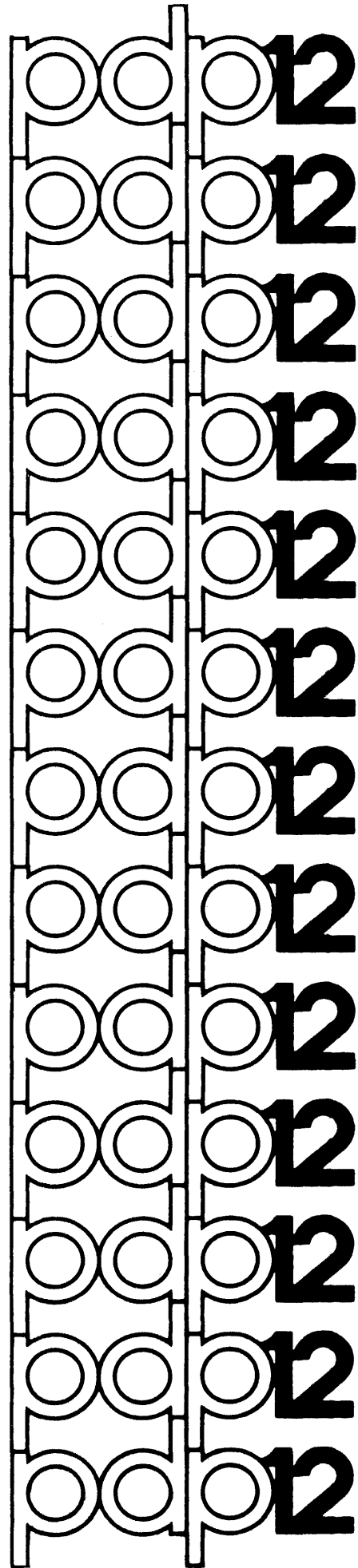


digital

FPP
SUPPORT LIBRARY
USER'S GUIDE



FPP

SUPPORT LIBRARY

USER'S GUIDE

For additional copies, order No. DEC-12-YEXA-D
from Program Library, Digital Equipment Corporation
Maynard, Massachusetts 01754 Price: \$5.00

May, 1971

Copyright © 1971 by Digital Equipment Corporation

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

The following are trademarks of Digital Equipment
Corporation, Maynard, Massachusetts:

DEC	PDP
FLIP CHIP	FOCAL
DIGITAL	COMPUTER LAB
OMNIBUS	UNIBUS

CONTENTS

	<u>Page</u>
1.0 Introduction	1
2.0 Hardware Requirements	1
3.0 Nucleus	2
3.1 Calls	2
3.2 Nucleus Functions	2
3.3 Interrupts	4
3.4 Location 005	5
3.5 Buffered I/O	6
4.0 Utility Routines	8
4.1 Description of the Utility Routines	9
4.2 Conversion Routines	10
5.0 Mathematical Routines	11
5.1 Implementing a Mathematical Routine	12
5.2 Program Library LINctapes	12
5.3 Example	13
5.4 Descriptions of the Mathematical Routines	14
6.0 Customizing the Library	16
6.1 Modifying the Assembly Sequence	16
6.2 Library Constants	17

1.0 INTRODUCTION

The FPP Support Library is a group of routines to handle in floating point format all I/O and mathematical calculations commonly required by the user, thereby performing all FPP hardware interfacing for him. The operating speed and system modularity of the Support Library facilitate all operations. Only those routines requested in the program are loaded in by the FPP Assembler; there is no need to load the entire Support Library. The Support Library relieves the programmer from the task of generating his own utility (I/O and math) routines by providing an easily implemented wide range of calculations for the Floating Point Processor.

This manual assumes complete familiarity with the FPP hardware (refer to the FPP User's Guide, DEC-12-GQZA-D) and the FPP Assembler (refer to DEC-12-AQZA-D).

The FPP routines are considered in three units:

1. Nucleus - handles special conditions so that the user can effectively control unusual events that may occur during his particular FPP application. (Division by zero and trapped instructions are examples of these special cases.) The Nucleus may be interrupt or non-interrupt driven. Optionally, a fully buffered Teletype¹ handler (input and output) may be loaded, increasing programming and running speed.
2. Format Conversion routines - convert ASCII code to floating point format and vice versa. The ASCII is left in core so that it may be displayed or printed. Three conversion formats are available, referred to as E, F, and P, as well as carriage return/line feed and Teletype output routines.
3. Mathematical functions - perform 22 different mathematical calculations, including trigonometric, logarithmic, and exponential functions.

2.0 HARDWARE REQUIREMENTS

The minimum hardware necessary to use the FPP Support Library is a PDP-12/40.

¹Teletype is a registered trademark of the Teletype Corporation.

3.0 NUCLEUS

The Nucleus of the FPP Support Library monitors the functioning of the Floating Point Processor. At the user's choice, it may be interrupt or non-interrupt driven. Although the FPP may be run in any field, the Nucleus of the Support Library handles only PDP-8 mode interrupts from field 0. The Nucleus occupies two pages of core and locations 0000-0004 in field 0. Location 0005 contains a pointer which may be used as such (refer to Section 3.4) or for any other function.

3.1 Calls

All calls to the Nucleus are of the form:

```
JMS I    4           /pointer inserted in location 4
                        /at assembly time.
FUNCTION
xxxxx           /Function is second word
                        /additional information
```

3.2 Nucleus Functions

At present, Nucleus functions 0-13 are implemented. (Note that these are Nucleus functions and not mathematical functions.) Most calls require a one-word argument; one requires two words; two require no argument.

The functions are described below according to function number.

Number	Use
0	Set the FPP program counter to the address specified by the next two words. This instruction is used before the FPP is started. When the next start command is given, the FPP will start at this location, effectively placing the address in the FPP active parameter table. The coding is: <pre>JMS I 4 0 /perform function 0 JA START /the JA gives the 15 bit address</pre>
1	When a divide by 0 occurs, JMS to the routine specified by the next word. The routine must be in field 0 and is user determined. The routine is called as if in an internal interrupt routine. To return, JMP indirectly to the subroutine entry point. The FPP is not restarted; it is up to the user to restart it. If a zero divide exit is not specified, the Nucleus sets the FAC to the largest number possible and restarts the FPP.

For example, assume the user desires to set the FAC to $\emptyset.\emptyset$ instead of letting the Nucleus handle it. The following code must appear as part of initialization:

```

JMS I 4 /call the nucleus
1
MYTHING /address of my routine on
/zero divide

```

Whenever a divide by $\emptyset.\emptyset$ occurs, the Nucleus JMS's to MYTHING. The routine which performs the above task may look like the following. (For the moment, do not pay too much attention to where the FPP active parameter table is in core or to restarting the FPP.)

```

MYTHING,  $\emptyset$  /enter here on divide by  $\emptyset$ 
DCA FPPEXP /zero out the exponent
DCA FPPHGH /and the most significant bits
DCA FPPLOW /and the low order bits
JMS I 4 /and start it going
1 $\emptyset$ 
JMP I MYTHING /now return to the Nucleus

```

Other tasks could be done here, such as changing the FPP program counter to go to a different location on starting. Note that the program counter is preserved on the interrupt and that when it is started it starts at the next instruction after the divide by $\emptyset.\emptyset$.

- 2 If an exponent overflow should ever occur, JMS to the location specified by the next location. This call is similar to the divide by zero exit, function 1. If no overflow exit is specified, the FAC is set to the biggest number possible and the FPP is restarted.
- 3 Set the underflow exit location. This function is similar to function 2, but will exit if an underflow occurs. If no return is specified, the FAC will be set to $\emptyset.\emptyset$ and the FPP will be restarted.
- 4 Set the fraction overflow exit. If an overflow occurs while in double precision mode, the Nucleus will JMS to the location specified. If none is specified, the FPP will be set to the largest integer fraction and the FPP will be restarted.
- 5 Set the instruction trap exit. If an instruction trap occurs, JMS to the location specified. At present, instructions 3-7 trap. Instruction trap 3 will be explained later. This function will only transfer control on traps 4-7. If no exit is provided, the FPP will be restarted at the location following the trap. In effect, the trap will be ignored.

- 6 Set the FPP FEXIT location. This is the "normal" exit condition and should always be specified. This location is the address of a user supplied subroutine to which the Nucleus issues a JMS at FPP exit time. A "normal" exit from this subroutine returns the user in-line to his non-FPP code. If no exit is provided, the FPP will not be restarted.

- 7 Set the FPP IOT exit. If an FPHLT is given, control is passed to the specified location. If none is specified, the FPP will be restarted. Remember that FPHLT is a PDP-8 IOT and not an FPP instruction.

- 10 Start the FPP running. This call takes no arguments; the FPP is started from information in the FPP active parameter table. The Nucleus then returns after it is started.

- 11 Clear Nucleus. This call causes the FPP to be halted and the mode reset to floating point. In addition, any buffers associated with the Nucleus at the time are cleared. This is a general system reset. There are no arguments in this call.

- 12 Set FPP "character received" exit. While using the library I/O routines, control will pass to the specified location with the character just typed on the console Teletype in the AC. This allows the user to examine input for control characters. If the user wishes to process this character, return indirect as described for function 1; if he wishes to have the routines ignore the character, ISZ the return location and then JMP. If the exit return is not specified, all characters will be processed. (This function does not concern the FPP hardware; therefore, do not try to start the FPP while in this routine.)

- 13 Set the FPP failed-to-start exit location. If a start command is issued to the FPP and it fails to start, control is passed to the specified location. This instruction is useful when running on non-FPP systems. If no exit is specified, the Nucleus will attempt to restart the FPP.

- 14-17 These commands are not used.

3.3 Interrupts

As mentioned previously, interrupts with the Nucleus are the user's option. If they are being used, a pointer to an interrupt exit routine must be placed in location 3. Additional device tests may be placed here if desired. When control is passed to this location, the AC, LINK, and MQ will be as they were at interrupt time so that it appears that

the interrupt was just received. (The AC, LINK, and MQ must be re-stored by the user.) It should be noted that location 3 will be assembled last, so that the user must place his pointer in it at execution time. The following code would suffice to clean up after an interrupt:

```

START,   TAD    POINT    /place pointer to my interrupt
          DCA    3        /routine
          xxx                    /in the correct location
          MYLOC  ION      /other code
MYLOC,   ION                    /where to go to exit
          ION                    /turn on interrupts, or test
          ION                    /other device
          JMP I  0        /and return to someplace

```

If interrupts are not to be used, all that is required is that the user JMS to 0 so that the FPP can check its flags. Location 3 does not have to be changed. The following demonstrates a brief program which divides 1 by 3 and then exits.

```

START,   ORG    200
          CLA                    /clear AC to be sure
          JMS I  4                /reset the FPP
          11
          JMS I  4                /set the FPP program counter
          0
          JA CODE                /where to start
          JMS I  4
          6
          ALLDUN                 /pointer
          JMS I  4
          10
          JMS  0                /wait for completion
          JMP  -1                /by looping

ALLDUN,  0                      /JMS's here in 8 mode when FEXIT
          HLT                    /just halt as an example
          JMS I  4
          10
          JMP I  ALLDUN         /and return

CODE,    SETX   4000            /users should always define indices,
          SETB   5000            /bases,
          STARTF                /and mode first

LOOP,    FLDA   (1.
          FDIV   (3.
          FEXIT
          JA     LOOP          /after FEXIT returns, do it again

```

3.4 Location 005

Frequently the user may wish to get to the active parameter table, MQ, etc., following an interrupt. This may be accomplished by using location 5 which points to the center of a table such as the following:

```

ORG      5
FPPTAB

ORG      SOMEPLACEELSE
Ø        /contains MQ on interrupt
Ø        /contains LINC on interrupt in bit Ø
Ø        /contains AC on interrupt
Ø        /contains FPP status if FPP
FPPTAB,  Ø        /interrupted
Ø        /contains command register to be used
Ø        /next time FPP is started
Ø        /extended bits of FPP active table
Ø        /low order bits of FPP
Ø        /low order bits of location of
Ø        /index Ø
Ø        /low order bits of base register
Ø        /low order bits of operand
Ø        /exponent of AC
Ø        /high order word of AC
Ø        /low order word of AC

```

As mentioned before (Section 3.2, function 5), the 3ØØØ trap instruction is treated differently. When a 3ØØØ trap instruction is encountered, the Nucleus will do a JMS in PDP-8 mode to the location specified by the address field of the trap instruction. This allows the user to call a PDP-8 program. The return is to JMP indirectly to the entry of the user routine. The FPP will be restarted automatically.

3.5 Buffered I/O

If the user wishes to have the buffered I/O routines in the Nucleus loaded into core, he issues the following dummy statement:

```
IO=BUFFERED
```

This will cause the buffered I/O routines to be loaded directly behind the Nucleus. The routines are approximately four pages long, including buffers. There are three FPP entry points:

PUTC Normalizes the FAC, fixes it, and then puts the low order 8 bits into the Teletype buffer. If the Teletype buffer is filled, waits until a location becomes vacant and then stores away the character and returns to the user.

GETC Gets a character from the buffer, providing the line is complete (ends in a carriage return). If there is no full line in the buffer, the routine will "hang" until a character is available. Thus, this routine provides automatic line editing because a

line is not available until a carriage return is typed. If the user is typing in a line, he may type "RUBOUT" to delete the last character in the present line. All "rubbed out" characters are enclosed in form characters (a back slash). CTRL/U deletes the current line. The next example illustrates what the user would see if he typed the following (NOTE: * means RUBOUT and \ is a backslash).

```
The quick drown*****brown fa*ox jumpep*d into
the truck.
```

```
The quick drown\ nword\ brown fa\ a\ ox jumpep\ p\ d into
the truck.
```

CTRL/U gives a closing form if it occurs in the middle of a RUBOUT sequence, as does a carriage return. If some other form of output is going on at the time, the system will echo characters sporadically. If the output buffer is full, the system will not echo the character. If the input buffers are full, the system will echo a "bell" instead of the character and the character is lost. If both input and output buffers are filled, and a character is typed, it will not be entered and the bell will not be echoed. Both the input and output buffers are each approximately 80 characters long. More than one line may be entered into the input buffer before any characters are removed.

IOOUTW This call will "hang" until all output is finished. This is useful for terminating a job when characters may still be in the buffers and have not yet been printed.

The buffered I/O routines are about four pages long. The Input/Output routines always immediately follow the Nucleus in core loading. If the user does not specify where to put the Nucleus, it will put itself at location 2000 with the buffered I/O right behind it. The user may specify where it is to go by setting the symbol FPPORG to the desired value. This setting only covers the Nucleus and the buffered I/O portion of the Library Support package. The mathematical and utility routines will always be put at the end of the user's program. To originate the Nucleus at the end of the user's code, he must just type

```
FPPORG=. /this will set the FPPORG to
         /the next free location
```

Remember that the Nucleus and the buffered I/O must be in field 0. All other routines may be anywhere (including crossing field boundaries).

4.0 UTILITY ROUTINES

The Library Support Utility routines are the link between the floating point mathematical world and the ASCII Teletype world. These routines do no actual I/O themselves, but call the routines in the buffered I/O package (PUTC and GETC) if needed. If the user is not using the buffered I/O package and he is using a utility routine which can call one of these routines, he must insert a routine which performs the same function. These routines are similar in function to the mathematical routines (refer to Section 5.0) in that the calling sequence and manner of loading are the same. It should be remembered that all functions are JSA'd with the argument in the accumulator. The second argument, if present, is in base register 10. It should also be noted that the routines are free to use base registers 0 and 10-17 and index registers 0-1. The conversion routines (F, P, E) also use index register 2. Therefore, the user should not depend on these locations having the same value on exit from a routine as they had on entry. Many routines require a pointer which always goes in base register 0. When doing input (or output) using the pointer, the characters are packed one ASCII character per word and the pointer points to the string of characters. The first word in this string is the count of the number of characters in the string. For example, to print on the Teletype

"FPP LIBRARY"

call the routine PUTSTR ("put string" which is described in Section 4.1) to put a string out on the Teletype which starts at the location to which base register 0 points, with the pointer in location 0. This code is:

	FLDA	MYP	/set up pointer to string
	FSTA	BASE0	/store in base register 0
	JSA	PUTSTR	/type it out now
	xxx		/more FPP code
MYP,	0		
	JA	OUTS	/pointer to output string
OUTS,	14		/12 ₁₀ = 14 ₈ characters long
	306		/F
	320		/P
	320		/P
	240		/space
	314		/L
	311		/I
	302		/B
	322		/R
	301		/A
	322		/R
	331		/Y
	256		/. "FPP LIBRARY."

Note that if the buffered I/O package is being used, the preceding example would probably return almost immediately, because the routines have approximately 80 characters of buffer for output. If the buffers are not full, then PUTSTR will call PUTC to store it in the buffer. This overlapped I/O generally greatly increases through-put.

4.1 Description of the Utility Routines

<u>Routine</u>	<u>Description</u>
CRLF	CRLF calls PUTSTR with a pointer set up to a 215,212 (a carriage return/line feed). Thus a JSA to CRLF produces a carriage return/line feed on the Teletype. No arguments are necessary.
GETNUM	This routine translates an input string of ASCII characters to a floating point number by calling GETC to get characters. It will continue to assemble characters as a floating point number until an illegal character is found. It then returns with the number in the FAC. The bad character is left as a floating point number in base register 17. For example, assume the user wants to get three floating point numbers from the Teletype. The following is acceptable:

```

JSA    GETNUM
FSTA  ONEN
JSA    GETNUM
FSTA  TWON
JSA    GETNUM
FSTA  THREEEN

```

Now consider the following Teletype input

```
1234,1.453,-.005E+09
```

The variable ONEN contains a 1234.; the variable TWON contains a 1.453; the variable THREEEN contains a -5000000. GETNUM ignores blanks (spaces). Note the effect of a misplaced + sign:

```
234 5.6E+00+9,,4.562,.2345
```

ONEN contains a 2345.6; TWON contains a 9 (because the second plus is illegal in that context); THREEEN contains a 0, because there are two illegal characters in a row (the two commas). The 4.562 and the .2345 are returned on subsequent calls to GETNUM.

PUTSTR	The put string routine types out a string of ASCII characters packed one per word on the Teletype by calling PUTC. The pointer is in base register 0. The first word pointed to by the pointer is the number of characters to print. When done, base register 0 is pointing to the location after the last character in the string. This is helpful for printing when using the conversion routine.
--------	---

4.2 Conversion Routines

Three sets of conversion routines are available to convert floating point numbers to ASCII; they are referred to as E, F, and P. E conversion is of the form Em.n, where the m is the total number of spaces and the n is the number of places after the decimal point. A number in E format has the syntax:

```
(m-n-8 spaces) (sign (blank or -)) (decimal point)
(n digits) (space) (E+xxx (where xxx is the exponent))
```

If insufficient space is provided (e.g., E10.13), asterisks are generated. To specify the values of m and n (here E15.8) use the following:

```
FLDA (8. /load second arg into FAC
FSTA BASE+30 /place in BASE register 10
FLDA (15. /get the total width
JSA ETYPE /call the conversion setup
```

The E conversion routine is now set up to use E15.8 as its standard format. Any conversions done will be in E15.8, but may be changed at any time by recalling ETYPE. There exists a routine for each conversion: FTYPE sets up the F conversion and PTYPE sets up the P conversion. At any time, the user may do a conversion by calling ECON (E conversion). Thus to convert 1/3, do the following:

```
FLDA (1./3.
JSA ECON /conversion is now done
```

The answer is left in core, packed one character to a word, starting at the location to which base register 0 pointed. Thus, if at some previous time a pointer to YYY was placed in base 0, YYY would contain the following after the conversion:

```
YYY, 17 /number of characters to follow (base 8)
256 /.
263 /3
263
263
263
263
263
263
263
264 /note slight error from rounding
240 /a space
305 /E
253 /+
260 /0
260 /0
260 /0
```


The FPP has approximately six digits of accuracy. Hence, as more digits are to be printed, some error will occur. In general, the extra digits will depend on the algorithm presently being used. The user should note that, in general, the seventh digit and those greater are not meaningful.

P conversion is similar to E conversion, except that the first digit is before the decimal point. This is referred to as scientific notation. Thus 1/3 printed in P format 2 \emptyset .1 \emptyset is

ss3.3333333333sE- $\emptyset\emptyset$ 1 (where s represents a space)

The call to convert a number for P conversion is PCON. The call for F conversion is FCON.

At the end of a conversion, base \emptyset is left pointing to the next character (i.e., one more than inserted).

Because base register \emptyset is always correctly updated, conversions can be performed and then printed without resetting base register \emptyset . The following example will convert and print three random numbers. Assume that FTYPE had been called to set up the F conversion for F13.7.

```

FLDA    MYPUNT    /set up correct pointer
FSTA    BASE $\emptyset$   /save in base  $\emptyset$ 
JSA     RAND      /generate a random number
JSA     FCON      /convert it
JSA     RAND      /second number
JSA     FCON
JSA     RAND
JSA     FCON
FLDA    MYPUNT    /reset the pointer again
FSTA    BASE $\emptyset$  /and store in  $\emptyset$ 
JSA     PUTSTR    /now output three strings
JSA     PUTSTR
JSA     PUTSTR

```

(Of course, the previous example could have been done in loop, but this expanded version better illustrates the calling method.)

5.0 MATHEMATICAL ROUTINES

The mathematical routines in the FPP Support Library perform commonly used math functions, such as sine and cosine of an angle, natural log, etc. These functions are, in general, independent of each other and the Nucleus and, therefore, may be used individually. Be sure to

check the description of the routine being used to ensure that the proper argument is given so that the correct answer will be returned.

When extracting a math routine to use it independently, note any routines that it calls or possible error traps that may result. There are no TRAP3 instructions in the mathematical package. However, there are TRAP7 instructions which are used to indicate error conditions for various functions. If the user is decoding the TRAP7 instructions, the operand address contains a number which corresponds to an error code. The user may then take appropriate action. He may set the FAC in the active parameter table so that the function will return with that value. If the user is not handling the TRAP7 instructions, the FAC will normally be a random value on exit from the function.

5.1 Implementing a Mathematical Routine

To use one of the mathematical functions in the "standard" manner, simply JSA to the function with the argument in the FAC. (The functions must be called in floating point mode [STARTF]). Additional arguments (if necessary) go in base register locations 10-17, respectively. For functions requiring a pointer, the pointer goes into location 0 on the base page. The mathematical functions are free to use base registers 0, 10-17, and index registers 0 and 1. Because of the modular nature of the FPP Support Library, new routines will be added and existing routines changed as improved and/or faster algorithms become available. Remember that the package uses the IFREF pseudo-op, so only those routines referenced will be assembled and loaded.

5.2 Program Library LINctapes

The FPP Mathematical Library supplied by the DEC Program Library resides in two files: FPPLIB and FPPLB1. The last statement in FPPLIB is CHAIN "FPPLB1"1 which causes an assembly of FPPLIB to chain to the rest of the source. Note that the FPP Support Library must be on unit 1 (or unit 11), unless the user changes the CHAIN command. In addition to these two files, there exist three more files called FPPLIBS, FPPLB1S, and FPPLB2S. These are similar to FPPLIB and FPPLB1, except that most of the symbols and comments have not been removed. The symbols were removed from FPPLIB and FPPLB1 to speed up assembly and to allow the user to have more symbols in his program.

5.3 Example

The following is a complete program to print the sines and cosines of angles from 1° to 360° in steps of 1° using the FPP Nucleus and the buffered I/O for output. All output conversion will be in the format F15.8.

NOTE

As presently implemented, the routines all use a set of common constants. By using these, much space is saved in duplicated literals. These constants are referred to as FPPCOM; if any routine is to be removed to be used stand-alone, check to see if it uses one of these constants. Also note that because these routines use the FPP, they are constrained by accuracy limitations. In addition, faulty arguments can cause underflow and overflow conditions to result, with resultant loss of accuracy. For the user's convenience, the symbol FPPNXT points to the first free location following the functions.

```

ORG      200      /start of my routine
JMS I   4        /call nucleus to clear all
11      /function 11
JMS I   4        /now set the starting PC
Ø      /
JA      START   /good address
JMS I   4        /now set the done return
6      /
ENDOFJ   /end of job exit
JMS I   4        /now start the FPP going
1Ø     /
JMS     Ø       /now wait for it to finish
JMP     .-1     /with an infinite loop
ENDOFJ, Ø       /entry point
HLT     /and halt when done

START,   SETX   14000 /set the index positions
        BASE   15000
        SETB   15000 /and the base positions
        STARTF /enter floating mode (good habit
                /in case of restart)
        FLDA   (8.   /set up the 8 for F15.8
        FSTA   15030 /save in base 1Ø
        FLDA   (15.  /get the 15
        JSA    FTYPE /set up F conversion
        JSA    CRLF  /give initial CR/LF
        LDS    -55Ø,7 /55Ø8=36Ø10
        LDX    Ø,6   /present 1Ø angle

LOOP,   ADDX   1,6   /increment angle
        XTA    6     /get angle in FAC
        JSA    MYPUT /send it out
        XTA    6     /recall it again
        JSA    MYPUT /print it
        XTA    6     /recall again

```

```

        JSA    COSD    /perform the cosine of angle
        JSA    MYPUT
        JSA    CRLF    /give final CR/LF
        JXN    LOOP,7+ /loop around until done
        JSA    IOOUTW  /wait for I/O to finish
        FEXIT
MYPUT,  JA      .      /entry for the little putter routine
        FLDA   150003  /save argument for a second
        FLDA   MYPST  /set up pointer for the convert
                        /routine to use
        FLDA   150000
        FLDA   150003  /recall the number
        JSA    FCON    /convert it
        FLDA   MYPST  /get pointer to where it was put
        FLDA   150000 /and reset it again
        JSA    PUTSTR  /output it to the Teletype
MYPST,  JA      MYPUT  /and return
        Ø      Ø      /necessary filler
        JA     177000  /working space for convert answer

        IO=BUFFERED  /tell it to load I/O routines
        FPPORG=.
        CHAIN "FPPLIB"1/and call in the library

```

5.4 Descriptions of the Mathematical Routines

A complete description of each of the FPP Support Library Mathematical routines, including implementation, follows.

- RAND** Random number generator. RAND does not require any arguments. The returning FAC contains a random number between 0.0 and 1.0.
- ABS** Absolute value function. This function returns with the absolute value of FAC in the FAC.
- SQRT** SQRT takes the square root of the FAC. The answer is returned in the FAC. If the FAC is minus on entry, a TRAP7 3 is issued. If no action is taken, the routine returns with the argument in the FAC.
- SIGN** SIGN returns with a -1, 0 or +1, depending on whether the calling FAC was minus, zero, or positive. This function is very useful for imparting the sign of one item to another item. If the user desires to have A have the sign of B, then the following code is sufficient:
- ```

 FLDA B
 JSA SIGN
 FMULM A

```
- AMOD** AMOD is a two argument function which returns the following in the FAC:  $FAC=A-INT(A/B)*B$ . Effectively this function returns A modulo B. The function is called with A in the FAC and B in base register 10. If it is impossible to integerize A/B, a TRAP7 11 is issued; if no action is taken, A/B is returned.

EXP3 EXP3 is used for raising a base to a power. This is commonly represented by  $A^{**}B$ , where A is the base and B is the power (e.g.,  $10^{**}3=1000$ ). EXP3 is called with the base in the FAC and the exponent in base register 10. The answer is returned in the FAC. This function uses ALOG and EXP and is, therefore, subject to the additional constraints imposed by these functions. Special cases:  $0^{**}X=0$ ,  $X^{**}0=1$ ,  $0^{**}0=0$ ; raising a negative number to a non-zero power will give a TRAP7 with an operand address of 0. If the FAC is not changed, AC will contain the negative base on exit following a TRAP7 0.

EXP This function raises E to the FAC. If the FAC is greater than  $2^{**}23$  (unable to fix it), a TRAP7 4 is issued. If no action is taken, this function returns with ABS(FAC). This function can get underflows and overflows for large values of the FAC, even though it can be fixed.

ALOG This routine takes the log of the FAC base E and returns with it in the FAC. If the FAC is less than zero, a TRAP7 6 is issued; if FAC=0, a TRAP7 7 is issued. In either case, if no action is taken, the argument is returned.

ALOG10 ALOG10 returns the log of the FAC base 10 in the FAC. The user must remember that ALOG returns the log of the FAC in base E, hence the reason for two log routines. This function calls ALOG.

SIN SIN returns with the sine of the FAC in the FAC. The FAC argument is in radians. If the absolute value of the FAC is greater than  $2^{**}23$  (unable to fix it), then a TRAP7 5 is issued. If no action is taken, the function returns with the ABS(FAC). This function will tend to produce better answers if the FAC is less than  $2\pi$  in magnitude.

SIND This routine takes the sine of the FAC. The FAC is in degrees as opposed to radians. This function calls SIN.

ASIN Arc-sine routine. This function returns with the arc-sine of the FAC in the FAC. Functions SQRT and ATAN are called. Answer is in radians. If the absolute value of the incoming argument is greater than 1, a TRAP7 2 is issued. If no action is taken, the function returns with ABS(FAC)-1 in the FAC.

SINH This routine returns with the hyperbolic sine of the FAC in the FAC. This routine calls EXP.

COS This routine takes the cosine of the FAC. The FAC is in radians. This routine calls SIN.

COSD COSD takes the cosine of the FAC. The FAC is in degrees as opposed to radians. This routine calls SIN.

ACOS Arc-cosine function. This function returns with the arc-cosine of the FAC in the FAC. Functions SQRT and ATAN are called. Answer is in radians. If the absolute value of the incoming argument is greater than 1, a TRAP7 1 is issued. If no action is taken, the function returns with ABS(FAC)-1 in the FAC.

COSH This routine returns with the hyperbolic cosine of the FAC in the FAC and calls EXP. If the ABS(FAC)-ALOT(2) is greater than 88.029, a TRAP7 12 is issued. If no action is taken, the largest possible number is returned.

TAN TAN takes the tangent of the FAC which is in radians. The routine calls SIN and COS. IF COS(FAC)=0, a TRAP7 10 is issued. If no action is taken the FAC=0 on return.

TAND TAND takes the tangent of the FAC which is in degrees. This routine calls TAN.

ATAN This function returns with the arc-tangent of the FAC in the FAC. The answer is in radians.

ATAN2 This function returns the arc-tangent of the FAC. The answer is returned in radians. This is a two argument function. The second argument (in base register 10) determines in which quadrant the user should be returned. Only the sign of the second argument is used. This function calls ATAN. The values of the answer can be summarized:

|     |                |       |            |
|-----|----------------|-------|------------|
| If: | $B > 0$        | Then: | ATAN(A)    |
|     | $B < 0, A < 0$ |       | ATAN(A)+PI |
|     | $B < 0, A > 0$ |       | ATAN(A)-PI |

TANH This routine returns the hyperbolic tangent of the FAC in the FAC. This routine calls SINH and COSH.

## 6.0 CUSTOMIZING THE LIBRARY

The information presented in this section is not necessary for using the FPP Support Library provided by the DEC Program Library; it is, however, useful for modifying the Support Library for individual requirements.

### 6.1 Modifying the Assembly Sequence

The FPP Support Library is provided in two separate files: the Nucleus (and buffered I/P) in file FPPLIB, and the Utility and Mathematical routines in file FPPLB1. It is possible to use one of these files

without the other. For example, the LDP system processes users' interrupts so that it is somewhat incompatible with the FPP Nucleus. Because the Utility and Math routines have only FPP code in them, they may be used with the LDP system. The Nucleus and buffered I/O need not be loaded. To accomplish this, only the statement

```
CHAIN "FPPLIB"1
```

need be changed to

```
CHAIN "FPPLB1"1
```

which will cause assembly to continue at FPPLB1 rather than at FPPLIB, which contains only the Nucleus and buffered I/O files. Note that some of the Utility routines call PUTC or GETC. If the Utility routines are being used without the Nucleus, the programmer must insert his own PUTC or GETC routines in their place. Also, some functions have error exits which the user must handle if a TRAP7 X instruction occurs. No action need be taken on it, but he must be prepared to restart the FPP. The TRAP7 error exits are:

| <u>Trap #</u> | <u>Cause</u>                                         |
|---------------|------------------------------------------------------|
| 0             | EXP3 - $A * B$ , where $A \neq 0.0$ and $B \neq 0.0$ |
| 1             | ACOS - $ABS(FAC) > 1.0$                              |
| 2             | ASIN - $ABS(FAC) > 1.0$                              |
| 3             | SQRT - $FAC < 0.0$                                   |
| 4             | EXP - FAC cannot be integerized                      |
| 5             | SIN - FAC divided by $2\pi$ cannot be integerized    |
| 6             | ALOG - $FAC < 0.0$                                   |
| 7             | ALOG - $FAC = 0.0$                                   |
| 10            | TAN - $FAC = \pi/2$ (90 degrees in radians)          |
| 11            | AMOD - $ABS(arg1/arg2)$ cannot be integerized        |
| 12            | COSH - $ABS(FAC) - ALOG(2) > 88.029$                 |

## 6.2 Library Constants

The FPP Support Library uses several constants which are always present and which the programmer may reference. It is good practice to use these constants in order to avoid wasting core space. The constants are:

| <u>Constant</u> | <u>Contents</u>    |
|-----------------|--------------------|
| FPPONE          | 1.0                |
| FPPTWO          | 2.0                |
| FPPPI           | $\pi$ (3.14159265) |
| FPP2PI          | $2\pi$             |
| FPPPI2          | $\pi/2$            |

They exist in the definition section, called FPPCOM; their ordering must not be changed nor can any of them be removed because they are not always referenced by name.





|                 |   |           |                                    |
|-----------------|---|-----------|------------------------------------|
| 0023            |   | EJECT     |                                    |
| 0024            | / |           |                                    |
| 0025            | / |           |                                    |
| 0026            | / |           |                                    |
| 0027            | / |           |                                    |
| 0030            | / |           |                                    |
| 0031            |   | FPPOLD#.  | /SAVE THE CURRENT LOCATIONS AS W   |
| 0032            |   |           | /TEMPORARILY GOING TO CHANGE THE   |
| 0033            | / |           |                                    |
| 0034            | / |           |                                    |
| 0035            | / |           |                                    |
| 0036            |   | ORG 0     | /SET UP COMMUNICATIONS POINTERS.   |
| 0037            | / | 0         |                                    |
| 0040000000 0000 |   | 0         | /FOR INTERRUPTS.                   |
| 0041 00001 5402 |   | JMP I .+1 | /ON INTERRUPT (OR PHONEY INTERRUPT |
| 004200002 0400  |   | FPPINT    | /GO AND SEE WHATS HAPPENING.       |
| 0043 00003 0346 |   | FPPXXX    | /USERS ADDITIONAL VECTORS.         |
| 004400004 0200  |   | FPPCTL    | /POINTER TO FPP SUPPORT OVERLORD   |
| 0045 00005 0353 |   | FPPTAB    | /POINTER TO FPP PARAMETER TABLE.   |
| 0046            | / |           |                                    |
| 0047            | / |           |                                    |
| 0050            | / |           |                                    |
| 0051            | / |           |                                    |

|      |    |       |                                         |
|------|----|-------|-----------------------------------------|
| 0052 |    | EJECT |                                         |
| 0053 | /  |       |                                         |
| 0054 | // |       |                                         |
| 0055 | /  |       |                                         |
| 0056 |    | ORG   | 200 /ORIGIN AT LOCATION 200             |
| 0057 | /  |       |                                         |
| 0058 | // |       |                                         |
| 0059 | /  |       |                                         |
| 0060 |    | IFREF | FPPORG /IF FPPORG IS DEFINED, ORIGIN IT |
| 0061 | // |       |                                         |
| 0062 | /  |       |                                         |
| 0063 | // |       |                                         |
| 0064 | /  |       |                                         |
| 0065 |    | ORG   | FPPORG                                  |
| 0066 | /  |       |                                         |
| 0067 | // |       |                                         |
| 0070 | /  |       |                                         |
| 0071 |    | \$    |                                         |
| 0072 | /  |       |                                         |
| 0073 | // |       |                                         |
| 0074 | /  |       |                                         |
| 0075 | // |       |                                         |
| 0076 | /  | PAGE  | /MUST BEGIN ON AN EVEN PAGE BOUN        |
| 0077 | // |       |                                         |
| 0100 | /  |       |                                         |
| 0101 | // |       |                                         |

|      |       |      |               |          |                                  |
|------|-------|------|---------------|----------|----------------------------------|
| 0102 |       |      | EJECT         |          |                                  |
| 0103 |       |      | /             |          |                                  |
| 0104 |       |      | /             |          |                                  |
| 0105 |       |      | /             |          |                                  |
| 0106 | 00200 | 0000 | FPPCTL, 0     |          | /FPP OVERLORD COMMAND DISPATCHER |
| 0107 | 00201 | 7200 | CLA           |          | /CLEAR THE AC TO BE SURE.        |
| 0110 | 00202 | 1600 | TAD I         | FPPCTL   | /PICK UP FUNCTION.               |
| 0111 | 00203 | 2200 | ISZ           | FPPCTL   | /SKIP PAST FUNCTION.             |
| 0112 | 00204 | 0377 | AND           | (17      | /ONLY USE BITS 8=11 TO GET FUNCT |
| 0113 | 00205 | 1210 | TAD           | FPPJMP   | /CREATE THE CORRECT JUMP INSTRU  |
| 0114 | 00206 | 3207 | DCA           | .+1      | /SAVE AWAY.                      |
| 0115 | 00207 | 0000 | 0             |          | /AND NOW DISPATCH.               |
| 0116 |       |      | /             |          |                                  |
| 0117 | 00210 | 5611 | FPPJMP, JMP I | .+1      | /PROTOTYPE INSTRUCTION.          |
| 0120 |       |      | /             |          |                                  |
| 0121 | 00211 | 0233 | FPPPCS        |          | /0=SET FPP P.C. IN PARAMETER TAB |
| 0122 | 00212 | 0245 | FPPDV0        |          | /1=SET DIVIDE BY 0 EXIT.         |
| 0123 | 00213 | 0247 | FPPOVR        |          | /2=SET EXPONENT OVERFLOW EXIT.   |
| 0124 | 00214 | 0251 | FPPUND        |          | /3=SET EXPONENT UNDERFLOW EXIT.  |
| 0125 | 00215 | 0253 | FPPFRO        |          | /4=SET FRACTION OVERFLOW EXIT.   |
| 0126 | 00216 | 0255 | FPPTRP        |          | /5=SET INSTRUCTION TRAP EXIT [E  |
| 0127 | 00217 | 0257 | FPPEXT        |          | /6=SET FPP EXIT EXIT (THE COMMON |
| 0130 | 00220 | 0261 | FPPIOT        |          | /7=SET FPP IOT EXIT.             |
| 0131 | 00221 | 0263 | FPPSRT        |          | /10=START FPP RUNNING.           |
| 0132 | 00222 | 0301 | FPPCLR        |          | /11=CLEAR FPP AND OWN I=0 SYSTEM |
| 0133 | 00223 | 0312 | FPPTY         |          | /12=SET FPP CHARACTER RECEIVED E |
| 0134 | 00224 | 0314 | FPPFAL        |          | /13=SET FPP FAIL TO START EXIT.  |
| 0135 | 00225 | 0231 | FPPKNO        |          | /14=17=NOT USED.                 |
| 0136 | 00226 | 0231 | FPPKNO        |          |                                  |
| 0137 | 00227 | 0231 | FPPKNO        |          |                                  |
| 0140 | 00230 | 0231 | FPPKNO        |          |                                  |
| 0141 |       |      | /             |          |                                  |
| 0142 |       |      | /             |          |                                  |
| 0143 |       |      | /             |          |                                  |
| 0144 |       |      | /             |          |                                  |
| 0145 |       |      | /             |          |                                  |
| 0146 | 00231 | 7200 | FPPKNO, CLA   |          | /CLA NECESSARY BECUASE FPPSRT NE |
| 0147 | 00232 | 5600 | JMP I         | FPPCTL   | /EXIT IF NOT A LEGAL COMMAND.    |
| 0150 |       |      | /             |          |                                  |
| 0151 |       |      | /             |          |                                  |
| 0152 |       |      | /             |          |                                  |
| 0153 |       |      | /             |          |                                  |
| 0154 |       |      | /             |          |                                  |
| 0155 | 00233 | 1354 | FPPPCS, TAD   | FPPTAB+1 | /SET FPP P.C.                    |
| 0156 | 00234 | 0376 | AND           | (7770    | /ZERO OUT HIGH ORDER BITS OF PC  |
| 0157 | 00235 | 3354 | DCA           | FPPTAB+1 |                                  |
| 0160 | 00236 | 4324 | JMS           | FPPARG   | /GET FIRST ARGUMENT OF CALL.     |
| 0161 | 00237 | 0375 | AND           | (7       | /REMOVE EXTRANEIOUS GARBAGE.     |
| 0162 | 00240 | 1354 | TAD           | FPPTAB+1 | /AND ADD IN OTHER FACTS.         |
| 0163 | 00241 | 3354 | DCA           | FPPTAB+1 | /AND STORE BACK.                 |
| 0164 | 00242 | 4324 | JMS           | FPPARG   | /NOW GET LOW ORDER BITS OF P.C.  |
| 0165 | 00243 | 3355 | DCA           | FPPTAB+2 | /AND STORE AWAY.                 |
| 0166 | 00244 | 5600 | JMP I         | FPPCTL   | /AND RETURN NOW.                 |
| 0167 |       |      | /             |          |                                  |
| 0170 |       |      | /             |          |                                  |
| 0171 |       |      | /             |          |                                  |
| 0172 | 00245 | 4316 | FPPDV0, JMS   | FPPRET   | /SET UP DIVIDE BY 0 RETURN.      |
| 0173 | 00246 | 0463 | FPPDER        |          |                                  |
| 0174 |       |      | /             |          |                                  |
| 0175 | 00247 | 4316 | FPPOVR, JMS   | FPPRET   | /AND OVERFLOW RETURN.            |
| 0176 | 00250 | 0533 | FPPOER        |          |                                  |

|      |       |      |         |          |           |                                  |
|------|-------|------|---------|----------|-----------|----------------------------------|
| 0177 |       |      | /       |          |           |                                  |
| 0200 | 00251 | 4316 | FPPUND, | JMS      | FPPRET    | /AND UNDERFLOW RETURN.           |
| 0201 | 00252 | 0440 |         | FPPUER   |           |                                  |
| 0202 |       |      | /       |          |           |                                  |
| 0203 | 00253 | 4316 | FPPFRO, | JMS      | FPPRET    | /AND FRACTIONAL OVERFLOW RETURN. |
| 0204 | 00254 | 0447 |         | FPPFER   |           |                                  |
| 0205 |       |      | /       |          |           |                                  |
| 0206 | 00255 | 4316 | FPPTRP, | JMS      | FPPRET    | /AND TRAP RETURN.                |
| 0207 | 00256 | 0531 |         | FPPTER   |           |                                  |
| 0210 |       |      | /       |          |           |                                  |
| 0211 | 00257 | 4316 | FPPEXT, | JMS      | FPPRET    | /AND FEXIT RETURN.               |
| 0212 | 00260 | 0430 |         | FPPEER   |           |                                  |
| 0213 |       |      | /       |          |           |                                  |
| 0214 | 00261 | 4316 | FPPLOT, | JMS      | FPPRET    | /AND IOT EXIT RETURN.            |
| 0215 | 00262 | 0465 |         | FPPIER   |           |                                  |
| 0216 |       |      | /       |          |           |                                  |
| 0217 |       |      | /       |          |           |                                  |
| 0220 | 00263 | 1353 | FPPSRT, | TAD      | FPPTAB    | /PREPARE TO START FPP RUNNING.   |
| 0221 | 00264 | 6553 |         | FPCCOM   |           | /LOAD COMMAND REGISTER NOW.      |
| 0222 | 00265 | 7200 |         | CLA      |           | /CLEAR AC BECAUSE OF PEEBLE CONT |
| 0223 | 00266 | 1374 |         | TAD      | (FPPTAB+1 | /GET ADDRESS OF ACTIVE PARAMETER |
| 0224 | 00267 | 6555 |         | FPST     |           | /START FPP RUNNING.              |
| 0225 | 00270 | 7610 |         | SKP CLA  |           | /IT DIDN T START. CLEAR AC BECAU |
| 0226 | 00271 | 5231 |         | JMP      | FPPKNO    | /IT DID. EXIT THIS WAY BECAUSE O |
| 0227 | 00272 | 4675 |         | JMS I    | FPPLER    | /FPP FAILED TO START. TAKE FAILI |
| 0230 | 00273 | 7200 |         | CLA      |           | /HE RETURNED. CLEAR AC TO BE SUR |
| 0231 | 00274 | 5263 |         | JMP      | FPPSRT    | /AND RETRY NOW.                  |
| 0232 |       |      | /       |          |           |                                  |
| 0233 |       |      | /       |          |           |                                  |
| 0234 |       |      | /       |          |           |                                  |
| 0235 | 00275 | 0276 | FPPLER, | .+1      |           | /INITIALLY POINTS TO DUMMY ROUTI |
| 0236 |       |      | /       |          |           |                                  |
| 0237 | 00270 | 0000 |         | 0        |           |                                  |
| 0240 | 00277 | 6552 |         | FPICL    |           | /DO A CLEAR ON THE FPP. THEN RET |
| 0241 | 00300 | 5676 |         | JMP I    | FPPLER+1  | /BY RETURNING.                   |
| 0242 |       |      | /       |          |           |                                  |
| 0243 |       |      | /       |          |           |                                  |
| 0244 | 00301 | 6552 | FPPCLR, | FPICL    |           | /A CLEAR COMMAND. CLEAR FPP FIRS |
| 0245 | 00302 | 1373 |         | TAD      | (2400     |                                  |
| 0246 | 00303 | 3353 |         | DCA      | FPPTAB    | /RESET STATUS REGISTER.          |
| 0247 | 00304 | 4700 |         | JMS I    | FPPO00    | /MJOR BUFFER RESETER.            |
| 0250 | 00305 | 5600 |         | JMP I    | FPPCTL    | /AND RETURN NOW.                 |
| 0251 |       |      | /       |          |           |                                  |
| 0252 | 00300 | 0310 | FPPO00, | .+2      |           | /DUMMY POINTER TO NOTHING ROUTIN |
| 0253 | 00307 | 0310 |         | .+1      |           | /I=0 IS NOT LOADED.              |
| 0254 |       |      | /       |          |           |                                  |
| 0255 | 00310 | 0000 |         | 0        |           |                                  |
| 0256 | 00311 | 5710 |         | JMP I    | .-1       | / REAL DO NOTHING ROUTINE.       |
| 0257 |       |      | /       |          |           |                                  |
| 0260 |       |      | /       |          |           |                                  |
| 0261 |       |      | /       |          |           |                                  |
| 0262 |       |      | /       |          |           |                                  |
| 0263 | 00312 | 4316 | FPPTYT, | JMS      | FPPRET    | /ESTABLISH TTY CHARACTER RETURN. |
| 0264 | 00313 | 0307 |         | FPPO00+1 |           | /WHERE TO PUT TTY POINTER.       |
| 0265 |       |      | /       |          |           |                                  |
| 0266 |       |      | /       |          |           |                                  |
| 0267 | 00314 | 4316 | FPPFAL, | JMS      | FPPRET    | /FAIL TO START POINTER           |
| 0270 | 00315 | 0275 |         | FPPLER   |           |                                  |
| 0271 |       |      | /       |          |           |                                  |
| 0272 |       |      | /       |          |           |                                  |
| 0273 | 00316 | 0000 | FPPRET, | 0        |           | /ROUTINE WHICH SETS UP POINTERS  |
| 0274 | 00317 | 1716 |         | TAD I    | FPPRET    | /GET WHERE TO STICK POINTER.     |

```

0361 EJECT
0362 /
0363 /
0364 /
0365 / MAJOR FIXED LOCATIONS WITH RESPECT TO "FPPTAB"
0366 /
0367 /
0370 /
0371 00347 0000 FPPMG, 0 /HOLDS MQ DURING INTERUPT.
0372 00350 0000 FPPLNC, 0 /HOLDS LINC DURING INTERUPT.
0373 00351 0000 FPPAC, 0 /HOLDS AC DURING INTERUPT.
0374 /
0375 00352 0000 0 /STATUS OF FPP AFTER AN INTERUPT.
0376 /
0377 00353 2400 FPPTAB, 2400 /FPP CPMMAND REGISTER.
0400 00354 0000 0 /FPP EXTENED BITS REGISTER.
0401 00355 0000 0 /FPP P.C. LOW ORDER REGISTER.
0402 00356 0000 0 /LOW ORDER BITS OF LOCATION OF INDEX 0.
0403 00357 0000 0 /LOW ORDER BITS OF BASE REGISTER.
0404 00360 0000 0 /LOW ORDER BITS OF OPERAND.
0405 00361 0000 0 /EXPONENT OF AC.
0406 00362 0000 0 /HIGH WORD OF AC.
0407 00363 0000 0 /LOW WORD OF AC.
0410 /
0411 /
0412 /
0413 /
0414 /
0415 /
 00373 2400
 00374 0354
 00375 0007
 00376 7770
 00377 0017

0416 PAGE
0417 /
0420 /

```

```

0316 EJECT
0317 /
0320 /
0321 /
0322 /
0323 /
0324 00330 0000 FPPDUT, 0 /ROUTINE TO DISPATCH TO CORRECT
0325 00331 7200 CLA /CLEAR THE AC TO BE SURE.
0326
0327 00332 1730 TAD I FPPDUT /GET THE POINTER.
0330 00333 3330 DCA FPPDUT /AND SAVE AWAY.
0331 00334 4730 JMS I FPPDUT /CALL THE CORRECT ROUTINE NOW.
0332 /
0333 /
0334 00335 7300 FPPDUT, CLA CLL /EXIT FROM INTERUPT ROUTINE. AC
0335 00336 1003 TAD 3 /GET USERS INTERUPT VECTOR.
0336 00337 3330 DCA FPPDUT /SAVE FOR A SECOND.
0337 00340 1347 TAD FPPMQ /RESOTRE THE MQ.
0340 00341 7421 7421 /MQL, LOAD MQ FROM AC. CLEAR AC.
0341 00342 1350 TAD FPPDUT /GET THE LINC.
0342 00343 7104 CLL RAL /AND RELOAD IT.
0343 00344 1351 TAD FPPAC /FINALLY RELOAD THE AC.
0344 00345 5730 JMP I FPPDUT /AND EXIT TO WHERE EVER WE HAVE
0345 /
0346 /
0347 /
0350 /
0351 /
0352 00346 5400 FPPXXX, JMP I 0 /PHONEY EXIT INCASE USER WANTS T
0353 /
0354 /
0355 /
0356 /
0357 /
0360 /

```

|      |       |      |         |        |                               |
|------|-------|------|---------|--------|-------------------------------|
| 0275 | 00320 | 3316 | DCA     | FPPRET | /AND SAVE                     |
| 0276 | 00321 | 4324 | JMS     | FPPARG | /GET THE USERS POINTER NOW.   |
| 0277 | 00322 | 3716 | DCA I   | FPPRET | /AND STASH IT AWAY CORRECTLY. |
| 0300 | 00323 | 5600 | JMP I   | FPPCTL | /AND RETURN TO THE USER.      |
| 0301 |       |      | /       |        |                               |
| 0302 |       |      | /       |        |                               |
| 0303 | 00324 | 0000 | FPPARG, | 0      | /GETS A USERS ARGUMENT.       |
| 0304 | 00325 | 1600 | TAD I   | FPPCTL | /GET HIS ARGUMENT             |
| 0305 | 00326 | 2200 | ISZ     | FPPCTL | /AND BOP UP THE POINTER.      |
| 0306 | 00327 | 5724 | JMP I   | FPPARG | /AND RETURN TO THE USER NOW.  |
| 0307 |       |      | /       |        |                               |
| 0310 |       |      | /       |        |                               |
| 0311 |       |      | /       |        |                               |
| 0312 |       |      | /       |        |                               |
| 0313 |       |      | /       |        |                               |
| 0314 |       |      | /       |        |                               |
| 0315 |       |      | /       |        |                               |



```

0421 EJECT
0422 /
0423 /
0424 /
0425 /
0426 00400 3777 FPPINT, DCA I (FPPAC /FPP INTERRUPT PRCESSOR, SAVE AC.
0427 00401 7010 RAR /AND LINC
0430 00402 3776 DCA I (FPPLNC /MGA, GET THE MQ INTO THE AC FOR
0431 00403 7701 7701 /AND SAVE THE MQ.
0432 00404 3775 DCA I (FPPMQ /CHECK FOR OTHER SUPPORT PACKAGE
0433 00405 4740 JMS I FPPITS /IS IT THE FPP?
0434 00406 6557 FPIST /NOPE, CONTINUE.
0435 00407 5774 JMP I (FPPOUT /YEP, SAVE THE STATUS NOW.
0436 00410 3773 DCA I (FPPTAB+1 /TURN ON THE DOUBLE MODE BIT.
0437 00411 7330 CLA CLL CML RAR /AND LEAVE IT ON IF IN DOULB E M
0440 00412 0773 AND I (FPPTAB-1 /PUT ON CORRECT COMMAND MODE BIT
0441 00413 1372 TAD (2400 /AND STASH AWAY.
0442 00414 3771 DCA I (FPPTAB /RETRIEVE STATUS AND TEST.
0443 00415 1773 TAD I (FPPTAB-1 /ROTATE TO LINC AND SIGN.
0444 00416 7006 RTL /TEST FOR INSTRUCTION TRAP
0445 00417 7430 SZL /ITS AN INSTRUCTION TRAP.
0446 00420 5272 JMP FPITRP /TEST FOR IOT EXIT.
0447 00421 7510 SPA /ITS AN IOT EXIT.
0450 00422 5264 JMP FPIOTT /ROTATE NEXT TWO BITS OVER.
0451 00423 7006 RTL /TEST FOR DIVIDE BY 0.
0452 00424 7430 SZL /ITS A DIVIDE BY 0.
0453 00425 5262 JMP FPDIV0 /TEST FOR FRACTION OVERFLOW.
0454 00426 7510 SPA /ITS A FRACTION OVERFLOW.
0455 00427 5246 JMP FPFRCO /GET THE NEXT TWO BITS.
0456 00430 7006 RTL /TEST FOR EXPONENT OVERFLOW
0457 00431 7430 SZL /ITS AN EXPONENT OVERFLOW
0460 00432 5332 JMP FPEXOV /TEST FOR UNDERFLOW.
0461 00433 7710 SPA CLA /ITS AN EXPONENT UNDERFLOW.
0462 00434 5237 JMP FPEXUD
0463 /
0464 00435 4770 JMS I (FPPDUT /ITS NONE OF THESE, THEN ITS AN
0465 00436 0310 FPPEER, FPP000+2 /ON EXIT DO NOTHING, JUST RETURN
0466 /
0467 /
0470 /
0471 00437 4770 FPEXUD, JMS I (FPPDUT /UNDERFLOW EXIT.
0472 00440 0441 FPPUER, .+1 /OVERLAYED BY COMMAND DECODER IF
0473 /
0474 00441 0000 0 /ROUTINE TO FIX UP UNDERFLOWS.
0475 00442 3767 DCA I (FPPTAB+6 /ZERO OUT THE FPP FAC.
0476 00443 3766 DCA I (FPPTAB+7
0477 00444 3765 DCA I (FPPTAB+10
0500 00445 5325 JMP FPPIST /AND NOW START FPP GOING AGAIN.
0501 /
0502 /
0503 00446 4770 FPFRCO, JMS I (FPPDUT /FRACTIONAL OVERFLOW PROCESSOR.
0504 00447 0450 FPPFER, .+1 /NO OVERLAYED RETURN(AS USUAL).
0505 /
0506 00450 0000 0 /FRACTIONAL OVERFLOW FIXER-UPPER
0507 00451 7330 CLA CLL CML RAR /TURN ON HIGH ORDER BIT OF AC IS
0510 00452 0766 AND I (FPPTAB+7 /LEAVE ON IF AC WAS NEGATIVE.
0511 00453 7500 SMA /WAS IT ON?
0512 00454 7350 CLA CLL CMA RAR /NOPE, SET TO A BIG NUMBER,
0513 00455 3766 DCA I (FPPTAB+7 /AND STORE AWAY.
0514 00456 7430 SZL /IF THE LINC IS ZERO IT SHOULD B
0515 00457 7040 CMA /IF ONE IT IS POSITIVE.

```

```

0516 00460 3765 DCA I (FPPTAB+10 /AND NOW SAVE AWAY AS LOW ORDER
0517 00461 5325 JMP FPPIST /AND THEN START FPP GOING AGAIN.
0520 /
0521 /
0522 /
0523 00462 4770 FPDIV0, JMS I (FPPDUT /A DIVIDE BY 0.
0524 00463 0534 FPPDER, FPPDER+1 /SET TO BIGGEST NUMBER [SAME AS
0525 /
0526 /
0527 /
0530 /
0531 /
0532 /
0533 00464 4770 FPIOTT, JMS I (FPPDUT /ITS AN IOT EXIT.
0534 00465 0466 FPPIER, .+1 /IOT EXIT POINTER. INITIALLY POI
0535 /
0536 00466 0000 0 /THIS ROUTINE JUST RESTARTS THE
0537 00467 4404 JMS I 4 /CALL THE COMMAND DECODER.
0540 00470 0010 10 /START CODE.
0541 00471 5666 JMP I FPPIER+1 /AND RETURN NOW.
0542 /
0543 /
0544 /
0545 /
0546 00472 7344 FPITRP, CLA CLL CMA RAL /ITS A TRAP. BACK UP PC NOW.
0547 00473 1764 TAD I (FPPTAB+2 /GET THE PC.
0550 00474 3266 DCA FPPIER+1 /SAVE FOR A SECOND.
0551 00475 7430 SZL /DID IT CHANGE FIELDS.
0552 00476 7040 CMA /YES. BACK UP THE FIELD ALSO.
0553 00477 1763 TAD I (FPPTAB+1 /GET THE FIELD BITS.
0554 00500 7006 RTL /
0555 00501 7004 RAL /ROTATE INTO A DESIRABLE POSITON
0556 00502 0362 AND (70 /CHOP OFF CRAP.
0557 00503 1307 TAD FPPCDF /CREATE THE CORRECT CDF INSTRUCT
0560 00504 3305 DCA .+1 /STORE AWAY, BUBBY.
0561 00505 0000 0 /
0562 00506 1666 TAD I FPPIER+1 /GET THE FPP TRAP INSTRUCTION.
0563 00507 6201 FPPCDF, CDF 0 /RESET TO THIS FIELD.
0564 00510 7710 SPA CLA /IF ITS OURS [3XXX] THEN AC IS P
0565 00511 5330 JMP FPPTTT /NOPE. LET USER TAKE CARE OF IT.
0566 00512 1763 TAD I (FPPTAB+1 /GET THE OPERAND FIELD BITS.
0567 00513 7012 RTR /
0570 00514 7012 RTR /
0571 00515 7012 RTR /
0572 00516 0362 AND (70 /ROATE TO A GOOD PLACE AND CHOP
0573 00517 1361 TAD (6203 /MAKE IT A GOOD CIF=CDF
0574 00520 3322 DCA .+2 /SAVE AWAY.
0575 00521 1760 TAD I (FPPTAB+5 /GET OPERAND ADDRESS.
0576 00522 0000 0 /SET TO CORRECT FIELD.
0577 00523 3266 DCA FPPIER+1 /SAVE AWAY ADDRESS.
0600 00524 4666 JMS I FPPIER+1 /AND GO TO IT.
0601 /
0602 00525 7200 FPPIST, CLA /CLEAR AC ON RETURN TO BE SURE.
0603 00526 4266 JMS FPPIER+1 /RESTART FPP.
0604 00527 5774 JMP I (FPPDUT /AND EXIT.
0605 /
0606 /
0607 /
0610 00530 4770 FPPTTT, JMS I (FPPDUT /USER TRAP.
0611 00531 0466 FPPTER, FPPIER+1 /ON TRAP JUST RESTART IT.
0612 /
0613 /

```

```

0614 /
0615 /
0616 00532 4770 FPEXOV, JMS I (FPPDUT /EXPONENT OVERFLOW.
0617 00533 0534 FPPOER, .+1 /SYSTEM FIXER=UPPER
0620 /
0621 00534 0000 0 /COME HERE IF HE WANTS US TO MAN
0622 00535 7350 CLA CLL CMA RAR /SET THE EXPONENT
0623 00536 3767 DCA I (FPPTAB+6 /TO A MAXIMUM
0624 00537 4250 JMS FPPFER+1 /AND THEN FAKE OUT FRACTIONAL OV
0625 /
0626 /
0627 /
0630 /
0631 /
0632 /
0633 00540 0310 FPPITS, FPP000+2 /PHONEY TTY CHECKER UNTIL OVERLA
0634 /
0635 /
0636 /
0637 /
0640 /
0641 /
0642 /
0643 /

```

0644  
0645  
0646  
0647  
0650  
0651  
0652  
0653  
0654  
0655  
0656  
0657

EJECT

00560 0360  
00561 6203  
00562 0070  
00563 0354  
00564 0355  
00565 0363  
00566 0362  
00567 0361  
00570 0330  
00571 0353  
00572 2400  
00573 0352  
00574 0335  
00575 0347  
00576 0350  
00577 0351

0660  
0661  
0662  
0663  
0664  
0665  
0666  
0667  
0670  
0671  
0672  
0673  
0674  
0675  
0676  
0677  
0700  
0701  
0702  
0703  
0704  
0705  
0706  
0707  
0710  
0711  
0712  
0713  
0714  
0715  
0716  
0717  
0720

PAGE

IFREF BUFFER

B U F F E R E D I O R O U T I N  
- - - - -

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSETTS.  
01754

WRITTEN BY JACK BURNES.

|      |   |                                           |                                  |
|------|---|-------------------------------------------|----------------------------------|
| 0721 | / |                                           |                                  |
| 0722 | / |                                           |                                  |
| 0723 | / |                                           |                                  |
| 0724 | / |                                           |                                  |
| 0725 | / |                                           |                                  |
| 0726 | / | EJECT                                     |                                  |
| 0727 | / |                                           |                                  |
| 0730 | / |                                           |                                  |
| 0731 | / |                                           |                                  |
| 0732 | / |                                           |                                  |
| 0733 | / | FPP INPUT AND OUTPUT BUFFERRING ROUTINES. |                                  |
| 0734 | / |                                           |                                  |
| 0735 | / |                                           |                                  |
| 0736 | / | COPYRIGHT 1971                            |                                  |
| 0737 | / | DIGITAL EQUIPMENT CORPORATION             |                                  |
| 0740 | / | MAYNARD, MASS.                            |                                  |
| 0741 | / |                                           | 01754                            |
| 0742 | / |                                           |                                  |
| 0743 | / |                                           |                                  |
| 0744 | / | WRITTEN BY JACK BURNES.                   |                                  |
| 0745 | / |                                           |                                  |
| 0746 | / |                                           |                                  |
| 0747 | / |                                           |                                  |
| 0750 | / |                                           |                                  |
| 0751 | / | EJECT                                     |                                  |
| 0752 | / |                                           |                                  |
| 0753 | / |                                           |                                  |
| 0754 | / |                                           |                                  |
| 0755 | / |                                           |                                  |
| 0756 | / | PAGE                                      | /ORIGIN PAST PREVIOUS LITTERALS. |
| 0757 | / |                                           |                                  |
| 0760 | / |                                           |                                  |
| 0761 | / |                                           |                                  |
| 0762 | / |                                           |                                  |
| 0763 | / |                                           |                                  |
| 0764 | / |                                           |                                  |
| 0765 | / |                                           |                                  |
| 0766 | / |                                           |                                  |
| 0767 | / |                                           |                                  |
| 0770 | / |                                           |                                  |
| 0771 | / | BUFFER=.                                  | /SAVE ORIGIN FOR LATER.          |
| 0772 | / |                                           |                                  |
| 0773 | / |                                           |                                  |
| 0774 | / |                                           |                                  |
| 0775 | / | ORG FPP000                                |                                  |
| 0776 | / |                                           |                                  |
| 0777 | / | GTREST                                    | /ADDRESS OF CLEAR                |
| 1000 | / |                                           |                                  |
| 1001 | / |                                           |                                  |
| 1002 | / | ORG FPPITS                                | /OVER ITERUPT VECTOR.            |
| 1003 | / |                                           |                                  |
| 1004 | / | GTGTGT                                    | /TO INTERUPT ROUTINE             |
| 1005 | / |                                           |                                  |
| 1006 | / |                                           |                                  |
| 1007 | / |                                           |                                  |
| 1010 | / |                                           |                                  |
| 1011 | / | ORG BUFFER                                | /RESOTRE OLD ORIGIN.             |
| 1012 | / |                                           |                                  |
| 1013 | / |                                           |                                  |
| 1014 | / |                                           |                                  |
| 1015 | / |                                           |                                  |
| 1016 | / |                                           |                                  |

|      |                       |         |                                   |
|------|-----------------------|---------|-----------------------------------|
| 1017 | GETCCC, 0             |         | /AFTER TRAP COMES HERE.           |
| 1020 | TAD I                 | GETCPP  | /GET A CHARACTER.                 |
| 1021 | DCA I                 | GETFPP  | /STORE IN THE FPP AC.             |
| 1022 | CLA CMA               |         | /DECREMENT THE CHARACTER COUNT.   |
| 1023 | TAD                   | GTCUNT  |                                   |
| 1024 | DCA                   | GTCUNT  |                                   |
| 1025 | TAD I                 | GETCPP  | /GET THE LAST CHARACTER.          |
| 1026 | TAD I                 | (GTM215 | /WAS IT A CARRIAGE RETURN.        |
| 1027 | SZA CLA               |         |                                   |
| 1030 | JMP                   | +4      | /NO. NOTHING SPECIAL.             |
| 1031 | CLA CMA               |         | /YES. DECREMENT NUMBER            |
| 1032 | TAD                   | GTLN    | /OF CARRIAGE RETURNS IN BUFFER.   |
| 1033 | DCA                   | GTLN    |                                   |
| 1034 | JMS I                 | (GPPBOP | /NOW BOP UP CHARACTER POINTER.    |
| 1035 | GETCPP, GPLOW         |         | /POINTER TO NEXT CHARACTER TO GE  |
| 1036 | JMP I                 | GETCCC  | /AND RETURN TO RESTART FPP.       |
| 1037 | /                     |         |                                   |
| 1040 | /                     |         |                                   |
| 1041 | /                     |         |                                   |
| 1042 | /                     |         |                                   |
| 1043 |                       | 27      | /EXPONENT FOR FIXED GTLN          |
| 1044 |                       | 0       |                                   |
| 1045 | GTLN, 0               |         | /IF YOU REMEMBER, GTLN WAS NUMBE  |
| 1046 | GTCUNT, 0             |         | /NUMBER OF CHARACTERS IN BUFFER.  |
| 1047 | GTMAX, GPLOW-GPHIGH-1 |         | /=MAXIMUM NUMBER OF CHARACTERS I  |
| 1050 | GTTCAR, 0             |         | /LAST CHARACTER TYPED IN.         |
| 1051 | GTRUBS, 0             |         | /WHETHER OR NOT WE RE RUBBING 0   |
| 1052 | GPPRST, GPLOW         |         | /FOR RESETTING BUFFER POINTERS WI |
| 1053 | /                     |         |                                   |
| 1054 | /                     |         |                                   |
| 1055 | /                     |         |                                   |
| 1056 | /                     |         |                                   |
| 1057 | GTREST, 0             |         | /THIS RESETS ALL POINTERS IN CAS  |
| 1060 | DCA                   | GTCUNT  |                                   |
| 1061 | DCA                   | GTLN    |                                   |
| 1062 | DCA                   | GTRUBS  |                                   |
| 1063 | TAD                   | GPPRST  |                                   |
| 1064 | DCA                   | GETCPP  |                                   |
| 1065 | TAD                   | GPPRST  |                                   |
| 1066 | DCA I                 | (GPPONT |                                   |
| 1067 | JMS I                 | (PUREST | /RESET OUTPUT NOW.                |
| 1070 | JMP I                 | GTREST  | /RETURN. ALL IS DONE.             |
| 1071 | /                     |         |                                   |
| 1072 | /                     |         |                                   |
| 1073 | /                     |         |                                   |
| 1074 | /                     |         |                                   |
| 1075 | /                     |         |                                   |
| 1076 | /                     |         |                                   |
| 1077 | /                     |         |                                   |
| 1100 | /                     |         |                                   |
| 1101 | /                     |         |                                   |
| 1102 | /                     |         |                                   |
| 1103 |                       | EJECT   |                                   |
| 1104 | /                     |         |                                   |
| 1105 | /                     |         |                                   |
| 1106 | /                     |         |                                   |
| 1107 | GTTEST, 0             |         | /THIS ROUTINES CHECKS             |
| 1110 | KSF                   |         | /AND GETS A CHARACTER FFROM THE   |
| 1111 | JMP I                 | GTTEST  | /NO CHARACTER, RETURN.            |
| 1112 | KRB                   |         | /YES, ITS A CHARACTER. READ IT I  |
| 1113 | AND                   | GT177   | /CHOP OFF STUFF.                  |
| 1114 | TAD                   | GT200   | /AND PUT IT BACK ON.              |

|      |  |             |         |                                  |
|------|--|-------------|---------|----------------------------------|
| 1115 |  | DCA         | GTTCAR  | /FINALLY SAVE IT AWAY.           |
| 1116 |  | TAD I       | GTTGOT  | /SET UP THE CHECKER.             |
| 1117 |  | DCA         | GTRST   | /AND SAVE FOR A SECOND           |
| 1120 |  | TAD         | GTTCAR  | /RECALL THE CHARACTER.           |
| 1121 |  | JMS I       | GTRST   | /GO DO THE SUBROUTINE NOW.       |
| 1122 |  | SKP CLA     |         | /ALL IS WELL.                    |
| 1123 |  | JMP         | GTTCLA  | /DONT PROCESS THIS CHARACTER.    |
| 1124 |  | TAD         | GTTCAR  | /PROCESS THIS CHARACTER.         |
| 1125 |  | TAD I       | (GTM215 | /IS IT A CARRIAGE RETURN?        |
| 1126 |  | SNA         |         |                                  |
| 1127 |  | JMP         | GTMCRF  | /YEP.                            |
| 1130 |  | TAD         | GTM10   | /NOPE. HOW ABOUT A CONTROL U <AU |
| 1131 |  | SNA         |         |                                  |
| 1132 |  | JMP I       | (GTMDEL | /YEP. DELETE THE LINE.           |
| 1133 |  | TAD         | GTM152  | /NOPE. HOW ABOUT A RUBOUT.       |
| 1134 |  | SNA CLA     |         |                                  |
| 1135 |  | JMP         | GTM RUB | /YEP. DELETE ONE CHARACTER.      |
| 1136 |  | TAD         | GTTCAR  | /RECALL THE CHARACTER.           |
| 1137 |  | JMS I       | (GPPUTI | /INSERT IT IN THE BUFFER.        |
| 1140 |  | TAD         | GTRUBS  | /CHECK THE SWITCH.               |
| 1141 |  | SNA CLA     |         |                                  |
| 1142 |  | JMP         | GTRPS   | /RUBOUTS NOT ON NOW.             |
| 1143 |  | DCA         | GTRUBS  | /RUBOUTS ON. TURN THEM OFF.      |
| 1144 |  | TAD         | GT334   | /SEND OUT A FORM CHARACTER.      |
| 1145 |  | JMS         | GTPRT   | /BYE BYE.                        |
| 1146 |  | /           |         |                                  |
| 1147 |  | GTRPS, TAD  | GTTCAR  | /RECALL THE CHARACTER.           |
| 1150 |  | JMS         | GTPRT   | /SEND IT OUT.                    |
| 1151 |  | GTTCLA, CLA |         | /CLA EXIT.                       |
| 1152 |  | JMP I       | GTTST   | /RETURN NOW TO NORM STRUCTURE.   |
| 1153 |  | /           |         |                                  |
| 1154 |  | /           |         |                                  |
| 1155 |  | /           |         |                                  |
| 1156 |  | /           |         |                                  |
| 1157 |  | /           |         |                                  |
| 1160 |  | EJECT       |         |                                  |
| 1161 |  | /           |         |                                  |
| 1162 |  | /           |         |                                  |
| 1163 |  | /           |         |                                  |
| 1164 |  | /           |         |                                  |
| 1165 |  | /           |         |                                  |
| 1166 |  | /           |         |                                  |
| 1167 |  | /           |         |                                  |
| 1170 |  | /           |         |                                  |
| 1171 |  | GTPRT, 0    |         | /THIS ROUTINE TYPES OUT A CHAR I |
| 1172 |  | DCA I       | (GTBACK | /SAVE CHARACTER FOR A SECOND.    |
| 1173 |  | TAD I       | GPUCUNT | /GET OUTPUT COUNT.               |
| 1174 |  | TAD I       | GPUMOST | /CHECK AGAINST UPPER LIMIT.      |
| 1175 |  | SNA CLA     |         | /HOW ABOUT IT.                   |
| 1176 |  | JMP I       | GTPRT   | /ITS BUSY. DONT WAIT FOR IT.     |
| 1177 |  | TAD I       | (GTBACK | /RECALL MY CHARACTER.            |
| 1200 |  | JMS I       | GPUTMY  | /PRINT IT.                       |
| 1201 |  | JMP I       | GTPRT   | /AND RETURN.                     |
| 1202 |  | /           |         |                                  |
| 1203 |  | /           |         |                                  |
| 1204 |  | /           |         |                                  |
| 1205 |  | /           |         |                                  |
| 1206 |  | /           |         |                                  |
| 1207 |  | /           |         |                                  |
| 1210 |  | /           |         |                                  |
| 1211 |  | GTMCRF, ISZ | GTLN    | /BOP UP NUMBER OF LINES ON A C.R |
| 1212 |  | TAD         | GT215   | /PLACE A CARRIAGE RETURN IN THE  |

```

1213 JMS I (GPPUTI /BY CALLING THIS ROUTINE.
1214 /
1215 GTEXXT, CLA /THIS ROUTINE TERMINATES A TTY LI
1216 TAD GTRUBS /ARE WE DOING RUBOUTS?
1217 SNA CLA /?
1220 JMP ,+4 /NOPE.
1221 DCA GTRUBS /YEP, CLEAR OUT FLAG.
1222 TAD GT334 /THEN TYPE A FORM.
1223 JMS GTPRT
1224 TAD GT215 /THEN PRINT A C.R. L.F.
1225 JMS GTPRT /ON THE TELETYPE.
1226 TAD GT212
1227 JMS GTPRT /OUT GOES THE LINE FEED.
1230 JMP I GTTEST /AND RETURN
1231 /
1232 /
1233 /
1234 /
1235 /
1236 GTMRUB, TAD GTRUBS /CHECK THE RUBOUT FLAG.
1237 SZA CLA
1240 JMP GTMRB2 /DON T FUDGE UP MULTIPLE RUBOUTS
1241 ISZ GTRUBS /SET RUBOUT FLAG.
1242 TAD GT334 /PRINT A LEADING SLASH.
1243 JMS GTPRT
1244 /
1245 GTMRB2, JMS I (GTBACK /BACK UP ONE CHARACTER.
1246 SPA SNA /ANY THERE?
1247 JMP GTEXXT /NOPE.
1250 JMS GTPRT /YEP, PRINT IT.
1251 JMP I GTTEST /AND EXIT
1252 /
1253 /
1254 /
1255 /
1256 /
1257 /
1260 /
1261 /
1262 /
1263 GPPFUL, TAD GT207 /RING BELL IF INPUT BUFFER FULL.
1264 JMS GTPRT
1265 JMP I GTTEST /AND EXIT FROM THIS ROUTINE.
1266 /
1267 /
1270 /
1271 /
1272 /
1273 GT177, 177
1274 GT200, 200
1275 GTM10, =10
1276 GTM152, =152
1277 GT334, 334
1300 GT215, 215
1301 GT212, 212
1302 GT207, 207
1303 GPUCUNT, PUCUNT
1304 GPUMOST, PUMOST
1305 GPUTMY, PUTMY
1306 GETFPP, FPPTAB+10
1307 GTTGOT, FPPOOD+1
1310 /

```



```

1311 /
1312 /
1313 /
1314 /
1315 /
1316 /
1317 / PAGE
1320 /
1321 /
1322 / EJECT
1323 /
1324 /
1325 /
1326 /
1327 /
1330 /
1331 GTMDEL, JMS GTBACK /BACK UP ONE CHARACTER,
1332 SMA SZA CLA /UNTIL THE END OF LINE HAS BEEN
1333 JMP .-2 /LOOP IF NOT DONE
1334 JMP I (GTRUB /WHEN DONE, SIMULATE ONE RUBOUT
1335 /
1336 /
1337 /
1340 /
1341 /
1342 /
1343 /
1344 /
1345 /
1346 /
1347 /
1350 /
1351 /
1352 / EJECT
1353 /
1354 /
1355 /
1356 /
1357 GPPUTI, 0 /THIS ROUTINE PUTS A CHARACTER I
1360 DCA I (GTPRT /SAVE FOR SECOND.
1361 TAD I (GTCUNT /TEST THE COUNT
1362 TAD I (GTMAX /AGAINST MAXIMUM ALLOWED.
1363 SNA CLA /TOOOO BIIGGG?
1364 JMP I (GPPFUL /YEP, RETURN WITHOUT STICKING IT
1365 TAD I (GTPRT /RECALL THE CHARACTER
1366 DCA I GPPONT /STICK IT IN.
1367 ISZ I (GTCUNT /BOP THE COUNTER.
1370 JMS GPPBOP /AND BOP THE POINTER
1371 GPPONT, GPPLOW /INITIAL VALUE.
1372 JMP I GPPUTI /AND THEN EXIT.
1373 /
1374 /
1375 /
1376 /
1377 /
1400 /
1401 /
1402 /
1403 /
1404 /
1405 /
1406 GPPBOP, 0 /POINTER BOPPER.

```

```

1407 ISZ I GPPBOP /BOP THE POINTER.
1410 TAD I GPPBOP /GET THE POINTER.
1411 TAD GPPUPL /CHECK AGAINST THE LIMIT.
1412 SZA CLA /OK?
1413 JMP GPPBEX /YEP, EXIT
1414 TAD I (GPPRST /RESET THE POINTER
1415 DCA I GPPBOP
1416 GppBEX, ISZ I GPPBOP /BOP OVER ARGUMENT
1417 JMP I GPPBOP /AND RETURN.
1420 /
1421 /
1422 /
1423 /
1424 /
1425 /
1426 /
1427 EJECT
1430 /
1431 /
1432 /
1433 GTBACK, 0 /THIS ROUTINE BACKS UP THE BUFFE
1434 CLA CMA /BOP DOWN THE COUNT
1435 TAD I (GTCUNT
1436 SPA /IF YOU CAN, THAT IS!
1437 JMP I GTBACK /ELSE RETURN.
1440 DCA I (GTCUNT /STORE BACK CORRECT COUNT.
1441 CLA CMA /BOP DOWN THE INPUT POINTER ALSO
1442 TAD GPPONT
1443 DCA GPPONT
1444 TAD GPPONT
1445 TAD GPTST1
1446 SZA CLA /TEST FOR LOWER END WRAP AROUND
1447 JMP ,+3 /=LOWER BUFFER-1
1450 TAD GPRST1 /TOO FAR?
1451 DCA GPPONT /NOPE.
1452 TAD I GPPONT /YEP, RESET POINTER.
1453 TAD GTM215
1454 SNA CLA /RECALL THE OLD CHARACTER.
1455 JMP I GPRINS /IS IT A CARRIAGE RETURN?
1456 TAD I GPPONT /YEP.
1457 JMP I GTBACK /NOPE. REFETCH CHAR AND RETURN.
1460 /
1461 GPRINS, TAD I (GT215 /TOO FAR, STICK C.R. BACK IN.
1462 JMS GPPUTI
1463 JMP I GTBACK /AND EXIT NOW.
1464 /
1465 /
1466 /
1467 /
1470 /
1471 /
1472 EJECT
1473 /
1474 /
1475 /
1476 /
1477 /
1500 NECESSARY CONSTANTS AND VALUES.
1501 /
1502 /
1503 /
1504 GTM215, =215

```

```

1505 GPRST1, GPHIGH
1506 GPTST1, -GPLOW+1
1507 GPPUPL, -GPHIGH-1
1510 /
1511 /
1512 /
1513 /
1514 /
1515 /
1516 /
1517 /
1520 /
1521 /
1522 /
1523 /
1524 /
1525 /
1526 /
1527 EJECT
1530 /
1531 /
1532 /
1533 /
1534 /
1535 /
1536 /
1537 /
1540 /
1541 /
1542 PUTMY2, 0
1543 TAD I (FPPTAB+10)
1544 JMS PUTMY
1545 JMP I PUTMY2
1546 /
1547 /
1550 /
1551 /
1552 PUTMY, 0
1553 DCA GTBACK
1554 TAD PUCUNT
1555 TAD PUMOST
1556 SNA CLA
1557 JMP PUBUSY
1560 TAD GTBACK
1561 DCA I PUPUNT
1562 ISZ PUCUNT
1563 JMS PUTEST
1564 JMS PUBOP
1565 PUPUNT, PULOWER
1566 JMP I PUTMY
1567 /
1570 /
1571 PUBUSY, JMS I 4
1572 0
1573 JA PUTDLY
1574 JMP I PUTMY
1575 /
1576 PUBOP, 0
1577 TAD I PUBOP
1600 TAD PUUPTS
1601 SNA
1602 TAD PURST

```

/THIS SHALL GET THE CHARACTER.  
/GET THE CHARACTER.  
/AND SEND IT OUT  
/AND RETURN.

/SAVE FOR A SECOND.  
/CHECK THE COUNT.

/ITS STILL FULL. DELAY.  
/RECALL THE CHARACTER.  
/SAVE IN BUFFER  
/BOP THE COUNT.  
/TEST ALL FLAGS.  
/BOP MY POINTER.

/AND RETURN.

/RESET FPP PC

/TO A DO NOTHING ROUTINE.  
/AND RETURN FOR NOW.

/GET THE POINTER.  
/CHECK THE UPPER LIMIT.

/ITS THERE RESET

```

1603 TAD PUFUDGE /AND STORE BACK AFTER CRRRECTING
1604 DCA I PUBOP /AWAY IT GOES.
1605 ISZ PUBOP /BOP PAST THE ARG,
1606 JMP I PUBOP /AND RETURN
1607 /
1610 /
1611 /
1612 /
1613 /
1614 PUCUNT, 0
1615 PUMOST, PULOWER=PUUPPER=1
1616 PUUPTS, =PUUPPER
1617 PURST, PULOWER=PUUPPER=1
1620 PUFUDGE, PUUPPER+1
1621 PUFLAG, 0
1622 /
1623 /
1624 /
1625 /
1626 /
1627 PUTEST, 0
1630 TSP
1631 JMP ,+3 /READY
1632 TCF
1633 DCA PUFLAG /NOT NOW.
1634 TAD PUCUNT /CLEAR IT
1635 SNA CLA
1636 JMP I PUTEST /AND RECARD SAID FACT.
1637 TAD PUFLAG /TEST BUFFER
1640 SZA CLA
1641 JMP I PUTEST /EXIT ON NO CHARS.
1642 TAD I PUPONT /TEST TTY FLAG
1643 TLS
1644 CLA CMA
1645 TAD PUCUNT /EXIT IF NOT READY.
1646 DCA PUCUNT /GET A CHARACTER
1647 ISZ PUFLAG /OUTPUT IT.
1650 JMS PUBOP /BOP DOWN THE COUNT
1651 PUPONT, PULOWER
1652 JMP I PUTEST
1653 /
1654 /
1655 /
1656 /
1657 /
1660 /
1661 /
1662 /
1663 /
1664 /
1665 /
1666 PUREST, 0
1667 TAD PUPUNT
1670 DCA PUPONT
1671 DCA PUCUNT
1672 DCA PUFLAG
1673 JMP I PUREST /TURN OFF ALL FLAGS.
1674 /
1675 /
1676 /
1677 /
1700 /

```

```

1701 /
1702 /
1703 /
1704 /
1705 /
1706 /
1707 /
1710 /
1711 / EJECT
1712 /
1713 /
1714 /
1715 / PAGE
1716 /
1717 /
1720 /
1721 /
1722 /
1723 /
1724 /
1725 /
1726 /
1727 /
1730 / GTTEST
1731 / PUTTEST
1732 /
1733 / GTGTGT, 0
1734 / JMS I GTGTGT+1 /CHECK BOTH BUFFERS
1735 / JMS I GTGTGT+2 /THEN EXIT WHEN DONE.
1736 / JMP I GTGTGT
1737 /
1740 /
1741 /
1742 /
1743 /
1744 / GETC, JA . /THIS ROUTINE GETS A CHARACTER F
1745 / FLDA GTLN-2 /GET NUMBER OF LINES IN BUFFER.
1746 / JEQ GETC+2 /IF NO LINES IN BUFFER LOOP.
1747 / TRAP3 GETCCC /IF THERE ARE CHARACTERS THERE,
1750 / FNORM /GET ONE BY TRAPPING AND BRING I
1751 / JA GETC /THEN RETURN WITH THE CHARACTER
1752 /
1753 /
1754 /
1755 /
1756 /
1757 /
1760 /
1761 /
1762 /
1763 /
1764 / PUTC, JA . /THE ROUTINE TO PUT A CHARACTER
1765 / ALN 0 /FIX THE CHARACTER.
1766 / PUTLOP, TRAP3 PUTMY2 /TRAP OUT NOW
1767 / JA PUTC /AND EXIT WHEN DONE.
1770 /
1771 /
1772 /
1773 /
1774 /
1775 / PUTDLY, LDX -1000,0 /THIS ROUTINE DELAYS IF NECESSAR
1776 / JXN .,0+ /BECAUSE THIS WILL LEAVE THE FPP

```

```

1777 JA PUTLOP /WHEN DONE, WE WILL TRAP AGAIN.
2000 /
2001 /
2002 /
2003 /
2004 IOOUTW, JA . /THIS WAITS FOR ALL OUTPUT TO BE
2005 STARTD /
2006 FLDA PUCUNT-1 /GET NUMBER OF CHARACTERS TO DO.
2007 ATX 0 /PLACE IN AN INDEX.
2010 STARTF /IN FLOATING MODE AGAIN.
2011 JXN IOOUTW+2,0 /WAIT IF NOT DONE.
2012 JA IOOUTW /EXIT IF OK.
2013 /
2014 /
2015 /
2016 /
2017 /
2020 /
2021 /
2022 /
2023 /
2024 /
2025 /
2026 /
2027 /
2030 GPLOW=,
2031 /
2032 PAGE
2033 /
2034 GPHIGH=,=1+13
2035 /
2036 PLLOWER=GPHIGH+1
2037 /
2040 PLUPPER=,+177
2041 /
2042 /
2043 /
2044 /
2045 0
2046 /
2047 /
2050 PAGE
2051 /
2052 /
2053 /
2054 /
2055 /
2056 /
2057 /
2060 S
2061 /
2062 /
2063 /
2064 /
2065 /

```

```
2066 EJECT
2067 /
2070 /
2071 /
2072 IFPOS FPPOLD- /FIND OUT WHERE WE ARE.
2073 /
2074 ORG FPPOLD /IF WE DON T OVERLAP, ORIGIN WHE
2075 /
2076 $
2077 /
2100 /
2101 /
2102 /
2103 /
2104 /
0000 CHAIN "FPPLB1S"1
0001 /
0002 /
0003 /
```





```

0046 EJECT
0047 BASE 77000
0050 /
0051 /
0052 /
0053 MCRTPB=77000
0054 MCRTPI=MCRTPB+30
0055 MCRTPI2=MCRTPI+3
0056 MCRTPI3=MCRTPI2+3
0057 MCRTPI4=MCRTPI3+3
0060 MCRTPI5=MCRTPI4+3
0061 MCRTPI6=MCRTPI5+3
0062 MCRTPI7=MCRTPI6+3
0063 MCRTPI8=MCRTPI7+3

```

```

0064 /
0065 /
0066 /
0067 /
0070 /
0071 /

```

```

0072 00600 0001 FPPONE, 1. /THESE ARE THE COMMONLY USED CON
 00601 2000
 00602 0000
0073 00603 0002 FPPTWO, 2.
 00604 2000
 00605 0000

```

```

0074 /
0075 /
0076 /

```

NOTE: DO NOT MOVE THE ORDER OF THESE AROUND.

```

0077 00606 0001 FPPPI2, 1 /PI DIVIDED BY 2.
0100 00607 3110 3110
0101 00610 3755 3755
0102 /
0103 00611 0002 FPPPI, 2
0104 00612 3110 3110
0105 00613 3755 3755 /PI
0106 /
0107 00614 0003 FPP2PI, 3
0110 00615 3110 3110
0111 00616 3755 3755 /TWO PI.
0112 /
0113 /
0114 /
0115 /
0116 /
0117 /
0120 /
0121 /

```



|      |         |        |          |                                      |
|------|---------|--------|----------|--------------------------------------|
| 0217 | FRNDPO, | LDA    | 0,0      |                                      |
| 0220 |         | FSTA   | FRZERO+1 | /SAVE THEM AWAY.                     |
| 0221 |         | FLDA   | FRNDTM   | /RECALL ADD-IN FACTOR                |
| 0222 |         | FADDM  | FRZERO   | /ADD IT IN NOW.                      |
| 0223 |         | FLDA   | FRNDX    | /RECALL RANDOM HIGH ORDER WORD       |
| 0224 |         | FSTA   | FRZERO-1 | /OVERLAY EXTRANEIOUS CARRY.          |
| 0225 |         | STARTF |          | /GET BACK INTO FLATING MODE.         |
| 0226 |         | FLDA   | FRZERO-1 | /RECALL RANDOM NUMBER                |
| 0227 |         | JXN    | FRNDOK,0 | /DO NOT FLIP AROUND .5               |
| 0230 |         | FNEG   |          | /DO FLIP AROUND .5                   |
| 0231 |         | FADD   | FPPTWO   | / 2.=RND(1 TO 2)                     |
| 0232 |         | JA     | +.4      |                                      |
| 0233 | FRNDOK, | FSUB   | FPPONE   | / RND(1 TO 2)-1.                     |
| 0234 |         | FMUL   | FPPTWO   | / BRING INTO THE 0-2 RANGE AGAIN     |
| 0235 |         | FSUB   | FPPONE   | / COMPENSATE FOR CENTER LINE ERRORS. |
| 0236 |         | JGE    | +.4      | /IT S STILL POSITIVE. EXIT.          |
| 0237 |         | FADD   | FPPONE   | / WE VE OVER CORRECTED. ADD IN.      |
| 0240 |         | FSTA   | FRNDX    | /AND SAVE FOR THE NEXT TIME AROUND.  |
| 0241 |         | JA     | RAND     | /RETURN NOW.                         |
| 0242 | /       |        |          |                                      |
| 0243 | /       |        |          |                                      |
| 0244 | /       |        |          |                                      |
| 0245 | /       |        |          |                                      |
| 0246 |         |        | 0        |                                      |
| 0247 | /       |        |          |                                      |
| 0250 | FRNDTM, |        | 0        | /DUMMY COUNTER WORD.                 |
| 0251 |         |        | 0        | /RANDOMIZING LOOP COUNTER.           |
| 0252 | /       |        |          |                                      |
| 0253 |         |        | 0        | /EXPONENT OVERLAY WORD.              |
| 0254 | FRNDX,  | -36    |          | /LAST RANDOM NUMBER.                 |
| 0255 |         | 3562   |          |                                      |
| 0256 |         | 7227   |          |                                      |
| 0257 | /       |        |          |                                      |
| 0258 | /       |        |          |                                      |
| 0261 | /       |        |          |                                      |
| 0262 | FRNDC1, | 22     |          | /RANDOMIZING CONSTANT (I HOPE).      |
| 0263 |         | 3224   |          |                                      |
| 0264 |         | 2342   |          |                                      |
| 0265 | /       |        |          |                                      |
| 0266 | /       |        |          |                                      |
| 0267 |         |        | 0        |                                      |
| 0270 | FRZEKO, |        | 0        |                                      |
| 0271 |         |        | 0        |                                      |
| 0272 |         |        | 0        |                                      |
| 0273 | /       |        |          |                                      |
| 0274 | /       |        |          |                                      |
| 0275 | /       |        |          |                                      |
| 0276 | FRNDCY, |        | 0        |                                      |
| 0277 |         |        | 0        |                                      |
| 0300 | /       |        |          |                                      |
| 0301 | /       |        |          |                                      |
| 0302 | /       |        |          |                                      |
| 0303 | /       |        |          |                                      |
| 0304 | /       |        |          |                                      |
| 0305 | /       |        |          |                                      |
| 0306 | /       |        |          |                                      |
| 0307 | /       |        |          |                                      |
| 0310 |         | \$     |          |                                      |
| 0311 |         | IFREF  | EXP3     |                                      |
| 0312 | /       |        |          |                                      |
| 0313 | /       |        |          |                                      |
| 0314 |         | EJECT  |          |                                      |

0315  
0316  
0317  
0320  
0321  
0322  
0323  
0324  
0325  
0326  
0327  
0330  
0331  
0332  
0333  
0334  
0335  
0336  
0337  
0340  
0341  
0342  
0343  
0344  
0345  
0346  
0347  
0350  
0351  
0352  
0353  
0354  
0355  
0356  
0357  
0360  
0361  
0362  
0363  
0364  
0365  
0366  
0367  
0370  
0371  
0372  
0373  
0374  
0375  
0376  
0377  
0400  
0401  
0402  
0403  
0404  
0405  
0406  
0407  
0410  
0411  
0412

E X P 3  
" " " "

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSSETTS,

01754

WRITTEN BY JACK BURNES.

EJECT

EXP3 RAISES A BASE TO A POWER.  
X=A\*\*B IN FORTRAN NOTATION.

|         |       |        |                          |
|---------|-------|--------|--------------------------|
| EXP3OK, | JSA   | ALOG   | /TAKE LOG OF THE NUMBER. |
|         | FMUL  | MCRTP8 | /MULTIPLY BY HIS POWER.  |
|         | JSA   | EXP    | /NOW RAISE IT.           |
| EXP3,   | JA    | .      | /RETURN AND ENTRY.       |
|         | JEQ   | EXP3   | /0AX=0                   |
|         | FSTA  | MCRTP2 | /SAVE BASE               |
|         | FLDA  | MCRTP1 | /GET POWER AND SAVE      |
|         | JEQ   | EXP3ON | /XA0=1                   |
|         | FSTA  | MCRTP8 |                          |
|         | FLDA  | MCRTP2 |                          |
|         | JGE   | EXP3OK | /ALL IS WELL.            |
|         | TRAP7 | 0      | /GIVE TRAP.              |
|         | JA    | EXP3   | /AND RETURN.             |
| EXP3ON, | FLDA  | FPPONE |                          |
|         | JA    | EXP3   | /AND RETURN              |





```

0607 /
0610 /
0611 /
0612 /
0613 /
0614 / COPYRIGHT 1971
0615 / DIGITAL EQUIPMENT CORPORATION
0616 / 146 MAIN STREET,
0617 / MAYNARD, MASSACHUSSETTS.
0620 / 01754
0621 /
0622 /
0623 /
0624 / WRITTEN BY JACK BURNES.
0625 /
0626 /
0627 /
0630 /
0631 /
0632 /
0633 /
0634 /
0635 /
0636 / EJECT
0637 /
0640 /
0641 /
0642 /
0643 /
0644 /
0645 / FLDA MCRT P7 /RESTORE AND EXIT.
0646 /
0647 /
0650 / SINH, JA . /SINH ROUTINE.
0651 / FSTA MCRT P8 /SAVE THE ARGUMENT.
0652 / JGE .+3 /MAKE IT POSITIVE.
0653 /
0654 / FSTA MCRT P7 /AND SAVE ABS VALUE IN CASE WE N
0655 / FSUB SIN H1 /IS IT LESS THEN .1?
0656 / JLE SIN H2 /YES, USE SERIES APPROXIMATION.
0657 / FSUB SIN H2 /IS IT GREATER THEN 88.029?
0660 / JGE SIN HAP /YES, USE LOG(2) APPROXIMATION.
0661 / FLDA MCRT P8 /RECALL THE NUMBER
0662 / JSA EXP /EXP(X)
0663 / FSTA MCRT P7
0664 / FLDA FPPONE
0665 / FDIV MCRT P7 /1/EXP(X)
0666 / FNEG /-1/EXP(X)
0667 / FADD MCRT P7 /EXP(X)-1/EXP(X)
0670 / FDIV FPPTWO / 1/2(EXP(X)-1/EXP(X))
0671 / JA SINH /AND RETURN NOW.
0672 /
0673 /
0674 / SINHAP, FLDA MCRT P7 /RECALL ABSOLUTE VALUE.
0675 / FSUB SIN HLG /ABS(X)=LN(2)
0676 / JSA EXP /EXP(ABS(X)=LN(2))
0677 / FSTA MCRT P7
0700 / FLDA MCRT P8 /GET SIGN OF ARGUMENT.
0701 / JGE SINH /LOAD POSITIVE IF ARG WAS POSITI
0702 / FNEG /ELSE NEGATE IT.
0703 / JA SINH /AND RETURN.
0704 /

```

|      |          |        |         |                      |       |
|------|----------|--------|---------|----------------------|-------|
| 0705 | /        |        |         |                      |       |
| 0706 | SINHSR,  | FLDA   | MC RTP8 | /X SERIES IF X4.1    |       |
| 0707 |          | FMUL   | MC RTP8 | /XA2                 |       |
| 0710 |          | FSTA   | MC RTP7 | /XA2                 |       |
| 0711 |          | FMUL   | MC RTP8 | /XA3                 |       |
| 0712 |          | FSTA   | MC RTP6 | /XA3                 |       |
| 0713 |          | FMULM  | MC RTP7 | /XA5                 |       |
| 0714 |          | FOIV   | SINH3   | /XA3/6               |       |
| 0715 |          | FADDM  | MC RTP8 | /X+XA3/6             |       |
| 0716 |          | FLDA   | MC RTP7 | /XA5                 |       |
| 0717 |          | FOIV   | SINH4   | /XA5/120             |       |
| 0720 |          | FADD   | MC RTP8 | /X+XA3/6+XA5/120     |       |
| 0721 |          | JA     | SINH    | /VOILA, WE ARE DONE. |       |
| 0722 | /        |        |         |                      |       |
| 0723 | /        |        |         |                      |       |
| 0724 | /        |        |         |                      |       |
| 0725 | SINH LG, | 0      |         |                      |       |
| 0726 |          | 2613   |         |                      |       |
| 0727 |          | 4412   |         |                      |       |
| 0730 | /        |        |         |                      |       |
| 0731 | SINH1,   | .1     |         |                      |       |
| 0732 | /        |        |         |                      |       |
| 0733 | SINH2,   | 87.929 |         |                      |       |
| 0734 | /        |        |         |                      |       |
| 0735 | SINH3,   | 6.     |         |                      |       |
| 0736 | /        |        |         |                      |       |
| 0737 | SINH4,   | 120.   |         |                      |       |
| 0740 | /        |        |         |                      |       |
| 0741 | /        |        |         |                      |       |
| 0742 | /        |        |         |                      |       |
| 0743 | /        |        |         |                      |       |
| 0744 | /        |        |         |                      |       |
| 0745 | /        |        |         |                      |       |
| 0746 | /        |        |         |                      |       |
| 0747 |          | \$     |         |                      |       |
| 0750 |          | IFREF  | COSH    |                      |       |
| 0751 | /        |        |         |                      |       |
| 0752 | /        |        |         |                      |       |
| 0753 | /        |        |         |                      |       |
| 0754 | /        |        |         |                      |       |
| 0755 | /        |        |         |                      |       |
| 0756 | /        |        |         |                      |       |
| 0757 | /        | C      | O       | S                    | H     |
| 0760 | /        | -      | -       | -                    | -     |
| 0761 | /        |        |         |                      |       |
| 0762 | /        |        |         |                      |       |
| 0763 | /        |        |         |                      |       |
| 0764 | /        |        |         |                      |       |
| 0765 | /        |        |         |                      |       |
| 0766 | /        |        |         |                      |       |
| 0767 | /        |        |         |                      |       |
| 0770 | /        |        |         |                      |       |
| 0771 | /        |        |         |                      |       |
| 0772 | /        |        |         |                      |       |
| 0773 | /        |        |         |                      | 01754 |
| 0774 | /        |        |         |                      |       |
| 0775 | /        |        |         |                      |       |
| 0776 | /        |        |         |                      |       |
| 0777 | /        |        |         |                      |       |
| 1000 | /        |        |         |                      |       |
| 1001 | /        |        |         |                      |       |
| 1002 | /        |        |         |                      |       |

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSETTS.

WRITTEN BY JACK BURNES.



```

1043 /
1044 /
1045 /
1046 /
1047 /
1010 /
1011 / EJECT
1012 /
1013 /
1014 /
1015 /
1016 /
1017 /
1020 /
1021 COSH, JA . /HYPERBOLIC COSINE.
1022 FSTA MC RTP8 /SAVE ARGUMENT
1023 JGE .+3 /ABS(X)
1024 FNEG
1025 FSTA MC RTP7
1026 FSUB COSM1 /TEST FOR LIMITS.
1027 JGE COSHBG
1030 FLDA MC RTP8 /X
1031 JSA EXP /EXP(X)
1032 FSTA MC RTP7
1033 FLDA PPPONE /1.
1034 FDIV MC RTP7 / 1./EXP(X)
1035 FADD MC RTP7 / EXP(X)+1./EXP(X)
1036 FDIV PPPTWO / (EXP(X)+1./EXP(X))2.
1037 JA COSH /AND THAT IS THE DEFINITION OF C
1040 /
1041 /
1042 COSHBG, FSUB COSHLG
1043 JGT COSHE
1044 FADD COSM1
1045 JSA EXP /SEE IF TOO BIG
1046 JA COSH /YEP, ERROR
1047 /
1050 /
1051 COSHE, FLDA COSHB
1052 TRAP7 12 /GIVE INFINITY IN CASE OF NO REC
1053 JA COSH /ERROR TRAP
1054 /
1055 /
1056 COSHLG, 0
1057 2613
1060 4412
1061 /
1062 COSHB, 3777
1063 3777
1064 7777
1065 /
1066 /
1067 COSM1, 88.029 /LIMIT FACTOR.
1070 /
1071 /
1072 /
1073 /
1074 /
1075 /
1076 /
1077 /
1100 /

```

```

1101 IFREF ACOS
1102 /
1103 /
1104 /
1105 A C O S
1106 * * * *
1107 /
1110 /
1111 /
1112 /
1113 /
1114 /
1115 /
1116 COPYRIGHT 1971
1117 DIGITAL EQUIPMENT CORPORATION
1118 146 MAIN STREET,
1119 MAYNARD, MASSACHUSSETTS.
1120
1121 01754
1122 /
1123 /
1124 /
1125 WRITTEN BY JACK BURNES.
1126 /
1127 /
1130 /
1131 /
1132 /
1133 /
1134 /
1135 /
1136 /
1137 EJECT
1140 /
1141 /
1142 /
1143 /
1144 /
1145 ACOSOK, FLDA MCRTPB /X
1146 FMUL MCRTPB /X^2
1147 FNEG /-X^2
1150 FADD FPPONE /1-X^2
1151 JSA SQRT /SQRT(1-X^2)
1152 FDIV MCRTPB /SQRT(1,X^2)/X
1153 JSA ATAN /ATAN(SQRT(1,-X^2)/X)
1154 JXN ACOS,0 /NO SIGN CHANGE NECESSARY.
1155 FADD FPPPI /ADD PI IF MINUS.
1156 /
1157 ACOS, JA . /ENTRY AND EXIT.
1160 /
1161 FSTA MCRTPB /STORE AWAY.
1162 JEQ ACOSOK /IF ZERO RETURN PI OVER 2.
1163 LDX =1,0 /JUMP TIME.
1164 JGE .+5
1165 LDX 0,0
1166 FNEG
1167 FSUB FPPONE /1-|X|
1170 JLE ACOSOK /IS IT LESS THEN 1.
1171 TRAP7 1 /NO. ERROR
1172 JA ACOS /AND RETURN.
1173 /
1174 /
1175 /
1176 ACOSOK, FLDA FPPPI2 /RETURN PI OVER 2 IF ZERO.

```

```

1177 JA ACOS /AND RETURN NOW.
1200 /
1201 /
1202 /
1203 /
1204 /
1205 S
1206 IFREF ASIN
1207 /
1210 /
1211 /
1212 /
1213 /
1214 A S I N
1215 - - - -
1216 /
1217 /
1220 /
1221 /
1222 /
1223 /
1224 /
1225 /
1226 /
1227 /
1230 /
1231 /
1232 /
1233 /
1234 /
1235 /
1236 /
1237 /
1240 /
1241 /
1242 /
1243 /
1244 /
1245 /
1246 /
1247 /
1250 /
1251 /
1252 /
1253 ASINOK, FLDA MCRTPB /X USES STRAIGHT TRIG RELATION
1254 FNEG
1255 FMUL MCRTPB /=X^2
1256 FADD FPPONE /1.=X^2
1257 JEQ ASINEG /IF 0,FAC=PI OVER 2
1260 JSA SQRT
1261 FSTA MCRTPB /SQRT(1.=X^2)
1262 FLDA MCRTPB
1263 FDIV MCRTPB /X/SQRT(1.X^2)
1264 JSA ATAN /TAKE THE ARCTANGENT.
1265 /
1266 ASIN, JA . /AND RETURN, BUBBY.
1267 /
1270 FSTA MCRTPB /STORE ARG AWAY.
1271 JGE .+3 /TAKE ABSOLUTE VALUE.
1272 FNEG
1273 FSUB FPPONE /SEE IF >1
1274 JLE ASINOK /CONTINUE PROCESS.

```

```

1275 TRAP7 2 /TRAP OUT.
1276 JA ASIN /RETURN.
1277 /
1300 /
1301 ASINEQ, FLDA FPPPI2 /RETURN PI OVER TWO.
1302 FMUL MCRTPB /TIMES ARG.
1303 JA ASIN /AND RETURN.
1304 /
1305 /
1306 /
1307 /
1310 S
1311 IFREF COSD
1312 /
1313 /
1314 /
1315 C O S D
1316 - - - -
1317 /
1320 /
1321 /
1322 /
1323 /
1324 /
1325 /
1326 /
1327 /
1330 /
1331 /
1332 /
1333 /
1334 /
1335 /
1336 /
1337 /
1340 /
1341 /
1342 /
1343 /
1344 /
1345 /
1346 /
1347 /
1350 /
1351 /
1352 /
1353 /
1354 /
1355 /
1356 /
1357 /
1360 /
1361 /
1362 /
1363 /
1364 /
1365 /
1366 /
1367 /
1370 /
1371 /
1372 /

```

COPYRIGHT 1971  
 DIGITAL EQUIPMENT CORPORATION  
 146 MAIN STREET,  
 MAYNARD, MASSACHUSSETTS.  
 01754

WRITTEN BY JACK BURNES.

```

1353 /
1354 COSD, JA . /COSINE IN DEGREES.
1355 FADD COSD90 /ADD IN 90.
1356 FDIV COSD91 /AND FUDGE UP A BIT (PI/180)
1357 JSA SIN /TAKE THE SINE NOW.
1360 JA COSD /AND RETURN.
1361 /
1362 /
1363 /
1364 COSD90, 90.
1365 COSD91, 6
1366 3451
1367 3561
1370 /
1371 /
1372 /

```





```

1567 /
1570 /
1571 /
1572 /
1573 /
1574 / EJECT
1575 /
1576 /
1577 /
1600 /
1601 / THIS IS A PRETTY FEEBLE TANGENT ROUTINE FOR THE PRESENT.
1602 /
1603 /
1604 /
1605 / TAN, JA , /ENTRY FOR THE TANGENT.
1606 / JEQ TAN /IF ZERO RETURN NOW.
1607 / FSTA MCRTPB /SAVE FOR A SECOND.
1610 / JSA COS /TAKE THE COSINE NOW.
1611 / JEQ TANER /COS#0, A NO-NO
1612 / FSTA MCRTPB /AND SAVE IT.
1613 / FLDA MCRTPB /NOW TAKE THE SINE OF IT.
1614 / JSA SIN /TAKE THE SINE NOW.
1615 / FDIV MCRTPB /DO THE DIVISION.
1616 / JA TAN /AND RETURN.
1617 /
1620 /
1621 / TANER, TRAP7 10 /GIVE THE TRAP INSTRUCTION.
1622 / JA TAN /AND RETURN.
1623 /
1624 /
1625 /
1626 /
1627 /
1630 /
1631 /
1632 /
1633 /
1634 /
1635 /
1636 / $
1637 / IFREF COS
1640 /
1641 /
1642 /
1643 / C 0 8
1644 / - - -
1645 /
1646 /
1647 /
1650 /
1651 /
1652 /
1653 / COPYRIGHT 1971
1654 / DIGITAL EQUIPMENT CORPORATION
1655 / 146 MAIN STREET,
1656 / MAYNARD, MASSACHUSSETTS.
1657 / 01754
1660 /
1661 /
1662 /
1663 / WRITTEN BY JACK BURNES.
1664 /

```

1665  
1666  
1667  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1760  
1761  
1762

/

EJECT

COS,

CA  
FADD FPPPI2  
CSA SIN  
CA COS

/ADD IN PI OVER 2.  
/AND TAKE THE SIN.  
/RETURN NOW.

S  
IFREF ATAN2

A T A N 2  
- - - - -

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSSETTS.  
01754

WRITTEN BY JACK BURNES.

EJECT

ATAN2 TAKES TWO ARGUMENTS. SECOND ARGUMENT DETERMINES TH



```

1763 /
1764 /
1765 /
1766 /
1767 FLDA MCRT P7 /COMES HERE IF SIGN IS OK.
1770 ATAN2, JA . /ENTRY.
1771 FSTA MCRT P7 /SAVE FOR A SECOND.
1772 FLDA MCRT P1 /RECALL THE QUADRANT.
1773 FSTA MCRT P8 /AND MOVE IT TO A SAVE PLACE.
1774 FLDA MCRT P7 /RECALL THE TAN.
1775 JSA ATAN /TAKE THE ARC-TANGENT OF IT.
1776 FSTA MCRT P7 /SAVE IT AWAY.
1777 FLDA MCRT P8 /RECALL THE QUADRANT.
2000 JGE ATAN2=1 /IF POSITIVE SIGN IS OK.
2001 FLDA MCRT P7 /RECALL ARGUMENT.
2002 JGE ATAN2A /IF POSITIVE DO FURTHER FUDGE
2003 FADD FPPPI /OTHERWISE ADD PI.
2004 JA ATAN2 /AND RETURN
2005 /
2006 ATAN2A, FSUR FPPPI /THIRD QUADRANT. SUBTRACT PI.
2007 JA ATAN2 /AND RETURN.
2010 /
2011 /
2012 /
2013 /
2014 /
2015 /
2016 $
2017 IFREF AMOD
2020 /
2021 /
2022 /
2023 A M O D
2024 - - - -
2025 /
2026 /
2027 /
2030 /
2031 /
2032 /
2033 /
2034 /
2035 /
2036 /
2037 /
2040 /
2041 /
2042 /
2043 /
2044 /
2045 /
2046 /
2047 /
2050 /
2051 /
2052 /
2053 /
2054 /
2055 /
2056 /
2057 /
2060 /
EJECT

```

COPYRIGHT 1971  
 DIGITAL EQUIPMENT CORPORATION  
 146 MAIN STREET,  
 MAYNARD, MASSACHUSSETTS.  
 01754

WRITTEN BY JACK BURNES.

```

2061 /
2062 /
2063 / AMOD(X,Y)=X-INT(X/Y)*Y
2064 /
2065 /
2066 AMOD, JA . /ENTRY
2067 FSTA MCRTPB /SAVE IN A TEMPORARY.
2070 FDIV MCRTPB /DIVIDE BY Y
2071 JAL AMODER /TOO BIG.
2072 ALN 0 /FIX IT UP NOW.
2073 FNORM
2074 FMUL MCRTPB /MULTIPLY IT.
2075 FNEG
2076 FADD MCRTPB /NEGATE IT.
2077 JA AMOD /AND ADD IN X.
 /AND RETURN.
2100 /
2101 /
2102 AMODER, TRAP7 11 /GIVE THERROR.
2103 JA AMOD /AND RETURN.
2104 /
2105 /
2106 /
2107 S
2110 IFREF SIGN
2111 /
2112 /
2113 /
2114 /
2115 S I G N
2116 - - - -
2117 /
2120 /
2121 /
2122 /
2123 /
2124 /
2125 /
2126 /
2127 / COPYRIGHT 1971
2130 / DIGITAL EQUIPMENT CORPORATION
2131 / 146 MAIN STREET,
2132 / MAYNARD, MASSACHUSSETTS.
2133 /
2134 /
2135 /
2136 / WRITTEN BY JACK BURNES.
2137 /
2140 /
2141 /
2142 /
2143 /
2144 /
2145 /
2146 /
2147 /
2150 / EJECT
2151 /
2152 /
2153 /
2154 /
2155 / SIGN RETURNS -1,0,+1 DEPENDING ON WHETHER
2156 / INPUT ARGUMENT IS <0, =0, >0

```

|      |       |       |        |                        |
|------|-------|-------|--------|------------------------|
| 2157 | /     |       |        |                        |
| 2160 | /     |       |        |                        |
| 2161 | /     |       |        |                        |
| 2162 |       | FLDA  | FPPONE | /GREATER THEN ZERO.    |
| 2163 | SIGN, | JA    | .      | /ENTRY AND RETURN.     |
| 2164 |       | JEQ   | SIGN   | /IF ZERO RETURN.       |
| 2165 |       | JGE   | SIGN-2 | /IF POSTIVE GIVE A +1. |
| 2166 |       | FLDA  | +.4    | /GIE THE MINUS 1.      |
| 2167 |       | JA    | SIGN   |                        |
| 2170 |       | 1     |        |                        |
| 2171 |       | 0000  |        |                        |
| 2172 |       | 0     |        |                        |
| 2173 | /     |       |        |                        |
| 2174 | /     |       |        |                        |
| 2175 | /     |       |        |                        |
| 2176 | /     |       |        |                        |
| 2177 |       | \$    |        |                        |
| 2200 |       | IFREF | CRLF   |                        |
| 2201 | /     |       |        |                        |
| 2202 | /     |       |        |                        |
| 2203 | /     |       |        |                        |
| 2204 |       | EJECT |        |                        |
| 2205 | /     |       |        |                        |
| 2206 | /     |       |        |                        |
| 2207 | /     |       |        |                        |
| 2210 | /     | C     | R      | L                      |
| 2211 | /     | .     | .      | .                      |
| 2212 | /     |       |        |                        |
| 2213 | /     |       |        |                        |
| 2214 | /     |       |        |                        |
| 2215 | /     |       |        |                        |
| 2216 | /     |       |        |                        |
| 2217 | /     |       |        |                        |
| 2220 | /     |       |        |                        |
| 2221 | /     |       |        |                        |
| 2222 | /     |       |        |                        |
| 2223 | /     |       |        |                        |
| 2224 | /     |       |        |                        |
| 2225 | /     |       |        |                        |
| 2226 | /     |       |        |                        |
| 2227 | /     |       |        |                        |
| 2230 | /     |       |        |                        |
| 2231 | /     |       |        |                        |
| 2232 | /     |       |        |                        |
| 2233 | /     |       |        |                        |
| 2234 | /     |       |        |                        |
| 2235 | /     |       |        |                        |
| 2236 | /     |       |        |                        |
| 2237 | /     |       |        |                        |
| 2240 | /     |       |        |                        |
| 2241 | /     |       |        |                        |
| 2242 | /     |       |        |                        |
| 2243 | /     |       |        |                        |
| 2244 | /     |       |        |                        |
| 2245 | /     |       |        |                        |
| 2246 | /     |       |        |                        |
| 2247 | /     |       |        |                        |
| 2250 | /     |       |        |                        |
| 2251 | CRLF, | JA    | .      |                        |
| 2252 |       | FLDA  | CRLFPT |                        |
| 2253 |       | FSTA  | MCRTPB |                        |
| 2254 |       | JSA   | PUTSTR |                        |

```
2255 JA CRLF
2256 /
2257 /
2260 2
2261 215
2262 CRLFPT, 212
2263 JA CRLFPT=2
2264 /
2265 /
2266 /
2267 /
2270 S
0000 CHAIN "FPPLB25"1
0001 /
0002 /
```

0003  
0004  
0005  
0006  
0007  
0010  
0011  
0012  
0013

/  
/  
/  
/  
/  
/  
/  
/  
/

EJECT

THIS IS THE SECOND PART OF THE FPP FUNCTION LIBRARY.

0014  
0015  
0016  
0017  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0100  
0101  
0102  
0103  
0104  
0105  
0106  
0107  
0110

/

EJECT  
IFREF GETNUM

EJECT

G E T N U M  
" - - - - "

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSSETTS.  
01754

WRITTEN BY JACK BURNES.

THIS ROUTINE WILL CONVERT A  
STRING OF ASCII CHARACTERS PACKED  
ONE PER WORD TO A FPP NUMBER.

THIS ROUTINE WILL CALL A ROUTINE CALLED "GETC" TO GET IS

|            |         |                                  |
|------------|---------|----------------------------------|
| GETNUM, JA | .       | /ENTRY AND RETURN POINT.         |
| FCLA       |         | /CLEAR FAC                       |
| FSTA       | MCRTPI  | /ZERO OUT ACCUMULATED NUMBER.    |
| FSTA       | MCRTPI3 | /ZERO OUT NUMBER OF DIGITS OFTER |
| ATX        | 0       | /CLEAR OUT DIGIT TOTAL           |
| ATX        | 1       | /SET TO ACCEPT SIGN (+ OR -)     |
| JSA        | GETCOL  | /ASSEMBLE A NUMBER NOW.          |

|      |         |       |          |                                  |
|------|---------|-------|----------|----------------------------------|
| 0111 |         | FLDA  | MCRTPB   | /GET THE BAD CHARACTER.          |
| 0112 |         | FSUB  | GETCOT   | /IS IT A PERIOD.                 |
| 0113 |         | JNE   | GETNDT   | /NO. NO DECIMAL POINT. CHECK FOR |
| 0114 |         | ATX   | 0        | /DECIMAL POINT. SET DIGIT TOTAL. |
| 0115 |         | FLDA  | MCRTP7   | /RESET ACCUMULATED SUM POSITIVE. |
| 0116 |         | FMULM | MCRTP1   | /BY MULTIPLY BY SIGN AGAIN.      |
| 0117 |         | JSA   | GETCOL   | /ACCULATE IN THE NEXT STRING.    |
| 0120 |         | XTA   | 0        | /RECALL TOTAL NUMBER OF DIGITS.  |
| 0121 |         | FSTA  | MCRTP3   | /AND SAVE AWAY FOR AWHILE.       |
| 0122 | /       |       |          |                                  |
| 0123 | GETNDT, | FLDA  | MCRTP1   | /MOVE OVER TOTAL BECAUSE         |
| 0124 |         | FSTA  | MCRTP2   | /THE EXPONENT WILL CREAM IT.     |
| 0125 |         | FLDA  | MCRTPB   | /RECALL THE BAD CHARACTER.       |
| 0126 |         | FSUB  | GETCEE   | /SEE IF ITS AN "E".              |
| 0127 |         | JNE   | GETEX1   | /NOT AN "E". NO EXPONENT.        |
| 0130 |         | FSTA  | MCRTP1   | /ZERO OUT ACCUMULATING REGISTER. |
| 0131 |         | ATX   | 1        | /AND ALSP RESET THE SIGN ACCEPT  |
| 0132 |         | JSA   | GETCOL   | /GET THE EXPONENT.               |
| 0133 |         | FLDA  | MCRTP1   | /RECALL EXPONENT.                |
| 0134 | GETNEX, | FSUB  | MCRTP3   | /SUBTRACT OFF THE DIGITS AFTER P |
| 0135 |         | JEQ   | GETDEQ   | /EXPONENTS BALANCE. ALL IS WELL. |
| 0136 |         | JGT   | GETDGT   | /GREATER THEN 0. MULITPLY.       |
| 0137 | /       |       |          |                                  |
| 0140 |         | ATX   | 1        | /NEGATIVE, SET UP COUNT.         |
| 0141 |         | FLDA  | MCRTP2   | /RECALL NUMBER.                  |
| 0142 |         | FDIV  | GETC10   | /DIVIDE BY 10.                   |
| 0143 |         | JXN   | ,=2,1+   | /LOOP UNTIL DONE.                |
| 0144 |         | JA    | GETNUM   | /RETURN WHEN DONE.               |
| 0145 | /       |       |          |                                  |
| 0146 | GETDEQ, | FLDA  | MCRTP2   | /RECALL THE NUMBER               |
| 0147 |         | JA    | GETNUM   | /AND EXIT.                       |
| 0150 | /       |       |          |                                  |
| 0151 | GETDGT, | FNEG  |          | /NEGATE NUMBER AND STICK         |
| 0152 |         | ATX   | 1        | /IT IN THE INDEX COUNTER.        |
| 0153 |         | FLDA  | MCRTP2   | /RECALL NUMBER                   |
| 0154 |         | FMUL  | GETC10   | /MULTIPLY IT BY 10.              |
| 0155 |         | JXN   | ,=2,1+   | /LOOP UNTIL DONE.                |
| 0156 |         | JA    | GETNUM   | /THEN RETURN.                    |
| 0157 | /       |       |          |                                  |
| 0160 | GETEX1, | FCLA  |          | /FAKE OUT ZERO EXPONENT.         |
| 0161 |         | JA    | GETNEX   | /AND REENTER.                    |
| 0162 | /       |       |          |                                  |
| 0163 | /       |       |          |                                  |
| 0164 | /       |       |          |                                  |
| 0165 | /       |       |          |                                  |
| 0166 | /       |       |          |                                  |
| 0167 | /       |       |          |                                  |
| 0170 | /       |       |          |                                  |
| 0171 | /       |       |          |                                  |
| 0172 | /       |       |          |                                  |
| 0173 | /       |       |          |                                  |
| 0174 | GETCOL, | JA    | .        | /ROUTINE TO ASSEMBLE A STRING OF |
| 0175 |         | JXN   | GETCLP,1 | /IF SIGN SET, BYPASS THIS CRUD.  |
| 0176 |         | FLDA  | FPPONE   | /SET THE SIGN TO ONE.            |
| 0177 |         | FSTA  | MCRTP7   | /7 IS THE SIGN REGISTER.         |
| 0200 | GETCLP, | JSA   | GETC     | /GET A CHARACTER NOW.            |
| 0201 |         | FSTA  | MCRTPB   | /SAVE AWAY.                      |
| 0202 |         | FSUB  | GETC00   | /TEST AGAINST "0".               |
| 0203 |         | JLT   | GETBD    | /ITS BAD.                        |
| 0204 |         | FSUB  | GETC10   | /TEST AGAINST "9".               |
| 0205 |         | JGE   | GETBD    | /NOPE. ITS BAD.                  |
| 0206 |         | ATX   | 1        | /SET THE SIGN TO SWITCH TO NON=A |

```

0207 FLDA GETC10 /MULTIPLY TOTAL.
0210 FMULM MCRTPI1 /AND STASH BACK.
0211 XTA 1 /RECALL MAGIC DIGIT.
0212 FADD GETC10 /FIX IT UP ABIT.
0213 FADDM MCRTPI1 /AND THEN STORE BACK.
0214 JXN GETCLP,0+ /GO BACK AND LOOP, BOP DIGIT COU
0215 /
0216 GETBD, FLDA MCRTPI8 /GET THE BAD CHARACTER.
0217 FSUB GETCPL /SEE IF ITS A PLUS SIGN
0220 JEQ GETP /ITS A PLUS SIGN.
0221 FSUB FPPTWO /SUBTRACT 2 AND SEE
0222 JEQ GETM /IF ITS A MINUS SIGN.
0223 FADD GETCSP /SEE IF ITS A SPACE
0224 JEQ GETCLP /AND LOOP IF IT IS.
0225 /
0226 GETXX, FLDA MCRTPI7 /RECALL THE SIGN.
0227 FMULM MCRTPI1 /AND FORCE IT IN.
0230 JA GETCOL /AND THEN EXIT.
0231 /
0232 GETM, FSUB FPPONE /MINUS ONE FOR THE SIGN WORD.
0233 JA .+4 /SKIP PAST THE CRAP.
0234 /
0235 GETP, FLDA FPPONE /PLUS ONE.
0236 JXN GETXX,1 /IF X IS ALREADY SET, EXIT.
0237 ATX 1 /ELSE RESET SWITCH.
0240 JA GETCLP-1 /AND LOOP.
0241 /
0242 /
0243 /
0244 /
0245 /
0246 /
0247 /
0250 /
0251 GETC00, 10
0252 2000
0253 0
0254 GETC10, 10.
0255 GETCEE, 10
0256 3050
0257 0
0260 GETCDT, 10
0261 2560
0262 0
0263 GETCPL, 10
0264 2530
0265 0
0266 GETCSP, 13.
0267 /
0270 /
0271 /
0272 /
0273 /
0274 S
0275 IFREF PCON
0276 /
0277 /
0300 /
0301 /
0302 /
0303 P C O N
0304 - - - -

```





```

0403 JA PBACK /AND GO IN AND FIGURE IT OUT.
0404 /
0405 /
0406 /
0407 /
0410 /
0411 /
0412 /
0413 /
0414 /
0415 PTYPE, JA . /SETS UP THE CONVERSION PROCESS.
0416 FSTA MCRTP6 /SAVE FOR A SECOND.
0417 FLDA PAFTER /CLEAR TOTALS AND AFTER.
0420 FSTA PTOTAL
0421 FSTA PAFTER
0422 FLDA MCRTP1 /RECALL TRAILING PLACES.
0423 JSA PTLIT /IS IT OK?
0424 FSTA PAFTER /YEP, STORE AWAY.
0425 FLDA MCRTP6 /RECALL TOTAL?
0426 JSA PTLIT /IS IT OK?
0427 FSTA PTOTAL /YEP.
0430 JA PTYPE /RETURN NOW.
0431 /
0432 /
0433 /
0434 /
0435 PTLIT, JA . /ARGUMENT CHECKER.
0436 JLT PTYPE /IF MINUS EXIT.
0437 JAL PTYPE /IF TOO BIG EXIT.
0440 ALN 0 /ELSE FIX IT.
0441 FNORM /AND FLOAT IT.
0442 JA PTLIT /RETURN NOW.
0443 /
0444 /
0445 /
0446 PSTARS, FLDA PTOTAL /GET THE TOTAL NUMBER PLACES
0447 JSA MCSTAR /AND GENERATE STARS.
0450 JA PCON /NOW EXIT.
0451 /
0452 /
0453 /
0454 /
0455 PTOTAL, 0.0
0456 PAFTER, 0.0
0457 PMCB, 8. /EXPON+.,+LEADING+MISC.
0460 /
0461 /
0462 /
0463 /
0464 /
0465 /
0466 /
0467 /
0470 /
0471 /
0472 S
0473 IPREF ETYPE
0474 /
0475 /
0476 /
0477 /
0500 /

```

```

0501 / E T Y P E
0502 / * - * - *
0503 /
0504 /
0505 /
0506 /
0507 /
0510 /
0511 / COPYRIGHT 1971
0512 / DIGITAL EQUIPMENT CORPORATION
0513 / 146 MAIN STREET,
0514 / MAYNARD, MASSACHUSSETTS.
0515 / 01754
0516 /
0517 /
0520 /
0521 / WRITTEN BY JACK BURNES.
0522 /
0523 /
0524 /
0525 /
0526 /
0527 /
0530 /
0531 /
0532 /
0533 / EJECT
0534 /
0535 /
0536 /
0537 / E - CONVERSION ROUTINE.
0540 /
0541 /
0542 / ECON, JA . /CONVERSION ENTRY. AQ=NUMBER. BA
0543 / FSTA MCRTP8 /SAVE NUMBER AWAY.
0544 / FLDA ETOTAL /GET TOTAL NUMBER OF SPACES FOR
0545 / JSA MCPUTC /PUT IT IN HIS BUFFER.
0546 / FLDA MCRTP8 /RECALL HIS NUMBER.
0547 / JLT EMINUS /ITS MINUS. FIGURE FOR SIGN.
0550 / FLDA ETOTAL /RECALL TOTAL NUMBER OF SPACES.
0551 /
0552 / EBACK, FSUB EMC7 /DEDUCT FOR SPACES,,E,ETC.
0553 / FSUB EAFTER /SUBTRACT PRECISION DIGITS.
0554 / JLT ESTARS /WONT FIT. GIVE STARS.
0555 / JSA MCLBLK /PRODUCE NECESSARY NUMBER F BLAN
0556 / FLDA MCRTP8 /RECALL MAGIC NUMBER
0557 / JSA MCCPMS /GIVE MINUS SIGN IF PRESENT.
0560 / JSA MCPDOT /PRINT THE DECIMAL POINT.
0561 / FLDA MCRTP8 /RECALL THE MAGIC NUMBER.
0562 / JSA MCDIGC /CONVERT IT,BABY.
0563 / FLDA MCRTP7 /GET THE EXPONENT.
0564 / FSTA MCRTP8 /SAVE FOR A SECOND.
0565 / FLDA EAFTER /GET NUMBER OF PRECISION BITS.
0566 / JSA MCPRTD /PRINT DIGITS.
0567 / JSA MCPEXP /PRINT EXPONENT.
0570 / JA ECON /AND RETURN NOW.
0571 /
0572 /
0573 /
0574 /
0575 /
0576 / ETOTAL, 0.0

```

```

0577 EAFTER, 0.0
0600 EMC7, 7.
0601 /
0602 /
0603 /
0604 EMINUS, FLDA ETOTAL /ALLOW ONE PLACE FOR MINUS SIGN.
0605 FSUB FPPONE /CHOP OFF.
0606 JA EBACK /GO BACK INTO IT,BUBBY.
0607 /
0610 /
0611 /
0612 /
0613 ETYPE, JA . /ROUTINE TO SET UP CONVERSION.
0614 FSTA MCRTPE /SAVE THE TOTAL NUMBER IN A TEMP
0615 FCLA /NOW CLEAR OUT BOTH DIGIT COUNTS
0616 FSTA ETOTAL
0617 FSTA EAFTER
0620 FLDA MCRTPE /GET THE NUMBER AFTER.
0621 JSA ETLIT /CHECK IT.
0622 FSTA EAFTER /STORE BACK.
0623 FLDA MCRTPE /RECALL TOTAL
0624 JSA ETLIT /CHECK IT.
0625 FSTA ETOTAL /SAVE IT, BUBBY.
0626 JA ETYPE /RETURN.
0627 /
0630 /
0631 /
0632 ETLIT, JA . /ENTRY POINT.
0633 JLT ETYPE /RETURN IF NOT POSITIVE.
0634 JAL ETYPE /RETURN IF TOO BIG.
0635 ALN 0 /FIX IT UP.
0636 FNORM
0637 JA ETLIT /AND RETURN NOW.
0640 /
0641 /
0642 /
0643 ESTARS, FLDA ETOTAL
0644 JSA MCSTAR
0645 JA ECON /GENERATE STARS WHEN REQUIRED.
 /BY CALLING THE STAR ROUTINE.
 /AND THEN RETURN.
0646 /
0647 /
0650 /
0651 /
0652 /
0653 /
0654 /
0655 /
0656 $
0657 IFREF FCON
0660 /
0661 /
0662 /
0663 /
0664 /
0665 /
0666 F C O N
0667 * - * -
0670 /
0671 /
0672 /
0673 /
0674 /

```



```

0773 JSA MCONTC /PUT IT INTO THE BUFFER.
0774 JXN .+4,0+ /CHECK WHETHER END.
0775 JA FCON /IT IS.
0776 JXN FCONLP,2+ /STILL SKIPPING.
0777 XTA 0 /RECALL SPACES LEFT
1000 FNEG /MAKE INTO POSITIVE TYPE NUMBER
1001 JA .+4 /AVOID BAD LOAD.
1002 /
1003 FCONOK, FLDA FAFTER /RECALL NUMBER AFTER DECIMAL POI
1004 JSA MCONTC /PRINT THEM.
1005 JA FCON /AND EXIT NOW.
1006 /
1007 /
1010 /
1011 /
1012 FTYPE, JA . /SET UP ENTRY.
1013 FSTA MCONTC /SAVE TOTAL FOR A SECOND.
1014 FCLA /RESET BOTH TO ZERO.
1015 FSTA FAFTER /RECALL THE AFTER PART.
1016 FSTA FTOTAL /TEST IT
1017 FLDA MCONTP1 /AND STORE IT AWAY
1020 JSA FTLIT /RECALL TOTAL
1021 FSTA FAFTER /AND RETURN.
1022 FLDA MCONTP6
1023 JSA FTLIT
1024 FSTA FTOTAL
1025 JA
1026 /
1027 /
1030 /
1031 FTLIT, JA . /LITTLE CHECKER.
1032 JLT FTYPE /IF NEGATIVE EXIT.
1033 JAL FTYPE /IF TOO BIG, EXIT.
1034 ALN 0 /FIX IT.
1035 FNORM /AND FLOAT IT.
1036 JA FTLIT /AND RETURN.
1037 /
1040 /
1041 /
1042 FSTARS, FLDA FTOTAL /GIVE STARS IF NECESSARY.
1043 JSA MCONTC /AND RETURN
1044 JA FCON
1045 /
1046 /
1047 /
1050 FTOTAL, 0.0
1051 FAFTER, 0.0
1052 /
1053 /
1054 /
1055 /
1056 /
1057 S
1060 IFREF MCONTC
1061 /
1062 /
1063 /
1064 /
1065 /
1066 /
1067 /
1070 /

```

C O N V E R S I O N - P R O T O T Y P  
- - - - -



```

1167 MC1LOP, FLDA MCRTF6 /REGET THE NUMBER.
1170 FSUB MC2TAB+36-30000,1 /COMPARE AGAINST GIVEN EXPONEN
1171 JGE MCGRTO /ITS GREATER THEN EXPONENT. DONT
1172 FLDA MC1TAB+36-30000,1 /UPDATE THE NUMBER WE RE WORKI
1173 FMULM MCRTF6 /BY DOING A MULT TO MEMORY.
1174 FLDA MCRTF7 /AND UPDATE THE EXPONENT ALSO.
1175 FSUB MC3TAB+36-30000,1 /SUBTRACT OFF CORRECT NUMBER
1176 FSTA MCRTF7 /AND STORE AWAY.
1177 MCGRTO, JXN MC1LOP,1+ /AND LOOP TO TRY NEXT EXPONENT I
1200 JA MCDUN /WE ARE DONE NOW. CLEAN UP A BIT
1201 /
1202 /
1203 /
1204 /
1205 /
1206 /
1207 / THIS ROUTINE PROCESSES A NUMBER IF ITS
1210 / GREATER THEN 1.
1211 /
1212 /
1213 MCTOOB, FLDA MCRTF6 /REGET THE NUMBER
1214 FSUB MC1TAB+36-30000,1 /TEST AGAINST EXPONENT.
1215 JLT MCLESO /ITS LESS THEN EXPONENT, SO AVOI
1216 FLDA MCRTF6 /NOW FIX UP THE NUMBER.
1217 FDIV MC1TAB+36-30000,1 /BY DIVIDING BY THE CORRECT TA
1220 FSTA MCRTF6 /STORE AWAY NOW.
1221 FLDA MC3TAB+36-30000,1 /ALSO UPDATE THE EXPONENT COUN
1222 FADDM MCRTF7 /ZAP BACK INTO MEMORY.
1223 MCLESO, JXN MCTOOB,1+ /AND LOOP AROUND TILL DUNE.
1224 FLDA MCRTF6 /WE NOW MUST DO ONE MORE CORRECT
1225 FDIV MC10 /
1226 FSTA MCRTF6 /AND NOW STASH BACK.
1227 FLDA FPPONE /AND NOW CORRECT THE EXPONENT.
1230 FADDM MCRTF7 /FOR THE EXTRA DIVIDE.
1231 /
1232 /
1233 /
1234 MCDUN, FLDA MC1PRT /SET UP POINTER FOR DIGIT PRODUC
1235 FSTA MCD01
1236 FLDA MCRTF6 /REGET THE CONVERTED NUMBER
1237 STARTO /GO INTO DOUBLE FOR A SECOND.
1240 XTA 0 /JAM THE LOW 12 BITS INTO AN IND
1241 STARTF /GET BACK INTO SUPER MODE.
1242 ATX 0 /NOW GET THE INDEX REGISTER AS A
1243 JEQ MCDIGC /IF ITS ZERO DON T BOTHER FUDGI
1244 FLDA MCFUD /CORRECT WITH A FUDGE FACTOR.
1245 FADDM MCRTF6 /AND STORE BACK.
1246 JA MCDIGC /AND EXIT FOR NOW.
1247 /
1250 MC1PRT#.=1 /REMEMBER AC IS A THREE WORD QUA
1251 JA MCRE$1
1252 /
1253 /
1254 /
1255 /
1256 EJECT
1257 /
1260 /
1261 /
1262 /
1263 / THIS ROUTINE CONVERTS NUMBERS IF THEY ARE AN EXACT INTEG
1264 / IT MAY BE REMOVED AT ANY TIME.

```



```

1265 /
1266 /
1267 MCEXAC, FLDA MC2PRT /RESET THE SWITCH FOR THE EXACT
1270 FSTA MCDD1 /AND STORE AWAY.
1271 FLDA MCRT6 /RECALL THE NUMBER.
1272 JEQ MCEXEX /IF ZERO EXIT IMMEDIATELY.
1273 LDX =10,1 /SET FOR A COUNT OF 7 (MOST NUMB
1274 MCELOP, FLDA MCRT6 /GET THE NUMBER AGAIN.
1275 FDIV MC4TAB+30-30000,1 /TEST THE RANGES.
1276 ALN 0 /FIX IT.
1277 FNORM /THEN NORMALIZE IT.
1300 JNE MCFOUN /ITS NON-ZERO. WE VE GOT THE COR
1301 JXN MCELOP,1+ /KEEP LOOKING FOR(THIS LOOP CAN
1302 /
1303 MCFOUN, XTA 1 /GET THE INDEX IN THE AC.
1304 FNEG /PLAY WITH IT.
1305 FSUB FPPONE /ADD IN A FUDGE FACTOR.
1306 FSTA MCRT7 /AND SAVE IT AS OUR EXPONENT.
1307 MCEXEX, JA MCDIGC /AND EXIT FROM THS ROUTINE.
1310 /
1311 /
1312 /
1313 /
1314 /
1315 MC2PRT=,=1 /REMEMBER THRE WORD ARITHMETIC.
1316 /
1317 JA MCRES2 /SECOND CONVERTER.
1320 /
1321 /
1322 /
1323 /
1324 EJECT
1325 /
1326 /
1327 /
1330 /
1331 / THIS ROUTINE ACTUALLY GETS A DIGIT AND MAKES IT A NUMBER
1332 /
1333 /
1334 /
1335 MCGET4, FADD MC260 /EXIT IS TO HERE, SO IT FALLS TH
1336 /
1337 MCGETD, JA . /ENTRY.
1340 /
1341 MCDD1=,-1 /THIS CORRECTS FOR THE THREE WOR
1342 JA . /THIS INSTRUCTION IS OVERLAYED.
1343 /
1344 /
1345 /
1346 /
1347 /
1350 MCRES1, FLDA MCRT6 /GET MY NUMBER.
1351 FMUL MC10 /BOP UP BY 10 TO GET NEXT DECIMA
1352 FSTA MCRT6 /AND STORE IT AWAY NOW.
1353 ALN 0 /FIX THE AC NOW.
1354 FNORM /AND LEAVE IT NORMALIZED.
1355 FNEG /NEGATE THE AC.
1356 FADDM MCRT6 /THEN UPDATE THE NUMBER.
1357 FNEG /AND FINALLY RESTORE IT TO ITS P
1360 JA MCGET4 /AND RETURN WITH IT IN THE AC.
1361 /
1362 /

```

```

1363 /
1364 /
1365 /
1366 /
1367 /
1370 /
1371 /
1372 MCRES2, FLDA MCRT P7 /GET THE EXPONENT.
1373 JEQ MCGET4 /IF ITS 0 RETURN NOW.
1374 FSUB MC1 /SUBTRACT ONE FROM IT.
1375 FSTA MCRT P7 /AND SAVE FOR NEXT TIME AROUND.
1376 FLDA MCRT P6 /RECALL MY NUMBER.
1377 FDIV MC4TAB+25-30000,1+ /CHOP DOWN ANOTHER NOTCH.
1400 ATX 0 /SAVE FOR A SECOND.
1401 XTA 0 /AND RECALL IT INTEGERIZED.
1402 FNEG /AND REMOVE IT FROM MCMYNU.
1403 FMUL MC4TAB+25-30000,1 /BY MULTIPLYING IT BY TO CORRE
1404 FADDM MCRT P6 /SO THAT THE CHOPPING PROCESS CO
1405 XTA 0 /FINALLY RECALL THE CORRECT DIGI
1406 JA MCGET4 /AND RETURN
1407 /
1410 /
1411 /
1412 /
1413 /
1414 /
1415 EJECT
1416 /
1417 /
1420 /
1421 /
1422 /
1423 /
1424 THIS ROUTINE INSERTS ONE CHARACTER INTO THE BUFFER (AS P
1425 /
1426 /
1427 /
1430 MCPUTC, JA . /STANDARD CALL.
1431 FSTA MCRT P5 /STORE AC FOR A SECOND.
1432 FLDA I MCRT PB /GET THREE CHARACTERS NOW.
1433 FSTA MCBLK /STORE IN THE MIDDLE OF A LITTLE
1434 FLDA MCRT P5 /RECALL ARGUMENT
1435 ALN 0 /FIX IT.
1436 FSTA MCBLK=2 /PLACE IT RIGHT AHEAD OF PREVIOU
1437 FLDA MCBLK /SO ONE CHARACTER OVERLAPS.
1440 FSTA I MCRT PB /FINALLY STORE BACK THREE CHARAC
1441 STARTD /NOW WE MUST UPDATE THE POINTER.
1442 FLDA MCMAGIC /ADD IN THE MAGIC NUMBER.
1443 FADDM MCRT PB /UPDATE POINTER
1444 STARTF /RETURN TO SUPER MODE.
1445 JA MCPUTC /AND EXIT.
1446 /
1447 /
1450 0 /FIVE WORD BLOCK FOR CHARACTER SCH
1451 0
1452 MCBLK, 0
1453 0
1454 0
1455 /
1456 /
1457 /
1460 /

```

```

1461 /
1462 /
1463 /
1464 /
1465 /
1466 /
1467 EJECT
1470 /
1471 /
1472 /
1473 /
1474 /
1475 /
1476 MCSTAR, JA . /THIS ROUTINE PRINTS "N" PUTS AS
1477 JLE MCSTAR /RETURN IF ZERO OR NEGATIVE.
1500 FNEG
1501 ATX 1 /PLACE NEGATED COUNT INTO AN IND
1502 MCSLOP, FLDA MC252 /A STAR.
1503 JSA MCPPUTC /OUT IT GOES.
1504 JXN MCSLOP,1+
1505 JA MCSTAR /RETURN NOW.
1506 /
1507 /
1510 /
1511 /
1512 /
1513 /
1514 /
1515 /
1516 MCLBLK, JA . /THIS ROUTINE PRINTS LEADING BLA
1517 JLE MCLBLK /IF ZERO RETURN NOW.
1520 FNEG /ELSE NEGATE THE AC.
1521 ATX 0 /AND PLACE NUMBER OF BLANKS IN I
1522 MCBBLP, FLDA MC240 /GET A BLANK IN THE AC.
1523 JSA MCPPUTC /AND STORE IT AWAY.
1524 JXN MCBBLP,0+ /LOOP UNTIL DONE.
1525 JA MCLBLK /THEN RETURN.
1526 /
1527 /
1530 /
1531 /
1532 /
1533 /
1534 /
1535 MCPEXP, JA . /THIS ROUTINE PRINTS THE EXPONEN
1536 FLDA MC1 /FIRST PRINT ONE SPACE.
1537 JSA MCLBLK /BY CALLING ONE OF OUR ROUTINES.
1540 FLDA MC305 /THEN PRINT THE E.
1541 JSA MCPPUTC /BY CALLING A LOW LEVEL ROUTINE.
1542 FLDA MCRTPB /RECALL THE EXPONENT NOW.
1543 JGE MCGPLS /GIVE A PLUS SIGN
1544 JSA MCCPMS /GIVE A MINUS SIGN.
1545 FLDA MCRTPB
1546 FNEG
1547 JA MCCPL2 /BYPASS PLUS SIGN.
1550 MCGPLS, FLDA MC253 /THE PLUS SIGN.
1551 JSA MCPPUTC /OUTPUT IT NOW.
1552 FLDA MCRTPB /RECALL THE EXPONENT NOW.
1553 MCCPL2, FADD MC4F /ADD IN 1000 TO FUDGE UP THE EXP
1554 JSA MCDIGC /CONVERT IT.
1555 JSA MCGETD /IGNORE THE 1000 DIGIT.
1556 FLDA MC3 /NOW PRINT THE LAST THREE DIGITS

```

```

1557 JSA MCPRTL /BY CALLING OUR LITTLE ROUTINE.
1560 JA MCPEXP /AND NOW RETURN TO THE CALLER.
1561 /
1562 /
1563 /
1564 /
1565 MCPDOT, JA . /ENTRY.
1566 FLDA MC255 /GET THE DOT.
1567 JSA MCPUTC /AND PRINT IT
1570 JA MCPDOT /AND RETURN NOW.
1571 /
1572 /
1573 /
1574 /
1575 MCCPMS, JA . /THIS ROUTINE CONDITIONALLY PRIN
1576 JGE MCCPMS /IF NOT NEGATIVE RETURN
1577 FLDA MC255 /ELSE PRINT THE MINUS SIGN.
1600 JSA MCPUTC /AND PRINT IT.
1601 JA MCCPMS /AND RETURN NOW.
1602 /
1603 /
1604 /
1605 /
1606 /
1607 MCPRTD, JA . /THIS ROUTINE CONVERTS XXX NUMBE
1610 FNEG /NEGATE THE NUMBER FOR THE INDEX
1611 ATX 2 /SAVE IN AN INDEX.
1612 JGE MCPRTD /IF NO PLACES THEN EXIT.
1613 MCPRTL, JSA MCGETD /GET A DIGIT NOW.
1614 JSA MCPUTC /PRINT IT.
1615 JXN MCPRTL,2+ /LOOP UNTIL ALL DONE.
1616 JA MCPRTD /THEN RETURN NOW.
1617 /
1620 /
1621 /
1622 /
1623 /
1624 /
1625 /
1626 /
1627 /
1630 /
1631 /
1632 EJECT
1633 /
1634 /
1635 /
1636 /
1637 /
1640 /
1641 /
1642 /
1643 / TABLES FOR THE NUMERIC CONVERSIONS.
1644 /
1645 /
1646 /
1647 /
1650 MC1TAB, 3245
1651 3430
1652 6320 /1.E512 THE FPP ASSEMBLER CAN T
1653
1654 1523

```

|      |         |       |                                  |
|------|---------|-------|----------------------------------|
| 1635 |         | 2523  |                                  |
| 1636 |         | 7566  | /NOR THIS ONE EITHER.            |
| 1637 | /       |       |                                  |
| 1660 |         | 0652  |                                  |
| 1661 |         | 2235  |                                  |
| 1662 |         | 6444  | /1.E128 NOT YET, BUT WE RE GETTI |
| 1663 | /       |       |                                  |
| 1664 |         | 0325  |                                  |
| 1665 |         | 3023  |                                  |
| 1666 |         | 6020  | /ITS ALRIGHT NOW.                |
| 1667 | /       |       |                                  |
| 1670 |         | 1.E32 |                                  |
| 1671 |         | 1.E16 |                                  |
| 1672 |         | 1.E8  |                                  |
| 1673 |         | 1.E4  |                                  |
| 1674 |         | 1.E2  |                                  |
| 1675 | MC10,   | 1.E1  |                                  |
| 1676 | /       |       |                                  |
| 1677 | /       |       |                                  |
| 1700 | MC2TAB, | 4534  | / 1.E-312                        |
| 1701 |         | 2202  |                                  |
| 1702 |         | 2367  |                                  |
| 1703 | /       |       |                                  |
| 1704 |         | 6256  | / 1.E-256                        |
| 1705 |         | 3001  |                                  |
| 1706 |         | 4242  |                                  |
| 1707 | /       |       |                                  |
| 1710 |         | 7127  | / 1.E-128                        |
| 1711 |         | 3356  |                                  |
| 1712 |         | 4043  |                                  |
| 1713 | /       |       |                                  |
| 1714 |         | 7454  | / 1.E-64                         |
| 1715 |         | 2503  |                                  |
| 1716 |         | 7765  |                                  |
| 1717 | /       |       |                                  |
| 1720 |         | 7626  | / 1.E-32                         |
| 1721 |         | 3175  |                                  |
| 1722 |         | 4217  |                                  |
| 1723 | /       |       |                                  |
| 1724 |         | 7713  | / 1.E-16                         |
| 1725 |         | 3464  |                                  |
| 1726 |         | 5312  |                                  |
| 1727 | /       |       |                                  |
| 1730 |         | 7746  | / 1.E-8                          |
| 1731 |         | 2536  |                                  |
| 1732 |         | 3074  |                                  |
| 1733 | /       |       |                                  |
| 1734 |         | 7763  | / 1.E-4                          |
| 1735 |         | 3215  |                                  |
| 1736 |         | 5614  |                                  |
| 1737 | /       |       |                                  |
| 1740 |         | 1.E-2 |                                  |
| 1741 |         | 1.E-1 |                                  |
| 1742 | /       |       |                                  |
| 1743 | MC3TAB, | 512.  |                                  |
| 1744 |         | 256.  |                                  |
| 1745 |         | 128.  |                                  |
| 1746 |         | 64.   |                                  |
| 1747 |         | 32.   |                                  |
| 1750 |         | 16.   |                                  |
| 1751 |         | 8.    |                                  |
| 1752 |         | 4.    |                                  |

```

1753 2.
1754 MC1, 1.
1755 /
1756 MCMAGIC=MC1-1
1757 /
1758 /
1759 /
1760 /
1761 MC4TAB, 1000000.
1762 100000.
1763 10000.
1764 MC4F, 1000.
1765 100.
1766 10.
1767 1.
1770
1771 /
1772 /
1773 /
1774 /
1775 /
1776 /
1777 /
2000 /
2001 /
2002 /
2003 /
2004 /
2005 /
2006 /
2007 MCFUD, 7750
2010 2000
2011 0000
2012 MC260, 176.
2013 MC252, 170.
2014 MC240, 160.
2015 MC305, 197.
2016 MC253, 171.
2017 MC3, 3.
2020 MC256, 174.
2021 MC255, 173.
2022 /
2023 /
2024 /
2025 /
2026 EJECT
2027 /
2030 /
2031 /
2032 /
2033 /
2034 S
2035 IFREF ALOG10
2036 /
2037 /
2040 /
2041 /
2042 EJECT
2043 /
2044 /
2045 /
2046 /
2047 /
2050 A L O G 1 0
 * - * - * -

```

/DIVIDING BY 1. IS RIDICULOUS,  
/BUT IT ELIMINATES A SPECIAL CAS

/MAGIC FORMAT FUDGE CONSTANT.

/THESE ARE REALLY OCTAL NUMBERS.

```

2051 /
2052 /
2053 /
2054 /
2055 /
2056 /
2057 /
2060 /
2061 /
2062 /
2063 /
2064 /
2065 /
2066 /
2067 /
2070 /
2071 /
2072 /
2073 /
2074 /
2075 /
2076 /
2077 /
2100 /
2101 /
2102 /
2103 /
2104 /
2105 /
2106 /
2107 /
2110 /
2111 /
2112 /
2113 /
2114 /
2115 /
2116 /
2117 /
2120 /
2121 /
2122 /
2123 /
2124 /
2125 /
2126 /
2127 /
2130 /
2131 /
2132 /
2133 /
2134 /
2135 /
2136 /
2137 /
2140 /
2141 /
2142 /
2143 /
2144 /
2145 /
2146 /

```

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSSETTS, 01754

WRITTEN BY JACK BURNES.

EJECT

|            |        |                                 |
|------------|--------|---------------------------------|
| ALOG10, JA | .      | /THIS ROUTINE WILL TAKE THE LOG |
| JSA        | ALOG   | /OF THE ARGUMENT.               |
| FMUL       | ALOG1C | /CORRECT FOR THE LOG BASE E.    |
| JA         | ALOG10 | /AND RETURN.                    |

ALOG1C, 7777 /FUDGE CONSTANT.  
3362  
6754

\$  
IFREF SORT

EJECT

S O R T  
- - - -

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,





```

2245 ALN 0 /FIX IT UP, NOW.
2246 FSTA MCRTPI /AND STORE IT BACK FOR LATER USE
2247 /
2250 / SQRTX IS NOW 1/4 <X< 1
2251 /
2252 FLDA SQRTX+1 /RECALL NUMBER.
2253 FSTA MCRTPI /SAVE IN A TEMP.
2254 /
2255 FMUL SQRTS1,1 /MULTIPLY BY CORRECT CONSTANT.
2256 FADD SQRTS2,1 /AND NOW ADD IN CORRECT CONSTANT
2257 /
2260 / NOTE: INITIAL APPROXIMATION DEPENDS ON WHETHER X
2261 / 1/2<X<1
2262 /
2263 FSTA MCRTPI /SAVE IN A SECOND TEMP.
2264 FLDA MCRTPI /RECALL INITIAL.
2265 FDIV MCRTPI /CALCULATE X(0)/X(1)
2266 FADD MCRTPI /X(1)+X(0)/X(1)
2267 FDIV FPPTWO /1/2(X(1)+X(0)/X(1))
2270 FSTA MCRTPI /SAVE AGAIN. NOW X(2)
2271 FLDA MCRTPI /RECALL ORIGINAL.
2272 FDIV MCRTPI /X(0)/X(2)
2273 FADD MCRTPI /X(2)+X(0)/X(2)
2274 FSTA SQRTX+1 /NOW STORE AWAY FOR FINAL EXPONE
2275 /
2276 STARTD
2277 /
2300 FCLA
2301 FSTA SQRTX-1 /ZERO HIGH ORDER EXPONENT PART.
2302 FLDA MCRTPI /RECALL MODIFIED EXPONENT.
2303 FADDM SQRTX /UPDATE FRACTIONAL EXPONENT.
2304 /
2305 STARTF
2306 / /RETRUN TO FLOATING MODE.
2307 FLDA SQRTX+1 /PICK UP THE ANSWER.
2310 /
2311 / AND RETURN.
2312 /
2313 /
2314 SORT, JA . /SQUARE ROOT ENTRY AND EXIT.
2315 JGT SQRTOK /IF GREATER THEN 0 PROCEED.
2316 JEQ SQRT /IF ZERO JUST RETURN.
2317 TRAP7 3 /IF LESS THEN 0 GIVE A TRAP
2320 JA SQRT /BYE BYE.
2321 /
2322 /
2323 /
2324 SORTSC, FSUB FPPONE /SPECIAL CASE FUDGE.
2325 FSTA MCRTPI /SET EXPONENT ADD ON TO -1.
2326 FNEG
2327 JA SQRTBK /AND SET ODD BIT ON.
2330 / /AND GO BACK UP.
2331 /
2332 /
2333 /
2334 /
2335 /
2336 SORTS1, 0 /IF BETWEEN 1/4 & 1/2
2337 3200
2340 0
2341 0 /IF BETWEEN 1/2 & 1
2342 2240

```

```

2343 0
2344 /
2345 /SQRTS2, 7777 /IF BETWEEN 1/4 & 1/2
2346 2327
2347 7772
2350 7777 /IF BETWEEN 1/2 & 1
2351 3300
2352 0
2353 /
2354 /
2355 /
2356 /
2357 0 /THE MANTISSA NO EXPONENT DIDDLE
2360 0
2361 /SQRTEX, 0
2362 0
2363 /SQRT13, 0
2364 0
2365 13 /PHONEY EXPONENT PATCH.
2366 /
2367 /
2370 /
2371 $
2372 /IFREF ALOG
2373 /
2374 /
2375 /
2376 / A L O G
2377 / - - - -
2400 /
2401 /
2402 /
2403 /
2404 /
2405 /
2406 / COPYRIGHT 1971
2407 / DIGITAL EQUIPMENT CORPORATION
2410 / 146 MAIN STREET,
2411 / MAYNARD, MASSACHUSSETTS.
2412 / 01754
2413 /
2414 /
2415 /
2416 / WRITTEN BY JACK BURNES.
2417 /
2420 /
2421 /
2422 /
2423 /
2424 /
2425 /
2426 /
2427 /
2430 / EJECT
2431 /
2432 /
2433 /
2434 /
2435 /
2436 /
2437 /ALOGOK, LDX =1,0 /SET UP FOR POSITIVE SIGN.
2440 / PSTA MCRTP1 /SAVE IN A TEMP.

```

|      |           |        |            |                                  |
|------|-----------|--------|------------|----------------------------------|
| 2441 |           | F SUB  | F P P ONE  | /KNOCK OFF ONE.                  |
| 2442 |           | J EQ   | A LOG      | /IF ZERO EXIT. LOG(1)=0          |
| 2443 |           | J GE   | A LOG ST   | /IF POSITIVE LOG>0               |
| 2444 |           | F LDA  | F P P ONE  | /NEGITE. INVERT IT.              |
| 2445 |           | F DIV  | M CRT P1   | /BY DIVIDING INTO ONE.           |
| 2446 |           | F STA  | M CRT P1   |                                  |
| 2447 |           | L DX   | 0,0        | /RESET SIGN TO NEGATIVE.         |
| 2450 |           | J A    | .,+3       | /AVOID USELESS LOAD INSTRUCTION. |
| 2451 | /         |        |            |                                  |
| 2452 | A LOG ST, | F LDA  | M CRT P1   | /RECALL NUMBER.                  |
| 2453 |           | F DIV  | F P P TWO  | /CUT IN HALF.                    |
| 2454 |           | F STA  | A LOG TM   | /PREPARE FOR EXPONENT DIDDLE.    |
| 2455 |           | F LDA  | A LOG MG   | /SET THE EXPONENT OF THE EXPONEN |
| 2456 |           | F STA  | A LOG TM-3 | /SO THAT NORMALIZE WILL DO JOB.  |
| 2457 |           | F STA  | A LOG TM+1 | /AND ALSO ZERO OUT LOW ORDER POA |
| 2460 |           | F LDA  | A LOG TM-1 | /RECALL THE NUMBER               |
| 2461 |           | F NORM |            | /NORMALIZE IT.                   |
| 2462 |           | F MUL  | A LOG E2   | /NOW MULTITPLY EXPONENT BY LOG E |
| 2463 |           | F STA  | M CRT P2   | /AND SAVE IT FOR A SECOND.       |
| 2464 |           | F LDA  | M CRT P1   | /RECALL THE NUMBER AGAIN.        |
| 2465 |           | F STA  | A LOG TM   | /STORE IN THE TEMPORARY WORKER.  |
| 2466 |           | F LDA  | F P P I2-2 | /RECALL WORD WITH LOW OR         |
| 2467 |           | F STA  | A LOG TM-2 | /STORE AWAY.                     |
| 2470 |           | F LDA  | A LOG TM   | /RECALL NUMBER WITH AN EXPONENT  |
| 2471 |           | F SUB  | F P P ONE  | /SUBTRACT AWAY.                  |
| 2472 |           | F STA  | M CRT P1   | /AND STORE                       |
| 2473 |           | F MUL  | A LOG L8   | /MULTIPLY BY THE CONSTANT.       |
| 2474 |           | F ADD  | A LOG L7   | /ADD IN                          |
| 2475 |           | F MUL  | M CRT P1   | /MULT.                           |
| 2476 |           | F ADD  | A LOG L6   | /AND SO ON DOWN THE LINE.        |
| 2477 |           | F MUL  | M CRT P1   |                                  |
| 2500 |           | F ADD  | A LOG L5   |                                  |
| 2501 |           | F MUL  | M CRT P1   |                                  |
| 2502 |           | F ADD  | A LOG L4   |                                  |
| 2503 |           | F MUL  | M CRT P1   |                                  |
| 2504 |           | F ADD  | A LOG L3   |                                  |
| 2505 |           | F MUL  | M CRT P1   |                                  |
| 2506 |           | F ADD  | A LOG L2   |                                  |
| 2507 |           | F MUL  | M CRT P1   |                                  |
| 2510 |           | F ADD  | A LOG L1   |                                  |
| 2511 |           | F MUL  | M CRT P1   |                                  |
| 2512 |           | F ADD  | M CRT P2   | /CORRECT NOW,ADD IN EXPONENT.    |
| 2513 |           | J XN   | A LOG,0    | /EXIT IF SIGN IS OK.             |
| 2514 |           | F NEG  |            | /ELSE NEGATE IT.                 |
| 2515 | /         |        |            |                                  |
| 2516 | A LOG,    | J A    | .          | /ENTRY AND EXIT.                 |
| 2517 |           | J GT   | A LOG OK   | /IF GREATER THEN 0, START DOING  |
| 2520 |           | J EQ   | A LOG 0    | /IF =0 THEN ERROR                |
| 2521 |           | TRAP 7 | 6          | /LESS THEN 0.                    |
| 2522 |           | J A    | A LOG      | /RETURN.                         |
| 2523 | /         |        |            |                                  |
| 2524 | A LOG 0,  | TRAP 7 | 7          | /ZERO TRAP.                      |
| 2525 |           | J A    | A LOG      | /RETURN NOW.                     |
| 2526 | /         |        |            |                                  |
| 2527 | /         |        |            |                                  |
| 2530 | /         |        |            |                                  |
| 2531 | /         |        |            |                                  |
| 2532 | /         |        |            |                                  |
| 2533 | /         |        |            |                                  |
| 2534 | /         |        |            |                                  |
| 2535 | /         |        |            |                                  |
| 2536 | /         |        |            |                                  |

|      |         |      |                                  |
|------|---------|------|----------------------------------|
| 2537 | /       |      |                                  |
| 2540 | /       |      |                                  |
| 2541 | /       |      |                                  |
| 2542 |         | 0    | /WORKING SPACE FOR EXPONENT DIDD |
| 2543 |         | 0    |                                  |
| 2544 |         | 0    |                                  |
| 2545 | ALOGTM, | 0    |                                  |
| 2546 |         | 0    |                                  |
| 2547 |         | 0    |                                  |
| 2550 |         | 0    |                                  |
| 2551 | /       |      |                                  |
| 2552 | /       |      |                                  |
| 2553 | /       |      |                                  |
| 2554 | ALOGMG, | 0    |                                  |
| 2555 |         | 0    |                                  |
| 2556 |         | 13   | /CORRECT EXPONENT DIDDLE.        |
| 2557 | /       |      |                                  |
| 2560 | /       |      |                                  |
| 2561 | /       |      |                                  |
| 2562 | /       |      |                                  |
| 2563 | ALOGL1, | 0    |                                  |
| 2564 |         | 3777 |                                  |
| 2565 |         | 7742 |                                  |
| 2566 | /       |      |                                  |
| 2567 | ALOGE2, | 0    |                                  |
| 2570 |         | 2613 |                                  |
| 2571 |         | 4414 |                                  |
| 2572 | /       |      |                                  |
| 2573 | ALOGL2, | 7777 |                                  |
| 2574 |         | 4000 |                                  |
| 2575 |         | 4100 |                                  |
| 2576 | /       |      |                                  |
| 2577 | ALOGL3, | 7777 |                                  |
| 2600 |         | 2517 |                                  |
| 2601 |         | 0310 |                                  |
| 2602 | /       |      |                                  |
| 2603 | ALOGL4, | 7776 |                                  |
| 2604 |         | 4113 |                                  |
| 2605 |         | 7211 |                                  |
| 2606 | /       |      |                                  |
| 2607 | ALOGL5, | 7776 |                                  |
| 2610 |         | 2535 |                                  |
| 2611 |         | 3301 |                                  |
| 2612 | /       |      |                                  |
| 2613 | ALOGL6, | 7775 |                                  |
| 2614 |         | 4746 |                                  |
| 2615 |         | 0771 |                                  |
| 2616 | /       |      |                                  |
| 2617 | ALOGL7, | 7774 |                                  |
| 2620 |         | 2236 |                                  |
| 2621 |         | 4304 |                                  |
| 2622 | /       |      |                                  |
| 2623 | ALOGL8, | 7771 |                                  |
| 2624 |         | 4544 |                                  |
| 2625 |         | 1735 |                                  |
| 2626 | /       |      |                                  |
| 2627 | /       |      |                                  |
| 2630 | /       |      |                                  |
| 2631 | /       |      |                                  |
| 2632 | /       |      |                                  |
| 2633 | /       |      |                                  |
| 2634 | /       |      |                                  |

```

2635 /
2636 /
2637 $
2640 IFREF SIN
2641 /
2642 /
2643 /
2644 /
2645 /
2646 S I N
2647 - - -
2650 /
2651 /
2652 /
2653 /
2654 /
2655 /
2656 /
2657 /
2660 /
2661 /
2662 /
2663 /
2664 /
2665 /
2666 /
2667 /
2670 /
2671 /
2672 /
2673 /
2674 /
2675 /
2676 /
2677 /
2700 /
2701 /
2702 /
2703 /
2704 /
2705 /
2706 /
2707 /
2710 /
2711 /
2712 /
2713 /
2714 /
2715 /
2716 /
2717 /
2720 /
2721 /
2722 /
2723 /
2724 /
2725 /
2726 /
2727 /
2730 /
2731 /
2732 /

```

|         |       |        |                                  |
|---------|-------|--------|----------------------------------|
| SINABT, | FLDA  | MCRTP2 | /RECALL NUMBER IF TOO SMALE      |
|         | JXN   | SIN,0  | /EXIT IF SAME SIGN.              |
|         | FNEG  |        | /ELSE NEGATE IT.                 |
| SIN,    | JA    | .      | /CALCULATES THE SIGN OF FAC IN R |
|         | LDX   | =1,0   | /SET SIGN TO POSITIVE.           |
|         | JGT   | SINMOD | /IF POSITIVE BYPASS FUDGE.       |
|         | JEQ   | SIN    | /IF ZERO EXIT.                   |
|         | FNEG  |        | /NEGATIVE. NEGATE AC. SIN(-X)=-S |
|         | LDX   | 0,0    | /SET SIGN TO MINUS.              |
| SINMOD, | JAL   | SINER  | /IF SIGN CAN T INT, THEN ERROR.  |
|         | FDIV  | FPP2PI | /REDUCE TO BELOW TWO PI.         |
|         | FSTA  | MCRTP1 | /SAVE IN A TEMP.                 |
|         | ALN   | 0      |                                  |
|         | FNORM |        | /INTERGIZE IT.                   |
|         | FNEG  |        |                                  |

|      |         |      |         |                                  |
|------|---------|------|---------|----------------------------------|
| 2733 |         | FADD | MC RTP1 | /RECALL NUMBER, AC NOW <0        |
| 2734 |         | FMUL | FPP2PI  | /NOW MULTIPLY BACK.              |
| 2735 |         | FSTA | MC RTP2 | /AND SAVE AWAY,                  |
| 2736 |         | FSUB | FPPPI   | /SUBTRACT OFF PI.                |
| 2737 |         | JLT  | SINP    | /LESS THEN PI.                   |
| 2740 |         | FSTA | MC RTP2 | /RESTORE AS 2.                   |
| 2741 |         | XTA  | 0       | /INVERT THE SIGN.                |
| 2742 |         | FNEG |         |                                  |
| 2743 |         | FSUB | FPPONE  | /SIN(X-PI)=-SIN(X)               |
| 2744 |         | ATX  | 0       | /AND PUT BACK.                   |
| 2745 | /       |      |         |                                  |
| 2746 | SINP,   | FLDA | MC RTP2 | /RECALL MAGIC GOODY.             |
| 2747 |         | FSUB | FPPPI2  | /TEST TO SEE IF X<PI/2           |
| 2750 |         | JLT  | SINPP   | /YEP.                            |
| 2751 | /       |      |         |                                  |
| 2752 |         | FLDA | FPPPI   | /SIN(X)=SIN(PI-X)                |
| 2753 |         | FSUB | MC RTP2 |                                  |
| 2754 |         | FSTA | MC RTP2 | /AND STORE IT BACK.              |
| 2755 | /       |      |         |                                  |
| 2756 | SINPP,  | FLDA | MC RTP2 | /GET THE MAGIC NUMBER.           |
| 2757 |         | FSUB | SINTST  | /SEE IF ITS CLOSE TO AN EDGE     |
| 2760 |         | JLE  | SINABT  | /IT IS, AVOID ITERATION, SIN(X)= |
| 2761 |         | FLDA | MC RTP2 | /RECALL NUMBER TO BE WORKED ON.  |
| 2762 |         | FDIV | FPPPI2  | /DIVIDE BY PI OVER TWO.          |
| 2763 |         | FSTA | MC RTP2 | /AND STORE BACK.                 |
| 2764 |         | FMUL | MC RTP2 | /MULTIPLY OUT.                   |
| 2765 |         | FSTA | MC RTP1 |                                  |
| 2766 |         | FMUL | SINC9   | /NOW DO THE STANDARD ITERATION.  |
| 2767 |         | FADD | SINC7   |                                  |
| 2770 |         | FMUL | MC RTP1 |                                  |
| 2771 |         | FADD | SINC5   |                                  |
| 2772 |         | FMUL | MC RTP1 |                                  |
| 2773 |         | FADD | SINC3   |                                  |
| 2774 |         | FMUL | MC RTP1 |                                  |
| 2775 |         | FADD | FPPPI2  | /ADD IN PI OVER 2                |
| 2776 |         | FMUL | MC RTP2 | /DO THE FINAL MULTIPLY.          |
| 2777 |         | JXN  | SIN,0   | /SHALL WE NEGATE                 |
| 3000 |         | FNEG |         | /YEP                             |
| 3001 |         | JA   | SIN     | /AND RETURN.                     |
| 3002 | /       |      |         |                                  |
| 3003 | /       |      |         |                                  |
| 3004 | /       |      |         |                                  |
| 3005 | /       |      |         |                                  |
| 3006 | SINC9,  | 7764 |         |                                  |
| 3007 |         | 2501 |         |                                  |
| 3010 |         | 7015 |         |                                  |
| 3011 | /       |      |         |                                  |
| 3012 | SINC7,  | 7771 |         |                                  |
| 3013 |         | 5464 |         |                                  |
| 3014 |         | 5515 |         |                                  |
| 3015 | /       |      |         |                                  |
| 3016 | SINC5,  | 7775 |         |                                  |
| 3017 |         | 2431 |         |                                  |
| 3020 |         | 5362 |         |                                  |
| 3021 | /       |      |         |                                  |
| 3022 | SINC3,  | 0000 |         |                                  |
| 3023 |         | 5325 |         |                                  |
| 3024 |         | 0414 |         |                                  |
| 3025 | /       |      |         |                                  |
| 3026 | SINTST, | 7770 |         |                                  |
| 3027 |         | 2000 |         |                                  |
| 3030 |         | 0000 |         |                                  |

```

3031 /
3032 /
3033 /
3034 /
3035 /
3036 SINER, TRAP7 5
3037 JA SIN /RETURN ON ERROR.
3040 /
3041 /
3042 /
3043 /
3044 /
3045 /
3046 /
3047 S
3050 IFREF ATAN
3051 /
3052 /
3053 /
3054 /
3055 /
3056 A T A N
3057 * * * *
3060 /
3061 /
3062 COPYRIGHT 1971
3063 DIGITAL EQUIPMENT CORPORATION
3064 146 MAIN STREET,
3065 MAYNARD, MASSACHUSSETTS.
3066 01754
3067 /
3070 /
3071 /
3072 WRITTEN BY JACK BURNES.
3073 /
3074 /
3075 /
3076 /
3077 /
3100 /
3101 /
3102 /
3103 /
3104 EJECT
3105 /
3106 /
3107 /
3110 /
3111 /
3112 /
3113 ATAN, JA . /NEW ARCTANGENT ROUTINE.
3114 LOX =1,0 /REMEMBER SIGN
3115 JGE .+5
3116 LOX 0,0 /SAVE THE SIGN.
3117 FNEG /NEGATE THE FAC [ABS]
3120 FSTA MCRT P1 /AND STORE AWAY.
3121 FSTA MCRT P2
3122 FSUB ATANC1 /TEST TO SEE IF TOO SMALL.
3123 JLE ATANBG /IT IS. ATAN(X)=X
3124 FSUB ATANC2 /TEST TO SEE IF TOO BIG.
3125 JLE ATANLW /IT ISNT.
3126 FLDA FPPONE /TO BIG. INVERT IT.

```

|      |          |        |         |                                  |
|------|----------|--------|---------|----------------------------------|
| 3127 |          | F DIV  | MCRT P1 |                                  |
| 3130 |          | F STA  | MCRT P1 |                                  |
| 3131 | /        |        |         |                                  |
| 3132 | ATANLW,  | F CLA  |         | /CLEAR OUT TEMP.                 |
| 3133 |          | F STA  | MCRT P3 |                                  |
| 3134 |          | F LDA  | MCRT P1 | /RECALL NUMBER.                  |
| 3135 |          | F SUB  | ATANC3  | /START THE KNOCKING OFF PROCESS. |
| 3136 |          | J LT   | ATANN T | /WRONG SECTOR.                   |
| 3137 |          | F LDA  | ATANC4  | /BOP UP ORIGINAL.                |
| 3140 |          | F ADDM | MCRT P1 |                                  |
| 3141 |          | F LOA  | ATANCJ  | /GET MAGIC NUMBER.               |
| 3142 |          | F DIV  | MCRT P1 |                                  |
| 3143 |          | F ADD  | ATANC4  |                                  |
| 3144 |          | F STA  | MCRT P1 |                                  |
| 3145 |          | F LDA  | ATANC5  |                                  |
| 3146 |          | F STA  | MCRT P3 |                                  |
| 3147 | /        |        |         |                                  |
| 3150 | ATANN T, | F LDA  | MCRT P1 | /RECALL AND SQUARE IT.           |
| 3151 |          | F MUL  | MCRT P1 |                                  |
| 3152 |          | F STA  | MCRT P4 | /YET ANOTHER TEMP.               |
| 3153 |          | F LDA  | ATANC6  |                                  |
| 3154 |          | F MUL  | MCRT P4 |                                  |
| 3155 |          | F ADD  | ATANC7  |                                  |
| 3156 |          | F MUL  | MCRT P4 |                                  |
| 3157 |          | F ADD  | ATANC8  |                                  |
| 3160 |          | F MUL  | MCRT P4 |                                  |
| 3161 |          | F ADD  | ATANC9  |                                  |
| 3162 |          | F MUL  | MCRT P4 |                                  |
| 3163 |          | F ADD  | FPPONE  |                                  |
| 3164 |          | F MUL  | MCRT P1 |                                  |
| 3165 |          | F ADD  | MCRT P3 |                                  |
| 3166 |          | F STA  | MCRT P1 |                                  |
| 3167 |          | F LDA  | MCRT P2 |                                  |
| 3170 |          | F SUB  | FPPONE  |                                  |
| 3171 |          | J LE   | ATANBG  |                                  |
| 3172 |          | F LDA  | ATANCH  |                                  |
| 3173 |          | F SUB  | MCRT P1 |                                  |
| 3174 |          | J A    | .+3     |                                  |
| 3175 | /        |        |         |                                  |
| 3176 | ATANBG,  | F LDA  | MCRT P1 |                                  |
| 3177 |          | J XN   | ATAN,0  |                                  |
| 3200 |          | F NEG  |         |                                  |
| 3201 |          | J A    | ATAN    |                                  |
| 3202 | /        |        |         |                                  |
| 3203 | /        |        |         |                                  |
| 3204 | /        |        |         |                                  |
| 3205 | /        |        |         |                                  |
| 3206 | /        |        |         |                                  |
| 3207 | ATANC1,  | +15    |         | /LOWER LIMIT TEST.               |
| 3210 |          | 2000   |         |                                  |
| 3211 |          | 0000   |         |                                  |
| 3212 | /        |        |         |                                  |
| 3213 | ATANC2,  | 0      |         | /UPPER LIMIT TEST.               |
| 3214 |          | 3777   |         |                                  |
| 3215 |          | 7000   |         |                                  |
| 3216 | /        |        |         |                                  |
| 3217 | ATANC3,  | +1     |         |                                  |
| 3220 |          | 2111   |         |                                  |
| 3221 |          | 4121   |         |                                  |
| 3222 | /        |        |         |                                  |
| 3223 | ATANC4,  | 1      |         |                                  |
| 3224 |          | 3355   |         |                                  |



3225 4754  
 3226 /  
 3227 ATANC5, 0  
 3230 2060  
 3231 2511  
 3232 /  
 3233 ATANC6, -3  
 3234 3023  
 3235 1227  
 3236 /  
 3237 ATANC7, -2  
 3240 5566  
 3241 7220  
 3242 /  
 3243 ATANC8, -2  
 3244 3146  
 3245 0740  
 3246 /  
 3247 ATANC9, -1  
 3250 5252  
 3251 5262  
 3252 /  
 3253 ATANCH, 1  
 3254 3110  
 3255 3755  
 3256 /  
 3257 ATANCJ, -4,  
 3260 /  
 3261 /  
 3262 /  
 3263 /  
 3264 /  
 3265 /  
 3266 \$  
 3267 IFREF EXP  
 3270 /  
 3271 /  
 3272 /  
 3273 /  
 3274 /  
 3275 E X P  
 3276 - - -  
 3277 /  
 3300 /  
 3301 /  
 3302 /  
 3303 /  
 3304 /  
 3305 /  
 3306 /  
 3307 /  
 3310 /  
 3311 /  
 3312 /  
 3313 /  
 3314 /  
 3315 /  
 3316 /  
 3317 /  
 3320 /  
 3321 /  
 3322 /

COPYRIGHT 1971  
 DIGITAL EQUIPMENT CORPORATION  
 146 MAIN STREET,  
 MAYNARD, MASSACHUSSETTS,  
 01754

WRITTEN BY JACK BURNES.

```

3323 /
3324 /
3325 /
3326 /
3327 /
3330 /
3331 EJECT
3332 /
3333 /
3334 /
3335 /
3336 /
3337 FLDA EXPFUD /AN EXIT.
3340 EXP, JA /ENTRY
3341 LDX =1,0 /PRESERVE SIGN.
3342 JGE .+5
3343 FNEG /IF NEGATIVE NEGATE IT.
3344 LDX 0,0 /AND REMEMBER IT,BUBBY.
3345 FMUL EXP2E /MULTIPLY TO BINARY TYPE.
3346 FSTA MC RTP1 /AND SAVE IT AWAY.
3347 JAL EXPR /CAN T FIX IT, ERROR.
3350 ALN 0 /FIX IT UP.
3351 FSTA MC RTP3 /AND SAVE IT.
3352 FNORM /NOW NORMALIZE FOR OUR COMPUTATI
3353 FNEG /NEGATE THE FAC
3354 FADD MC RTP1 /ADD IN BEFORE NORMAL.
3355 FSTA MC RTP1 /AND STORE BACK, NO FADDM
3356 FMUL MC RTP1 /NOW SQUARE IT.
3357 FSTA MC RTP2 /AND SAVE IT.
3360 FADD EXPDF /START THE ITERATION.
3361 FSTA MC RTP4 /SAVE IN ANOTHER TEMP.
3362 FLDA EXPCF /NEXT CONSTANT.
3363 FDIV MC RTP4 /AND DIVIDE INTO IT.
3364 FSUB MC RTP1 /SUBTRACT BACK NOW.
3365 FADD EXPAF /NEXT CONSTANT.
3366 FSTA MC RTP4 /AND SAVE AGAIN. KEEP THIS UP.
3367 FLDA EXPBF
3370 FMUL MC RTP2
3371 FADDM MC RTP4
3372 FLDA MC RTP1
3373 FDIV MC RTP4
3374 FMUL FPP2W0
3375 FADD FPPONE
3376 FSTA EXPFUD /NOW FIDDLE THE EXPONENT.
3377 STARTD
3400 FLDA MC RTP3
3401 FADDM EXPFUD-1 /EXPONENT UPDATE.
3402 STARTF
3403 JKN EXP-2,0 /NO INVERSION NECESSARY. RETURN.
3404 FLDA FPPONE /INVERT IT
3405 FDIV EXPFUD
3406 JA EXP
3407 /
3410 /
3411 /
3412 EXPAF, 4
3413 2372
3414 1402
3415 /
3416 EXPBF, 7774
3417 2157
3420 5157

```

```

3421 /
3422 EXPCF, 12
3423 5454
3424 343
3425 /
3426 EXPCF, 7
3427 2566
3430 5341
3431 /
3432 0
3433 EXPFUD, 0
3434 0
3435 0
3436 /
3437 /
3440 /
3441 EXPER, TRAP7 4
3442 JA EXP
3443 /
3444 /
3445 /
3446 EXP2E, 1
3447 2705
3450 2435
3451 /
3452 /
3453 /
3454 /
3455 5
3456 IFREF PUTSTR
3457 /
3460 EJECT
3461 /
3462 /
3463 /
3464 /
3465 /
3466 /
3467 P U T S T R
3470 - - - - -
3471 /
3472 /
3473 /
3474 /
3475 /
3476 /
3477 /
3500 /
3501 /
3502 /
3503 /
3504 /
3505 /
3506 /
3507 /
3510 /
3511 /
3512 /
3513 /
3514 /
3515 /
3516 /

```

COPYRIGHT 1971  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET,  
MAYNARD, MASSACHUSSETTS. 01754

WRITTEN BY JACK BURNES.

```

3517 /
3520 / EJECT
3521 /
3522 /
3523 /
3524 /
3525 /
3526 / THIS ROUTINE UNPACKS A STRING AND SENDS IT TO "P
3527 /
3530 /
3531 /
3532 /
3533 / PUTSTR, JA . /ENTRY POINT.
3534 / JSA PUTSRL /GET A CHARACTER NOW.
3535 / JEQ PUTSTR /IF ZERO EXIT.
3536 / FNEG /ELSE SEND IT TO AN INDEX REGIST
3537 / ATX 1
3540 / JSA PUTSRL /GET ANOTHER CHAR.
3541 / JSA PUTC /OUTPUT IT.
3542 / JXN ,=4,1+ /LOOP AROUND FOR AWHILE.
3543 / JA PUTSTR /AND THEN RETURN.
3544 /
3545 /
3546 /
3547 / PUTSRL, JA . /THE ENTRY POINT.
3550 / STARTD /GET INTO MINI-MODE
3551 / FLDA I MCRTPB /GET A DOUBLE WORD.
3552 / FSTA PUTSRB /AND SAVE AWAY.
3553 / FLDA FPPPI2-1 /NOW ADD "00000001" TO THE POINT
3554 / FADDM MCRTPB /AND STORE BACK.
3555 / STARTF /GO BACK INTO FLOATING MODE.
3556 / FLDA PUTSRB-2 /RECALL MYSTERY WORD.
3557 / FNORM /NORMALIZE IT.
3560 / JA PUTSRL /AND EXIT NOW.
3561 /
3562 /
3563 /
3564 /
3565 / 27 /NORMALIZATION FUDGE.
3566 / 0
3567 / PUTSRB, 0
3570 / 0
3571 /
3572 /
3573 /
3574 /
3575 / S
3576 /
3577 /
3600 /
3601 / FPPNXT.
3602 /
3603 /
3604 /

```

0000 ERRORS

FPOIV0 00462  
FPXLV 00532  
FPXUD 00437  
FPPRCO 00446  
FPIOTT 00464  
FPIIKP 00472  
FPPAC 00351  
FPPARG 00324  
FPPCDF 00507  
FPPCLR 00301  
FPPCTL 00200  
FPPDER 00463  
FPPDUT 00330  
FPPDV0 00245  
FPPEER 00436  
FPPEXT 00257  
FPPFAL 00314  
FPPFER 00447  
FPPFKO 00253  
FPPIER 00465  
FPPINI 00400  
FPPIGT 00261  
FPPIST 00525  
FPPITS 00540  
FPPJMP 00210  
FPPKNO 00231  
FPPLER 00275  
FPPLNC 00350  
FPPMG 00347  
FPPNXT 00617  
FPPOER 00533  
FPPOLD 00200  
FPPONE 00600  
FPPOGO 00306  
FPPOUT 00335  
FPPOVR 00247  
FPPPCS 00233  
FPPPI 00611  
FPPPI2 00606  
FPPRET 00316  
FPPSKT 00263  
FPPTAB 00353  
FPPTER 00531  
FPPTKP 00255  
FPPTTT 00530  
FPPTY 00312  
FPPTW0 00603  
FPPUER 00440  
FPPUND 00251  
FPPXXX 00346  
FPP2PI 00614  
MC RTP 77000  
MC RTP1 77030  
MC RTP2 77033  
MC RTP3 77036  
MC RTP4 77041  
MC RTP5 77044  
MC RTP6 77047  
MC RTP7 77052  
MC RTP8 77055



## INDEX

- Active parameter table, 5
- ASCII
  - character packing, 8
  - code conversion, 1
- Asterisk (\*) usage
  - printout, 10
  - RUBOUT, 7
- Automatic line editing, 6, 7
  
- Back slash ( \ ) usage, 7
- Bell signal, 7
- Buffered I/O routines, 6
  - length, 7
  - loading priority, 7
  
- Call format, 2
- Calling PDP-8 program, 8
- Carriage return (in GETC), 6, 7
- Chaining files, 12
- Character deletion, 7
- Character received exit function, 4
- Clear Nucleus function, 4
- Code conversion, 1
- Conditions, special, 1
- Constants, library, 17, 18
- Conversion
  - ASCII characters to floating point number, 9
  - code, 1
  - routines, 10, 11
- Counter, setting program, 2
- CRLF routine, 9
- CTRL/U, 7
- Customizing the library, 16
  - library constants, 16
  - modifying assembly sequence, 16
  
- Deletion of
  - characters, 7
  - line, 7
- Description of FPP Support Library, 1
- Division by zero, 2
  
- E conversion or format, 10
- Error condition indicators, 12
- Error exits, 17
- Errors, typing, 7
- Example program, 12, 13
- Exits, 3, 4
  - error, 17
  - zero divide, 2
- Exponent overflow, 3
  
- F conversion or format, 10
- Fail-to-start exit location, 4
- FEXIT, normal exit condition, 4
- Field limitation, 2
- Format
  - E, 10
  - F, 10
  - P, 11
- FPHLT, 4
- FPPORG, 7
- Fraction overflow exit, 3
- Functions, mathematical, 1
  
- GETC, 6, 17
- GETNUM routine, 9
  
- Hardware requirements, 1
  
- Implementing mathematical routines, 12
- Instruction trap exit, 3
- Interrupts, 4, 5
  - PDP-8 mode, 2
- IOOUTW, 7
- IOT exit, 4
  
- Job termination, 7
  
- Library
  - constants, 17, 18
  - customizing, 16
  - modification, 16
- LINtapes, program library, 12
- Line
  - deletion, 7
  - editing, automatic, 6, 7
- Loading
  - buffered I/O, 6
  - location, 7
  - routines, 7
  
- Mathematical functions, 1
- Mathematical routines, 7
  - description, 11 through 16
  - implementation, 12
  - location, 7
  
- Normal exit condition (FEXIT), 4
- Nucleus, 1, 2
  - functions, 2, 3, 4

Output buffer full signal, 7  
 Overflow  
     condition, 13  
     exponent, 3  
     fraction exit, 3  
  
 P conversion or format, 11  
 PDP-8  
     mode interrupts, 2  
     program call, 8  
 Pointer, 4, 8  
 Program example, 12, 13  
 Program library LINCTapes, 12  
 PUTC, 6, 17  
 PUTSTR routine, 9  
  
 Routines, mathematical, 11  
 RUBOUT, 7  
  
 Scientific notation, 11  
 Set program counter, 2  
 Special conditions, 1  
 Start FPP function, 4  
  
 Termination of job, 7  
 Trap instructions, 3  
 TRAP3 instructions, 12  
 TRAP7  
     error exits, 17  
     instructions, 12  
  
 Underflow  
     condition, 13  
     exit, 3  
 Utility routines, 7, 8  
     description, 8, 9  
     exit locations, 8  
     location, 7  
     used without Nucleus, 17  
 Unusual events, 1  
  
 Zero divide, 2, 3  
     exit, 2



## 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



READER'S COMMENTS

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 \_\_\_\_\_

