OTHERWISE WERE DONE WHEN ACH(0)<>ACH(1)
0257 00324 5200          JMP F7600
0260 00325 4241          JMS AL1         /LEFT SHIFT AC ONE PLACE
0261 00326 7240          CLA CMA
0262 00327 1112          TAD ACE
0263 00330 3112          DCA ACE         /REDUCE EXPONENT
0264 00331 5315          JMP NORMLP      /KEEP GOING
0265 00332 0531          FPDIV           /7000
0266 00333 0462          FPMUL           /6000
0267 00334 0416          FPSUB           /5000
0270 00335 0417          FPADD           /4000
0271 00336 0300          FPLAC           /3000
0272 00337 0270          FPSTO           /2000
0273 00340 0265          FPJMP           /1000
0274 00341 0254          FPOPER          /0000 FEXIT
0275                                     /0010 FNOP
0276                                     /0020 FSNA
0277                                     /0030 FSZA
```

```
0300                                        /0060 FSPA
0301                                        /0070 FSMA
0302                                        /0120 FSPASNA
0303                                        /0130 FSMASZA
0304 00342 0000 AR1,    0                   /RIGHT SHIFT AC ONE PLACE
0305 00343 7300         CLA CLL
0306 00344 1117         TAD ACH
0307 00345 7510         SPA
0310 00346 7020         CML
0311 00347 7010         RAR
0312 00350 3117         DCA ACH
0313 00351 1116         TAD ACL
0314 00352 7010         RAR
0315 00353 3116         DCA ACL
0316 00354 7010         RAR
0317 00355 3115         DCA ACO             /ONLY SAVE ONE BIT OF OVERFLOW WORD
0320 00356 2112         ISZ ACE
0321 00357 0120 POPO,   OPO
0322 00360 5742 FPJUMP, JMP I AR1
0323 00361 3112 ADCEXP, DCA ACE
0324 00362 5200         JMP F7600           /ZERO EXPONENT AND GET NEXT INST.
0325            FADR=.
0326 00363 0000 FNEGX,  0                   /TRIPLE-WORD NEGATION
0327 00364 1357         TAD POPO
0330 00365 3241         DCA AL1             /AC=0 OR -3 ON ENTRY
0331 00366 7346         CLA CLL CMA RTL
0332 00367 3342         DCA AR1
0333 00370 7024 NEGLP,  CML RAL
0334 00371 1641         TAD I AL1
0335 00372 7141         CLL CIA
0336 00373 3641         DCA I AL1
0337 00374 2241         ISZ AL1
0340 00375 2342         ISZ AR1
0341 00376 5370         JMP NEGLP
0342 00377 5763         JMP I FNEGX
```

```
0343                    EJECT
0344                    ORG 400          /SECOND PAGE OF FLOATING POINT PACKAGE
0345 00400 0000 OADD,   0                /ADD OP TO AC
0346 00401 7100         CLL
0347 00402 1120         TAD OPO
0350 00403 1115         TAD ACO
0351 00404 3115         DCA ACO
0352 00405 7004         RAL
0353 00406 1121         TAD OPL
0354 00407 1116         TAD ACL
0355 00410 3116         DCA ACL
0356 00411 7004         RAL
0357 00412 1122         TAD OPH
0360 00413 1117         TAD ACH
0361 00414 3117         DCA ACH
0362 00415 5600         JMP I OADD
0363 00416 4775 FPSUB,  JMS I PNEG
0364 00417 1122 FPADD,  TAD OPH
0365 00420 7650         SNA CLA
0366 00421 5774         JMP I PANORM     /OP=0 - NOP
0367 00422 1117         TAD ACH
0370 00423 7650         SNA CLA
0371 00424 5241         JMP DOADDS+2     /AC=0 - SET AC TO OP
0372 00425 1112 FPADLP, TAD ACE
0373 00426 7141         CIA CLL
0374 00427 1113         TAD OPE
0375 00430 7450         SNA              /COMPARE AC EXP TO OP EXP
0376 00431 5237         JMP DOADDS       /EQUAL - GO ADD
0377 00432 7500         SMA
0400 00433 4776         JMS I PAR1       /OPE>ACE
0401 00434 7710         SPA CLA
0402 00435 4243         JMS OR1          /ACE>OPE
0403 00436 5225         JMP FPADLP
0404 00437 4243 DOADDS, JMS OR1
0405 00440 4776         JMS I PAR1       /UNNORMALIZE BOTH BY ONE PLACE
0406 00441 4200         JMS OADD
0407 00442 5727         JMP I PFPLCE     /SET ACE=OPE (IN CASE AC WAS 0)
0410 00443 0000 OR1,    0                /SHIFT OPERAND RIGHT ONE PLACE
0411 00444 7300         CLA CLL
0412 00445 1122         TAD OPH
0413 00446 7510         SPA
0414 00447 7020         CML
0415 00450 7010         RAR
0416 00451 3122         DCA OPH
0417 00452 1121         TAD OPL
0420 00453 7010         RAR
0421 00454 3121         DCA OPL
0422 00455 7010         RAR
0423 00456 3120         DCA OPO          /OP OVFLO WD
0424 00457 2113         ISZ OPE
0425 00460 0000 FITCNT, 0
0426 00461 5643         JMP I OR1
0427 00462 1122 FPMUL,  TAD OPH
```

```
0430 00463 7650          SNA CLA
0431 00464 5773          JMP I PFPLAC
0432 00465 1117          TAD ACH
0433 00466 7650          SNA CLA
0434 00467 5774          JMP I PANORM     /EITHER ARG=0 MEANS RESULT=0
0435 00470 1112          TAD ACE
0436 00471 1113          TAD OPE
0437 00472 3113          DCA OPE
0440 00473 1117          TAD ACH
0441 00474 3243          DCA OR1
0442 00475 1116          TAD ACL
0443 00476 3114          DCA TMP
0444 00477 1330          TAD FM30         /ITERATION COUNTER
0445 00500 3260          DCA FITCNT
0446 00501 3117          DCA ACH
0447 00502 3116          DCA ACL
0450 00503 7100          CLL
0451 00504 5314          JMP FPMDCD
0452 00505 4776 FPMLLP,  JMS I PAR1       /RIGHT SHIFT AC ONE PLACE
0453 00506 1243          TAD OR1
0454 00507 7010          RAR
0455 00510 3243          DCA OR1
0456 00511 1114          TAD TMP
0457 00512 7010          RAR
0460 00513 3114          DCA TMP          /RIGHT SHIFT MULTIPLIER
0461 00514 1114 FPMDCD,  TAD TMP
0462 00515 7004          RAL
0463 00516 7001          IAC
0464 00517 7012          RTR
0465 00520 7620          SNL CLA          /DOES BIT SHIFTED OUT = NEXT BIT?
0466 00521 5324          JMP FMNOAD       /YES - DO NOTHING
0467 00522 4775          JMS I PNEG       /NEGATE MULTIPLICAND
0470 00523 4200          JMS DADD         /ADD TO PRODUCT
0471 00524 2260 FMNOAD,  ISZ FITCNT
0472 00525 5305          JMP FPMLLP
0473 00526 5727          JMP I .+1        /NOW GO SET ACE = OPE
0474 00527 0304 PFPLCE,  FPLACE
0475 00530 7750 FM30,    -30
0476 00531 1122 FPDIV,   TAD OPH
0477 00532 7450          SNA
0500 00533 5773          JMP I PFPLAC     /DIVIDE BY ZERO RESULTS IN ZERO!
0501 00534 7700          SMA CLA          /CHECK DIVISOR SIGN
0502 00535 5340          JMP .+3
0503 00536 7346          CLA CLL CMA RTL /IF NEGATIVE,
0504 00537 4775          JMS I PNEG       /NEGATE AC
0505 00540 4776          JMS I PAR1       /UNNORMALIZE DIVIDEND ONE PLACE
0506 00541 1113          TAD OPE
0507 00542 7041          CIA
0510 00543 1112          TAD ACE
0511 00544 3113          DCA OPE          /SET UP RESULTANT EXPONENT
0512 00545 1330          TAD FM30
0513 00546 3260          DCA FITCNT       /SET UP ITERATION COUNT
0514 00547 1117          TAD ACH
0515 00550 7004          RAL              /INITIALIZE LINK
```

```
0515 00551 7210 FPDVLP, CLA RAR          /GET DIVIDEND SIGN FROM LINK AFTER LEFTS
0517 00552 1122          TAD OPH
0520 00553 7700          SMA CLA         /DOES DIVIDEND SIGN = DIVISOR SIGN?
0521 00554 4775          JMS I PNEG      /YES = NEGATE DIVISOR
0522 00555 4200          JMS DADD        /ADD DIVISOR TO DIVIDEND
0523 00556 1112          TAD ACE
0524 00557 7004          RAL             /ADD CARRY INTO QUOTIENT
0525 00560 3112          DCA ACE
0526 00561 1114          TAD TMP
0527 00562 7004          RAL
0530 00563 3114          DCA TMP
0531 00564 4777          JMS I PAL1      /LEFT SHIFT DIVIDEND
0532 00565 2260          ISZ FITCNT
0533 00566 5351          JMP FPDVLP      /LOOP
0534 00567 1114          TAD TMP
0535 00570 3122          DCA OPH
0536 00571 1112          TAD ACE
0537 00572 5773          JMP I .+1
0540 00573 0301 PFPLAC, FPLAC+1          /PUT RESULT INTO AC
0541 00574 0306 PANORM, ANORM
0542 00575 0363 PNEG,   FNEGX
0543 00576 0342 PAR1,   AR1
0544 00577 0241 PAL1,   AL1
```

```
0545                    EJECT
0546                    CPLBUF=6000
0547                    LINE=6200
0550                    USETBL=6340
0551                    BINARY=6400
0552                    SOURCE=7000
0553                    P0LBUF=7400
0554                    USEB=14
0555                    BINARB=15
0556                    SOURCB=16
0557                    PAGE
0560 00600 7240 NEXTST, CLA CMA                 /CHECK PASS NUMBER
0561 00601 1056         TAD     PASSNO
0562 00602 7650         SNA CLA
0563 00603 5222         JMP     START           /IF PASS 1 THEN NO LISTING
0564 00604 1061         TAD     LISTSW          /CHECK LIST STATUS
0565 00605 7650         SNA CLA
0566 00606 5222         JMP     START           /NO, DONT LIST
0567 00607 1063         TAD     OUTSWT          /DID THIS LINE GENERATE OUTPUT?
0570 00610 7640         SZA CLA
0571 00611 5222         JMP     START           /YES, NO PRINT NECESSARY
0572 00612 4576         JMS I   [PRINTC         /PRINT CR/LF
0573 00613 1060         TAD     LINENO          /PRINT LINE NUMBER
0574 00614 4575         JMS I   [OCTOUT
0575 00615 1174         TAD     [247
0576 00616 4576         JMS I   [PRINTC         /THEN TAB
0577 00617 1174         TAD     [247            /ANOTHER TAB
0600 00620 4576         JMS I   [PRINTC
0601 00621 4777         JMS I   [PRNTLN         /THEN PRINT LINE
0602 00622 4573 START,  JMS I   [GETCHR         /ANY MORE CHARS ?
0603 00623 5226         JMP     NOTEG
0604 00624 4572         JMS I   [ERMSG          /EXTRA GARBAGE ON LAST LINE
0605 00625 0507         0507                    /*EG*
0606 00626 1064 NOTEG,  TAD     SCSWT           /DID LAST LINE END WITH SEMICOLO
0607 00627 7650         SNA CLA
0610 00630 5235         JMP     .+5             /NO
0611 00631 3064         DCA     SCSWT           /KILL SC SWITCH
0612 00632 2017         ISZ     CHRPTR          /SKIP OVER SEMICOLON
0613 00633 2020         ISZ     NCHARS
0614 00634 5776         JMP I   [ASMBL          /DONT READ NEW LINE
0615 00635 1171         TAD     [LINE-1         /RESET POINTER
0616 00636 3017         DCA     CHRPTR
0617 00637 1375         TAD     (-137           /95 CHARACTERS ONLY
0620 00640 3066         DCA     MAXLIN
0621 00641 3063         DCA     OUTSWT          /CLEAR OUTPUT SWITCH
0622 00642 3361         DCA     LTMSG           /SET LT MESSAGE SWITCH
0623 00643 2024 RDLOOP, ISZ     CHRCNT          /ANY MORE CHARS IN THIS BLOCK?
0624 00644 5274         JMP     NOREAD          /YES, GO GET IT
0625 00645 1335         TAD     M1000           /NUMBER OF CHARS PER BLOCK
0626 00646 3024         DCA     CHRCNT
0627 00647 1170         TAD     [SOURCE         /SOURCE FILE BUFFER
0630 00650 3362         DCA     WDPTR           /INTO POINTER
0631 00651 3360         DCA     DDDEVN          /START WITH EVEN CHAR
```

```
0632 00652 1105          TAD     SBLOCK          /FIND BLOCK NUMBER
0633 00653 1106          TAD     SFUDGE
0634 00654 3270          DCA     RDBLOK
0635 00655 1107          TAD     STAR20          /CHECK FOR FIRST READ
0636 00656 7640          SZA CLA
0637 00657 5265          JMP     .+6
0640 00660 2107          ISZ     STAR20          /SET SWITCH
0641 00661 1374          TAD     (-774           /IGNORE FIRST FOUR CHARACTERS
0642 00662 3024          DCA     CHRCNT
0643 00663 1373          TAD     (SOURCE+2       /SKIP FIRST TWO WORDS
0644 00664 3362          DCA     WDPTR
0645 00665 4501          JMS I   DIALRD          /DIAL READ
0646 00666 0000 SRCUNT,  0                       /SOURCE UNIT
0647 00667 0016          SOURCB                  /CORE LOC OVER 256
0650 00670 0000 RDBLOK,  0                       /BLOCK NUMBER
0651 00671 0001          1                       /NUMBER OF BLOCKS
0652 00672 7200          CLA                     /DOES DIAL RETURN ZERO AC?
0653 00673 2105          ISZ     SBLOCK          /INCREMENT RELATIVE BLOCK NUMBER
0654 00674 1360 NOREAD,  TAD     ODDEVN          /WHICH CHAR?
0655 00675 7640          SZA CLA
0656 00676 5330          JMP     ODDCHR          /ODD ONE
0657 00677 2360          ISZ     ODDEVN          /FLIP SWITCH
0660 00700 1762          TAD I   WDPTR           /GET EVEN CHAR
0661 00701 7012          RTR
0662 00702 7012          RTR
0663 00703 7012          RTR
0664 00704 0167 DOCHR,   AND     [77             /SIX BITS
0665 00705 7450          SNA                     /ZERO IS EOF
0666 00706 5772          JMP I   (ENDX           /SO DO "END"
0667 00707 1166          TAD     [-43            /DIAL CR
0670 00710 7450          SNA
0671 00711 5334          JMP     ENDLIN          /BUMP LINE NUMBER
0672 00712 1371          TAD     (3              /CONVERT TO ASCII
0673 00713 7510          SPA
0674 00714 1165          TAD     [100
0675 00715 1164          TAD     [240
0676 00716 3417          DCA I   CHRPTR
0677 00717 2066          ISZ     MAXLIN          /TEST FOR LINE TOO LONG
0700 00720 5243          JMP     RDLOOP          /PUT CHAR AWAY AND GET NEXT 1
0701 00721 7240          CLA CMA
0702 00722 3066          DCA     MAXLIN          /IGNORE REST OF LINE
0703 00723 7240          CLA CMA
0704 00724 1017          TAD     CHRPTR          /BACK UP BUFFER
0705 00725 3017          DCA     CHRPTR
0706 00726 2361          ISZ     LTMSG           /SET SWITCH
0707 00727 5243          JMP     RDLOOP
0710 00730 3360 ODDCHR,  DCA     ODDEVN
0711 00731 1762          TAD I   WDPTR           /GET ODD CHAR
0712 00732 2362          ISZ     WDPTR           /BUMP WORD POINTER
0713 00733 5304          JMP     DOCHR
0714 00734 2060 ENDLIN,  ISZ     LINENO          /BUMP LINE NUMBER
0715 00735 7000 M1000,   NOP
0716 00736 1060          TAD     LINENO
0717 00737 7421          7421                    /PUT LINE NUM INTO MQ
```

```
0720 00740 7200          CLA                    /NO BUG ON NON-MQ MACHINES
0721 00741 1017          TAD     CHRPTR         /FIND - NUMBER OF CHARS - 1
0722 00742 7040          CMA
0723 00743 1171          TAD     [LINE-1
0724 00744 3020          DCA     NCHARS
0725 00745 7001          IAC                    /SAVE SIZE OF LINE FOR PRINT
0726 00746 1020          TAD     NCHARS
0727 00747 3023          DCA     LINSIZ
0730 00750 1171          TAD     [LINE-1
0731 00751 3017          DCA     CHRPTR         /SET POINTER
0732 00752 1361          TAD     LTMSG          /CHECK FOR LINE TOO LONG MESSAGE
0733 00753 7650          SNA CLA
0734 00754 5776          JMP I   (ASMBL         /NONE, OK
0735 00755 4572          JMS I   [ERMSG         /PRINT LINE TOO LONG MESSAGE
0736 00756 1424          1424                   /*LT*
0737 00757 5776          JMP I   (ASMBL
0740                     MAXLIN=LTEMP
0741 00760 0000 OUDEVN,  0
0742 00761 0000 LTMSG,   0
0743 00762 0000 WDPTR,   0
```

```
0744                            EJECT
     00771 0003
     00772 4600
     00773 7002
     00774 7004
     00775 7641
     00776 1000
     00777 2000
0745                            PAGE
0746 01000 4573 ASMBL,  JMS I   [GETCHR         /LOOK FOR A CHARACTER
0747 01001 5563         JMP I   [NEXTST
0750 01002 1162         TAD     [-257           /IS IT SLASH ?
0751 01003 7450         SNA
0752 01004 5220         JMP     NOASM           /YES, COOL IT
0753 01005 1377         TAD     (257-244        /IS IT $
0754 01006 7450         SNA
0755 01007 5223         JMP     ISDOLR
0756 01010 1376         TAD     (244-240        /IS IT BLANK OR TAB ?
0757 01011 7650         SNA CLA
0760 01012 5200         JMP     ASMBL           /YES, TRY AGAIN
0761 01013 4561         JMS I   [BACK1          /NO, PUT IT BACK
0762 01014 4560         JMS I   [CKKILL         /CHECK FOR ABORT
0763 01015 1057         TAD     ASMOF           /IS ASSEMBLY SWITCHED OFF ?
0764 01016 7650         SNA CLA
0765 01017 5775         JMP I   (LUNAME         /ASSEMBLE STMT
0766 01020 7240 NOASM,  CLA CMA
0767 01021 3020         DCA     NCHARS          /DONT ASSEMBLE THIS LINE
0770 01022 5563         JMP I   [NEXTST         /(PREVENTING *EG* MESSAGE)
0771 01023 3057 ISDOLR, DCA     ASMOF           /TURN ASSEMBLY BACK ON (IF IT WA
0772 01024 5563         JMP I   [NEXTST
0773 01025 0000 OVER3,  0                       /DIVIDE AC BY THREE
0774 01026 3070         DCA     EXTMP2          /MQ
0775 01027 1374         TAD     (-15            /SET SHIFT COUNT
0776 01030 3066         DCA     LTEMP
0777 01031 7100 DIVLUP, CLL                     /ZERO LINK
1000 01032 1373         TAD     (-3             /SUBTRACT DIVISOR FROM AC
1001 01033 7430         SZL                     /IF AC>=3 SET LINK TO 1
1002 01034 5237         JMP     .+3             /OK, DONT RESTORE
1003 01035 1372         TAD     (3              /TOO SMALL, RESTORE AC
1004 01036 7100         CLL                     /SET LINK BACK TO 0
1005 01037 3067         DCA     EXTMP           /SAVE AC
1006 01040 1070         TAD     EXTMP2          /ROTATE MQ-AC LEFT, PUT LINK INT
1007 01041 7004         RAL
1010 01042 3070         DCA     EXTMP2          /SAVE MQ
1011 01043 1067         TAD     EXTMP           /GET BACK AC
1012 01044 7004         RAL                     /COMPLETE SHIFT
1013 01045 2066         ISZ     LTEMP           /TEST COUNT
1014 01046 5231         JMP     DIVLUP          /KEEP GOING
1015 01047 3067         DCA     EXTMP           /THIS IS REMAINDER
1016 01050 1070         TAD     EXTMP2          /RETURN QUOTIENT
1017 01051 5625         JMP I   OVER3
1020 01052 0000 OCTOUT, 0
1021 01053 3005         DCA     QTEMP           /SAVE WORD
```

```
1022 01054 1157          TAD      [-4
1023 01055 3006          DCA      OCNT            /FOUR DIGITS
1024 01056 1005 OLOOP,   TAD      OTEMP
1025 01057 7106          CLL RTL
1026 01060 7004          RAL
1027 01061 3005          DCA      OTEMP           /SAVE SHIFTED WORD
1030 01062 1005          TAD      OTEMP
1031 01063 7004          RAL                      /SHIFT REST OF THE WAY
1032 01064 0156          AND      [7
1033 01065 1155          TAD      [260            /CONVERT TO ASCII
1034 01066 4576          JMS I    [PRINTC
1035 01067 2006          ISZ      OCNT
1036 01070 5256          JMP      OLOOP
1037 01071 1164          TAD      [240            /PRINT BLANK
1040 01072 4576          JMS I    [PRINTC
1041 01073 5652          JMP I    OCTOUT
1042 01074 4561 ENDEXP,  JMS I    [BACK1          /END OF EXPR, PUT BACK CHAR
1043 01075 1100          TAD      LITRL           /LITERAL ?
1044 01076 7640          SZA CLA
1045 01077 4771          JMS I    (CRLIT          /GO CREATE LITERAL
1046 01100 1040          TAD      EXPSW           /DONT SKIP IF NO EXPRESSION
1047 01101 7640          SZA CLA
1050 01102 5770          JMP I    (BAD            /BECAUSE ITS AN ERROR
1051 01103 1055          TAD      LASTOP          /WAS THERE A TRAILING OPERATOR ?
1052 01104 7450          SNA
1053 01105 5767          JMP I    (OKEXP          /NO, JUST RETURN
1054 01106 1366          TAD      (-1             /WAS IT PLUS ?
1055 01107 7450          SNA
1056 01110 1050          TAD      XINCR           /AND IS IT LEGAL ?
1057 01111 7650          SNA CLA
1060 01112 5767          JMP I    (OKEXP          /YES TO BOTH
1061 01113 5770          JMP I    (BAD
1062 01114 4554 LITORX,  JMS I    [ADRGET         /GET ORIGIN
1063 01115 1030          TAD      LITRG1          /PREVIOUS LITORG ?
1064 01116 7700          SMA CLA
1065 01117 5563          JMP I    [NEXTST         /YES, IGNORE THIS ONE
1066 01120 1035          TAD      EXPVAL+1
1067 01121 0156          AND      [7
1070 01122 3030          DCA      LITRG1
1071 01123 1036          TAD      EXPVAL+2
1072 01124 3031          DCA      LITRG2
1073 01125 5563          JMP I    [NEXTST
```

```
1074                            EJECT
     01166 7777
     01167 2444
     01170 2445
     01171 3032
     01172 0003
     01173 7775
     01174 7763
     01175 1200
     01176 0004
     01177 0013
1075                            PAGE
1076 01200 1017 LUNAME, TAD     CHRPTR          /SAVE CHAR STUFF
1077 01201 3021         DCA     CPTMP
1100 01202 1020         TAD     NCHARS
1101 01203 3022         DCA     NCTMP
1102 01204 4553         JMS I   [GETNAM         /LOOK FOR NAME
1103 01205 5754         JMP I   EXPGET          /NONE, MIGHT BE EXPRESSION
1104 01206 4573         JMS I   [GETCHR         /LOOK FOR COMMA
1105 01207 5343         JMP     JSTONE          /ITS JUST ONE SYMBOL
1106 01210 1377         TAD     (-254           /COMMA TEST
1107 01211 7440         SZA
1110 01212 5246         JMP     TRYEQU          /NO COMMA, CHECK FOR EQUAL
1111 01213 4552         JMS I   [LOOKUP         /LOOK UP SYMBOL
1112 01214 5234         JMP     NEWLBL          /ITS COMPLETELY NEW
1113 01215 7650         SNA CLA
1114 01216 5237         JMP     DEFLBL          /ITS UNDEFINED
1115 01217 1410         TAD I   X10             /CHECK LOCCTR AGAINST OLD DEFINI
1116 01220 7041         CIA
1117 01221 1026         TAD     LOCTR1          /FIRST UPPERR HALF
1120 01222 7640         SZA CLA
1121 01223 5231         JMP     .+6
1122 01224 1410         TAD I   X10
1123 01225 7041         CIA
1124 01226 1027         TAD     LOCTR2          /THEN LOWER HALF
1125 01227 7650         SNA CLA
1126 01230 5244         JMP     DEFIND
1127 01231 4572         JMS I   [ERMSG          /MULTIPLY DEFINED
1130 01232 1504         1504                    /*MD*
1131 01233 5776         JMP I   (ASMBL          /FIELD IS OK
1132 01234 7126 NEWLBL, CLL CML RTL             /BUMP NEXT BY 2
1133 01235 1016         TAD     NEXT            /TO MAKE ROOM
1134 01236 3016         DCA     NEXT            /FOR NEW SYMBOL
1135 01237 2466 DEFLBL, ISZ I   LTEMP          /SET TYPE TO 1 (USER ADDR)
1136 01240 1026         TAD     LOCTR1          /PUT LOCATION COUNTER
1137 01241 3410         DCA I   X10             /INTO VALUE
1140 01242 1027         TAD     LOCTR2
1141 01243 3410         DCA I   X10
1142 01244 6201 DEFIND, CDF     FLD0            /GO LOOK FOR ANOTHER TAG
1143 01245 5776         JMP I   (ASMBL
1144 01246 1375 TRYEQU, TAD     (-21            /CHECK FOR EQUAL SIGN
1145 01247 7440         SZA
1146 01250 5340         JMP     TRYBLK          /NO, TRY BLANK
```

```
1147 01251 1052          TAD      NAME1
1150 01252 3071          DCA      EQUN              /SAVE 6 CHARACTER NAME
1151 01253 1053          TAD      NAME2
1152 01254 3072          DCA      EQUN+1
1153 01255 1054          TAD      NAME3
1154 01256 3073          DCA      EQUN+2
1155 01257 1051          TAD      BUCKET
1156 01260 3074          DCA      EQUN+3
1157 01261 4551          JMS I    [EXPR             /GET VALUE RIGHT OF EQUALS
1160 01262 5326          JMP      EQUERR            /BAD EQU
1161 01263 1037          TAD      EXPTYP            /IS EXPR UNDEFINED ?
1162 01264 7650          SNA CLA
1163 01265 5563          JMP I    [NEXTST           /YES, LEAVE SYMBOL UNDEFINED
1164 01266 1071          TAD      EQUN              /RESTORE NAME
1165 01267 3052          DCA      NAME1
1166 01270 1072          TAD      EQUN+1
1167 01271 3053          DCA      NAME2
1170 01272 1073          TAD      EQUN+2
1171 01273 3054          DCA      NAME3
1172 01274 1074          TAD      EQUN+3
1173 01275 3051          DCA      BUCKET
1174 01276 4552          JMS I    [LOOKUP           /LOOKUP SYMBOL
1175 01277 5331          JMP      NEWSYM            /NEW SYMBOL
1176 01300 7450          SNA
1177 01301 7001          IAC                        /REFNCD BUT UNDEF, ASSUME ADDR T
1200 01302 7041          CIA
1201 01303 1037          TAD      EXPTYP            /COMPARE TYPES
1202 01304 7640          SZA CLA
1203 01305 5326          JMP      EQUERR            /TYPE CONFLICT
1204 01306 7144 PUTVAL,  CLL CMA RAL               /-2
1205 01307 1037          TAD      EXPTYP            /GO TO PROPER PLACE
1206 01310 7750          SPA SNA CLA
1207 01311 5314          JMP      MOV2WD
1210 01312 1034          TAD      EXPVAL            /F.P. SYMBOL
1211 01313 3410          DCA I    X10
1212 01314 1035 MOV2WD,  TAD      EXPVAL+1          /D.P. OR ADDRESS SYMBOL
1213 01315 3410          DCA I    X10
1214 01316 1036          TAD      EXPVAL+2
1215 01317 3410          DCA I    X10
1216 01320 1466          TAD I    LTEMP             /NOW GET TYPE WORD
1217 01321 0374          AND      (7740             /ZERO OLD TYPE (PRESERVING FORCE
1220 01322 1037          TAD      EXPTYP            /PUT IN NEW
1221 01323 3466          DCA I    LTEMP             /RESTORE WORD
1222 01324 6201          CDF      FLD0
1223 01325 5563          JMP I    [NEXTST           /GO GET NEXT STMT
1224 01326 4572 EQUERR,  JMS I    [ERMSG            /BAD EQU
1225 01327 0205          0205                       /*BE*
1226 01330 5563          JMP I    [NEXTST
1227 01331 7146 NEWSYM,  CLL CMA RTL               /-3
1230 01332 1037          TAD      EXPTYP            /BUMP NEXT BY CORRECT NUMBER
1231 01333 7700          SMA CLA
1232 01334 2016          ISZ      NEXT              /THRICE FOR FLOATING
1233 01335 2016          ISZ      NEXT              /TWICE FOR DP AND ADDRESS
1234 01336 2016          ISZ      NEXT
```

```
1235 01337 5306          JMP      PUTVAL
1236 01340 1373 TRYBLK,  TAD      (35              /CHECK FOR BLANK
1237 01341 7640          SZA CLA
1240 01342 5754          JMP I    EXPGET           /NO BLANK, GO TRY FOR EXPRESSION
1241 01343 4552 JSTONE,  JMS I    [LOOKUP          /LOOKUP SYMBOL
1242 01344 5772          JMP I    (NEWONE          /ITS A NEW SYMBOL
1243 01345 1353          TAD      OPCTBL           /CREATE JUMP THRU TABLE
1244 01346 3352          DCA      OPCJMP           /SAVE IT
1245 01347 1410          TAD I    X10              /PICK UP FIRST WORD OF VALUE
1246 01350 3045          DCA      OPCODE           /ITS AN OPCODE-MAYBE?
1247 01351 6201          CDF      FLD0
1250 01352 0000 OPCJMP,  0                         /JUMP SOMEWHERE
1251 01353 5754 OPCTBL,  JMP I    .+1
1252 01354 1403 EXPGET,  GETEXP                    /UNDEFINED
1253 01355 1403          GETEXP                    /USER ADDRESS
1254 01356 1403          GETEXP                    /USER DP
1255 01357 1403          GETEXP                    /USER FP
1256 01360 1403          GETEXP                    /PDP 8 OPERATE
1257 01361 1555          PSEUDO                    /PSEUDO OP
1260 01362 1600          PDP8MR                    /PDP8   MR
1261 01363 1431          FPPMR                     /FPPMR
1262 01364 1534          FPPS1                     /OTHER FPP OPCODES
1263 01365 1637          FPPS2
1264 01366 1552          FPPS3
1265 01367 1647          FPPS4
1266 01370 1655          FPPS5
1267 01371 1473          FPPMRL                    /FPP 2 WORD MR FORMAT
```

```
1270                        EJECT
     01372 1400
     01373 0035
     01374 7740
     01375 7757
     01376 1000
     01377 7524
1271                        PAGE
1272 01400 3416 NEWONE,  DCA I   NEXT          /RESERVE 2 WORDS
1273 01401 3416          DCA I   NEXT          /THUS ASSUMING ADDR TYPE
1274 01402 6201          CDF     FLD0
1275 01403 1021 GETEXP,  TAD     CPTMP         /RESTORE CHARACTER POINTER
1276 01404 3017          DCA     CHRPTR
1277 01405 1022          TAD     NCTMP         /TO JUST AFTER TAG (IF ANY)
1300 01406 3020          DCA     NCHARS
1301 01407 4551          JMS I   [EXPR         /TRY FOR AN EXPRESSION
1302 01410 5226          JMP     BADEXP        /IF NONE, ERROR
1303 01411 7144          CLL CMA RAL
1304 01412 1057          TAD     EXPTYP        /CHECK TYPE
1305 01413 7450          SNA
1306 01414 5221          JMP     OUT2WD        /D,P,, OUTPUT 2 WORDS
1307 01415 7710          SPA CLA
1310 01416 5223          JMP     OUT1WD        /ADDRESS, OUTPUT 1 WORD
1311 01417 1034 OUT3WD,  TAD     EXPVAL        /F,P,, OUTPUT 3 WORDS
1312 01420 4550          JMS I   [OUTWRD
1313 01421 1035 OUT2WD,  TAD     EXPVAL+1
1314 01422 4550          JMS I   [OUTWRD
1315 01423 1036 OUT1WD,  TAD     EXPVAL+2
1316 01424 4550          JMS I   [OUTWRD
1317 01425 5563          JMP I   [NEXTST       /GO DO NEXT STMT
1320 01426 4572 BADEXP,  JMS I   [ERMSG        /BAD EXPRESSION
1321 01427 0230          0230
1322 01430 5563          JMP I   [NEXTST       /DO NEXT STMT
1323 01431 4547 FPPMR,   JMS I   [CHKIND       /CHECK FOR INDIRECT
1324 01432 7201          CLA IAC               /SET SWITCH
1325 01433 3047          DCA     INDRCT
1326 01434 4546          JMS I   [GETADR       /GO GET ADDRESS AND INDEX
1327 01435 1077          TAD     FPPWD2        /CHECK FOR FORCED 2 WORD ADDR
1330 01436 7640          SZA CLA
1331 01437 5277          JMP     FORMT1        /FORWD REFNCE, USE 2 WORD FORMAT
1332 01440 1033          TAD     BASER+1
1333 01441 7161          CLL CML CIA           /COMPARE BASE WITH ADDR
1334 01442 1044          TAD     FPPADR+1
1335 01443 3042          DCA     WORD2         /BY DOUBLE SUBTRACTION
1336 01444 7004          RAL
1337 01445 1032          TAD     BASER
1340 01446 7041          CIA
1341 01447 1043          TAD     FPPADR
1342 01450 7640          SZA CLA
1343 01451 5277          JMP     FORMT1        /IF HIGH ORDER WORD NOT 0, LONG
1344 01452 1047          TAD     INDRCT        /IF INDIRECT, USE SHORTEST FORM
1345 01453 7650          SNA CLA
1346 01454 5315          JMP     FORMT3
```

```
1347 01455 1046          TAD     INDEX           /IF INDEX USED, MUST USE LONG
1350 01456 7640          SZA CLA
1351 01457 5277          JMP     FORMT1
1352 01460 1042          TAD     WORD2           /COMPARE ADDR-BASE
1353 01461 7100          CLL
1354 01462 1377          TAD     C-600           /IF <= 128*3, CAN USE SHORT
1355 01463 7630          SZL CLA
1356 01464 5277          JMP     FORMT1          /OTHERWISE USE 2 WORD FMT
1357 01465 1042 FORMT2,  TAD     WORD2           /DIVIDE DISPLACEMENT BY THREE
1360 01466 4776          JMS I   COVER3
1361 01467 1045          TAD     OPCODE          /ADD OPCODE TO DISPLACEMENT
1362 01470 1145          TAD     C200            /TURN ON A BIT
1363 01471 4550          JMS I   COUTWRD         /OUTPUT IT
1364 01472 5563          JMP I   CNEXTST
1365 01473 4547 FPPMRL,  JMS I   CCHKIND         /CHECK FOR "I" (INDIRECT)
1366 01474 7201          CLA IAC                 /SET SWITCH
1367 01475 3047          DCA     INDRCT
1370 01476 4546          JMS I   CGETADR         /GET ADDRESS FIELD
1371 01477 1047 FORMT1,  TAD     INDRCT          /CHECK FOR INDIRECT
1372 01500 7640          SZA CLA                 /IF INDIRECT, THIS IS A NO-NO
1373 01501 5305          JMP     .+4
1374 01502 4572          JMS I   CERMSG          /ILLEGAL INDIRECT
1375 01503 1111          1111                    /*II*
1376 01504 3047          DCA     INDRCT          /CLEAR SWITCH
1377 01505 4544          JMS I   CFIXOPC         /GO PUT IN INDEX AND INCREMENT
1400 01506 1043          TAD     FPPADR          /GET ADDRESS EXTENSION
1401 01507 0156          AND     C7
1402 01510 1045          TAD     OPCODE          /PLUS OPCODE
1403 01511 4550          JMS I   COUTWRD         /OUTPUT IT
1404 01512 1044          TAD     FPPADR+1
1405 01513 4550          JMS I   COUTWRD         /THEN REST OF ADDRESS
1406 01514 5563          JMP I   CNEXTST
1407 01515 1042 FORMT3,  TAD     WORD2           /TRY INDIRECT FORMAT
1410 01516 7100          CLL
1411 01517 1143          TAD     C-30            /DISPLACEMENT CAN BE < OR = TO 7
1412 01520 7620          SNL CLA
1413 01521 5325          JMP     .+4
1414 01522 3042          DCA     WORD2           /SET DISPLACEMENT TO 0
1415 01523 4572          JMS I   CERMSG          /ILLEGAL INDIRECT
1416 01524 1111          1111                    /*II*
1417 01525 4544          JMS I   CFIXOPC         /STICK IN INDEX AND INCR
1420 01526 1042          TAD     WORD2           /GET DISPLACEMENT
1421 01527 4776          JMS I   COVER3          /DIVIDE BY THREE
1422 01530 1045          TAD     OPCODE          /PLUS OPCODE
1423 01531 1145          TAD     C200            /PLUS ANOTHER BIT
1424 01532 4550          JMS I   COUTWRD         /OUTPUT
1425 01533 5563          JMP I   CNEXTST
1426 01534 4547 FPPS1,   JMS I   CCHKIND         /WAS INDIRECT ASKED FOR ?
1427 01535 5340          JMP     .+3             /NO
1430 01536 4572          JMS I   CERMSG          /*II*
1431 01537 1111          1111
1432 01540 4546          JMS I   CGETADR         /GET ADDR, AND INDEX
1433 01541 4544          JMS I   CFIXOPC         /PUT OPCODE TOGETHER
1434 01542 1043          TAD     FPPADR          /GET ADDR EXTENSION
```

```
1435 01543 0156          AND     [7
1436 01544 1045          TAD     OPCODE          /WITH TOGETHER OPCODE
1437 01545 0375          AND     [7377           /WITHDRAW ONE BIT
1440 01546 4550          JMS I   [OUTWRD         /OUTPUT THIS
1441 01547 1044          TAD     FPPADR+1
1442 01550 4550          JMS I   [OUTWRD         /NOW OUTPUT REST OF ADDRESS
1443 01551 5563          JMP I   [NEXTST
1444 01552 1045 FPPS3,   TAD     OPCODE          /JUST PUT OUT OPCODE
1445 01553 4550          JMS I   [OUTWRD
1446 01554 5563          JMP I   [NEXTST
1447 01555 5445 PSEUDO,  JMP I   OPCODE          /GO HANDLE PSEUDO OP
```

```
1450                            EJECT
     01575 7377
     01576 1025
     01577 7200
1451                            PAGE
1452 01600 4271 PUP8MR, JMS     CHKIND          /CHECK FOR INDIRECT
1453 01601 5205         JMP     .+4             /NO
1454 01602 1045         TAD     OPCODE          /PUT INDIRECT INTO OPCODE
1455 01603 1377         TAD     (400
1456 01604 3045         DCA     OPCODE
1457 01605 4323         JMS     ADRGET          /PICK UP ADDRESS FIELD
1460 01606 1036         TAD     EXPVAL+2        /CHECK PAGE OF ADDRESS
1461 01607 0142         AND     [7600
1462 01610 7450         SNA
1463 01611 5233         JMP     PAGE0           /ITS IN PAGE 0
1464 01612 7041         CIA
1465 01613 1027         TAD     LOCTR2          /COMPARE WITH CURRENT PAGE
1466 01614 0142         AND     [7600
1467 01615 7650         SNA CLA
1470 01616 5227         JMP     THSPAG          /OK, ITS THIS PAGE
1471 01617 1045         TAD     OPCODE          /CAN WE USE A LINK ?
1472 01620 0377         AND     (400            /IS INDIRECT BIT ON ?
1473 01621 7650         SNA CLA
1474 01622 5226         JMP     MAKLNK          /YES, GO MAKE LINK
1475 01623 4572         JMS I   [ERMSG          /NOPE, ITS AN ILLEGAL REFERENCE
1476 01624 1122         1122                    /*IR*
1477 01625 5230         JMP     THSPAG+1
1500 01626 4776 MAKLNK, JMS I   (CRLINK         /YES, CREATE LINK
1501 01627 1036 THSPAG, TAD     EXPVAL+2        /GET ADDRESS
1502 01630 0141         AND     [177            /LOWER 7 BITS
1503 01631 1145         TAD     [200            /PUT IN PAGE BIT
1504 01632 7410         SKP
1505 01633 1036 PAGE0,  TAD     EXPVAL+2        /GET ADDRESS (UPPER 5 BITS ZERO)
1506 01634 1045         TAD     OPCODE          /PLUS OPCODE
1507 01635 4550         JMS I   [OUTWRD         /OUTPUT WORD
1510 01636 5563         JMP I   [NEXTST
1511 01637 4323 FPPS2,  JMS     ADRGET          /GET ADDRESS FIELD
1512 01640 1035         TAD     EXPVAL+1        /PUT EXTENSION
1513 01641 0156         AND     [7
1514 01642 1045         TAD     OPCODE          /WITH OPCODE
1515 01643 4550         JMS I   [OUTWRD         /OUT
1516 01644 1036         TAD     EXPVAL+2
1517 01645 4550         JMS I   [OUTWRD         /REST OF ADDR
1520 01646 5563         JMP I   [NEXTST
1521 01647 4323 FPPS4,  JMS     ADRGET          /GET INDEX REG EXPRESSION
1522 01650 1036         TAD     EXPVAL+2        /GET LOWER 3 BITS
1523 01651 0156         AND     [7              /OF INDEX REG EXPR
1524 01652 1045         TAD     OPCODE          /WITH OPCODE
1525 01653 4550         JMS I   [OUTWRD         /OUT
1526 01654 5563         JMP I   [NEXTST
1527 01655 7201 FPPS5,  CLA IAC                 /SET INDEX INCR SWITCH OFF
1530 01656 4337         JMS     GETADR          /GET ADR AND INDEX FIELDS
1531 01657 1046         TAD     INDEX           /WAS THERE AN INDEX?
```

```
1532 01660 7650          SNA CLA
1533 01661 5264          JMP     .+3             /NO
1534 01662 1036          TAD     EXPVAL+2        /YES, GET 3 BITS
1535 01663 0156          AND     [7
1536 01664 1045          TAD     OPCODE          /GET OPCODE
1537 01665 4550          JMS I   [OUTWRD         /OUTPUT
1540 01666 1044          TAD     FPPADR+1        /NOW OUTPUT LOWER 12 BITS
1541 01667 4550          JMS I   [OUTWRD         /OF ADDRESS
1542 01670 5563          JMP I   [NEXTST
1543             XITEMP,
1544 01671 0000 CHKIND,  0                       /CHECK FOR "I"
1545 01672 1017          TAD     CHRPTR          /SAVE CHAR POSITION
1546 01673 3021          DCA     CPTMP
1547 01674 1020          TAD     NCHARS
1550 01675 3022          DCA     NCTMP
1551 01676 4553          JMS I   [GETNAM         /LOOK FOR NAME "I"
1552 01677 5671          JMP I   CHKIND          /IF NO NAME, NO INDIRECT
1553 01700 1051          TAD     BUCKET          /DID IT START WITH "I"?
1554 01701 1375          TAD     (-11
1555 01702 7640          SZA CLA
1556 01703 5316          JMP     NOTIND          /NO, GO AWAY
1557 01704 1052          TAD     NAME1           /WAS "I" ENTIRE NAME?
1560 01705 7640          SZA CLA
1561 01706 5316          JMP     NOTIND          /NO, GO AWAY
1562 01707 2271          ISZ     CHKIND          /YES, SKIP ON RETURN
1563 01710 4573          JMS I   [GETCHR         /LOOK FOR BLANK
1564 01711 5671          JMP I   CHKIND          /NONE LEFT, RETURN
1565 01712 1140          TAD     [-240
1566 01713 7640          SZA CLA
1567 01714 4561          JMS I   [BACK1          /NOT BLANK, BACKUP
1570 01715 5671          JMP I   CHKIND          /RETURN
1571 01716 1021 NOTIND,  TAD     CPTMP           /RESET CHAR POSITION
1572 01717 3017          DCA     CHRPTR
1573 01720 1022          TAD     NCTMP
1574 01721 3020          DCA     NCHARS
1575 01722 5671          JMP I   CHKIND          /RETURN, NO SKIP
1576 01723 0000 ADRGET,  0                       /GET ADDRESS EXPR AND CHECK TYPE
1577 01724 4551          JMS I   [EXPR           /GET EXPR
1600 01725 5332          JMP     ERR1
1601 01726 7344          CLA CLL CMA RAL         /MUST BE TYPE 1 OR 0 (ADR OR UND
1602 01727 1037          TAD     EXPTYP
1603 01730 7750          SPA SNA CLA
1604 01731 5723          JMP I   ADRGET
1605 01732 4572 ERR1,    JMS I   [ERMSG          /BAD ADDR EXPR
1606 01733 0230          0230                    /*BX*
1607 01734 3035          DCA     EXPVAL+1        /SET EXPR TO 0
1610 01735 3036          DCA     EXPVAL+2
1611 01736 5723          JMP I   ADRGET
1612 01737 0000 GETADR,  0                       /GET ADDR, INDEX
1613 01740 3271          DCA     XITEMP          /SAVE INDEX INCREMENT SWITCH
1614 01741 2075          ISZ     FPPSWT          /TELL EXPR ITS AN FPP INSTR
1615 01742 4323          JMS     ADRGET          /GET ADDR
1616 01743 3075          DCA     FPPSWT          /KILL FPP SWITCH
1617 01744 1037          TAD     EXPTYP          /IF EXPR WAS UNDEFINED
```

```
1620 01745 7650          SNA CLA
1621 01746 7001          IAC                     /OR FORCE BIT WAS SET
1622 01747 1476          TAD     FPP2WD
1623 01750 3077          DCA     FPP%D2          /FORCE 2 WORD FORMAT
1624 01751 3046          DCA     INDEX           /ZERO INDEX SWT
1625 01752 1435          TAD     EXPVAL+1        /SAVE ADDRESS VALUE
1626 01753 3043          DCA     FPPADR
1627 01754 1436          TAD     EXPVAL+2
1630 01755 3044          DCA     FPPADR+1
1631 01756 4573          JMS I   [GETCHR         /LOOK FOR COMMA
1632 01757 5737          JMP I   GETADR          /NO INDEX
1633 01760 1374          TAD     (-254
1634 01761 7640          SZA CLA
1635 01762 5371          JMP     ERR2            /NON COMMA IS BAD
1636 01763 2046          ISZ     INDEX           /SET INDEX SWITCH
1637 01764 1271          TAD     XITEMP          /SET INDEX INCREMENT SWITCH
1640 01765 3050          DCA     XINCR
1641 01766 4323          JMS     ADRGET
1642 01767 2050          ISZ     XINCR           /CLEAR INDEX INCREMENT SWITCH
1643 01770 5737          JMP I   GETADR
1644 01771 4572 ERR2,    JMS I   [ERMSG          /BAD ADDR EXPR
1645 01772 0230          0230                    /*BX*
1646 01773 5737          JMP I   GETADR
```

```
1647                    EJECT
     01774 7524
     01775 7767
     01776 3017
     01777 0400
1650                    PAGE
1651 02000 0000 PRNTLN, 0                       /PRINT THE LINE
1652 02001 1063         TAD     OUTSWT          /HAS THE LINE BEEN PRINTED YET?
1653 02002 7640         SZA CLA
1654 02003 5600         JMP I   PRNTLN          /YES, COOL IT
1655 02004 2063         ISZ     OUTSWT          /SET SWITCH
1656 02005 1171         TAD     [LINE-1         /POINTER TO LINE
1657 02006 3010         DCA     X10
1660 02007 1410         TAD I   X10             /GET A CHAR
1661 02010 4576         JMS I   [PRINTC         /PRINT IT
1662 02011 2023         ISZ     LINSIZ          /BUMP COUNT
1663 02012 5207         JMP     .-3
1664 02013 5600         JMP I   PRNTLN
1665 02014 0000 GETUNT, 0                       /GET UNIT FOR CHAIN
1666 02015 4573         JMS I   [GETCHR         /IS ANYTHING THERE ?
1667 02016 5230         JMP     NOUNIT          /TREAT IT AS UNIT 0
1670 02017 4561         JMS I   [BACK1          /PUT BACK EXPR CHAR
1671 02020 4554         JMS I   [ADRGET         /GET UNIT EXPR
1672 02021 1036         TAD     EXPVAL+2
1673 02022 1377         TAD     (-10            /CHECK RANGE
1674 02023 7500         SMA                     /CHECK RANGE
1675 02024 5776         JMP I   [CHERR          /TOO BIG
1676 02025 1375         TAD     (10
1677 02026 7510         SPA
1700 02027 5776         JMP I   [CHERR          /TOO SMALL
1701 02030 3774 NOUNIT, DCA I   [UNIT          /SAVE UNIT
1702 02031 5614         JMP I   GETUNT          /RETURN
1703 02032 1040 FP,     TAD     EXPSW           /FIRST TIME THRU ?
1704 02033 7650         SNA CLA
1705 02034 5237         JMP     .+3             /NO
1706 02035 1373         TAD     (3              /SET TYPE TO 3 (FP)
1707 02036 3037         DCA     EXPTYP
1710 02037 1037         TAD     EXPTYP          /CHECK TYPE OF EXPR SO FAR
1711 02040 7440         SZA                     /0 (UND) IS OK
1712 02041 1372         TAD     (-3             /SO IS 3 (FP)
1713 02042 7640         SZA CLA
1714 02043 5771         JMP I   [MXDTYP         /ANYTHING ELSE IS A NO-NO
1715 02044 1410         TAD I   X10             /GET OPERAND
1716 02045 3355         DCA     OPRAND
1717 02046 1410         TAD I   X10
1720 02047 3356         DCA     OPRAND+1
1721 02050 1410         TAD I   X10
1722 02051 3357         DCA     OPRAND+2
1723 02052 6201         CDF     FLD0
1724 02053 3040         DCA     EXPSW           /CLEAR FIRST TIME SWITCH
1725 02054 1370         TAD     (XFPTBL-1       /GET CORRECT FP OPERATION
1726 02055 1055         TAD     LASTOP          /LAST OPERATOR
1727 02056 3263         DCA     XFOPR           /ADDR OF CORRECT OPERATION
```

```
1730 02057 1663          TAD I   XFOPR           /GET CORRECT OPERATION
1731 02060 3263          DCA     XFOPR           /STORE IT
1732 02061 4177          JMS     177             /ENTER FPPRL
1733 02062 3034          3000    EXPVAL          /FPLAC EXPVAL
1734 02063 0000 XFOPR,   0                       /FPXXX OPRAND
1735 02064 2034          2000    EXPVAL          /FPSTO EXPVAL
1736 02065 0000          0                       /FPEXIT
1737 02066 5537          JMP I   [OPR8R          /GO GET OPERATOR
1740                     XOPRND=OPRAND&177       /PAGE ADDR OF OPRAND
1741                     XBAD=BADX&177           /PAGE ADDR OF BADX
1742 02067 4355 XFPTBL,  4200    XOPRND          /FPADD OPRAND
1743 02070 5355          5200    XOPRND          /FPSUB OPRAND
1744 02071 6355          6200    XOPRND          /FPMUL OPRAND
1745 02072 7355          7200    XOPRND          /FPDIV OPRAND
1746 02073 1275          1200    XBAD            /FPJMP BADX
1747 02074 1275          1200    XBAD            /FPJMP BADX
1750 02075 0000 BADX,    0                       /FPEXIT
1751 02076 5767          JMP I   (BAD            /& ! BAD FOR FP OR DP
1752 02077 4766 INFP,    JMS I   (FLINTP         /GET FP OR DP NUMBER
1753 02100 5232          JMP     FP              /ITS FP
1754 02101 1410          TAD I   X10             /ITS DP, BUT NORMALIZED
1755 02102 3355 DP,      DCA     OPRAND          /PUT EXPONENT
1756 02103 1410          TAD I   X10             /THEN HIGH PART
1757 02104 3356          DCA     OPRAND+1
1760 02105 1410          TAD I   X10             /THEN LOW PART
1761 02106 3357          DCA     OPRAND+2
1762 02107 6201          CDF     FLD0
1763 02110 1040          TAD     EXPSW           /TYPE CHECK
1764 02111 7650          SNA CLA
1765 02112 5315          JMP     .+3
1766 02113 7126          CLL CML RTL             /FIRST TIME SETS TYPE
1767 02114 3037          DCA     EXPTYP
1770 02115 1037          TAD     EXPTYP          /CHECK
1771 02116 7440          SZA                     /0 OR 2 OK
1772 02117 1365          TAD     (-2
1773 02120 7640          SZA CLA
1774 02121 5771          JMP I   (MXDTYP         /MIXED TYPES
1775 02122 3040          DCA     EXPSW
1776 02123 1055          TAD     LASTOP          /GET CORRECT OPERATION
1777 02124 1370          TAD     (XFPTBL-1
2000 02125 3334          DCA     XDOPR
2001 02126 1734          TAD I   XDOPR
2002 02127 3334          DCA     XDOPR
2003 02130 4177          JMS     177             /ENTER FPPRL
2004 02131 3355          3200    XOPRND          /FPLAC OPRAND
2005 02132 2355          2200    XOPRND          /FPSTO OPRAND (NORMALIZES)
2006 02133 3034          3000    EXPVAL          /FPLAC EXPVAL
2007 02134 0000 XDOPR,   0                       /FPXXX OPRAND
2010 02135 0000          0                       /FEXIT
2011 02136 1112 UNNORM,  TAD     ACE             /CHECK EXPONENT
2012 02137 7450          SNA
2013 02140 5347          JMP     MOVEDP          /DONT SHIFT IF EXP 0
2014 02141 7500          SMA
2015 02142 1364          TAD     (-27            /IF POS EXP, INTEGERIZE
```

```
2016 02143 7700          SMA CLA
2017 02144 5347          JMP     MOVEDP          /IF STILL POS DONT SHIFT
2020 02145 4763          JMS I   (AR1            /SHIFT AC RIGHT 1
2021 02146 5336          JMP     UNNORM          /CONTINUE UN-NORMALIZATION
2022 02147 3034 MOVEDP,  DCA     EXPVAL          /ZERO EXPONENT
2023 02150 1116          TAD     ACL
2024 02151 3036          DCA     EXPVAL+2        /LOWER WORD
2025 02152 1117          TAD     ACH
2026 02153 3035          DCA     EXPVAL+1        /UPPER WORD
2027 02154 5537          JMP I   [OPR8R          /GET OPERATOR
2030 02155 0000 OPRAND,  0.0
     02156 0000
     02157 0000
```

```
2031                            EJECT
     02163 0342
     02164 7751
     02165 7776
     02166 2200
     02167 2445
     02170 2066
     02171 2554
     02172 7775
     02173 0003
     02174 5243
     02175 0010
     02176 5347
     02177 7770
2032                            PAGE
2033 02200 0000 FLINTP, 0
2034 02201 3324         DCA     PRSW            /SET PERIOD SWITCH
2035 02202 4326         JMS     DECONV
2036 02203 4573         JMS I   [GETCHR         /GET CHAR
2037 02204 5217         JMP     ENDFPN+1
2040 02205 1136         TAD     [-256           /IS IT "."
2041 02206 7640         SZA CLA
2042 02207 5216         JMP     ENDFPN          /IF NOT, LOOK FOR EXPONENT
2043 02210 1324         TAD     PRSW            /PERIOD FOUND PREVIOUSLY ?
2044 02211 7640         SZA CLA
2045 02212 5221         JMP     PER2            /YES, SECOND PERIOD
2046 02213 3361         DCA     DNUMBR          /ZERO DIGIT COUNT
2047 02214 2324         ISZ     PRSW            /SET PERIOD SWITCH
2050 02215 5334         JMP     DECON           /CONVERT REST OF STRING
2051 02216 4561 ENDFPN, JMS I   [BACK1          /PUT BACK NON PERIOD
2052 02217 1324         TAD     PRSW            /PERIOD READ IN PREVIOUSLY ?
2053 02220 7640         SZA CLA
2054 02221 1361 PER2,   TAD     DNUMBR          /YES, -NUMBER OF DIGITS TO SEXP
2055 02222 7041         CIA                     /NO, ZERO TO SEXP
2056 02223 3325         DCA     SEXP
2057 02224 1065         TAD     RADIX           /SAVE RADIX
2060 02225 3326         DCA     DECONV
2061 02226 2065         ISZ     RADIX           /AND SET RADIX TO DECIMAL
2062 02227 4573         JMS I   [GETCHR         /CHECK TERMINATOR
2063 02230 5267         JMP     FPFIX           /END OF LINE
2064 02231 1377         TAD     (-305           /"E"?
2065 02232 7450         SNA
2066 02233 5237         JMP     .+4             /YES
2067 02234 7001         IAC                     /"D"?
2070 02235 7450         SNA
2071 02236 2200         ISZ     FLINTP          /SKIP ON RETURN IF DP
2072 02237 7640         SZA CLA                 /"D" OR "E" ?
2073 02240 5266         JMP     FPFIX-1         /NO, END OF NUMBER
2074 02241 4573         JMS I   [GETCHR         /LOOK FOR SIGN
2075 02242 5267         JMP     FPFIX           /NO SIGN
2076 02243 1376         TAD     (-253           /IS IT +
2077 02244 7450         SNA
2100 02245 5252         JMP     ISPLUS          /YES
```

```
2101 02246 1375          TAD     (253-255        /IS IT -
2102 02247 7650          SNA CLA
2103 02250 5253          JMP     ISPLUS+1        /YES
2104 02251 4561          JMS I   [BACK1          /PUT IT BACK
2105 02252 7001 ISPLUS,  IAC
2106 02253 3373          DCA     FPESGN          /SET EXP SIGN
2107 02254 4535          JMS I   [NUMBER         /GET EXP
2110 02255 5267          JMP     FPFIX           /NO EXP
2111 02256 1373          TAD     FPESGN          /GET SIGN INDICATOR
2112 02257 7110          CLL RAR                 /INTO LINK
2113 02260 1042          TAD     WORD2           /EXPONENT VALUE
2114 02261 7420          SNL
2115 02262 7041          CIA                     /COMPLEMENT IF SIGN -
2116 02263 1325          TAD     SEXP            /NUMBER OF DIGITS RIGHT OF .
2117 02264 3325          DCA     SEXP            /GIVES MODIFIED EXPONENT
2120 02265 7410          SKP
2121 02266 4561          JMS I   [BACK1          /RETURN CHAR
2122 02267 1325 FPFIX,   TAD     SEXP            /KEEP GOING ?
2123 02270 7450          SNA
2124 02271 5311          JMP     ENDFPI          /NO, RETURN
2125 02272 7700          SMA CLA
2126 02273 5302          JMP     MULT10          /MULT BY 10.
2127 02274 2325          ISZ     SEXP            /INCREMENT BY 1
2130 02275 7000          NOP
2131 02276 4177          JMS     177
2132 02277 7321          TEN&177+7200            /DIVIDE BY 10.
2133 02300 0000          0
2134 02301 5267          JMP     FPFIX
2135 02302 7240 MULT10,  CLA CMA                 /DECREMENT BY 1
2136 02303 1325          TAD     SEXP
2137 02304 3325          DCA     SEXP
2140 02305 4177          JMS     177
2141 02306 6321          TEN&177+6200            /MULTIPLY BY 10.
2142 02307 0000          0
2143 02310 5267          JMP     FPFIX
2144 02311 4177 ENDFPI,  JMS     177             /PUT NUMBER INTO FPTMP
2145 02312 2365          FPTMP&177+2200
2146 02313 0000          0
2147 02314 1326          TAD     DECONV          /RESTORE RADIX
2150 02315 3065          DCA     RADIX
2151 02316 1374          TAD     (FPTMP-1
2152 02317 3010          DCA     X10             /SET POINTER IN X10
2153 02320 5600          JMP I   FLINTP          /RETURN
2154 02321 0004 TEN,     4;2400;0                /10.
     02322 2400
     02323 0000
2155 02324 0000 PRSW,    0
2156 02325 0000 SEXP,    0
2157 02326 0000 DECONV,  0
2160 02327 3112          DCA     ACE             /ZERO FAC
2161 02330 3116          DCA     ACL
2162 02331 3117          DCA     ACH
2163 02332 3115          DCA     ACO
2164 02333 3361          DCA     DNUMBR          /ZERO DIGIT COUNT
```

```
2165 02334 4573 DECON,  JMS I   [GETCHR         /GET A CHARACTER
2166 02335 5726         JMP I   DECONV          /RETURN
2167 02336 1134         TAD     [-272           /TEST FOR DIGIT
2170 02337 7500         SMA
2171 02340 5357         JMP     NOTFPD          /NOT A DIGIT
2172 02341 1133         TAD     [12
2173 02342 7510         SPA
2174 02343 5357         JMP     NOTFPD
2175 02344 3364         DCA     FPDIGT+2        /STORE IN LOW PART OF FP NUMBER
2176 02345 4177         JMS     177
2177 02346 2365         FPTMP&177+2200          /SAVE FAC
2200 02347 3362         FPDIGT&177+3200         /NORMALIZE DIGIT
2201 02350 2370         FPTMP2&177+2200
2202 02351 3365         FPTMP&177+3200          /GET FAC
2203 02352 6321         TEN&177+6200            /MULT BY 10.
2204 02353 4370         FPTMP2&177+4200         /ADD NEW DIGIT
2205 02354 0000         0                       /FEXIT
2206 02355 2361         ISZ     DNUMBR          /INCR DIGIT COUNT
2207 02356 5334         JMP     DECON           /LOOP
2210 02357 4561 NOTFPD, JMS I   [BACK1
2211 02360 5726         JMP I   DECONV
2212 02361 0000 DNUMBR, 0
2213 02362 0027 FPDIGT, 27;0;0
     02363 0000
     02364 0000
2214 02365 0000 FPTMP,  0;0;0
     02366 0000
     02367 0000
2215 02370 0000 FPTMP2, 0;0;0
     02371 0000
     02372 0000
2216 02373 0000 FPESGN, 0
0000                    CHAIN   "FPPASM2"
```

```
0001                    EJECT
     02374 2364
     02375 7776
     02376 7525
     02377 7473
0002                    PAGE
0003 02400 0000 EXPR,   0                       /GET EXPRESSION
0004 02401 3034         DCA     EXPVAL          /ZERO EXPR VALUE
0005 02402 3035         DCA     EXPVAL+1
0006 02403 3036         DCA     EXPVAL+2
0007 02404 3037         DCA     EXPTYP          /AND TYPE
0010 02405 7201         CLA IAC                 /SET EXPR SWITCH TO NO EXPR
0011 02406 3040         DCA     EXPSW
0012 02407 3076         DCA     FPP2WD          /SET FORCE SWITCH OFF
0013 02410 7201         CLA IAC                 /SET LASTOP TO +
0014 02411 3055         DCA     LASTOP
0015 02412 4532         JMS I   [CHKLIT         /GO CHECK FOR LITERAL
0016 02413 4573         JMS I   [GETCHR         /LOOK FOR UNARY+ -
0017 02414 5600         JMP I   EXPR            /NO EXPRESSION
0020 02415 1377         TAD     (-255           /IS IT -
0021 02416 7450         SNA
0022 02417 2055         ISZ     LASTOP          /SET LAST OPER8R TO -
0023 02420 7440         SZA
0024 02421 1376         TAD     (255-253        /IS IT + OR -
0025 02422 7640         SZA CLA
0026 02423 4561         JMS I   [BACK1          /NO, PUT CHAR BACK
0027 02424 4553 SYMBOL, JMS I   [GETNAM         /NOW PICK UP NAME
0030 02425 5247         JMP     NOSYM           /NONE, TRY OTHER
0031 02426 4552         JMS I   [LOOKUP         /LOOK IT UP
0032 02427 5357         JMP     DEFSYM          /NEW SYMBOL
0033 02430 1375         TAD     (-5             /CHECK FOR BADDIES
0034 02431 7500         SMA
0035 02432 5245         JMP     BAD             /BAD EXPRESSION
0036 02433 1236         TAD     SYMTYP
0037 02434 3235         DCA     .+1             /STORE JUMP
0040 02435 0000         0                       /JUMP THRU TABLE
0041 02436 5644 SYMTYP, JMP I   .+6
0042 02437 2600 PUNDEF, UNDEF                   /UNDEFINED
0043 02440 2520         ADR                     /USER ADDRESS
0044 02441 2102         DP                      /USER D.P.
0045 02442 2032         FP                      /USER F.P.
0046 02443 2525         ONE                     /PDP-8 OPERATE
0047 02444 2200 OKEXP,  ISZ     EXPR            /GOOD EXPR, BUMP RETURN
0050 02445 7200 BAD,    CLA
0051 02446 5600         JMP I   EXPR            /RETURN
0052 02447 1017 NOSYM,  TAD     CHRPTR          /NO NAME, SAVE CHAR POS
0053 02450 3067         DCA     EXTMP
0054 02451 1020         TAD     NCHARS
0055 02452 3070         DCA     EXTMP2
0056 02453 4774         JMS I   [NUMBER         /LOOK FOR A NUMBER
0057 02454 5275         JMP     NOTNUM          /NO NUMBER
0060 02455 4573         JMS I   [GETCHR         /LOOK AT NEXT CHAR
0061 02456 5331         JMP     ADREXP          /NO NEXT CHAR, USE NUMBER
```

```
0062 02457 1136          TAD     [-256           /IS CHAR "."
0063 02460 7440          SZA
0064 02461 1373          TAD     (256-305        /IS IT "E" ?
0065 02462 7440          SZA
0066 02463 7001          IAC                     /IS IT "D" ?
0067 02464 7650          SNA CLA
0070 02465 5270          JMP     GETFPN          /TRY FOR DP NUM
0071 02466 4561          JMS I   [BACK1          /OTHERWISE PUT IT BACK
0072 02467 5331          JMP     ADREXP          /GO USE IT
0073 02470 1067 GETFPN,  TAD     EXTMP           /RESET CHAR POINTER
0074 02471 3017          DCA     CHRPTR
0075 02472 1070          TAD     EXTMP2
0076 02473 3020          DCA     NCHARS
0077 02474 5772          JMP I   (INFP           /TRY FOR FP OR DP NUMBER
0100 02475 4573 NOTNUM,  JMS I   [GETCHR         /NOT A NUMBER, GET A CHAR
0101 02476 5771          JMP I   (ENDEXP+1       /NONE LEFT, END
0102 02477 1136          TAD     [-256           /IS IT "." ?
0103 02500 7640          SZA CLA
0104 02501 5770          JMP I   (ENDEXP         /NO, END EXPR
0105 02502 4573          JMS I   [GETCHR         /LOOK AT NEXT CHAR
0106 02503 5313          JMP     ISDOT           /NO NEXT CHAR, ITS LOC SYMBOL
0107 02504 1134          TAD     [-272           /IS NEXT CHAR A DIGIT
0110 02505 7500          SMA
0111 02506 5312          JMP     ISDOT-1         /NO
0112 02507 1133          TAD     [12
0113 02510 7700          SMA CLA
0114 02511 5270          JMP     GETFPN          /YES, TREAT ".N" AS FP NUMBER
0115 02512 4561          JMS I   [BACK1          /OTHERWISE PUT IT BACK
0116 02513 1026 ISDOT,   TAD     LOCTR1          /THIS WAS LOC SYMBOL
0117 02514 3041          DCA     WORD1           /PUT VALUE INTO WORD1,2
0120 02515 1027          TAD     LOCTR2
0121 02516 3042          DCA     WORD2
0122 02517 5331          JMP     ADREXP          /AND USE VALUE
0123 02520 1466 ADR,     TAD I   LTEMP           /CHECK FORCE BIT FOR THIS SYMBOL
0124 02521 0131          AND     [40
0125 02522 7640          SZA CLA
0126 02523 2076          ISZ     FPP2WD          /AND SET SWITCH IF BIT ON
0127 02524 1410          TAD I   X10             /GET FIRST WORD OF VALUE
0130 02525 3041 ONE,     DCA     WORD1           /SINGLE WORD SYMBOL, HIGH=0
0131 02526 1410          TAD I   X10             /GET REST OF SYMBOL
0132 02527 3042          DCA     WORD2
0133 02530 6201          CDF     FLD0            /FIX FIELD
0134 02531 1040 ADREXP,  TAD     EXPSW           /FIRST TIME THROUGH ?
0135 02532 7440          SZA                     /IF SO THEN SET TYPE TO ADDR
0136 02533 3037          DCA     EXPTYP          /OTHERWISE LEAVE IT ALONE
0137 02534 7144          CLL CMA RAL
0140 02535 1037          TAD     EXPTYP          /CHECK FOR TYPE CONFLICT
0141 02536 7700          SMA CLA
0142 02537 5354          JMP     MXDTYP          /NOT - MEANS FP OR DP
0143 02540 3040          DCA     EXPSW           /KILL FIRST TIME SWITCH
0144 02541 1055          TAD     LASTOP          /PICK UP LAST OPERATOR
0145 02542 1345          TAD     ADROP           /MAKE A JMP I
0146 02543 3344          DCA     .+1
0147 02544 0000          0                       /DO IT
```

```
0150 02545 5745 ADROP,  JMP I
0151 02546 2641         ADRADD
0152 02547 2652         ADRSUB
0153 02550 2706         ADRMUL
0154 02551 2445         BAD                     /INTEGER DIVIDE NOT IMPLEMENTED
0155 02552 2664         ADRAND
0156 02553 2673         ADROR
0157 02554 4572 MXDTYP, JMS I   [ERMSG          /MIXED TYPES
0160 02555 1524         1524                    /*MT*
0161 02556 5537         JMP I   [OPR8R
0162 02557 3416 DEFSYM, DCA I   NEXT            /NEW SYMBOL, ALLOCATE 2 WORDS
0163 02560 3416         DCA I   NEXT
0164 02561 5637         JMP I   PUNDEF          /THEN TREAT LIKE UNDEFINED
```

```
0165                            EJECT
     02570 1074
     02571 1075
     02572 2077
     02573 7751
     02574 3475
     02575 7773
     02576 0002
     02577 7523
0166                            PAGE
0167 02600 1475 UNDEF,  TAD     FPPSWT          /IS THIS AN FPP ADDR ?
0170 02601 7650         SNA CLA
0171 02602 5207         JMP     .+5             /NO, SKIP AROUND
0172 02603 1466         TAD I   LTEMP           /TURN ON FORCE BIT
0173 02604 0377         AND     (7737           /FOR THIS SYMBOL
0174 02605 1131         TAD     (40
0175 02606 3466         DCA I   LTEMP
0176 02607 3037         DCA     EXPTYP          /SET TYPE TO UNDEFINED
0177 02610 6201         CDF     FLD0            /FIX FIELD
0200 02611 3040         DCA     EXPSW           /KILL FIRST TIME SWITCH
0201 02612 7144         CLL CMA RAL             /=2
0202 02613 1056         TAD     PASSNO          /IF > PASS 1, UNDEFINED ERROR
0203 02614 7710         SPA CLA
0204 02615 5220         JMP     OPR8R           /UNDEFINED HAS VALUE 0
0205 02616 4572         JMS I   (ERMSG
0206 02617 2523         2523                    /*US*
0207 02620 1376 OPR8R,  TAD     (OPR8RS-1       /SET POINTER
0210 02621 3011         DCA     X11             /TO OPERATOR TABLE
0211 02622 3055         DCA     LASTOP          /ZERO LASTOP
0212 02623 4573         JMS I   (GETCHR         /GET CHAR
0213 02624 5775         JMP I   (ENDEXP+1       /NONE, DONE
0214 02625 3067         DCA     EXTMP           /SAVE IT
0215 02626 1411 FINDOP, TAD I   X11             /GET NEXT LIST ENTRY
0216 02627 7450         SNA
0217 02630 5774         JMP I   (ENDEXP         /ZERO IS END OF LIST
0220 02631 1067         TAD     EXTMP           /COMPARE
0221 02632 7650         SNA CLA
0222 02633 5236         JMP     GOTOP           /THIS IS THE OPERATOR
0223 02634 2011         ISZ     X11             /NO, BUMP PTR
0224 02635 5226         JMP     FINDOP          /LOOP
0225 02636 1411 GOTOP,  TAD I   X11             /PICK UP OTHER VALUE
0226 02637 3055         DCA     LASTOP          /SAVE IN "LASTOP"
0227 02640 5773         JMP I   (SYMBOL         /LOOK FOR OPERAND
0230 02641 1036 ADRADD, TAD     EXPVAL+2        /ADD FOR 15 BIT ADDRESS
0231 02642 7100         CLL                     /ZERO LINK
0232 02643 1042         TAD     WORD2           /ADD LOW WORDS
0233 02644 3036         DCA     EXPVAL+2        /SAVE RESULT
0234 02645 7004         RAL                     /PUT CARRY INTO BIT 11
0235 02646 1035         TAD     EXPVAL+1        /ADD HIGH
0236 02647 1041         TAD     WORD1           /ORDER WORDS
0237 02650 3035         DCA     EXPVAL+1        /SAVE RESULTT
0240 02651 5220         JMP     OPR8R           /LOOK FOR OPERATOR
0241 02652 1042 ADRSUB, TAD     WORD2           /SUBTRACT LOW 12 BITS
```

```
0242 02653 7161          CLL CML CIA
0243 02654 1036          TAD     EXPVAL+2
0244 02655 3036          DCA     EXPVAL+2        /SAVE LOW HALF
0245 02656 7004          RAL
0246 02657 1041          TAD     WORD1           /SUBTRACT HIGH HALF
0247 02660 7041          CIA
0250 02661 1035          TAD     EXPVAL+1
0251 02662 3035          DCA     EXPVAL+1        /SAVE HIGH HALF
0252 02663 5220          JMP     OPR8R           /GET OPERATOR
0253 02664 1041 ADRAND,  TAD     WORD1           /AND
0254 02665 0035          AND     EXPVAL+1        /HIGH
0255 02666 3035          DCA     EXPVAL+1        /HALF
0256 02667 1042          TAD     WORD2           /THEN
0257 02670 0036          AND     EXPVAL+2        /LOW
0260 02671 3036          DCA     EXPVAL+2        /HALF
0261 02672 5220          JMP     OPR8R           /THEN COOL IT
0262 02673 1041 ADROR,   TAD     WORD1           /OR IS THE SAME AS
0263 02674 7040          CMA                     /SETTING THE BITS
0264 02675 0035          AND     EXPVAL+1        /THAT ARE ON IN B AND NOT ON IN
0265 02676 1041          TAD     WORD1           /AND THEN SETTING THE BITS
0266 02677 3035          DCA     EXPVAL+1        /THAT ARE ON IN A
0267 02700 1042          TAD     WORD2
0270 02701 7040          CMA
0271 02702 0036          AND     EXPVAL+2
0272 02703 1042          TAD     WORD2
0273 02704 3036          DCA     EXPVAL+2
0274 02705 5220          JMP     OPR8R           /LOOK FOR OPERATOR
0275 02706 3066 ADRMUL,  DCA     SIGNX           /ZERO SWITCH
0276 02707 1035          TAD     EXPVAL+1        /IS EXPR NEGATIVE
0277 02710 7710          SPA CLA
0300 02711 4352          JMS     COMPL           /YES, COMPLEMENT IT
0301 02712 1372          TAD     (-31
0302 02713 3010          DCA     NBITS           /BIT COUNT
0303 02714 3067          DCA     EXTMP           /ZERO RESULT (HIGH 24 BITS)
0304 02715 3070          DCA     EXTMP2          /RECEPTACLE
0305 02716 1067 MULOOP,  TAD     EXTMP           /ROTATE 48 BIT REGISTER
0306 02717 7110          CLL RAR                 /RIGHT ONE
0307 02720 3067          DCA     EXTMP
0310 02721 1070          TAD     EXTMP2
0311 02722 7010          RAR
0312 02723 3070          DCA     EXTMP2
0313 02724 1035          TAD     EXPVAL+1
0314 02725 7010          RAR
0315 02726 3035          DCA     EXPVAL+1
0316 02727 1036          TAD     EXPVAL+2
0317 02730 7010          RAR
0320 02731 3036          DCA     EXPVAL+2
0321 02732 7420          SNL                     /IS LINK ON?
0322 02733 5344          JMP     NOADD           /NO, DONT ADD
0323 02734 1070          TAD     EXTMP2          /YES, ADD MULTIPLIER
0324 02735 7100          CLL                     /TO HIGH HALF OF 48 BIT
0325 02736 1042          TAD     WORD2           /RESULT
0326 02737 3070          DCA     EXTMP2          /(THIS IS A 2 WORD ADD)
0327 02740 7004          RAL
```

```
0330 02741 1041          TAD     WORD1
0331 02742 1067          TAD     EXTMP
0332 02743 3067          DCA     EXTMP
0333 02744 2010 NOADD,   ISZ     NBITS          /INCREMENT COUNTER
0334 02745 5316          JMP     MULOOP         /LOOP
0335 02746 1066          TAD     SIGNX          /CHECK FOR RE-COMPLEMENT
0336 02747 7640          SZA CLA
0337 02750 4352          JMS     COMPL          /YES, GO DO IT
0340 02751 5220          JMP     OPR6R          /LOOK FOR OPERATOR
0341 02752 0000 COMPL,   0
0342 02753 2066          ISZ     SIGNX          /SET SWITCH
0343 02754 1036          TAD     EXPVAL+2       /COMPLEMENT
0344 02755 7161          CLL CML CIA            /THE
0345 02756 3036          DCA     EXPVAL+2       /TWO
0346 02757 7004          RAL
0347 02760 1035          TAD     EXPVAL+1       /WORD
0350 02761 7041          CIA                    /THING
0351 02762 3035          DCA     EXPVAL+1
0352 02763 5752          JMP I   COMPL          /RETURN
0353                     SIGNX=LTEMP
0354                     NBITS=X10
```

```
0355                            EJECT
     02772 7747
     02773 2424
     02774 1074
     02775 1075
     02776 4053
     02777 7737
0356                            PAGE
0357 03000 0000 CHKLIT, 0                       /CHECK FOR LITERAL
0360 03001 3362         DCA     PAGENO          /ZERO PAGE NUMBER
0361 03002 3100         DCA     LITRL
0362 03003 4573         JMS I   [GETCHR         /GET CHARACTER
0363 03004 5600         JMP I   CHKLIT          /NO LITERAL
0364 03005 1377         TAD     (-250           /CHECK FOR (
0365 03006 7450         SNA
0366 03007 2362         ISZ     PAGENO          /CURRENT PAGE LITERAL
0367 03010 7440         SZA                     /SKIP IF ALREADY ZERO
0370 03011 1376         TAD     (-63            /CHECK FOR [
0371 03012 7450         SNA
0372 03013 2100         ISZ     LITRL           /SET SWITCH
0373 03014 7640         SZA CLA
0374 03015 4561         JMS I   [BACK1          /PUT BACK NON ([
0375 03016 5600         JMP I   CHKLIT
0376 03017 0000 CRLINK, 0                       /CREATE LINK
0377 03020 1217         TAD     CRLINK          /FIX RETURN ADDR
0400 03021 3232         DCA     CRLIT
0401 03022 1045         TAD     OPCODE          /SET INDIRECT BIT
0402 03023 1375         TAD     (400
0403 03024 3045         DCA     OPCODE
0404 03025 7201         CLA IAC
0405 03026 3362         DCA     PAGENO          /SET INDICATOR
0406 03027 4572         JMS I   [ERMSG          /*LG*
0407 03030 1407         1407
0410 03031 5245         JMP     NOTP0
0411 03032 0000 CRLIT,  0                       /CREATE LITERAL WITH TYPE EXPTYP
0412                                            /VALUE:EXPVAL, IN PAGE:PAGENO
0413 03033 1075         TAD     FPPSWT          /FP LITERAL ?
0414 03034 7640         SZA CLA
0415 03035 5774         JMP I   (FPPLIT
0416 03036 1362         TAD     PAGENO          /CHECK FOR PAGE 0
0417 03037 7640         SZA CLA
0420 03040 5245         JMP     NOTP0           /NOT A PAGE 0 LITERAL
0421 03041 1373         TAD     (P0LBUF         /SET BASE TO PAGE 0 LIT BUFFER
0422 03042 3364         DCA     LITBAS
0423 03043 1372         TAD     (17             /ASSUME FIRST 20 WORDS USED
0424 03044 5251         JMP     DOLIT           /GO DO LITERAL
0425 03045 1371 NOTP0,  TAD     (CPLBUF         /CURRENT PAGE LIT BUFFER
0426 03046 3364         DCA     LITBAS
0427 03047 1027         TAD     LOCTR2          /NUMBER OF WORDS USED IN PAGE
0430 03050 0141         AND     [177
0431 03051 3365 DOLIT,  DCA     NWUSED
0432 03052 1362         TAD     PAGENO          /GET POINTER TO
0433 03053 1130         TAD     [P0LIT          /LITERAL BOUNDARY
```

```
0434 03054 3363          DCA      XPAGE
0435 03055 1763          TAD I    XPAGE          /DISPLACEMENT OF LIT BUFR - 1
0436 03056 3366          DCA      LITPTR         /INTO LITPTR
0437 03057 1037          TAD      EXPTYP         /TYPE IS ACTUALLY SIZE OF LITERA
0440 03060 7440          SZA                     /CHECK FOR UNDEFINED
0441 03061 5264          JMP      .+3            /OK, ITS DEFINED
0442 03062 3036          DCA      EXPVAL+2       /UNDEFINED, ASSUME ZERO
0443 03063 7001          IAC                     /AND ADDRESS TYPE
0444 03064 7041          CIA
0445 03065 3037          DCA      EXPTYP         /GET THE COMPLEMENT
0446 03066 1370          TAD      (EXPVAL+2
0447 03067 1037          TAD      EXPTYP         /GET POINTER TO VALUE
0450 03070 3012          DCA      X12            /MINUS ONE INTO X12
0451 03071 1037 NOTIT,   TAD      EXPTYP         /CHECK FOR END OF TABLE
0452 03072 7041          CIA
0453 03073 1366          TAD      LITPTR         /POINTER+SIZE
0454 03074 0145          AND      [200           /SHOULD BE LESS THAN 200
0455 03075 7640          SZA CLA
0456 03076 5331          JMP      NEWLIT         /ENTER NEW LITERAL
0457 03077 1037          TAD      EXPTYP         /PUT COPY OF LENGTH
0460 03100 3066          DCA      LTEMP          /INTO LTEMP
0461 03101 1012          TAD      X12            /AND COPY OF PROTOTYPE POINTER
0462 03102 3015          DCA      X15            /INTO X15
0463 03103 1366          TAD      LITPTR         /NOW GET POINTER
0464 03104 1364          TAD      LITBAS         /TO TABLE
0465 03105 3011          DCA      X11            /FOR COMPARISON
0466 03106 2366          ISZ      LITPTR         /INCREMENT POINTER
0467 03107 1411 TSTLIT,  TAD I    X11            /GET WORD OF LITERAL
0470 03110 7041          CIA
0471 03111 1415          TAD I    X15            /COMPARE WITH PROTOTYPE
0472 03112 7640          SZA CLA
0473 03113 5271          JMP      NOTIT          /NOT IT, SLIDE POINTER AND RETRY
0474 03114 2066          ISZ      LTEMP          /BUMP COUNTER
0475 03115 5307          JMP      TSTLIT         /LOOP IF MORE
0476 03116 1362 LITADR,  TAD      PAGENO         /PAGE 0 ?
0477 03117 7640          SZA CLA
0500 03120 1027          TAD      LOCTR2         /NO, CURRENT PAGE, GET ADDRESS
0501 03121 0142          AND      [7600
0502 03122 1366          TAD      LITPTR         /PLUS PAGE DISPLACEMENT
0503 03123 3036          DCA      EXPVAL+2       /INTO VALUE
0504 03124 1026          TAD      LOCTR1
0505 03125 3035 RETLIT,  DCA      EXPVAL+1
0506 03126 7201          CLA IAC                 /SET TYPE TO ADDRESS
0507 03127 3037          DCA      EXPTYP
0510 03130 5632          JMP I    CRLIT
0511 03131 1763 NEWLIT,  TAD I    XPAGE          /MOVE LITERAL BOUNDARY DOWN
0512 03132 1037          TAD      EXPTYP
0513 03133 3010          DCA      X10            /ADDRESS OF NEW LITERAL
0514 03134 1365          TAD      NWUSED         /CHECK FOR PAGE OVERFULL
0515 03135 7041          CIA
0516 03136 1010          TAD      X10
0517 03137 7700          SMA CLA
0520 03140 5345          JMP      .+5            /NOT FULL
0521 03141 4572          JMS I    [ERMSG         /*PO*
```

```
0522 03142 2017          2017
0523 03143 3036          DCA      EXPVAL+2        /ZERO ADDRESS
0524 03144 5325          JMP      RETLIT
0525 03145 1010          TAD      X10
0526 03146 3763          DCA I    XPAGE
0527 03147 1763          TAD I    XPAGE           /SET UP POINTER FOR MOVE
0530 03150 1364          TAD      LITBAS
0531 03151 3010          DCA      X10
0532 03152 1412          TAD I    X12             /MOVE LITERAL INTO PLACE
0533 03153 3410          DCA I    X10
0534 03154 2037          ISZ      EXPTYP
0535 03155 5352          JMP      .-3
0536 03156 1763          TAD I    XPAGE           /SET UP LITERAL ADDRESS
0537 03157 7001          IAC
0540 03160 3366          DCA      LITPTR
0541 03161 5316          JMP      LITADR          /RETURN LITERAL ADDRESS
0542 03162 0000 PAGENO,  0
0543 03163 0000 XPAGE,   0
0544 03164 0000 LITBAS,  0
0545 03165 0000 NWUSED,  0
0546 03166 0000 LITPTR,  0
```

```
0547                            EJECT
     03170 0036
     03171 6000
     03172 0017
     03173 7400
     03174 3200
     03175 0400
     03176 7715
     03177 7530
0550                            PAGE
0551 03200 7201 FPPLIT, CLA IAC
0552 03201 3076          DCA    FPP2WD          /FORCE 2 WORD INSTRUCTION
0553 03202 7240          CLA CMA                /WHAT PASS ?
0554 03203 1056          TAD    PASSNO
0555 03204 7650          SNA CLA
0556 03205 5777          JMP I  (RETLIT         /PASS 1 DONT GENERATE FP LITERAL
0557 03206 4355          JMS    TYPE3           /SKIP IF FP OR ADDR TYPE
0560 03207 7001          IAC                    /DP TYPE
0561 03210 1376          TAD    (FPLIST         /POINTER TO FP LITERAL BUCKET
0562 03211 6211          CDF    FLD1
0563 03212 3353 FPLOOK, DCA     OLDFP           /START ADDR OF CURRENT BLOCK
0564 03213 1753          TAD I  OLDFP           /ADDR OF NEXT BLOCK
0565 03214 7450          SNA
0566 03215 5256          JMP    NEWFPL          /IF 0 THEN END OF LIST
0567 03216 3010          DCA    X10
0570 03217 1410          TAD I  X10             /GET DISPLACEMENT
0571 03220 3066          DCA    LTEMP           /SAVE IT
0572 03221 4355          JMS    TYPE3
0573 03222 5230          JMP    CMP2WD          /DP, ONLUT 2 WORDS
0574 03223 1034          TAD    EXPVAL          /DO 3 WORD COMPARE
0575 03224 7141          CIA CLL
0576 03225 1410          TAD I  X10
0577 03226 7640          SZA CLA
0600 03227 5252          JMP    DIFFPL          /NOT THE SAME
0601 03230 1035 CMP2WD, TAD     EXPVAL+1
0602 03231 7141          CIA CLL
0603 03232 1410          TAD I  X10
0604 03233 7640          SZA CLA
0605 03234 5252          JMP    DIFFPL
0606 03235 1036          TAD    EXPVAL+2
0607 03236 7141          CIA CLL
0610 03237 1410          TAD I  X10
0611 03240 7640          SZA CLA
0612 03241 5252          JMP    DIFFPL
0613 03242 1066 RETFPL, TAD     LTEMP           /GET LITERAL DISPLACEMENT
0614 03243 6201          CDF    FLD0
0615 03244 7100          CLL                    /ADD LITORG ADDR
0616 03245 1031          TAD    LITRG2
0617 03246 3036          DCA    EXPVAL+2        /AND MAKE IT THE NEW EXPVAL
0620 03247 7004          RAL
0621 03250 1030          TAD    LITRG1
0622 03251 5777          JMP I  (RETLIT
0623 03252 7430 DIFFPL, SZL
```

```
0624 03253 5256          JMP      NEWFPL          /INSERT NEW LITERAL HERE
0625 03254 1753          TAD I    OLDFP           /GET START ADDR OF THIS BLOCK
0626 03255 5212          JMP      FPLOOK
0627 03256 1753 NEWFPL,  TAD I    OLDFP           /HOOK IN NEW FP LITERAL
0630 03257 3416          DCA I    NEXT
0631 03260 1016          TAD      NEXT
0632 03261 3753          DCA I    OLDFP
0633 03262 1354          TAD      FPLNUM          /PUT IN DISPLACEMENT
0634 03263 3416          DCA I    NEXT
0635 03264 4355          JMS      TYPE3           /2 OR 3 WORDS
0636 03265 5270          JMP      .+3
0637 03266 1034          TAD      EXPVAL          /STORE VALUE
0640 03267 3416          DCA I    NEXT
0641 03270 1035          TAD      EXPVAL+1
0642 03271 3416          DCA I    NEXT
0643 03272 1036          TAD      EXPVAL+2
0644 03273 3416          DCA I    NEXT
0645 03274 1016          TAD      NEXT            /CHECK FOR ST OVERFLOW
0646 03275 7100          CLL
0647 03276 1133          TAD      [12
0650 03277 7620          SNL CLA
0651 03300 5304          JMP      .+4             /OK, NOT FULL
0652 03301 4527          JMS I    [ERMSG1
0653 03302 2324          2324                     /*ST*
0654 03303 5775          JMP I    [RETSYS         /SINCE ITS FATAL, ABORT
0655 03304 1354          TAD      FPLNUM          /SAVE DISPLACEMENT
0656 03305 3066          DCA      LTEMP
0657 03306 2354          ISZ      FPLNUM          /BUMP DISPLACEMENT BY 3
0660 03307 2354          ISZ      FPLNUM
0661 03310 4355          JMS      TYPE3           /OR MAYBE 2
0662 03311 7410          SKP
0663 03312 2354          ISZ      FPLNUM
0664 03313 5242          JMP      RETFPL
0665 03314 0000 DLITS,   0                        /DUMP FP LITERALS
0666 03315 7144          CLL CMA RAL              /2 LISTS
0667 03316 3355          DCA      TYPE3           /SAVE COUNT IN SUBR ENTRY
0670 03317 1376 DLITS2,  TAD      [FPLIST         /FP LITERAL BUCKET
0671 03320 6211          CDF      FLD1
0672 03321 3353 FPLDMP,  DCA      OLDFP
0673 03322 1753          TAD I    OLDFP           /GET ADDR OF NEXT FP LITERAL
0674 03323 7440          SZA
0675 03324 5333          JMP      MORFPL
0676 03325 6201          CDF      FLD0
0677 03326 7201          CLA IAC                  /SET AC=1
0700 03327 2355          ISZ      TYPE3           /FINISHED YET
0701 03330 5317          JMP      DLITS2          /NO, DO OP LIST
0702 03331 7200          CLA
0703 03332 5714          JMP I    DLITS           /YES, RETURN
0704 03333 3010 MORFPL,  DCA      X10             /SAVE POINTER
0705 03334 1410          TAD I    X10             /GET DISPLACEMENT
0706 03335 7100          CLL
0707 03336 1031          TAD      LITRG2          /ADD LITERAL ORG
0710 03337 3027          DCA      LOCTR2          /AND PUT IT INTO LOCATION COUNTE
0711 03340 7004          RAL
```

```
0712 03341 1030          TAD      LITRG1
0713 03342 3026          DCA      LOCTR1
0714 03343 1355          TAD      TYPE3          /2 OR 3 WORDS ?
0715 03344 7001          IAC
0716 03345 7640          SZA CLA
0717 03346 4363          JMS      OUTFPL         /THREE
0720 03347 4363          JMS      OUTFPL
0721 03350 4363          JMS      OUTFPL
0722 03351 1753          TAD I    OLDFP          /POINTER TO NEXT BLOCK
0723 03352 5321          JMP      FPLDMP
0724 03353 0000 OLDFP,   0
0725 03354 0000 FPLNUM,  0
0726 03355 0000 TYPE3,   0                      /SKIP ON ADDR OR FP
0727 03356 7144          CLL CMA RAL            /-2
0730 03357 1037          TAD      EXPTYP
0731 03360 7640          SZA CLA
0732 03361 2355          ISZ      TYPE3          /NOT TYPE 2
0733 03362 5755          JMP I    TYPE3
0734 03363 0000 OUTFPL,  0                      /PUT WORD FROM LITERALS
0735 03364 1410          TAD I    X10            /GET WORD
0736 03365 6201          CDF      FLD0
0737 03366 4550          JMS I    [OUTWRD
0740 03367 6211          CDF      FLD1
0741 03370 5763          JMP I    OUTFPL
```

```
0742                    EJECT
     03375 4131
     03376 0020
     03377 3125
0743                    PAGE
0744 03400 0000 LOOKUP, 0                       /SYMBOL TABLE LOOKUP
0745 03401 1051         TAD     BUCKET          /GET BUCKET ADDRESS
0746 03402 1377         TAD     (BUCKTS-1
0747 03403 3051         DCA     BUCKET          /INTO "BUCKET"
0750 03404 1451         TAD I   BUCKET          /GET ADDR OF BUCKET BOTTOM
0751 03405 6211         CDF     FLD1            /GO TO FIELD 1
0752 03406 3004 LOOK,   DCA     OLDN3           /THIS IS PTR OF PREV ENTRY
0753 03407 1404         TAD I   OLDN3           /THIS IS ADR OF NEXT ENTRY
0754 03410 7450         SNA                     /IF ZERO, THEN
0755 03411 5247         JMP     HOOKIN          /GO HOOK IN AT THE END
0756 03412 3010         DCA     X10             /SAVE NEXT NAME PTR
0757 03413 1052         TAD     NAME1           /COMPARE NAMES
0760 03414 7141         CIA CLL
0761 03415 1410         TAD I   X10             /WORD 1
0762 03416 7640         SZA CLA
0763 03417 5243         JMP     NOTSAM
0764 03420 1053         TAD     NAME2
0765 03421 7141         CIA CLL
0766 03422 1410         TAD I   X10             /WORD2
0767 03423 7640         SZA CLA
0770 03424 5243         JMP     NOTSAM
0771 03425 1410         TAD I   X10
0772 03426 0376         AND     (7700           /WORD 3 (ONLY UPPER HALF)
0773 03427 3066         DCA     LTEMP
0774 03430 1054         TAD     NAME3
0775 03431 7141         CIA CLL
0776 03432 1066         TAD     LTEMP
0777 03433 7640         SZA CLA
1000 03434 5243         JMP     NOTSAM
1001 03435 2200         ISZ     LOOKUP          /IF FOUND BUMP RETURN
1002 03436 1010         TAD     X10
1003 03437 3066         DCA     LTEMP           /ADDR OF TYPE WORD
1004 03440 1466         TAD I   LTEMP           /GET TYPE INTO AC
1005 03441 0126         AND     [37             /WITHOUT FORCE BIT
1006 03442 5600         JMP I   LOOKUP          /RETURN
1007 03443 7430 NOTSAM, SZL                     /NAMES DIFFER, IS NAME 1,2,3 < E
1010 03444 5247         JMP     HOOKIN          /YES, HOOK IN HERE
1011 03445 1404         TAD I   OLDN3           /GET ADDR OF LINK INTO AC
1012 03446 5206         JMP     LOOK            /LOOP
1013 03447 1404 HOOKIN, TAD I   OLDN3           /GET FORWARD LINK TO
1014 03450 3416         DCA I   NEXT            /NEXT ENTRY INTO NEW ENTRY
1015 03451 1016         TAD     NEXT            /PUT FORWARD LINK TO NEW
1016 03452 3404         DCA I   OLDN3           /ENTRY INTO PREVIOUS ENTRY
1017 03453 1052         TAD     NAME1           /PUT IN NAME
1020 03454 3416         DCA I   NEXT
1021 03455 1053         TAD     NAME2
1022 03456 3416         DCA I   NEXT
1023 03457 1054         TAD     NAME3
```

```
1024 03460 3416          DCA I    NEXT
1025 03461 1016          TAD      NEXT            /X10=NEXT
1026 03462 3010          DCA      X10
1027 03463 1016          TAD      NEXT            /LTEMP=NEXT
1030 03464 3066          DCA      LTEMP
1031 03465 1016          TAD      NEXT            /CHECK FOR TABLE FULL
1032 03466 7100          CLL
1033 03467 1133          TAD      [12
1034 03470 7620          SNL CLA
1035 03471 5600          JMP I    LOOKUP          /NO PROBLEMS, RETURN (NO SKIP)
1036 03472 4527          JMS I    [ERMSG1
1037 03473 2324          2324                     /*ST*
1040 03474 5775          JMP I    [RETSYS         /ST FULL, ABORT
1041 03475 0000 NUMBER,  0                        /GET INTEGER NUMBER (NO SIGN)
1042 03476 3352          DCA      NSWTCH          /CLEAR SWITCH
1043 03477 3041          DCA      WORD1           /CLEAR 24 BIT NUMBER
1044 03500 3042          DCA      WORD2
1045 03501 4573 NUMLUP,  JMS I    [GETCHR         /GET A CHAR
1046 03502 5346          JMP      NODGT+1         /NONE LEFT
1047 03503 1134          TAD      [-272           /IS IT A DIGIT?
1050 03504 7500          SMA
1051 03505 5345          JMP      NODGT           /NO, TOO BIT
1052 03506 1133          TAD      [12
1053 03507 7510          SPA
1054 03510 5345          JMP      NODGT           /NO, TOO SMALL
1055 03511 3353          DCA      NUM             /YES, SAVE IT
1056 03512 1041          TAD      WORD1           /SAVE CURRENT VALUE
1057 03513 3354          DCA      NUM1            /OF NUMBER
1060 03514 1042          TAD      WORD2
1061 03515 3355          DCA      NUM2
1062 03516 4356          JMS      SHIFT           /SHIFT WORD1,2, LEFT 1 (MULT BY
1063 03517 4356          JMS      SHIFT           /DO IT AGAIN (MULT BY 4)
1064 03520 1065          TAD      RADIX           /LOOK AT RADIX (1=DECIMAL)
1065 03521 7650          SNA CLA
1066 03522 5333          JMP      OCTNUM          /ITS OCTAL
1067 03523 7100          CLL                      /DECIMAL, ADD IN NUMBER
1070 03524 1355          TAD      NUM2
1071 03525 1042          TAD      WORD2           /THUS MULTIPLYING BY 5
1072 03526 3042          DCA      WORD2
1073 03527 7004          RAL
1074 03530 1354          TAD      NUM1
1075 03531 1041          TAD      WORD1
1076 03532 3041          DCA      WORD1
1077 03533 4356 OCTNUM,  JMS      SHIFT           /SHIFT LEFT 1 AGAIN, THUS
1100 03534 1042          TAD      WORD2           /MULTIPLYING BY 8 OR 10
1101 03535 7100          CLL                      /THEN ADD IN NEW DIGIT
1102 03536 1353          TAD      NUM
1103 03537 3042          DCA      WORD2
1104 03540 7004          RAL
1105 03541 1041          TAD      WORD1
1106 03542 3041          DCA      WORD1
1107 03543 2352          ISZ      NSWTCH          /SET SWITCH
1110 03544 5301          JMP      NUMLUP          /LOOP
1111 03545 4561 NODGT,   JMS I    [BACK1          /PUT BACK NON-DIGIT
```

```
1112 03546 1352          TAD     NSWTCH          /WAS THERE A NUMBER
1113 03547 7640          SZA CLA
1114 03550 2275          ISZ     NUMBER          /YES, SKIP
1115 03551 5675          JMP I   NUMBER          /RETURN
1116 03552 0000 NSWTCH,  0
1117 03553 0000 NUM,     0
1120 03554 0000 NUM1,    0
1121 03555 0000 NUM2,    0
1122 03556 0000 SHIFT,   0                       /SHIFT DOUBLE WORD LEFT 1
1123 03557 1042          TAD     WORD2
1124 03560 7104          CLL RAL
1125 03561 3042          DCA     WORD2
1126 03562 1041          TAD     WORD1
1127 03563 7004          RAL
1130 03564 3041          DCA     WORD1
1131 03565 5756          JMP I   SHIFT
```

```
1132                            EJECT
     03575 4131
     03576 7700
     03577 4021
1133                            PAGE
1134 03600 0000 BACK1,  0
1135 03601 7240         CLA CMA                 /BACKUP COUNT
1136 03602 1020         TAD     NCHARS
1137 03603 3020         DCA     NCHARS
1140 03604 7240         CLA CMA                 /AND POINTER
1141 03605 1017         TAD     CHRPTR
1142 03606 3017         DCA     CHRPTR
1143 03607 5600         JMP I   BACK1
1144 03610 0000 GETNAM, 0                       /GET A NAME
1145 03611 4311         JMS     GETCHR          /GET FIRST CHAR
1146 03612 5610         JMP I   GETNAM          /NO CHAR, NO NAME
1147 03613 4255         JMS     LETTER          /FIRST CHAR MUST BE LETTER
1150 03614 5251         JMP     NONAME          /OTHERWISE, NO NAME
1151 03615 0167         AND     (77
1152 03616 3051         DCA     BUCKET          /THIS CHAR IS BUCKET
1153 03617 3052         DCA     NAME1
1154 03620 3053         DCA     NAME2           /ZERO REST
1155 03621 3054         DCA     NAME3           /OF NAME
1156 03622 1377         TAD     (NAME1          /SET
1157 03623 3253         DCA     NPTR            /POINTER
1160 03624 7146         CLL CMA RTL             /AND
1161 03625 3254         DCA     NCNT            /COUNTER
1162 03626 4270 PAKLUP, JMS     GNC             /GET NAME CHAR (LETTER OR DIGIT)
1163 03627 7106         CLL RTL                 /SHIFT TO UPPER HALF
1164 03630 7006         RTL
1165 03631 7006         RTL
1166 03632 3653         DCA I   NPTR            /SAVE HALF
1167 03633 4270         JMS     GNC             /GET NEXT CHAR
1170 03634 1653         TAD I   NPTR            /PUT 2 TOGETHER
1171 03635 3653         DCA I   NPTR            /STORE
1172 03636 2253         ISZ     NPTR            /BUMP POINTER
1173 03637 2254         ISZ     NCNT            /AND COUNTER
1174 03640 5226         JMP     PAKLUP          /LOOP
1175 03641 4270         JMS     GNC             /SKIP AMY EXTRA CHARS
1176 03642 7200         CLA
1177 03643 5241         JMP     .-2
1200 03644 1054 NOONE,  TAD     NAME3           /ZERO TYPE FIELD
1201 03645 0376         AND     (7700
1202 03646 3054         DCA     NAME3
1203 03647 2210         ISZ     GETNAM          /SKIP
1204 03650 5610         JMP I   GETNAM          /RETURN
1205 03651 4200 NONAME, JMS     BACK1           /PUT CHAR BACK
1206 03652 5610         JMP I   GETNAM          /NO-SKIP, RETURN
1207 03653 0000 NPTR,   0
1210 03654 0000 NCNT,   0
1211 03655 0000 LETTER, 0                       /IS AC A LETTER?
1212 03656 1375         TAD     (-301
1213 03657 7510         SPA
```

```
1214 03660 5266          JMP      NLETR          /NO, TOO SMALL
1215 03661 1374          TAD      (-32
1216 03662 7510          SPA
1217 03663 2255          ISZ      LETTER         /YES, INCR RETURN
1220 03664 1373          TAD      (333           /RESTORE CHAR
1221 03665 5655          JMP I    LETTER         /RETURN
1222 03666 1372 NLETR,   TAD      (301           /RESTORE NON-LETTER
1223 03667 5655          JMP I    LETTER         /RETURN
1224 03670 0000 GNC,     0                       /GET A CHAR IF LETTER OR DIGIT
1225 03671 4311          JMS      GETCHR
1226 03672 5244          JMP      NDONE          /NONE LEFT
1227 03673 4255          JMS      LETTER         /IS IT A LETTER?
1230 03674 5277          JMP      .+3            /NO
1231 03675 0167          AND      [77            /ONLY 6 BITS
1232 03676 5670          JMP I    GNC            /RETURN
1233 03677 1134          TAD      [-272          /CHECK FOR DIGIT
1234 03700 7500          SMA
1235 03701 5307          JMP      .+6            /NO, TOO BIG
1236 03702 1133          TAD      [12
1237 03703 7510          SPA
1240 03704 5307          JMP      .+3            /NO, TOO SMALL
1241 03705 1371          TAD      (60            /OK, MAKE IT 60-71
1242 03706 5670          JMP I    GNC            /RETURN
1243 03707 4200          JMS      BACK1          /PUT BACK NON LETTER/DIGIT
1244 03710 5244          JMP      NDONE          /NAME DONE
1245 03711 0000 GETCHR,  0                       /GET NEXT CHAR
1246 03712 2020          ISZ      NCHARS         /BUMP COUNT
1247 03713 5317          JMP      .+4            /YES VIRGINIA, THERE ARE MORE
1250 03714 7240 GETC2,   CLA CMA                 /RESET COUNT
1251 03715 3020          DCA      NCHARS         /TO MINUS1
1252 03716 5711          JMP I    GETCHR         /RETURN WITH NO SKIP
1253 03717 1417          TAD I    CHRPTR         /GET CHAR
1254 03720 1140          TAD      [-240          /CHECK FOR BLANK
1255 03721 7450          SNA
1256 03722 5334          JMP      BLANK          /YES, BLANK
1257 03723 1370          TAD      (-7            /CHECK FOR TAB
1260 03724 7450          SNA
1261 03725 5334          JMP      BLANK          /SAME AS BLANK
1262 03726 1367          TAD      (247-273       /CHECK FOR SEMI COLON
1263 03727 7450          SNA
1264 03730 5360          JMP      SEMICL         /YUP!
1265 03731 1366          TAD      (273           /FIX CHAR
1266 03732 2311          ISZ      GETCHR         /INCR RETURN
1267 03733 5711          JMP I    GETCHR         /RETURN
1270 03734 2020 BLANK,   ISZ      NCHARS         /GET NEXT CHAR
1271 03735 7410          SKP
1272 03736 5314          JMP      GETC2          /BLANKS AT END OF LINE =CR
1273 03737 1417          TAD I    CHRPTR
1274 03740 1140          TAD      [-240          /IS IT BLANK?
1275 03741 7450          SNA
1276 03742 5334          JMP      BLANK          /YES, TRY AGAIN
1277 03743 1370          TAD      (-7            /IS IT TAB ?
1300 03744 7450          SNA
1301 03745 5334          JMP      BLANK          /YES, TRY AGAIN
```

```
1302 03746 1365          TAD     (-10            /IS IT SLASH ?
1303 03747 7450          SNA
1304 03750 5314          JMP     GETC2           /YES, (BLANK.OR.TAB) SLASH=CR
1305 03751 1364          TAD     (257-273        /IS IT SEMI COLON ?
1306 03752 7650          SNA CLA
1307 03753 5360          JMP     SEMICL          /YES
1310 03754 4200          JMS     BACK1           /NONE OF THESE, PUT IT BACK
1311 03755 1164          TAD     [240            /AND RETURN A SINGLE BLANK
1312 03756 2311          ISZ     GETCHR
1313 03757 5711          JMP I   GETCHR
1314 03760 7201 SEMICL,  CLA IAC                 /SET SEMI COLON SWITCH
1315 03761 3054          DCA     SCSWT
1316 03762 4200          JMS     BACK1           /PUT BACK SEMI COLON
1317 03763 5711          JMP I   GETCHR
```

```
1320                            EJECT
     03764 7764
     03765 7770
     03766 0273
     03767 7754
     03770 7771
     03771 0060
     03772 0301
     03773 0333
     03774 7746
     03775 7477
     03776 7700
     03777 0052
1321                            PAGE
1322 04000 0000 FIXOPC, 0                       /COMBINE OPCODE PARTS
1323 04001 1046         TAD     INDEX           /CHECK INDEX SWITCH
1324 04002 7650         SNA CLA
1325 04003 5216         JMP     ZRONDX          /IF ZERO, NO INDEX REG
1326 04004 7240         CLA CMA
1327 04005 1055         TAD     LASTOP          /IF INDEX, CHECK FOR INCR
1330 04006 7650         SNA CLA
1331 04007 1165         TAD     [100            /YES, PUT + BIT ON
1332 04010 1045         TAD     OPCODE          /COMBINE WITH OPCODE
1333 04011 3045         DCA     OPCODE
1334 04012 1036         TAD     EXPVAL+2        /GET INDEX REG, EXPR
1335 04013 0156         AND     [7              /ONLY 3 BITS
1336 04014 7106         CLL RTL                 /SHIFT INTO POSITION
1337 04015 7004         RAL
1340 04016 1045 ZRONDX, TAD     OPCODE          /ADD OPCODE
1341 04017 1377         TAD     (400            /TURN ON TYPE BIT
1342 04020 3045         DCA     OPCODE          /SAVE OPCODE
1343 04021 5600         JMP I   FIXOPC          /RETURN
1344 04022 0022 BUCKTS, ALIST                   /BUCKET LIST
1345 04023 0047         BLIST
1346 04024 0055         CLIST
1347 04025 0126         DLIST
1350 04026 0141         ELIST
1351 04027 0154         FLIST
1352 04030 0376         GLIST
1353 04031 0377         HLIST
1354 04032 0405         ILIST
1355 04033 0451         JLIST
1356 04034 0560         KLIST
1357 04035 0605         LLIST
1360 04036 0637         MLIST
1361 04037 0640         NLIST
1362 04040 0646         OLIST
1363 04041 0666         PLIST
1364 04042 0674         QLIST
1365 04043 0675         RLIST
1366 04044 0746         SLIST
1367 04045 1036         TLIST
1370 04046 1126         ULIST
```

```
1371 04047 1127          VLIST
1372 04050 1130          WLIST
1373 04051 1131          LISTX
1374 04052 1137          YLIST
1375 04053 1140          ZLIST
1376 04054 7540 OPR8RS,  -240
1377 04055 0006          6
1400 04056 7525          -253
1401 04057 0001          1
1402 04060 7523          -255
1403 04061 0002          2
1404 04062 7526          -252
1405 04063 0003          3
1406 04064 7521          -257
1407 04065 0004          4
1410 04066 7532          -246
1411 04067 0005          5
1412 04070 7537          -241
1413 04071 0006          6
1414 04072 0000          0
1415 04073 0000 CKKILL,  0                       /TEST FOR CHAR ON TTY
1416 04074 6031          KSF                     /CHAR TYPED ?
1417 04075 5673          JMP I   CKKILL          /NO, RETURN
1420 04076 6036          KRB                     /READ CHAR
1421 04077 1376          TAD     (-214           /CONTROL L?
1422 04100 7440          SZA
1423 04101 5305          JMP     .+4             /NO
1424 04102 3061          DCA     LISTSW          /YES, STOP LISTING
1425 04103 3062          DCA     LSTON
1426 04104 5673          JMP I   CKKILL          /RETURN
1427 04105 1375          TAD     (214-204        /CONTROL D?
1430 04106 7640          SZA CLA
1431 04107 5673          JMP I   CKKILL          /NO, RETURN
1432 04110 6201          CDF     FLD0
1433 04111 1103 ENDXX,   TAD     BBLOCK          /GET REAL BLOCK NUMBER
1434 04112 1104          TAD     BFUDGE          /OF LAST BLOCK
1435 04113 3317          DCA     ENDBLK          /AND WRITE IT OUT
1436 04114 4502          JMS I   DIALWR
1437 04115 0001          1
1440 04116 0015          BINARB
1441 04117 0000 ENDBLK,  0
1442 04120 0001          1
1443 04121 1374          TAD     (57             /WRITE OUT CORE USE TABLE
1444 04122 1104          TAD     BFUDGE
1445 04123 3327          DCA     USEBLK
1446 04124 4502          JMS I   DIALWR
1447 04125 0001          1
1450 04126 0014          USEB
1451 04127 0000 USEBLK,  0
1452 04130 0001          1
1453 04131 7200 RETSYS,  CLA
1454 04132 4773          JMS I   (PAGEJ          /EJECT PAGE
1455 04133 6041          TSF                     /WAIT FOR TTY
1456 04134 5333          JMP     .-1
```

```
1457 04135 1106          TAD     SFUDGE          /COMPUTE FUDGED BLOCK FOR SYSTEM
1460 04136 1372          TAD     (-46            /ONLY J.B. KNOWS THE REASON FOR
1461 04137 3343          DCA     SYSBLK
1462 04140 4501          JMS I   DIALRD          /READ IN SYSTEM
1463 04141 0000          0
1464 04142 0036          36                      /INTO 17000
1465 04143 0000 SYSBLK,  0
1466 04144 0002          2
1467 04145 6213          6213
1470 04146 5771          JMP I   (7777           /RETURN TO SYSTEM
1471 04147 0000 P1,      0                       /CONVERT TO ASCII AND PRINT
1472 04150 0167          AND     [77
1473 04151 7450          SNA
1474 04152 5356          JMP     .+4             /ZERO CHAR PRINTS AS BLANK
1475 04153 1370          TAD     (-40
1476 04154 7510          SPA
1477 04155 1165          TAD     [100
1500 04156 1164          TAD     [240
1501 04157 4576          JMS I   [PRINTC
1502 04160 5747          JMP I   P1
```

```
1503                            EJECT
     04170 7740
     04171 7777
     04172 7732
     04173 5154
     04174 0057
     04175 0010
     04176 7564
     04177 0400
1504                            PAGE
1505 04200 0000 PRINTC,  0                      /PRINT CHAR ON ANALEX, LP08 OR T
1506 04201 7450          SNA
1507 04202 5223          JMP     CRLF           /ZERO IS CR-LF
1510 04203 1377          TAD     (-247
1511 04204 7440          SZA
1512 04205 5215          JMP     NOTAB
1513 04206 1164          TAD     [240           /DO TAB
1514 04207 4245          JMS     PCHR
1515 04210 2241          ISZ     TABCNT
1516 04211 5206          JMP     .-3            /KEEP GOING UNTIL TAB STOP
1517 04212 1376 SETTAB,  TAD     (-10
1520 04213 3241          DCA     TABCNT         /SET TAB COUNTER
1521 04214 5600          JMP I   PRINTC
1522 04215 1174 NOTAB,   TAD     [247           /FIX CHAR
1523 04216 4245          JMS     PCHR
1524 04217 4560          JMS I   [CKKILL        /CHECK FOR ABORT
1525 04220 2241          ISZ     TABCNT
1526 04221 5600          JMP I   PRINTC         /RETURN
1527 04222 5212          JMP     SETTAB         /RESET TAB COUNT
1530 04223 3242 CRLF,    DCA     LSIZE          /MAKE SURE THERES ROOM
1531 04224 1375          TAD     (215           /OUTPUT CR-LF
1532 04225 4245          JMS     PCHR
1533 04226 1374          TAD     (212
1534 04227 4245          JMS     PCHR
1535 04230 2007          ISZ     PAGSIZ         /NEW PAGE ?
1536 04231 5236          JMP     NEJECT         /NO
1537 04232 1025          TAD     SIZPAG         /RESET COUNT
1540 04233 3007          DCA     PAGSIZ
1541 04234 1373          TAD     (214
1542 04235 4245          JMS     PCHR           /START NEW PAGE
1543 04236 1243 NEJECT,  TAD     WIDTH          /SET WIDTH OF PRINTER
1544 04237 3242          DCA     LSIZE
1545 04240 5212          JMP     SETTAB         /RESET TAB SETTING
1546 04241 7770 TABCNT,  -10
1547 04242 0000 LSIZE,   0
1550 04243 7667 WIDTH,   -111                   /WIDTH OF PRINTER IN CHARS
1551 04244 4311 PC,      TTY
1552 04245 0000 PCHR,    0                      /TAKE CARE OF PRINTER WIDTH
1553 04246 3257          DCA     ACHR           /SAVE CHAR
1554 04247 2242          ISZ     LSIZE          /ANY ROOM LEFT
1555 04250 5254          JMP     NOCRLF         /NO, GO
1556 04251 7240          CLA CMA                /TRUNCATE LINE
1557 04252 3242          DCA     LSIZE
```

```
1560 04253 5645          JMP I   PCHR
1561 04254 1257 NOCRLF,  TAD     ACHR
1562 04255 4644          JMS I   PC
1563 04256 5645          JMP I   PCHR
1564 04257 0000 ACHR,    0
1565 04260 0000 ANALEX,  0                       /ANALEX PRINTER HANDLER
1566 04261 3257          DCA     ACHR
1567 04262 1257          TAD     ACHR            /LINE FEED?
1570 04263 1372          TAD     (-212
1571 04264 7650          SNA CLA
1572 04265 5660          JMP I   ANALEX          /YES,IGNORE IT
1573 04266 6661          6661                    /WAIT FOR PRINTER
1574 04267 5266          JMP     .-1
1575 04270 6652          6652                    /CLEAR FLAG
1576 04271 1257          TAD     ACHR            /GET CHAR BACK
1577 04272 1371          TAD     (-215           /END LINE ON CR
1600 04273 7450          SNA
1601 04274 5305          JMP     FINLPB
1602 04275 7001          IAC                     /CHECK FOR FORM
1603 04276 7650          SNA CLA
1604 04277 5304          JMP     FINLPB-1        /PAGE EJECT ON FORM
1605 04300 1257          TAD     ACHR            /PRINT CHAR
1606 04301 6654          6654
1607 04302 7200          CLA
1610 04303 5660          JMP I   ANALEX
1611 04304 1156          TAD     [7              /SET CONTROL
1612 04305 1370 FINLPB,  TAD     (10
1613 04306 6664          6664
1614 04307 7200          CLA
1615 04310 5660          JMP I   ANALEX
1616 04311 0000 TTY,     0
1617 04312 6041          TSF
1620 04313 5312          JMP     .-1
1621 04314 6046          TLS
1622 04315 7200          CLA
1623 04316 5711          JMP I   TTY
1624 04317 0000 LP08,    0
1625 04320 6661          6661
1626 04321 5320          JMP     .-1
1627 04322 6666          6666
1630 04323 7200          CLA
1631 04324 5717          JMP I   LP08
1632 04325 0000 ERMSG1,  0                       /PASS 1 MESSAGES
1633 04326 7200          CLA
1634 04327 6201          CDF     FLD0
1635 04330 1325          TAD     ERMSG1
1636 04331 3336          DCA     ERMSG           /COMMONIZE CALL
1637 04332 4200          JMS     PRINTC          /CR-LF
1640 04333 1060          TAD     LINENO          /PRINT LINE NUMBER
1641 04334 4575          JMS I   [OCTOUT         /ON PASS 1 MESSAGE
1642 04335 5352          JMP     PRMSG           /SKIP OVER PASS TEST
1643 04336 0000 ERMSG,   0
1644 04337 6201          CDF     FLD0            /FIX FIELD
1645 04340 7240          CLA CMA                 /NO MESSAGE ON PASS 1
```

```
1646 04341 1056          TAD     PASSNO
1647 04342 7650          SNA CLA
1650 04343 5366          JMP     MSGDUN
1651 04344 4200          JMS     PRINTC          /PRINT CR-LF
1652 04345 1061          TAD     LISTSW          /IS LIST ON ?
1653 04346 7640          SZA CLA
1654 04347 5352          JMP     .+3             /YES
1655 04350 1060          TAD     LINENO          /NO, PRINT LINE NUMBER
1656 04351 4575          JMS I   [OCTOUT
1657 04352 1367 PRMSG,   TAD     (252            /PRINT *
1660 04353 4200          JMS     PRINTC
1661 04354 1736          TAD I   ERMSG           /FIRST CHAR OF CODE
1662 04355 4766          JMS I   (PRINT2         /PRINT THE MESSAGE
1663 04356 1367          TAD     (252            /PRINT *
1664 04357 4200          JMS     PRINTC
1665 04360 2336 MSGDUN,  ISZ     ERMSG
1666 04361 2765          ISZ I   (ERRORS         /BUMP ERROR COUNT
1667 04362 5736          JMP I   ERMSG
1670 04363 5736          JMP I   ERMSG
```

```
1671                            EJECT
     04365 4762
     04366 5714
     04367 0252
     04370 0010
     04371 7563
     04372 7566
     04373 0214
     04374 0212
     04375 0215
     04376 7770
     04377 7531
1672                            PAGE
1673 04400 0000 OUTWRD, 0                       /OUTPUT ROUTINE
1674 04401 3356         DCA     WRD             /SAVE WORD
1675 04402 7240         CLA CMA
1676 04403 1056         TAD     PASSNO          /CHECK PASS
1677 04404 7650         SNA CLA
1700 04405 5351         JMP     ENDOUT
1701 04406 1027         TAD     LOCTR2          /GET LOW 12 BITS OF LOCATION
1702 04407 7006         RTL
1703 04410 7006         RTL
1704 04411 7006         RTL
1705 04412 0126         AND     [37             /GET PAGE NUMBER (WITHIN FIELD)
1706 04413 3005         DCA     OTEMP           /SAVE PAGE NUMBER
1707 04414 1005         TAD     OTEMP
1710 04415 7640         SZA CLA                 /POINTER TO P0LIT OR CPLIT
1711 04416 7001         IAC
1712 04417 1130         TAD     [P0LIT
1713 04420 3357         DCA     OWTEMP
1714 04421 1027         TAD     LOCTR2          /GET CURRENT ADDRESS DISPLACEMEN
1715 04422 0141         AND     [177
1716 04423 7041         CIA                     /COMPARE WITH LITERAL BOUNDARY
1717 04424 1757         TAD I   OWTEMP
1720 04425 7700         SMA CLA
1721 04426 5232         JMP     .+4             /NO PAGE OVER FLOW
1722 04427 4572         JMS I   [ERMSG
1723 04430 2017         2017                    /*P0*
1724 04431 5322         JMP     PRNTST          /DONT STORE IF PAGE OVERFLOW
1725 04432 1005         TAD     OTEMP           /NOW GET SUPER-PAGE NUMBER
1726 04433 7110         CLL RAR
1727 04434 3005         DCA     OTEMP           /AND SAVE IT
1730 04435 1026         TAD     LOCTR1          /GET FIELD BITS
1731 04436 0272         AND     01              /ONLY ONE BIT FOR DIAL CRAP
1732 04437 7106         CLL RTL
1733 04440 7006         RTL
1734 04441 1005         TAD     OTEMP           /THIS GIVES TAPE BLOCK NUMBER
1735 04442 3005         DCA     OTEMP
1736 04443 1103         TAD     BBLOCK          /GET CURRENT BLOCK
1737 04444 7041         CIA
1740 04445 1005         TAD     OTEMP           /COMPARE WITH DESIRED BLOCK
1741 04446 7650         SNA CLA
1742 04447 5314         JMP     SAMBLK          /SAME BLOCK
```

```
1743 04450 1103          TAD     BBLOCK          /FIND REAL BLOCK NUMBER
1744 04451 1104          TAD     BFUDGE
1745 04452 3256          DCA     WRBLOK
1746 04453 4502          JMS I   DIALWR          /WRITE OUT OLD BLOCK
1747 04454 0001          1
1750 04455 0015          BINARB
1751 04456 0000 WRBLOK,  0
1752 04457 0001          1
1753 04460 1005          TAD     OTEMP           /CHECK FOR THIS BLOCK ALREADY US
1754 04461 1377          TAD     (USETBL
1755 04462 3006          DCA     OCNT            /POINTER INTO USE TABLE
1756 04463 1406          TAD I   OCNT            /GET, INDICATOR
1757 04464 7650          SNA CLA
1760 04465 5277          JMP     NEWBLK          /FIRST TIME USED
1761 04466 1005          TAD     OTEMP           /BLOCK WAS USED, FIND REAL BLOCK
1762 04467 1104          TAD     BFUDGE          /NUMBER AND READ BLOCK
1763 04470 3274          DCA     BLOCKN
1764 04471 4501          JMS I   DIALRD
1765 04472 0001 01,      1
1766 04473 0015          BINARB
1767 04474 0000 BLOCKN,  0
1770 04475 0001          1
1771 04476 5310          JMP     DONT0
1772 04477 2776 NEWBLK,  ISZ I   (USETBL-1       /INCREMENT COUNT OF BLKS IN USE
1773 04500 1375          TAD     (BINARY
1774 04501 3360          DCA     CLRBIN          /CLEAR OUT BUFFER
1775 04502 1374          TAD     (-400
1776 04503 3357          DCA     OWTEMP
1777 04504 3760          DCA I   CLRBIN
2000 04505 2360          ISZ     CLRBIN
2001 04506 2357          ISZ     OWTEMP
2002 04507 5304          JMP     .-3
2003 04510 1005 DONT0,   TAD     OTEMP           /RESET CURRENT BLOCK
2004 04511 3103          DCA     BBLOCK
2005 04512 7240          CLA CMA                 /SET BLOCK IN USE
2006 04513 3406          DCA I   OCNT
2007 04514 1027 SAMBLK,  TAD     LOCTR2          /GET POINTER FOR STORING WORD
2010 04515 0373          AND     (377
2011 04516 1375          TAD     (BINARY
2012 04517 3005          DCA     OTEMP
2013 04520 1356          TAD     WRD             /GET WORD
2014 04521 3405          DCA I   OTEMP           /STORE IT
2015 04522 1061 PRNTST,  TAD     LISTSW          /IS LIST ON ?
2016 04523 7650          SNA CLA
2017 04524 5351          JMP     ENDOUT          /NO, DONT PRINT
2020 04525 4576          JMS I   [PRINTC         /PUT CR-LF
2021 04526 1063          TAD     OUTSWT          /WAS LINE NUM PRINTED YET?
2022 04527 7640          SZA CLA
2023 04530 5334          JMP     .+4
2024 04531 1060          TAD     LINENO          /NO, PRINT IT
2025 04532 4575          JMS I   [OCTOUT
2026 04533 5340          JMP     NOBLNK          /SKIP OVER BLANKS
2027 04534 4772          JMS I   (PRINT2         /2 BLANKS
2030 04535 4772          JMS I   (PRINT2         /2 MORE
```

```
2031 04536 1164          TAD     [240
2032 04537 4576          JMS I   [PRINTC          /1 MORE 4 5
2033 04540 1026 NOBLNK,  TAD     LOCTR1          /PRINT LOCATION COUNTER
2034 04541 0156          AND     [7
2035 04542 1155          TAD     [260            /FIRST DIGIT
2036 04543 4576          JMS I   [PRINTC
2037 04544 1027          TAD     LOCTR2          /NEXT FOUR DIGITS
2040 04545 4575          JMS I   [OCTOUT
2041 04546 1356          TAD     WRD             /NOW WORD
2042 04547 4575          JMS I   [OCTOUT
2043 04550 4771          JMS I   (PRNTLN         /PRINT LINE IF NECESSARY
2044 04551 2027 ENDOUT,  ISZ     LOCTR2          /BUMP LOCATION COUNTER
2045 04552 5600          JMP I   OUTWRD
2046 04553 2026          ISZ     LOCTR1          /BUMP SECOND WORD
2047 04554 5600          JMP I   OUTWRD
2050 04555 5600          JMP I   OUTWRD
2051 04556 0000 WRD,     0
2052 04557 0000 OWTEMP,  0
2053 04560 0000 CLRBIN,  0
0000                     CHAIN   "FPPASM3"
```

```
0001                            EJECT
     04571 2000
     04572 5714
     04573 0377
     04574 7400
     04575 6400
     04576 0337
     04577 6340
0002                            PAGE
0003 04600 3065 ENDX,   DCA     RADIX           /SET DEFAULT CONDITIONS
0004 04601 7240         CLA CMA                 /END OF WHICH PASS ?
0005 04602 1056         TAD     PASSNO
0006 04603 7640         SZA CLA
0007 04604 5243         JMP     EOP2            /END OF PASS TWO
0010 04605 4777         JMS I   (PAGEJ          /EJECT AFTER PASS ONE
0011 04606 3107         DCA     STAR2           /RESET STUPIDITY SWITCH
0012 04607 3060         DCA     LINENO          /ZERO LINE NUMBER
0013 04610 3057         DCA     ASMOF           /ZERO CONDITIONAL SWITCH
0014 04611 3064         DCA     SCSWT           /ZERO SEMICOLON SWITCH
0015 04612 1062         TAD     LSTON           /SET LIST SWITCH
0016 04613 3061         DCA     LISTSW
0017 04614 1030         TAD     LITRG1          /IF LITORG HAS BEEN SET
0020 04615 7700         SMA CLA
0021 04616 5223         JMP     .+5             /DONT CHANGE IT
0022 04617 1026         TAD     LOCTR1          /SET LITORG INCASE NO COMMAND
0023 04620 3030         DCA     LITRG1
0024 04621 1027         TAD     LOCTR2
0025 04622 3031         DCA     LITRG2
0026 04623 1145         TAD     [200
0027 04624 3027         DCA     LOCTR2          /LOCATION COUNTER
0030 04625 3026         DCA     LOCTR1
0031 04626 7130         CLL CML RAR             /4000
0032 04627 3032         DCA     BASER           /SET BASE BEYOND BELIEF
0033 04630 1141         TAD     [177            /INITIALIZE LITERAL BOUNDARIES
0034 04631 3110         DCA     P0LIT
0035 04632 1141         TAD     [177
0036 04633 3111         DCA     CPLIT
0037 04634 7040         CMA                     /OPEN INPUT FILE
0040 04635 3024         DCA     CHRCNT
0041 04636 3105         DCA     SBLOCK
0042 04637 3776         DCA I   (SRCUNT         /START AT UNIT 0
0043 04640 3362         DCA     ERRORS          /ZERO ERROR COUNT
0044 04641 2056         ISZ     PASSNO          /BUMP PASS NUMBER
0045 04642 5525         JMP I   [START          /DO NEXT PASS
0046 04643 7201 EOP2,   CLA IAC                 /DUMP CURRENT PAGE LITERALS
0047 04644 4775         JMS I   (DMPLIT
0050 04645 4775         JMS I   (DMPLIT         /THEN DUMP PAGE 0 LITERALS
0051 04646 2063         ISZ     OUTSWT          /DONT PRINT LINE WITH FP LITERAL
0052 04647 4774         JMS I   (DLITS          /THEN FLOATING POINT LITERALS
0053 04650 4777         JMS I   (PAGEJ          /EJECT
0054 04651 1362         TAD     ERRORS          /ERROR COUNT
0055 04652 4575         JMS I   [OCTOUT
0056 04653 1373         TAD     (0522           /PRINT ER
```

```
0057 04654 4772          JMS I   (PRINT2
0060 04655 1371          TAD     (2217          /PRINT RO
0061 04656 4772          JMS I   (PRINT2
0062 04657 1370          TAD     (2223          /PRINT RS
0063 04660 4772          JMS I   (PRINT2
0064 04661 4777          JMS I   (PAGEJ         /EJECT
0065 04662 1367          TAD     (BUCKTS-1      /SET UP FOR SYMBOL TABLE
0066 04663 3015          DCA     X15
0067 04664 1366          TAD     (-32           /26 BUCKETS
0070 04665 3066          DCA     LTEMP
0071 04666 1365          TAD     (301           /BUCKET CHARACTER
0072 04667 3051          DCA     BUCKET
0073 04670 1415 STPRNT,  TAD I   X15            /GET NEXT BUCKET
0074 04671 6211          CDF     FLD1
0075 04672 3067          DCA     EXTMP          /BUCKET START ADDRESS
0076 04673 7200 LUPBKT,  CLA
0077 04674 1467          TAD I   EXTMP          /WAS TTAT LAST SYMBOL ?
0100 04675 7450          SNA
0101 04676 5353          JMP     NXTBKT         /YES, GO GET NEXT BUCKET
0102 04677 3067          DCA     EXTMP          /SAVE LINK ADDR
0103 04700 1067          TAD     EXTMP
0104 04701 3014          DCA     X14            /SET UP POINTER FOR NAME
0105 04702 1414          TAD I   X14            /PICK UP THE NAME
0106 04703 3052          DCA     NAME1
0107 04704 1414          TAD I   X14
0110 04705 3053          DCA     NAME2
0111 04706 1414          TAD I   X14
0112 04707 3054          DCA     NAME3
0113 04710 1054          TAD     NAME3          /LOOK AT THE TYPE
0114 04711 0126          AND     [37            /LOSE FORCE BIT
0115 04712 1157          TAD     [-4            /IS AN OPCODE ?
0116 04713 7500          SMA
0117 04714 5273          JMP     LUPBKT         /YES, GET NEXT SYMBOL
0120 04715 1332          TAD     SETTYP         /GET JUMP THRU TABLE
0121 04716 3331          DCA     SETTYP-1
0122 04717 4576          JMS I   [PRINTC        /PRINT CR-LF
0123 04720 1051          TAD     BUCKET
0124 04721 4576          JMS I   [PRINTC        /PRINT FIRST CHAR
0125 04722 1052          TAD     NAME1
0126 04723 4772          JMS I   (PRINT2        /PRINT 2 AND 3
0127 04724 1053          TAD     NAME2
0130 04725 4772          JMS I   (PRINT2        /PRINT 4 AND 5
0131 04726 1054          TAD     NAME3
0132 04727 0364          AND     (7700          /PRINT 6 AND BLANK
0133 04730 4772          JMS I   (PRINT2
0134 04731 0000          0
0135 04732 5737 SETTYP,  JMP I   .+5
0136 04733 4673          LUPBKT                 /UNDEFINED, PRINT NOTHING
0137 04734 4746          SYM1
0140 04735 4741          SYM2
0141 04736 4737          SYM3
0142 04737 1414 SYM3,    TAD I   X14            /PRINT 3 WORD THING
0143 04740 4575          JMS I   [OCTOUT
0144 04741 1414 SYM2,    TAD I   X14            /PRINT 2 WORD THING
```

```
0145 04742 4575          JMS I   [OCTOUT
0146 04743 1414          TAD I   X14
0147 04744 4575          JMS I   [OCTOUT
0150 04745 5273          JMP     LUPBKT          /GET NEXT SYMBOL
0151 04746 1414 SYM1,    TAD I   X14             /PRINT FIRST DIGIT
0152 04747 0156          AND     [7
0153 04750 1155          TAD     [260
0154 04751 4576          JMS I   [PRINTC
0155 04752 5343          JMP     SYM2+2          /PRINT NEXT FOUR DIGITS
0156 04753 2051 NXTBKT,  ISZ     BUCKET          /NEXT BUCKET CHAR
0157 04754 6201          CDF     FLD0
0160 04755 2066          ISZ     LTEMP           /INCREMENT COUNT
0161 04756 5270          JMP     STPRNT
0162 04757 4777          JMS I   (PAGEJ          /EJECT
0163 04760 4576          JMS I   [PRINTC         /FINAL CR/LF
0164 04761 5763          JMP I   (ENDXX          /FINISH IT OFF
0165 04762 0000 ERRORS,  0
```

```
0166                            EJECT
     04763 4111
     04764 7700
     04765 0301
     04766 7746
     04767 4021
     04770 2223
     04771 2217
     04772 5714
     04773 0522
     04774 3314
     04775 5075
     04776 0666
     04777 5154
0167                            PAGE
0170 05000 4554 BASEX,  JMS I   [ADRGET         /GET ADDRESS EXPR VALUE
0171 05001 1037         TAD     EXPTYP          /WAS THING DEFINED
0172 05002 7650         SNA CLA
0173 05003 5220         JMP     ORGERR          /NO, GIVE *UO* ERROR
0174 05004 1035         TAD     EXPVAL+1        /PUT INTO BASER
0175 05005 0156         AND     [7
0176 05006 3032         DCA     BASER
0177 05007 1036         TAD     EXPVAL+2
0200 05010 3033         DCA     BASER+1
0201 05011 5563         JMP I   [NEXTST
0202 05012 4551 ORGX,   JMS I   [EXPR           /GET ORG EXPR
0203 05013 5220         JMP     ORGERR          /BAD IF NO EXPR
0204 05014 7240         CLA CMA                 /CHECK TYPE
0205 05015 1037         TAD     EXPTYP
0206 05016 7650         SNA CLA
0207 05017 5223         JMP     FIXORG          /TYPE OK
0210 05020 4572 ORGERR, JMS I   [ERMSG
0211 05021 2517         2517                    /*UO*
0212 05022 5563         JMP I   [NEXTST
0213 05023 1026 FIXORG, TAD     LOCTR1          /CHECK FOR NEW FIELD
0214 05024 7041         CIA
0215 05025 1035         TAD     EXPVAL+1
0216 05026 7650         SNA CLA
0217 05027 5236         JMP     SAMFLD          /NOT A DIFFERENT FIELD
0220 05030 7201         CLA IAC
0221 05031 4275         JMS     DMPLIT          /DUMP CURRENT PAGE LITERALS
0222 05032 4275         JMS     DMPLIT          /DUMP PAGE 0 LITERALS
0223 05033 1141         TAD     [177            /RESET PAGE 0 LIT BOUNDARY
0224 05034 3110         DCA     P0LIT
0225 05035 5251         JMP     SAMPAG-2        /DO THE SAME FOR CURRENT PAGE
0226 05036 1027 SAMFLD, TAD     LOCTR2
0227 05037 0142         AND     [7600           /CHECK FOR SAME PAGE
0230 05040 3066         DCA     LTEMP
0231 05041 1036         TAD     EXPVAL+2
0232 05042 0142         AND     [7600
0233 05043 7041         CIA
0234 05044 1066         TAD     LTEMP
0235 05045 7650         SNA CLA
```

```
0236 05046 5253          JMP     SAMPAG          /PAGE IS THE SAME
0237 05047 7201          CLA IAC
0240 05050 4275          JMS     DMPLIT          /DUMP CURRENT PAGE LITERALS
0241 05051 1141          TAD     [177            /RESET BOUNDARY
0242 05052 3111          DCA     CPLIT
0243 05053 1035 SAMPAG,  TAD     EXPVAL+1        /PUT ORG VALUE
0244 05054 0156          AND     [7
0245 05055 3026          DCA     LOCTR1          /INTO LOCATION COUTER
0246 05056 1036          TAD     EXPVAL+2
0247 05057 3027          DCA     LOCTR2
0250 05060 5563          JMP I   [NEXTST
0251 05061 1062 LSTONX,  TAD     LSTON
0252 05062 3061 LSTOFX,  DCA     LISTSW          /SET LIST STATUS
0253 05063 5563          JMP I   [NEXTST
0254 05064 1027 PAGEX,   TAD     LOCTR2          /ADVANCE TO NEXT PAGE
0255 05065 7100          CLL
0256 05066 1141          TAD     [177
0257 05067 0142          AND     [7600
0260 05070 3036          DCA     EXPVAL+2
0261 05071 7004          RAL
0262 05072 1026          TAD     LOCTR1
0263 05073 3035          DCA     EXPVAL+1
0264 05074 5223          JMP     FIXORG
0265                     SWTOUT=OPE
0266 05075 0000 DMPLIT,  0
0267 05076 3112          DCA     PAGEN           /SAVE PAGE INDICATOR
0270 05077 1063          TAD     OUTSWT          /SAVE OUTPUT SWITCH
0271 05100 3113          DCA     SWTOUT
0272 05101 2063          ISZ     OUTSWT          /DONT PRINT LINE WITH LITERALS
0273 05102 1112          TAD     PAGEN
0274 05103 1130          TAD     [P0LIT          /GET BOUNDARY POINTER
0275 05104 3066          DCA     LTEMP
0276 05105 1112          TAD     PAGEN           /WHICH LITERAL BUFFER ?
0277 05106 7650          SNA CLA
0300 05107 1377          TAD     (P0LBUF-CPLBUF  /PAGE 0 BUFFER
0301 05110 1376          TAD     (CPLBUF         /CURRENT PAGE BUFFER
0302 05111 1466          TAD I   LTEMP           /PLUS  PAGE ADDRESS
0303 05112 3010          DCA     X10             /GIVES START OF LITERALS -1
0304 05113 1112          TAD     PAGEN
0305 05114 7640          SZA CLA
0306 05115 1027          TAD     LOCTR2          /UPPER FIVE BITS OF ADDRESS
0307 05116 0142          AND     [7600
0310 05117 1466          TAD I   LTEMP           /PLUS LOWER SEVEN
0311 05120 7001          IAC                     /PLUS ONE
0312 05121 3027          DCA     LOCTR2          /GIVES LOCATION COUNTER
0313 05122 1466          TAD I   LTEMP           /SAVE OLD LITERAL BOUNDARY
0314 05123 3112          DCA     PAGEN
0315 05124 1141          TAD     [177            /STORE SPURIOUS LITERAL BOUNDARY
0316 05125 3466          DCA I   LTEMP           /TO PREVENT FALLACIOUS *P0* MESS
0317 05126 1010 LITLUP,  TAD     X10             /END OF LITERALS
0320 05127 0141          AND     [177            /GET DISPLACEMENT -1
0321 05130 1375          TAD     (-177           /IS IT .GE. 177 ?
0322 05131 7700          SMA CLA
0323 05132 5336          JMP     DMPFIN          /GO RESTORE LITERAL BOUNDARY
```

```
0324 05133 1410          TAD I   X10             /NO, GET NEXT LITERAL
0325 05134 4550          JMS I   [OUTWRD         /OUTPUT WORD AND BUMP LC
0326 05135 5326          JMP     LITLUP          /LOOP
0327 05136 1112 DMPFIN,  TAD     PAGEN           /GET CORRECT LITERAL BOUNDARY
0330 05137 3466          DCA I   LTEMP           /PUT IT IN ITS PLACE
0331 05140 1113          TAD     SWTOUT          /RESTORE OUTPUT SWITCH
0332 05141 3063          DCA     OUTSWT
0333 05142 5675          JMP I   DMPLIT          /ALL DONE
0334                     PAGEN=ACE
0335 05143 7240 EJECTX,  CLA CMA                 /NO EJECT ON PASS 1
0336 05144 1056          TAD     PASSNO
0337 05145 7650          SNA CLA
0340 05146 5563          JMP I   [NEXTST
0341 05147 1061          TAD     LISTSW          /OR LIST OFF
0342 05150 7650          SNA CLA
0343 05151 5563          JMP I   [NEXTST
0344 05152 4354          JMS     PAGEJ
0345 05153 5563          JMP I   [NEXTST
0346 05154 0000 PAGEJ,   0                       /PAGE EJECT SUBROUTINE
0347 05155 4576          JMS I   [PRINTC         /CR=LF
0350 05156 1374          TAD     (214
0351 05157 4576          JMS I   [PRINTC         /FORM FEED
0352 05160 1025          TAD     SIZPAG          /FIX PAGE COUNTER
0353 05161 3007          DCA     PAGSIZ
0354 05162 5754          JMP I   PAGEJ
```

```
0355                        EJECT
     05174 0214
     05175 7001
     05176 6000
     05177 1400
0356                        PAGE
0357 05200 4573 CHAINX, JMS I   [GETCHR          /LOOK FOR FIRST "
0360 05201 5347         JMP     CHERR           /MISSING FILE NAME
0361 05202 1377         TAD     (-242
0362 05203 7640         SZA CLA
0363 05204 5200         JMP     CHAINX          /KEEP LOOKING
0364 05205 1157         TAD     [-4             /NAME WORD COUNT
0365 05206 3066         DCA     LTEMP
0366 05207 1376         TAD     (FNAME          /NAME POINTER
0367 05210 3067         DCA     EXTMP
0370 05211 3353         DCA     CBSWIT          /SET CAT BLOCK SWITCH
0371 05212 4775 CNLOOP, JMS I   (GETCN          /GET CHAR
0372 05213 7106         CLL RTL
0373 05214 7006         RTL
0374 05215 7006         RTL
0375 05216 3467         DCA I   EXTMP           /SAVE UPPER HALF
0376 05217 4775         JMS I   (GETCN
0377 05220 1467         TAD I   EXTMP           /UNITE HALVES
0400 05221 3467         DCA I   EXTMP
0401 05222 2067         ISZ     EXTMP           /BUMP POINTER
0402 05223 2066         ISZ     LTEMP           /AND COUNT
0403 05224 5212         JMP     CNLOOP
0404 05225 4573         JMS I   [GETCHR         /LOOK FOR CLOSE QUOTE
0405 05226 5347         JMP     CHERR
0406 05227 1377         TAD     (-242
0407 05230 7640         SZA CLA
0410 05231 5347         JMP     CHERR
0411 05232 4774         JMS I   (GETUNT         /GO GET UNIT NUMBER
0412 05233 7144         CLL CMA RAL             /2 BLOCKS OF CAT
0413 05234 3067         DCA     EXTMP
0414 05235 1373         TAD     (-40            /32 ENTRIES PER BLOCK
0415 05236 3352         DCA     NFILES
0416 05237 1372         TAD     (-22            /COMPUTE CAT BLOCK NUMBER
0417 05240 1106         TAD     SFUDGE
0420 05241 3245         DCA     DIRBLK
0421 05242 4501 DIRLUP, JMS I   DIALRD          /READ CATALOG
0422 05243 0000 UNIT,   0
0423 05244 0016         SOURCB                  /INTO SOURCE BUFFER
0424 05245 0000 DIRBLK, 0
0425 05246 0001         1
0426 05247 2245         ISZ     DIRBLK          /INCR BLOCK
0427 05250 1371         TAD     (SOURCE-1       /GET CAT POINTER
0430 05251 3010         DCA     X10
0431 05252 1353         TAD     CBSWIT          /IS THIS SECOND HALF OF CAT ?
0432 05253 2353         ISZ     CBSWIT          /SET SWITCH
0433 05254 7640         SZA CLA
0434 05255 5263         JMP     FILLUP          /YES
0435 05256 1570         TAD I   [SOURCE         /CHECK FIRST NAME FOR "////////"
```

```
0436 05257 1370          TAD     (-5757
0437 05260 7640          SZA CLA
0440 05261 5347          JMP     CHERR           /FILE NOT FOUND
0441 05262 5340          JMP     NOTFIL          /GO BUMP NAME POINTER
0442 05263 1367 FILLUP,  TAD     (FNAME-1        /NAME POINTER
0443 05264 3011          DCA     X11
0444 05265 1010          TAD     X10             /SAVE CAT POINTER
0445 05266 3012          DCA     X12
0446 05267 1157          TAD     [-4             /NAME SIZE
0447 05270 3066          DCA     LTEMP
0450 05271 1412 FNLOOP,  TAD I   X12             /COMPARE NAMES
0451 05272 7041          CIA
0452 05273 1411          TAD I   X11
0453 05274 7640          SZA CLA
0454 05275 5340          JMP     NOTFIL          /NOT THIS ONE
0455 05276 2066          ISZ     LTEMP
0456 05277 5271          JMP     FNLOOP          /NOT DONE
0457 05300 1106          TAD     SFUDGE          /GET RELATIVE BLOCK NUM
0460 05301 7041          CIA
0461 05302 1412          TAD I   X12
0462 05303 3105          DCA     SBLOCK
0463 05304 1412          TAD I   X12             /CHECK SIZE
0464 05305 7710          SPA CLA
0465 05306 5347          JMP     CHERR           /IF NEG, NO SUCH FILE
0466 05307 1243          TAD     UNIT            /SAVE UNIT
0467 05310 3766          DCA I   (SRCUNT
0470 05311 7040          CMA
0471 05312 3024          DCA     CHRCNT          /SET COUNT TO -1
0472 05313 3060          DCA     LINENO          /ZERO LINE NUMBER
0473 05314 3064          DCA     SCSWT           /ZERO SEMICOLON SWITCH
0474 05315 3107          DCA     STAR20          /ZERO STUPIDITY SWITCH
0475 05316 7240          CLA CMA
0476 05317 3020          DCA     NCHARS          /REMOVE FALSE *EG* MESSAGE
0477 05320 7240          CLA CMA                 /DO WE PRINT MESSAGE
0500 05321 1056          TAD     PASSNO
0501 05322 7640          SZA CLA
0502 05323 1061          TAD     LISTSW
0503 05324 7640          SZA CLA
0504 05325 5563          JMP I   [NEXTST         /ONLY IF PASS1, OR PASS2 & NOLIS
0505 05326 4576          JMS I   [PRINTC         /CR-LF
0506 05327 1365          TAD     (-12            /MESSAGE SIZE
0507 05330 3066          DCA     LTEMP
0510 05331 1364          TAD     (CMSG-1         /AND POINTER
0511 05332 3010          DCA     X10
0512 05333 1410          TAD I   X10             /GET WORD OF MESSAGE
0513 05334 4763          JMS I   (PRNT2          /PRINT 2 CHARS
0514 05335 2066          ISZ     LTEMP
0515 05336 5333          JMP     .-3
0516 05337 5563          JMP I   [NEXTST         /PRINT CHAIN COMMAND
0517 05340 1010 NOTFIL,  TAD     X10             /BUMP CAT POINTER
0520 05341 1362          TAD     (10             /BY 8
0521 05342 3010          DCA     X10
0522 05343 2352          ISZ     NFILES          /MORE FILES IN BLOCK ?
0523 05344 5263          JMP     FILLUP          /YES
```

```
0524 05345 2067          ISZ     EXTMP          /ANOTHER BLOCK ?
0525 05346 5242          JMP     DIRLUP         /YES
0526 05347 4527 CHERR,   JMS I   [ERMSG1
0527 05350 0310          0310                   /*CH*
0530 05351 5761          JMP I   (RETSYS        /FATAL ERROR
0531 05352 0000 NFILES,  0
0532 05353 0000 CBSWIT,  0
```

```
0533                            EJECT
     05361 4131
     05362 0010
     05363 5726
     05364 5753
     05365 7766
     05366 0666
     05367 5761
     05370 2021
     05371 6777
     05372 7756
     05373 7740
     05374 2014
     05375 5400
     05376 5762
     05377 7536
0534                            PAGE
0535 05400 0000 GETCN,  0                       /RETURN NEXT CHAR OR 77
0536 05401 4573         JMS I   [GETCHR         /GET CHAR
0537 05402 5777         JMP I   (CHERR          /NONE, GIVE ERROR
0540 05403 1376         TAD     (-242           /IS IT "
0541 05404 7450         SNA
0542 05405 5211         JMP     ISQ             /YES
0543 05406 1375         TAD     (242            /FIX CHAR
0544 05407 0167         AND     [77
0545 05410 5600         JMP I   GETCN           /RETURN IT
0546 05411 4561 ISQ,    JMS I   [BACK1          /PUT BACK "
0547 05412 1167         TAD     [77             /RETURN 77
0550 05413 5600         JMP I   GETCN
```

```
0551                        EJECT
     05575 0242
     05576 7536
     05577 5347
0552                        PAGE
0553 05600 4573 TEXTX,  JMS I   [GETCHR         /GET DELIMITER
0554 05601 5563         JMP I   [NEXTST         /NULL STMT
0555 05602 7041         CIA
0556 05603 3067         DCA     EXTMP           /SAVE - DELIM
0557 05604 4222 LOOP68, JMS     GETCHT          /GET HIG ORDER CHAR
0560 05605 5563         JMP I   [NEXTST
0561 05606 7106         CLL RTL
0562 05607 7006         RTL
0563 05610 7006         RTL                     /SHIFT IT UP
0564 05611 3066         DCA     LTEMP           /SAVE HALF
0565 05612 4222         JMS     GETCHT          /GET LOWER CHAR
0566 05613 5217         JMP     OUTTXT          /GO PUT LAST
0567 05614 1066         TAD     LTEMP           /PUT 2 CHARS TOGETHER
0570 05615 4550         JMS I   [OUTWRD         /OUTPUT WORD
0571 05616 5204         JMP     LOOP68          /LOOP
0572 05617 1066 OUTTXT, TAD     LTEMP           /PUT OUT HALF WORD
0573 05620 4550         JMS I   [OUTWRD         /OR ZERO WORD
0574 05621 5563         JMP I   [NEXTST
0575 05622 0000 GETCHT, 0                       /GET CHAR FOR TEXT STMT
0576 05623 2020         ISZ     NCHARS          /BUMP COUNT
0577 05624 7410         SKP
0600 05625 5622         JMP I   GETCHT          /END OF TEXT
0601 05626 1417         TAD I   CHRPTR          /GET CHAR
0602 05627 3051         DCA     BUCKET          /SAVE IT
0603 05630 1051         TAD     BUCKET          /IS IT THE DELIM ?
0604 05631 1067         TAD     EXTMP
0605 05632 7650         SNA CLA
0606 05633 5622         JMP I   GETCHT          /YES, RETURN NO SKIP
0607 05634 2222         ISZ     GETCHT          /BUMP RETURN
0610 05635 1051         TAD     BUCKET          /GET CHAR
0611 05636 0167         AND     [77             /LOW 6 BITS
0612 05637 5622         JMP I   GETCHT          /RETURN
0613 05640 4554 IFNZRX, JMS I   [ADRGET         /GET EXPR FOR IFNZRO
0614 05641 1035         TAD     EXPVAL+1
0615 05642 7450         SNA
0616 05643 1036         TAD     EXPVAL+2
0617 05644 7650         SNA CLA
0620 05645 2057         ISZ     ASMOF           /IF BOTH HALVES 0, SET SWITCH
0621 05646 5563         JMP I   [NEXTST
0622 05647 4554 IFZROX, JMS I   [ADRGET         /GET EXPR FOR IFZERO
0623 05650 1035         TAD     EXPVAL+1
0624 05651 7450         SNA
0625 05652 1036         TAD     EXPVAL+2
0626 05653 7640         SZA CLA
0627 05654 2057         ISZ     ASMOF           /IF BOTH HALVES NON ZERO, SET SW
0630 05655 5563         JMP I   [NEXTST
0631 05656 4554 IFNEGX, JMS I   [ADRGET         /GET EXPR FOR IFNEG
0632 05657 1035         TAD     EXPVAL+1        /CHECK SIGN
```

```
0633 05660 7700          SMA CLA
0634 05661 2057          ISZ     ASMOF          /SET SWITCH IF POSITIVE
0635 05662 5563          JMP I   [NEXTST
0636 05663 4554 IFPOSX,  JMS I   [ADRGET        /GET EXPR FOR IFPOS
0637 05664 1035          TAD     EXPVAL+1       /CHECK SIGN
0640 05665 7710          SPA CLA
0641 05666 2057          ISZ     ASMOF          /SET SWITCH IF NEGATIVE
0642 05667 5563          JMP I   [NEXTST
0643 05670 4553 IFREFX,  JMS I   [GETNAM        /GET SYMBOL NAME
0644 05671 5563          JMP I   [NEXTST        /RETURN IF NONE
0645 05672 4552          JMS I   [LOOKUP        /S.T. LOOKUP
0646 05673 5300          JMP     NOTREF         /NOT REFERENCED YET
0647 05674 7200          CLA                    /REFERENCED, ASSEMBLY ON
0650 05675 3057          DCA     ASMOF
0651 05676 6201          CDF     FLD0           /FIX DATA FIELD
0652 05677 5563          JMP I   [NEXTST
0653 05700 1016 NOTREF,  TAD     NEXT           /SYMBOL WAS JUST ENTERRED
0654 05701 1157          TAD     [-4
0655 05702 3016          DCA     NEXT           /FIRST FIX NEXT
0656 05703 1016          TAD     NEXT
0657 05704 3010          DCA     X10            /SET UP POINTER
0660 05705 1410          TAD I   X10            /GET FWD POINTER
0661 05706 3404          DCA I   OLDN3          /INTO PREVIOUS ENTRY
0662 05707 6201          CDF     FLD0
0663 05710 7201          CLA IAC                /SET ASSEMBLY OFF
0664 05711 3057          DCA     ASMOF
0665 05712 5563          JMP I   [NEXTST
0666 05713 0000 P2,      0
0667            P3,
0670 05714 0000 PRINT2,  0                      /PRINT TWO PACKED CHARS
0671 05715 3313          DCA     P2
0672 05716 1313          TAD     P2
0673 05717 7012          RTR
0674 05720 7012          RTR
0675 05721 7012          RTR
0676 05722 4777          JMS I   (P1
0677 05723 1313          TAD     P2
0700 05724 4777          JMS I   (P1
0701 05725 5714          JMP I   PRINT2
0702 05726 0000 PRNT2,   0                      /SPECIAL PRINT FOR CHAIN
0703 05727 3313          DCA     P2             /ELIMINATES ?
0704 05730 1313          TAD     P2
0705 05731 7012          RTR
0706 05732 7012          RTR
0707 05733 7012          RTR
0710 05734 3314          DCA     P3
0711 05735 1314          TAD     P3
0712 05736 0376          AND     (77
0713 05737 1375          TAD     (-77
0714 05740 7650          SNA CLA
0715 05741 5726          JMP I   PRNT2
0716 05742 1314          TAD     P3
0717 05743 4777          JMS I   (P1
0720 05744 1313          TAD     P2
```

```
0721 05745 0376          AND     (77
0722 05746 1575          TAD     (-77
0723 05747 7650          SNA CLA
0724 05750 5726          JMP I   PRNT2
0725 05751 1613          TAD     P2
0726 05752 4777          JMS I   (P1
0727 05753 5726          JMP I   PRNT2
0730 05754 0310 CMSG,    TEXT    "CHAINING TO "
     05755 6111
     05756 1611
     05757 1607
     05760 4024
     05761 1740
0731 05762 0000 FNAME,   0;0;0;0
     05763 0000
     05764 0000
     05765 0000
0732 05766 7201 DECX,    CLA IAC                 /RADIX=DECIMAL
0733 05767 3065 OCTALX,  DCA     RADIX           /RADIX=OCTAL
0734 05770 5563          JMP I   (NEXTST
```

```
0735                        EJECT
     05775 7701
     05776 0077
     05777 4147
0736                        ORG     USETBL-1
0737 06337 0001             1                           /1 BLOCK USED
0740 06340 7777             -1                          /PAGE 200 BLOCK
0741 06341 0000             0
0742 06342 0000             0
0743 06343 0000             0
0744 06344 0000             0
0745 06345 0000             0
0746 06346 0000             0
0747 06347 0000             0
0750 06350 0000             0
0751 06351 0000             0
0752 06352 0000             0
0753 06353 0000             0
0754 06354 0000             0
0755 06355 0000             0
0756 06356 0000             0
0757 06357 0000             0
0760 06360 0000             0
0761 06361 0000             0
0762 06362 0000             0
0763 06363 0000             0
0764 06364 0000             0
0765 06365 0000             0
0766 06366 0000             0
0767 06367 0000             0
0770 06370 0000             0
0771 06371 0000             0
0772 06372 0000             0
0773 06373 0000             0
0774 06374 0000             0
0775 06375 0000             0
0776 06376 0000             0
0777 06377 0000             0
```

```
1000                            EJECT
1001                            PDPOP=4
1002                            PSJDD=5
1003                            PDPMR=6
1004                            FPPMRF=7
1005                            FPPSF1=10
1006                            FPPSF2=11
1007                            FPPSF3=12
1010                            FPPSF4=13
1011                            FPPSF5=14
1012                            FPMRFL=15
      00125 0622
      00126 0037
      00127 4325
      00130 0110
      00131 0040
      00132 3000
      00133 0012
      00134 7506
      00135 3475
      00136 7522
      00137 2620
      00140 7540
      00141 0177
      00142 7600
      00143 7750
      00144 4000
      00145 0200
      00146 1737
      00147 1671
      00150 4400
      00151 2400
      00152 3400
      00153 3610
      00154 1723
      00155 0260
      00156 0007
      00157 7774
      00160 4073
      00161 3600
      00162 7521
      00163 0600
      00164 0240
      00165 0100
      00166 7735
      00167 0077
      00170 7000
      00171 6177
      00172 4336
      00173 3711
      00174 0247
      00175 1052
      00176 4200
```

```
     00177 0000
1013                       ORG      10020          /INITIAL SYMBOL TABLE
1014 10020 0000 FPLIST,    0                       /FP LITERAL BUCKET
1015 10021 0000 DPLIST,    0                       /DP LITERAL BUCKET
1016 10022 0023 ALIST,     .+1
1017 10023 0030            .+5                     /ADDX
1020 10024 0404            0404;3000
     10025 3000
1021 10026 0014            FPPSF5
1022 10027 0110            0110
1023 10030 0035            .+5                     /ALN
1024 10031 1416            1416;0
     10032 0000
1025 10033 0013            FPPSF4
1026 10034 0010            0010
1027 10035 0042            .+5                     /AND
1030 10036 1604            1604;0
     10037 0000
1031 10040 0006            PDPMR
1032 10041 0000            AND 0
1033 10042 0000            0                       /ATX
1034 10043 2430            2430;0
     10044 0000
1035 10045 0013            FPPSF4
1036 10046 0020            0020
1037 10047 0050 BLIST,     .+1
1040 10050 0000            0                       /BASE
1041 10051 0123            0123;0500
     10052 0500
1042 10053 0005            PSUDO
1043 10054 5000            BASEX
1044 10055 0056 CLIST,     .+1
1045 10056 0063            .+5                     /CDF
1046 10057 0406            0406;0
     10060 0000
1047 10061 0004            PDPOP
1050 10062 6201            CDF
1051 10063 0070            .+5                     /CHAIN
1052 10064 1001            1001;1116
     10065 1116
1053 10066 0005            PSUDO
1054 10067 5200            CHAINX
1055 10070 0075            .+5                     /CIA
1056 10071 1101            1101;0
     10072 0000
1057 10073 0004            PDPOP
1060 10074 7041            CIA
1061 10075 0102            .+5                     /CIF
1062 10076 1106            1106;0
     10077 0000
1063 10100 0004            PDPOP
1064 10101 6202            CIF
1065 10102 0107            .+5                     /CLA
1066 10103 1401            1401;0
```

```
     10104 0000
1067 10105 0004          PDPOP
1070 10106 7200          CLA
1071 10107 0114          .+5                        /CLL
1072 10110 1414          1414;0
     10111 0000
1073 10112 0004          PDPOP
1074 10113 7100          CLL
1075 10114 0121          .+5                        /CMA
1076 10115 1501          1501;0
     10116 0000
1077 10117 0004          PDPOP
1100 10120 7040          CMA
1101 10121 0000          0                          /CML
1102 10122 1514          1514;0
     10123 0000
1103 10124 0004          PDPOP
1104 10125 7020          CML
1105 10126 0127 DLIST,   .+1
1106 10127 0134          .+5                        /DCA
1107 10130 0301          0301;0
     10131 0000
1110 10132 0006          PDPMR
1111 10133 3000          DCA 0
1112 10134 0000          0                          /DECIMAL
1113 10135 0503          0503;1115
     10136 1115
1114 10137 0105          PSUDO 0100
1115 10140 5766          DECX
1116 10141 0142 ELIST,   .+1
1117 10142 0147          .+5                        /EJECT
1120 10143 1205          1205;0324
     10144 0324
1121 10145 0005          PSUDO
1122 10146 5143          EJECTX
1123 10147 0000          0                          /END
1124 10150 1604          1604;0
     10151 0000
1125 10152 0005          PSUDO
1126 10153 4600          ENDX
1127 10154 0155 FLIST,   .+1
1130 10155 0162          .+5                        /FADD
1131 10156 0104          0104;0400
     10157 0400
1132 10160 0007          FPPMRF
1133 10161 1000          1000
1134 10162 0167          .+5                        /FADDL
1135 10163 0104          0104;0414
     10164 0414
1136 10165 0015          FPMRFL
1137 10166 1000          1000
1140 10167 0174          .+5                        /FADDM
1141 10170 0104          0104;0415
     10171 0415
```

```
1142 10172 0007          FPPMRF
1143 10173 5000          5000
1144 10174 0201          .+5                    /FADDML
1145 10175 0104          0104;0415
     10176 0415
1146 10177 1415          FPMRFL+1400
1147 10200 5000          5000
1150 10201 0206          .+5                    /FCLA
1151 10202 0314          0314;0100
     10203 0100
1152 10204 0012          FPPSF3
1153 10205 0002          0002
1154 10206 0213          .+5                    /FDIV
1155 10207 0411          0411;2600
     10210 2600
1156 10211 0007          FPPMRF
1157 10212 3000          3000
1160 10213 0220          .+5                    /FDIVL
1161 10214 0411          0411;2614
     10215 2614
1162 10216 0015          FPMRFL
1163 10217 3000          3000
1164 10220 0225          .+5                    /FEXIT
1165 10221 0530          0530;1124
     10222 1124
1166 10223 0012          FPPSF3
1167 10224 0000          0000
1170 10225 0232          .+5                    /FLDA
1171 10226 1404          1404;0100
     10227 0100
1172 10230 0007          FPPMRF
1173 10231 0000          0000
1174 10232 0237          .+5                    /FLDAL
1175 10233 1404          1404;0114
     10234 0114
1176 10235 0015          FPMRFL
1177 10236 0000          0000
1200 10237 0244          .+5                    /FMUL
1201 10240 1525          1525;1400
     10241 1400
1202 10242 0007          FPPMRF
1203 10243 4000          4000
1204 10244 0251          .+5                    /FMULL
1205 10245 1525          1525;1414
     10246 1414
1206 10247 0015          FPMRFL
1207 10250 4000          4000
1210 10251 0256          .+5                    /FMULM
1211 10252 1525          1525;1415
     10253 1415
1212 10254 0007          FPPMRF
1213 10255 7000          7000
1214 10256 0263          .+5                    /FMULML
1215 10257 1525          1525;1415
```

```
     10260 1415
1216 10261 1415          FPMRFL+1400
1217 10262 7000          7000
1220 10263 0270          .+5                        /FNEG
1221 10264 1605          1605;0700
     10265 0700
1222 10266 0012          FPPSF3
1223 10267 0003          0003
1224 10270 0275          .+5                        /FNOP
1225 10271 1617          1617;2000
     10272 2000
1226 10273 0012          FPPSF3
1227 10274 0040          0040
1230 10275 0302          .+5                        /FNORM
1231 10276 1617          1617;2215
     10277 2215
1232 10300 0012          FPPSF3
1233 10301 0004          0004
1234 10302 0307          .+5                        /FPAUSE
1235 10303 2001          2001;2523
     10304 2523
1236 10305 0512          FPPSF3+0500
1237 10306 0001          0001
1240 10307 0314          .+5                        /FPCOM
1241 10310 2003          2003;1715
     10311 1715
1242 10312 0004          PDPOP
1243 10313 6553          6553
1244 10314 0321          .+5                        /FPHLT
1245 10315 2010          2010;1424
     10316 1424
1246 10317 0004          PDPOP
1247 10320 6554          6554
1250 10321 0326          .+5                        /FPICL
1251 10322 2011          2011;0314
     10323 0314
1252 10324 0004          PDPOP
1253 10325 6552          6552
1254 10326 0333          .+5                        /FPINT
1255 10327 2011          2011;1624
     10330 1624
1256 10331 0004          PDPOP
1257 10332 6551          6551
1260 10333 0340          .+5                        /FPIST
1261 10334 2011          2011;2324
     10335 2324
1262 10336 0004          PDPOP
1263 10337 6557          6557
1264 10340 0345          .+5                        /FPRST
1265 10341 2022          2022;2324
     10342 2324
1266 10343 0004          PDPOP
1267 10344 6556          6556
1270 10345 0352          .+5                        /FPST
```

```
1271 10346 2023          2023;2400
     10347 2400
1272 10350 0004          PDPOP
1273 10351 6555          6555
1274 10352 0357          .+5                        /FSTA
1275 10353 2324          2324;0100
     10354 0100
1276 10355 0007          FPPMRF
1277 10356 6000          6000
1300 10357 0364          .+5                        /FSTAL
1301 10360 2324          2324;0114
     10361 0114
1302 10362 0015          FPMRFL
1303 10363 6000          6000
1304 10364 0371          .+5                        /FSUB
1305 10365 2325          2325;0200
     10366 0200
1306 10367 0007          FPPMRF
1307 10370 2000          2000
1310 10371 0000          0                          /FSUBL
1311 10372 2325          2325;0214
     10373 0214
1312 10374 0015          FPMRFL
1313 10375 2000          2000
1314 10376 0000 GLIST,   0
1315 10377 0400 HLIST,   .+1
1316 10400 0000          0                          /HLT
1317 10401 1424          1424;0
     10402 0000
1320 10403 0004          PDPOP
1321 10404 7402          HLT
1322 10405 0406 ILIST,   .+1
1323 10406 0413          .+5                        /IAC
1324 10407 0103          0103;0
     10410 0000
1325 10411 0004          PDPOP
1326 10412 7001          IAC
1327 10413 0420          .+5                        /IFNEG
1330 10414 0616          0616;0507
     10415 0507
1331 10416 0005          PSUDO
1332 10417 5656          IFNEGX
1333 10420 0425          .+5                        /IFNZRO
1334 10421 0616          0616;3222
     10422 3222
1335 10423 1705          PSUDO+1700
1336 10424 5640          IFNZRX
1337 10425 0432          .+5                        /IFPOS
1340 10426 0620          0620;1723
     10427 1723
1341 10430 0005          PSUDO
1342 10431 5663          IFPOSX
1343 10432 0437          .+5                        /IFREF
1344 10433 0622          0622;0506
```

```
     10434 0576
1345 10435 0005          PSUDO
1346 10436 5670          IFREFX
1347 10437 0444          .+5                        /IFZERO
1350 10440 0632          0632;0522
     10441 0522
1351 10442 1705          PSUDO+1700
1352 10443 5647          IFZROX
1353 10444 0000          0                          /ISZ
1354 10445 2332          2332;0
     10446 0000
1355 10447 0006          PDPMR
1356 10450 2000          ISZ 0
1357 10451 0452 JLIST,   .+1
1360 10452 0457          .+5                        /JA
1361 10453 0100          0100;0
     10454 0000
1362 10455 0011          FPPSF2
1363 10456 1030          1030
1364 10457 0464          .+5                        /JAC
1365 10460 0103          0103;0
     10461 0000
1366 10462 0012          FPPSF3
1367 10463 0007          0007
1370 10464 0471          .+5                        /JAL
1371 10465 0114          0114;0
     10466 0000
1372 10467 0011          FPPSF2
1373 10470 1070          1070
1374 10471 0476          .+5                        /JEQ
1375 10472 0521          0521;0
     10473 0000
1376 10474 0011          FPPSF2
1377 10475 1000          1000
1400 10476 0503          .+5                        /JGE
1401 10477 0705          0705;0
     10500 0000
1402 10501 0011          FPPSF2
1403 10502 1010          1010
1404 10503 0510          .+5                        /JGT
1405 10504 0724          0724;0
     10505 0000
1406 10506 0011          FPPSF2
1407 10507 1060          1060
1410 10510 0515          .+5                        /JLE
1411 10511 1405          1405;0
     10512 0000
1412 10513 0011          FPPSF2
1413 10514 1020          1020
1414 10515 0522          .+5                        /JLT
1415 10516 1424          1424;0
     10517 0000
1416 10520 0011          FPPSF2
1417 10521 1050          1050
```

```
1420 10522 0527          .+5                    /JMP
1421 10523 1520          1520;0
     10524 0000
1422 10525 0006          PDPMR
1423 10526 5000          JMP 0
1424 10527 0534          .+5                    /JMS
1425 10530 1523          1523;0
     10531 0000
1426 10532 0006          PDPMR
1427 10533 4000          JMS 0
1430 10534 0541          .+5                    /JNE
1431 10535 1605          1605;0
     10536 0000
1432 10537 0011          FPPSF2
1433 10540 1040          1040
1434 10541 0546          .+5                    /JSA
1435 10542 2301          2301;0
     10543 0000
1436 10544 0011          FPPSF2
1437 10545 1120          1120
1440 10546 0553          .+5                    /JSR
1441 10547 2322          2322;0
     10550 0000
1442 10551 0011          FPPSF2
1443 10552 1130          1130
1444 10553 0000          0                      /JXN
1445 10554 3016          3016;0
     10555 0000
1446 10556 0010          FPPSF1
1447 10557 2000          2000
1450 10560 0561 KLIST,   .+1
1451 10561 0566          .+5                    /KCC
1452 10562 0303          0303;0
     10563 0000
1453 10564 0004          PDPOP
1454 10565 6032          KCC
1455 10566 0573          .+5                    /KRB
1456 10567 2202          2202;0
     10570 0000
1457 10571 0004          PDPOP
1460 10572 6036          KRB
1461 10573 0600          .+5                    /KRS
1462 10574 2223          2223;0
     10575 0000
1463 10576 0004          PDPOP
1464 10577 6034          KRS
1465 10600 0000          0                      /KSF
1466 10601 2306          2306;0
     10602 0000
1467 10603 0004          PDPOP
1470 10604 6031          KSF
1471 10605 0606 LLIST,   .+1
1472 10606 0613          .+5                    /LAS
1473 10607 0123          0123;0
```

```
     10610 0000
1474 10611 0004          PDPOP
1475 10612 7604          LAS
1476 10613 0620          .+5                        /LDX
1477 10614 0430          0430;0
     10615 0000
1500 10616 0014          FPPSF5
1501 10617 0100          0100
1502 10620 0625          .+5                        /LISTOFF
1503 10621 1123          1123;2417
     10622 2417
1504 10623 0605          PSUDO+0600
1505 10624 5062          LSTOFX
1506 10625 0632          .+5                        /LISTON
1507 10626 1123          1123;2417
     10627 2417
1510 10630 1605          PSUDO+1600
1511 10631 5061          LSTONX
1512 10632 0000          0                          /LITORG
1513 10633 1124          1124;1722
     10634 1722
1514 10635 0705          PSUDO+00700
1515 10636 1114          LITORX
1516 10637 0000 MLIST,   0
1517 10640 0641 NLIST,   .+1
1520 10641 0000          0                          /NOP
1521 10642 1720          1720;0
     10643 0000
1522 10644 0004          PDPOP
1523 10645 7000          NOP
1524 10646 0647 OLIST,   .+1
1525 10647 0654          .+5                        /OCTAL
1526 10650 0324          0324;0114
     10651 0114
1527 10652 0005          PSUDO
1530 10653 5767          OCTALX
1531 10654 0661          .+5                        /ORG
1532 10655 2207          2207;0
     10656 0000
1533 10657 0005          PSUDO
1534 10660 5012          ORGX
1535 10661 0000          0                          /OSR
1536 10662 2322          2322;0
     10663 0000
1537 10664 0004          PDPOP
1540 10665 7404          OSR
1541 10666 0667 PLIST,   .+1
1542 10667 0000          0                          /PAGE
1543 10670 0107          0107;0500
     10671 0500
1544 10672 0005          PSUDO
1545 10673 5064          PAGEX
1546 10674 0000 QLIST,   0
1547 10675 0676 RLIST,   .+1
```

```
1550 10676 0703          .+5                    /RAL
1551 10677 0114          0114;0
     10700 0000
1552 10701 0004          PDPOP
1553 10702 7004          RAL
1554 10703 0710          .+5                    /RAR
1555 10704 0122          0122;0
     10705 0000
1556 10706 0004          PDPOP
1557 10707 7010          RAR
1560 10710 0715          .+5                    /RDF
1561 10711 0406          0406;0
     10712 0000
1562 10713 0004          PDPOP
1563 10714 6214          RDF
1564 10715 0722          .+5                    /RIB
1565 10716 1102          1102;0
     10717 0000
1566 10720 0004          PDPOP
1567 10721 6234          RIB
1570 10722 0727          .+5                    /RIF
1571 10723 1106          1106;0
     10724 0000
1572 10725 0004          PDPOP
1573 10726 6224          RIF
1574 10727 0734          .+5                    /RMF
1575 10730 1506          1506;0
     10731 0000
1576 10732 0004          PDPOP
1577 10733 6244          RMF
1600 10734 0741          .+5                    /RTL
1601 10735 2414          2414;0
     10736 0000
1602 10737 0004          PDPOP
1603 10740 7006          RTL
1604 10741 0000          0                      /RTR
1605 10742 2422          2422;0
     10743 0000
1606 10744 0004          PDPOP
1607 10745 7012          RTR
1610 10746 0747 SLIST,   .+1
1611 10747 0754          .+5                    /SETB
1612 10750 0524          0524;0200
     10751 0200
1613 10752 0011          FPPSF2
1614 10753 1110          1110
1615 10754 0761          .+5                    /SETX
1616 10755 0524          0524;3000
     10756 3000
1617 10757 0011          FPPSF2
1620 10760 1100          1100
1621 10761 0766          .+5                    /SKP
1622 10762 1320          1320;0
     10763 0000
```

```
1623 10764 0004          PDPOP
1624 10765 7410          SKP
1625 10766 0773          .+5                        /SMA
1626 10767 1501          1501;0
     10770 0000
1627 10771 0004          PDPOP
1630 10772 7500          SMA
1631 10773 1000          .+5                        /SNA
1632 10774 1501          1601;0
     10775 0000
1633 10776 0004          PDPOP
1634 10777 7450          SNA
1635 11000 1005          .+5                        /SNL
1636 11001 1614          1614;0
     11002 0000
1637 11003 0004          PDPOP
1640 11004 7420          SNL
1641 11005 1012          .+5                        /SPA
1642 11006 2001          2001;0
     11007 0000
1643 11010 0004          PDPOP
1644 11011 7510          SPA
1645 11012 1017          .+5                        /STARTD
1646 11013 2401          2401;2224
     11014 2224
1647 11015 0412          FPPSF3+0400
1650 11016 0006          0006
1651 11017 1024          .+5                        /STARTF
1652 11020 2401          2401;2224
     11021 2224
1653 11022 0612          FPPSF3+0600
1654 11023 0005          0005
1655 11024 1031          .+5                        /SZA
1656 11025 3201          3201;0
     11026 0000
1657 11027 0004          PDPOP
1660 11030 7440          SZA
1661 11031 0000          0                          /SZL
1662 11032 3214          3214;0
     11033 0000
1663 11034 0004          PDPOP
1664 11035 7430          SZL
1665 11036 1037 TLIST,   .+1
1666 11037 1044          .+5                        /TAD
1667 11040 0104          0104;0
     11041 0000
1670 11042 0006          PDPMR
1671 11043 1000          TAD 0
1672 11044 1051          .+5                        /TCF
1673 11045 0306          0306;0
     11046 0000
1674 11047 0004          PDPOP
1675 11050 6042          TCF
1676 11051 1056          .+5                        /TEXT
```

```
1677 11052 0530          0530;2400
     11053 2400
1700 11054 0005          PSUDO
1701 11055 5600          TEXTX
1702 11056 1063          .+5                        /TLS
1703 11057 1423          1423;0
     11060 0000
1704 11061 0004          PDPOP
1705 11062 6046          TLS
1706 11063 1070          .+5                        /TPC
1707 11064 2003          2003;0
     11065 0000
1710 11066 0004          PDPOP
1711 11067 6044          TPC
1712 11070 1075          .+5                        /TRAP3
1713 11071 2201          2201;2063
     11072 2063
1714 11073 0010          FPPSF1
1715 11074 3000          3000
1716 11075 1102          .+5                        /TRAP4
1717 11076 2201          2201;2064
     11077 2064
1720 11100 0010          FPPSF1
1721 11101 4000          4000
1722 11102 1107          .+5                        /TRAP5
1723 11103 2201          2201;2065
     11104 2065
1724 11105 0010          FPPSF1
1725 11106 5000          5000
1726 11107 1114          .+5                        /TRAP6
1727 11110 2201          2201;2066
     11111 2066
1730 11112 0010          FPPSF1
1731 11113 6000          6000
1732 11114 1121          .+5                        /TRAP7
1733 11115 2201          2201;2067
     11116 2067
1734 11117 0010          FPPSF1
1735 11120 7000          7000
1736 11121 0000          0                          /TSF
1737 11122 2306          2306;0
     11123 0000
1740 11124 0004          PDPOP
1741 11125 6041          TSF
1742 11126 0000 ULIST,   0
1743 11127 0000 VLIST,   0
1744 11130 0000 WLIST,   0
1745 11131 1132 LISTX,   .+1
1746 11132 0000          0                          /XTA
1747 11133 2401          2401;0
     11134 0000
1750 11135 0013          FPPSF4
1751 11136 0030          0030
1752 11137 0000 YLIST,   0
```

```
1753 11140 0000 ZLIST,  0
1754            FREE,
```

```
1755                            EJECT
1756                            PAGE
1757 11200 7200 INITAL,         CLA
1760 11201 6211                 CDF     FLD1
1761 11202 3777                 DCA I   (7775          /FIX UP DIAL I/O ROUTINES
1762 11203 1376                 TAD     (5772
1763 11204 3775                 DCA I   (7776
1764 11205 1374                 TAD     (5773
1765 11206 3773                 DCA I   (7777
1766 11207 4772                 JMS I   (7774          /READ I COPLETE COPY OF I/O STUF
1767 11210 1273                 RDSYS                  /POINTER TO ARGS FOR SYS READ IN
1770 11211 4771                 JMS I   (4200          /MOVE ROUTINE UP
1771 11212 6211                 CDF     FLD1
1772 11213 4000                 4000
1773 11214 6211                 CDF     FLD1
1774 11215 7000                 7000
1775 11216 1000                 1000
1776 11217 4772                 JMS I   (7774          /FIND FUDGE FACTOR FOR SOURCE
1777 11220 1265                 DUMYS
2000 11221 1770                 TAD I   (7770          /GET READ ENTRY POINT
2001 11222 3300                 DCA     RDDIAL
2002 11223 1767                 TAD I   (7771          /SOURCE FUDGE NUMBER
2003 11224 3277                 DCA     FUDGES
2004 11225 4772                 JMS I   (7774          /FIND STUFF FOR BINARY
2005 11226 1267                 DUMYB
2006 11227 1770                 TAD I   (7770          /MAKE SURE ITS THE SAME HANDLER
2007 11230 7041                 CIA
2010 11231 1300                 TAD     RDDIAL
2011 11232 7640                 SZA CLA
2012 11233 7402                 HLT                    /TWO DIFFERENT HANDLERS
2013 11234 1767                 TAD I   (7771          /STORE BINARY FUDGE
2014 11235 6201                 CDF     FLD0
2015 11236 3766                 DCA I   (BFUDGE
2016 11237 1277                 TAD     FUDGES         /NOW SOURCE FUDGE
2017 11240 3765                 DCA I   (SFUDGE
2020 11241 1300                 TAD     RDDIAL         /READ ENTRY POINT
2021 11242 0364                 AND     (177           /GET PAGE DISPLACEMENT
2022 11243 1363                 TAD     (7600          /PLUS BASE
2023 11244 3762                 DCA I   (DIALRD
2024 11245 7126                 CLL CML RTL
2025 11246 1762                 TAD I   (DIALRD        /NOW WRITE ENTRY POINT
2026 11247 3761                 DCA I   (DIALWR
2027 11250 6211                 CDF     FLD1
2030 11251 1300                 TAD     RDDIAL
2031 11252 0363                 AND     (7600          /GET PAGE ADDR OF I/O ROUTINE
2032 11253 3256                 DCA     .+3
2033 11254 4760                 JMS I   (7200          /MOVE ROUTINE TO FIELD 0
2034 11255 6211                 CDF     FLD1
2035 11256 0000                 0
2036 11257 6201                 CDF     FLD0
2037 11260 7600                 7600
2040 11261 0200                 200
2041 11262 5757                 JMP I   (GETLPT        /GO FIND LINE PRINTER
```

```
2042 11263 0110 DUMYS,  110;SOURCE;0;1
     11264 0016
     11265 0026
     11266 0001
2043 11267 0111 DUMYB,  111;BINARB;0;1
     11270 0015
     11271 0020
     11272 0001
2044 11273 0100 RDSYS,  100;30;22;2
     11274 0030
     11275 0022
     11276 0002
2045 11277 2000 FUDGES, 0
2046 11300 0000 RDDIAL, 0
```

```
2047                    EJECT
     11357 1400
     11360 7200
     11361 0102
     11362 0101
     11363 7600
     11364 0177
     11365 0106
     11366 0104
     11367 7771
     11370 7770
     11371 4200
     11372 7774
     11373 7777
     11374 5773
     11375 7776
     11376 5772
     11377 7775
2050                    PAGE
2051 11400 6201 GETLPT, CDF     FLD0            /NOW FIND OUT ...
2052 11401 6652          6652                   /WHICH PRINTER ?
2053 11402 6662          6662
2054 11403 1377          TAD     (-4
2055 11404 3245          DCA     LPT
2056 11405 3246          DCA     LPT2
2057 11406 6046          TLS                    /INITIALIZE TTY
2060 11407 6661 ALPT,    6661                   /CHECK FOR ANALEX
2061 11410 7410          SKP
2062 11411 5233          JMP     ISANAL         /ITS THE 645
2063 11412 2246          ISZ     LPT2           /INCREMENT TIMER
2064 11413 5207          JMP     ALPT
2065 11414 2245          ISZ     LPT
2066 11415 5207          JMP     ALPT
2067 11416 1377          TAD     (-4            /RESET TIMER FOR LP08 TRY
2070 11417 3245          DCA     LPT
2071 11420 6666          6666
2072 11421 7200          CLA
2073 11422 6661 LLPT,    6661                   /TEST LP08 FLAG
2074 11423 7410          SKP
2075 11424 5240          JMP     ISLP08         /ITS AN LP08
2076 11425 2246          ISZ     LPT2           /INCREMENT TIMER
2077 11426 5222          JMP     LLPT
2100 11427 2245          ISZ     LPT
2101 11430 5222          JMP     LLPT
2102 11431 6203 TSTLST, CDF CIF
2103 11432 5776          JMP I   (START
2104 11433 1375 ISANAL,  TAD     (ANALEX        /SET PRINTER HANDLER PTR
2105 11434 3774          DCA I   (PC            /TO ANALEX PRINTER HANDLER
2106 11435 1373          TAD     (-201          /SET WIDTH TO 128 CHARS
2107 11436 3772          DCA I   (WIDTH
2110 11437 5231          JMP     TSTLST
2111 11440 1371 ISLP08,  TAD     (LP08
2112 11441 3774          DCA I   (PC
```

```
2113 11442 1370          TAD      (-121           /SET WIDTH TO 80
2114 11443 3772          DCA I    (WIDTH
2115 11444 5231          JMP      TSTLST
2116 11445 0000 LPT,     0
2117 11446 0000 LPT2,    0
     11570 7657
     11571 4317
     11572 4243
     11573 7577
     11574 4244
     11575 4266
     11576 0622
     11577 7774
```

0000 ERRORS

```
ACE     00112
ACH     00117
ACHR    04257
ACL     00116
ACO     00115
ADCEXP  00361
ADR     02520
ADRADD  02641
ADRAND  02664
ADREXP  02531
ADRGET  01723
ADRMUL  02706
ADROP   02545
ADROR   02673
ADRSUB  02652
ALIST   10022
ALPT    11407
AL1     00241
ANALEX  04260
ANORM   00306
AR1     00342
ASMBL   01000
ASMOF   00057
BACK1   03600
BAD     02445
BADEXP  01426
BADX    02075
BASER   00032
BASEX   05000
BBLOCK  00103
BFUDGE  00104
BINARB  00015
BINARY  06400
BLANK   03734
BLIST   10047
BLOCKN  04474
BUCKET  00051
BUCKTS  04022
CBSWIT  05353
CHAINX  05200
CHERR   05347
CHKIND  01671
CHKLIT  03000
CHRCNT  00024
CHRPTR  00017
CKKILL  04073
CLIST   10055
CLRBIN  04560
CMP2WD  03230
CMSG    05754
CNLOOP  05212
COMPL   02752
CPLBUF  06000
```

```
CPLIT   00111
CPTMP   00021
CRLF    04223
CRLINK  03017
CRLIT   03032
DECON   02334
DECONV  02326
DECX    05766
DEFINO  01244
DEFLBL  01237
DEFSYM  02557
DIALRD  00101
DIALWR  00102
DIFFPL  03252
DIRBLK  05245
DIRLUP  05242
DIVLUP  01031
DLIST   10126
DLITS   03314
DLITS2  03317
DMPFIN  05136
DMPLIT  05075
DNUMBR  02361
DUADDS  00437
DOCHR   00704
DOLIT   03051
DONTØ   04510
DP      02102
DPLIST  10021
DUMYB   11267
DUMYS   11263
EJECTX  05143
ELIST   10141
ENDBLK  04117
ENDEXP  01074
ENDFPI  02311
ENDFPN  02216
ENDLIN  00734
ENDOUT  04551
ENDX    04600
ENDXX   04111
EOP2    04643
EQUERR  01326
EQUN    00071
ERMSG   04336
ERMSG1  04325
ERRORS  04762
ERR1    01732
ERR2    01771
EXPGET  01354
EXPR    02400
EXPSW   00040
EXPTYP  00037
EXPVAL  00034
```

```
EXTMP   00067
EXTMP2  00070
FADR    00363
FILLUP  05263
FINDOP  02626
FINLPB  04305
FITCNT  00460
FIXOPC  04000
FIXORG  05023
FLD0    00000
FLD1    00010
FLINTP  02200
FLIST   10154
FMNOAD  00524
FM30    00530
FNAME   05762
FNEGX   00363
FNLOOP  05271
FORMT1  01477
FORMT2  01465
FORMT3  01515
FP      02032
FPADD   00417
FPADLP  00425
FPDIGT  02362
FPDIV   00531
FPDVLP  00551
FPESGN  02373
FPFIX   02267
FPGOTO  00241
FPJMP   00265
FPJUMP  00360
FPLAC   00300
FPLACE  00304
FPLDMP  03321
FPLIST  10020
FPLNUM  03354
FPLOOK  03212
FPMDCD  00514
FPMLLP  00505
FPMRFL  00015
FPMUL   00462
FPOPER  00254
FPPADR  00043
FPPLIT  03200
FPPMR   01431
FPPMRF  00007
FPPMRL  01473
FPPSF1  00010
FPPSF2  00011
FPPSF3  00012
FPPSF4  00013
FPPSF5  00014
FPPSWT  00075
```

```
FPPS1   01534
FPPS2   01637
FPPS3   01552
FPPS4   01647
FPPS5   01655
FPPWD2  00077
FPP2WD  00076
FPSKIP  00262
FPSNL   00222
FPSTO   00270
FPSUB   00416
FPT     00177
FPTMP   02365
FPTMP2  02370
FREE    11141
FUDGES  11277
F177    00210
F200    00215
F7400   00200
F7600   00200
GETADR  01737
GETCHR  03711
GETCHT  05622
GETCN   05400
GETC2   03714
GETEXP  01403
GETFPN  02470
GETLPT  11400
GETNAM  03610
GETUNT  02014
GLIST   10376
GNC     03670
GOTOP   02636
HLIST   10377
HOOKIN  03447
IFNEGX  05656
IFNZRX  05640
IFPOSX  05663
IFREFX  05670
IFZROX  05647
ILIST   10405
INDEX   00046
INDRCT  00047
INFP    02077
INITAL  11200
ISANAL  11433
ISDOLR  01023
ISDUT   02513
ISLP08  11440
ISPLUS  02252
ISQ     05411
JLIST   10451
JSTONE  01343
KLIST   10560
```

```
LASTOP 00055
LETTER 03655
LINE   06200
LINENO 00060
LINSIZ 00023
LISTSW 00061
LISTX  11131
LITADR 03116
LITBAS 03164
LITLUP 05126
LITORX 01114
LITPTR 03166
LITRG1 00030
LITRG2 00031
LITRL  00100
LLIST  10605
LLPT   11422
LOCTR1 00026
LOCTR2 00027
LOCK   03406
LOOKUP 03400
LOOP6B 05604
LPT    11445
LPT2   11446
LP08   04317
LSIZE  04242
LSTOFX 05062
LSTON  00062
LSTONX 05061
LTEMP  00066
LTMSG  00761
LUNAME 01200
LUPBKT 04673
MAKLNK 01626
MAXLIN 00066
MLIST  10637
MORFPL 03333
MOVEUP 02147
MOV2WD 01314
MSGDUN 04360
MULOOP 02716
MULT10 02302
MXDTYP 02554
M1000  00735
NAME1  00052
NAME2  00053
NAME3  00054
NBITS  00010
NCHARS 00020
NCNT   03654
NCTMP  00022
NDONE  03644
NEGLP  00370
NEJECT 04236
```

```
NEWBLK 04477
NEWFPL 03256
NEWLBL 01234
NEWLIT 03131
NEWONE 01400
NEWSYM 01331
NEXT   00016
NEXTST 00600
NFILES 05352
NLETR  03666
NLIST  10640
NOADD  02744
NOASM  01020
NOBLNK 04540
NOCRLF 04254
NODGT  03545
NONAME 03651
NOREAD 00674
NORMLP 00315
NOSYM  02447
NOTAB  04215
NOTEG  00626
NOTFIL 05340
NOTFPD 02357
NOTIND 01716
NOTIT  03071
NOTNUM 02475
NOTP0  03045
NOTREF 05700
NOTSAM 03443
NOUNIT 02030
NPTR   03653
NSWTCH 03552
NUM    03553
NUMBER 03475
NUMLUP 03501
NUM1   03554
NUM2   03555
NWUSED 03165
NXTBKT 04753
OADD   00400
OCNT   00006
OCTALX 05767
OCTNUM 03533
OCTOUT 01052
ODDCHR 00730
ODDEVN 00760
OKEXP  02444
OLDFP  03353
OLDN3  00004
OLIST  10646
OLOOP  01056
ONE    02525
OPCJMP 01352
```

```
OPCODE 00045
OPCTBL 01353
OPE    00113
OPH    00122
OPL    00121
OPO    0012?
OPRAND 02155
OPRBR  02620
OPRBRS 04054
ORGERR 0502?
ORGX   05012
OR1    00443
OTEMP  00005
OUTFPL 03363
OUTSWT 00063
OUTTXT 05617
OUTWRD 04409
OUT1WD 01423
OUT2WD 01421
OUT3WD 01417
OVER3  01025
OWTEMP 04557
O1     04472
PAGEJ  05154
PAGEN  00112
PAGEND 03162
PAGEX  05064
PAGE0  01633
PAGSIZ 00007
PAKLUP 03626
PAL1   00577
PANORM 00574
PAR1   00576
PASSNO 00056
PC     04244
PCHR   04245
PDPMR  00006
PDPOP  00004
PDP8MR 0160?
PER2   02221
PFPLAC 00573
PFPLCE 00527
PLIST  10666
PNEG   00575
POPO   00357
PRINTC 0420?
PRINT2 05714
PRMSG  04352
PRNTLN 0200?
PRNTST 04522
PRNT2  05726
PRSW   02324
PSEUDO 01555
PSUDO  00005
```

```
PUNDEF 02437
PUTVAL 01306
P0LBUF 07400
P0LIT  00110
P1     04147
P2     05713
P3     05714
QLIST  10674
RADIX  00065
RDBLOK 00670
RDDIAL 11300
RDLOOP 00643
RDSYS  11273
RETPPL 03242
RETLIT 03125
RETSYS 04131
RLIST  10675
SAMBLK 04514
SAMFLD 05036
SAMPAG 05053
SBLOCK 00105
SCSWT  00064
SEMICL 03760
SETTAB 04212
SETTYP 04732
SEXP   02325
SFUDGE 00106
SHIFT  03556
SIGNX  00066
SIZPAG 00025
SLIST  10746
SOURCB 00016
SOURCE 07000
SRCUNT 00666
START  00622
STAR20 00107
STPRNT 04670
SWTOUT 00113
SYMBOL 02424
SYMTYP 02436
SYM1   04746
SYM2   04741
SYM3   04737
SYSBLK 04143
TABCNT 04241
TEN    02321
TEXTX  05600
THSPAG 01627
TLIST  11036
TMP    00114
TRYBLK 01340
TRYEQU 01246
TSTLIT 03107
TSTLST 11431
```

```
TTY     04311
TYPE3   03355
ULIST   11126
UNDEF   02600
UNIT    05243
UNNORM  02136
USEB    00014
USEBLK  04127
USETBL  0634Ø
VERS    00003
VLIST   11127
WDPTR   00762
WIDTH   04243
WLIST   11130
WORD1   00041
WORD2   00042
WRBLOK  04456
WRD     04556
XBAD    00075
XDOPR   02134
XFOPR   02063
XFPTBL  02067
XINCR   00050
XITEMP  01671
XOPRND  00155
XPAGE   03163
X10     00010
X11     00011
X12     00012
X13     00013
X14     00014
X15     00015
YLIST   11137
ZLIST   11140
ZRONOX  04016
```

INDEX

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 & PDP-12
Digital Software News for the PDP-11
Digital Software News for the PDP-9/15 Family

These newsletters contain information applicable to software available from Digital's Program Library, Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning Digital's Software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problem to:

Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

These forms which are provided in the software kit should be fully filled out and accompanied by teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software and manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest Digital Field office or representative. U.S.A. customers may order directly from the Program Library in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

DECUS
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

— — — — — — — — — — — — — — — — — Fold Here —

— — — — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

READER'S COMMENTS

FPP Assemblers
User's Manual
DEC-12-AQZA-D

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability and readability.

Did you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Please state your position. _____ Date: _____

Name: _____ Organization: _____

Street: _____ Department: _____

City: _____ State: _____ Zip or Country _____

— — — — — — — — — — — — — — — — — — Fold Here — — — — — — — — — — — — — — — — — —

— — — — — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754