DECUS
PROGRAM LIBRARY

DECUS NO.	8-103A
TITLE	FOUR WORD FLOATING POINT ROUTINES - FUNCTION PACKAGE
AUTHOR	D. A. Dalby and D. E. Wells
COMPANY	Bedford Institute of Oceanography Dartmouth, Nova Scotia, Canada
DATE	November, 1967
SOURCE LANGUAGE	PAL III

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

FOUR WORD FLOATING POINT FUNCTION PACKAGE

D. A. Dalby and D. E. Wells
Bedford Institute of Oceanography
Dartmouth, Nova Scotia, Canada
November, 1967

ABSTRACT

The program package was written for use with Digital 8-20-F Four Word Floating Point Package (now DEC-08-FMHA-D) and includes subroutines to evaluate square, square root, sine, cosine, arctangent, natural logarithm and exponential functions.

3. REQUIREMENTS

3.1 Storage

This package occupies locations 63, 4600-5577 5772-5777, 6166-6176, 6752-6776 and 7173.

The FOUR WORD FLOATING POINT PACKAGE (Digital 8-20-F) occupies locations 7, 40-61 and nearly all of 5600-7577.

Total storage requirements are 7, 40-61, 63 and practically all of 4600-7577.

3.3 Equipment

Standard PDP-8, ASR-33 Teletype.

4. USAGE

4.1 Loading

This package is loaded with the binary loader (Digital 8-2-U). The binary tape supplied contains both the Four Word Floating Point Package (Digital 8-20-F) and the Four Word Floating Point Function Package (BIO A-07-01). The symbolic tape contains only the Function Package (BIO A-07-01).

4.2 Calling Sequence

The function subroutines of this program are called through the "pseudo" operation code "0" special exits in the Four Word Floating Point Package (Digital 8-20-F) and the operation is performed on the current contents of the FAC (locations 44, 45, 46, 47).

The following instruction codes are used:

Function	Mnemonic	F.P. Instruction
Square	FSQ	0001
Square Root	FSQRT	0002
Sine	FSIN	0003
Cosine	FCOS	0004
Arctangent	FATN	0005
Natural logarithm	FLOG	0006
Exponential	FEXP	0007

5. RESTRICTIONS

5.1 Maximum Arguments

The exponential routine is limited to arguments less than $2047 \times \ln 2$.

The sine-cosine routine is limited to arguments less than $2047 \times 2\pi$.

6. DESCRIPTION

6.1 Discussion

See Digital 8-5-S Floating Point System manual and Digital 8-20-F Four Word Floating Point Package.

7. METHODS

7.1 Discussion

7.1.1 Square Root

The Newton-Raphson method of iteration is used and iteration proceeds until the result is within the least significant bit. A negative argument is signalled by C (61) $\neq$ 0.

7.1.2 Sine

The sine function is evaluated from the series

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}$$

The following identities are used:

$\sin(2\pi I + F) = \sin(F)$ I is an integer

$\sin(F) = -\sin(F - \pi)$ $F > \pi$

$\sin(F) = \sin(\pi - F)$ $F > \frac{\pi}{2}$

7.1.3 Cosine

The cosine function is evaluated from the sine series using the identity

$$\cos F = \sin\left(F + \frac{\pi}{2}\right)$$

7.1.4 Arctangent

A Chebyshev polynomial approximation is used to evaluate the arctangent function. The following series is used:

$$\arctan(x) = \sum_{K=0}^{11} C_{2K+1}\, x^{2K+1} \qquad -1 \leq x \leq +1$$

For $|x| > 1$ the relation

$$\arctan x = \frac{\pi}{2} - \arctan\left(\frac{1}{x}\right)$$ is used.

The constants used are:

$C_1 = 0.99999\ 99994\ 327$
$C_3 = -0.33333\ 32705\ 833$
$C_5 = 0.19999\ 79391\ 013$

C_7 = -0.14282 55494 507
C_9 = 0.11083 68974 645
C_{11} = -0.08941 17658 345
C_{13} = 0.07143 18607 648
C_{15} = -0.05251 58539 068
C_{17} = 0.03223 34073 423
C_{19} = -0.01472 14242 826
C_{21} = 0.00429 35922 689
C_{23} = -0.00058 76689 411

7.1.5 Natural Logarithm

The following Chebyshev polynomial approximation is used:

$$\ln (1 + x) = \sum_{k=1}^{12} C_k x^k \quad 0 \leq x \leq 1$$

For arguments greater than 2

$$\ln (y) = \ln (2^n \cdot F) = N \ln 2 + \ln F$$
$1 \leq F < 2$
N is an integer

For arguments less than 1

$$\ln (y) = - \ln (1/y) \quad 0 \leq y < 1$$

A negative argument is signalled by C (61) $\neq$ 0.

The constants used are:

C_1 = 0.99999 99936 903
C_2 = -0.49999 96251 745
C_3 = 0.33332 45482 811
C_4 = -0.24989 22048 312
C_5 = 0.19920 31876 192
C_6 = -0.16281 38105 985
C_7 = 0.12999 97465 301
C_8 = -0.09421 41492 545
C_9 = 0.05639 79927 825
C_{10} = -0.02497 11807 794
C_{11} = 0.00704 59951 115
C_{12} = -0.00093 33128 518

7.1.6 Exponential

The following Chebyshev polynomial is used:

$$\exp (- x) = \sum_{k=0}^{7} C_k x^k \quad 0 \leq x \leq \ln 2$$

For $x > \ln 2$

$$e^{-x} = \frac{1}{e^x} = \frac{1}{2^{N+F}} = \frac{1}{2^N} \cdot \frac{1}{2^F}$$

$$= \frac{1}{2^N} \cdot \frac{1}{e^{\ln 2 \cdot F}} = \frac{1}{2^N} \quad e^{-(\ln 2 \cdot F)}$$

For $x < 0$

$$e^x = \frac{1}{e^{-x}}$$

The constants used are:

$C_0 = 0.99999\ 99999\ 708$
$C_1 = -0.99999\ 99945\ 846$
$C_2 = 0.49999\ 98348\ 308$
$C_3 = -0.16666\ 47432\ 214$
$C_4 = 0.04165\ 56272\ 118$
$C_5 = -0.00829\ 87080\ 685$
$C_6 = 0.00132\ 79607\ 352$
$C_7 = -0.00014\ 08264\ 616$

7.3 Accuracy

The routines for square root, arctangent, natural logarithm and exponential yield better than 8 significant figures.

The sine-cosine routine yields at least 8 significant digits for arguments up to approximately 10π. The error can affect the fifth digit for arguments as large as 1000π.

8. FORMAT (not applicable)

9. EXECUTION TIME

Approximate execution times are the following:

Square Root	- less than 90 milliseconds for arguments less than 10^6.
Sine - Cosine	- 55 milliseconds.
Arctangent	- less than 70 milliseconds for arguments less than 10^3.
Logarithm	- 55 milliseconds.
Exponential	- 45 milliseconds.

10. PROGRAM

10.4 Program Listing follows this documentation.

11. DIAGRAMS

11.1 Flow Charts follow the program listing.

12. REFERENCES

12.2 Digital Manuals

Digital 8-5-S Floating Point System.

12.4 Textbooks

D.D. McCracken and W.S. Dorn, Numerical Methods and Fortran Programming, Wiley, 1964.

12.5 Periodicals

A.W. Duijvestijn and A.J. Dekkers, "Chebyshev Approximations of Some Transcendental Functions for Use in Digital Computing", Phillips Res. Rept., 16, (April 1961).

```
                /A-07-01

                /FOUR WORD FLOATING POINT
                /FUNCTION PACKAGE

                FSQ=0001
                FSQRT=0002
                FSIN=0003
                FCOS=0004
                FATN=0005
                FLOG=0006
                FEXP=0007
                FNEG=0016

                FPAC=0052


                /ADDITIONS, CHANGES, AND CORRECTIONS
                /TO FOUR WORD FLOATING POINT PACKAGE

                *63
0063   6261     NEG,        6261

                /TABLE6 ADDITIONS
                *5751
5751   5772                 SQUARE
5752   5000                 ROOT
5753   5200                 SINE
5754   5355                 COS
5755   5400                 ARTN
5756   4600                 LOG
5757   5076                 EXP

                *5766
5766   6261                 6261                    /NEGATE

                /SQUARE FAC
                *5772
5772   0000     SQUARE,     0
5773   4407                 JMS I 7
5774   6052                 FPUT FPAC
5775   3052                 FMPY FPAC
5776   0000                 FEXT
5777   5772                 JMP I SQUARE


                /CORRECTION TO FSUB ROUTINE
                *6030
6030   5364                 JMP 6164
                *6164
6164   4200                 JMS 6000
6165   5626                 JMP I 6026

                /RELOCATE PRCHAR ROUTINE FROM 6767
                *6166
6166   0000     PRCHAR,     0
6167   1057                 TAD 57
6170   7650                 SNA CLA
6171   5766                 JMP I PRCHAR
6172   1375                 TAD LFED
6173   4776                 JMS I OPUT
```

```
6174  5766              JMP I PRCHAR
6175  0212  LFED,       0212
6176  7345  OPUT,       7345

                /GET INTEGRAL PORTION OF FAC INTO FAC
                *6752
6752  0000  INT,        0
6753  1044              TAD 44
6754  7550              SPA SNA
6755  5365              JMP INT2
6756  1375              TAD M13
6757  3044              DCA 44
6760  1044  INT1,       TAD 44
6761  7700              SMA CLA
6762  5367              JMP INT3
6763  4777              JMS I SHFT
6764  5360              JMP INT1
6765  7200  INT2,       CLA
6766  3045              DCA 45
6767  3046  INT3,       DCA 46
6770  3047              DCA 47
6771  1376              TAD P13
6772  3044              DCA 44
6773  3050              DCA 50
6774  5752              JMP I INT

6775  7765  M13,        -13
6776  0013  P13,        13
6777  6116  SHFT,       6116

                /CALL PRCHAR AT NEW LOCATION
                *7173
7173  6166              6166
```

```
                    /FLOATING SQUARE ROOT
                    *5000
5000  0000  ROOT,    0
5001  1045           TAD 45
5002  7450           SNA
5003  5600           JMP I ROOT          /ZERO INPUT
5004  7700           SMA CLA
5005  5210           JMP ROOT1
5006  4463           JMS I NEG           /NEGATIVE INPUT
5007  7001           IAC
5010  3061  ROOT1,   DCA 61              /SET INPUT ERROR FLAG
5011  1044           TAD 44              /PUT INITIAL APPROX INTO ITER
5012  7100           CLL
5013  7510           SPA
5014  7020           CML
5015  7010           RAR
5016  3272           DCA ITER
5017  1210           TAD ROOT1
5020  3273           DCA ITER +1
5021  3274           DCA ITER+2
5022  3275           DCA ITER+3
5023  4407           JMS I 7             /PUT INPUT INTO FPAC
5024  6052           FPUT FPAC
5025  0000           FEXT
5026  5232           JMP ROOT3
5027  4407  ROOT2,   JMS I 7             /PUT FAC INTO ITER
5030  6272           FPUT ITER
5031  0000           FEXT
5032  4407  ROOT3,   JMS I 7             /GENERATE NEXT APPROX IN FAC
5033  5052           FGET FPAC
5034  4272           FDIV ITER
5035  1272           FADD ITER
5036  0000           FEXT
5037  7240           CLA CMA
5040  1044           TAD 44
5041  3044           DCA 44
5042  1044           TAD 44              /ITER=FAC WITHIN ONE BIT
5043  7041           CMA IAC
5044  1272           TAD ITER
5045  7640           SZA CLA
5046  5227           JMP ROOT2
5047  1045           TAD 45
5050  7041           CMA IAC
5051  1273           TAD ITER+1
5052  7640           SZA CLA
5053  5227           JMP ROOT2
5054  1046           TAD 46
5055  7041           CMA IAC
5056  1274           TAD ITER+2
5057  7640           SZA CLA
5060  5227           JMP ROOT2
5061  1047           TAD 47
5062  7041           CMA IAC
5063  1275           TAD ITER+3
5064  7500           SMA
5065  7041           CMA IAC
5066  7001           IAC
5067  7710           SPA CLA
5070  5227           JMP ROOT2
5071  5600           JMP I ROOT          /YES
```

```
                ITER,   0
5073    0000            0
5074    0000            0
5075    0000            0



                /FLOATING EXPONENT
5076    0000    EXP,    0
5077    1045            TAD 45
5100    7700            SMA CLA             /IS ARGUMENT NEGATIVE
5101    5304            JMP EXP1
5102    4463            JMS I NEG           /YES, NEGATE
5103    7240            CLA CMA
5104    7040    EXP1,   CMA
5105    3272            DCA ITER            /NEGATIVE ARGUMENT FLAG
5106    4407            JMS I 7
5107    4756            FDIV I ELGE2        /GET N+F
5110    6052            FPUT FPAC
5111    0000            FEXT
5112    4755            JMS I EINT          /DROP F
5113    1045            TAD 45
5114    7041            CMA IAC
5115    3273            DCA ITER+1          /STORE INTEGER -N
5116    4463            JMS I NEG           /FORM FLOATING (-N).
5117    4407            JMS I 7
5120    1052            FADD FPAC           /FORM F
5121    3756            FMPY I ELGE2
5122    6052            FPUT FPAC
5123    3766            FMPY I EK7
5124    1767            FADD I EK6
5125    3052            FMPY FPAC
5126    1770            FADD I EK5
5127    3052            FMPY FPAC
5130    1771            FADD I EK4
5131    3052            FMPY FPAC
5132    1772            FADD I EK3
5133    3052            FMPY FPAC
5134    1773            FADD I EK2
5135    3052            FMPY FPAC
5136    1774            FADD I EK1
5137    3052            FMPY FPAC
5140    1362            FADD EK0
5141    0000            FEXT
5142    1273            TAD ITER+1
5143    1044            TAD 44
5144    3044            DCA 44              /DIVIDE SUM BY 2 TO THE N.
5145    2272            ISZ ITER
5146    5676            JMP I EXP
5147    4407            JMS I 7               /POSITIVE ARGUMENT,
5150    6052            FPUT FPAC             /TAKE RECIPROCAL.
5151    5357            FGET ONE
5152    4052            FDIV FPAC
5153    0000            FEXT
5154    5676            JMP I EXP
```

```
5155  6752  EINT,     INT
5156  4714  ELGE2,    LGE2
5157  0001  ONE,      0001
5160  2000             2000
5161  0000             0000
5162  0000  EK0,      0000
5163  3777             3777
5164  7777             7777
5165  7775             7775

                /POINTERS TO EXPONENT SERIES CONSTANTS
5166  5364  EK7,      X7
5167  5370  EK6,      X6
5170  5374  EK5,      X5
5171  5560  EK4,      X4
5172  5564  EK3,      X3
5173  5570  EK2,      X2
5174  5574  EK1,      X1



                /FLOATING SINE
                *5200
5200  0000  SINE,     0
5201  1045             TAD 45
5202  7740             SMA SZA CLA
5203  5211             JMP SINI
5204  1045             TAD 45
5205  7700             SMA CLA
5206  5600             JMP I SINE          /ZERO INPUT
5207  4463             JMS I NEG           /NEGATIVE INPUT
5210  7240             CLA CMA
5211  3306  SINI,     DCA PNTR
5212  4407             JMS I 7               /REDUCE TO MODULO 2PI
5213  4315             FDIV PI
5214  4321             FDIV TWO
5215  6710             FPUT I XSQR
5216  0000             FEXT
5217  4707             JMS I SINT          /GET INTEGRAL PART OF FAC
5220  4463             JMS I NEG
5221  4407             JMS I 7
5222  1710             FADD I XSQR           /GET FRACTIONAL PART
5223  3315             FMPY PI
5224  3321             FMPY TWO
5225  6052             FPUT FPAC           /SIN(2 PI I + F)=SIN(F)
5226  2315             FSUB PI               /REDUCE TO MODULO PI
5227  0000             FEXT
5230  1045             TAD 45
5231  7710             SPA CLA
5232  5242             JMP SIN2
5233  4407             JMS I 7             /F G.T. PI. SIN (F-PI)=-SIN(F)
5234  6052             FPUT FPAC
5235  0000             FEXT
5236  1306             TAD PNTR
5237  7650             SNA CLA
5240  7040             CMA
5241  3306             DCA PNTR
5242  4407  SIN2,     JMS I 7               /REDUCE MODULO PI/2
```

```
5243  5052            FGET FPAC
5244  2311            FSUB PIOT
5245  0000            FEXT
5246  1045            TAD 45
5247  7710            SPA CLA
5250  5256            JMP SIN3
5251  4407            JMS I 7              /F G.T. PI/2. SIN(PI-F)=SIN(F)
5252  5315            FGET PI
5253  2052            FSUB FPAC
5254  6052            FPUT FPAC
5255  0000            FEXT
5256  4407  SIN3,     JMS I 7              /EVALUATE SERIES
5257  5052            FGET FPAC
5260  4311            FDIV PIOT              /REDUCE ARGUMENT L.T. 1
5261  6052            FPUT FPAC
5262  3052            FMPY FPAC
5263  6710            FPUT I XSQR
5264  3325            FMPY C13
5265  1331            FADD C11
5266  3710            FMPY I XSQR
5267  1335            FADD C9
5270  3710            FMPY I XSQR
5271  1341            FADD C7
5272  3710            FMPY I XSQR
5273  1345            FADD C5
5274  3710            FMPY I XSQR
5275  1351            FADD C3
5276  3710            FMPY I XSQR
5277  1311            FADD PIOT
5300  3052            FMPY FPAC
5301  0000            FEXT                 /SERIES EVALUATED
5302  2306            ISZ PNTR
5303  5600            JMP I SINE
5304  4463            JMS I NEG
5305  5600            JMP I SINE

5306  0000  PNTR,     0                    /NEGATION FLAG
5307  6752  SINT,     INT
5310  5072  XSQR,     IIER

5311  0001  PIOT,     0001
5312  3110            3110
5313  3755            3755
5314  2421            2421
5315  0002  PI,       0002
5316  3110            3110
5317  3755            3755
5320  2421            2421
5321  0002  TWO,      0002
5322  2000            2000
5323  0000            0000
5324  0000            0000

                /SINE CONSTANTS
5325  7750  C13,      7750
5326  3643            3643
5327  6415            6415
5330  1500            1500
5331  7756  C11,      7756
5332  4163            4163
5333  7054            7054
```

```
5335  7764  C9,       7764
5336  2501            2501
5337  7015            7015
5340  1042            1042
5341  7771  C7,       7771
5342  5464            5464
5343  5514            5514
5344  6150            6150
5345  7775  C5,       7775
5346  2431            2431
5347  5361            5361
5350  4736            4736
5351  0000  C3,       0000
5352  5325            5325
5353  0414            0414
5354  3167            3167

            /FLOATING COSINE
5355  0000  COS,      0
5356  4463            JMS I NEG
5357  4407            JMS I /
5360  1311            FADD PIOT          /COS(F)=SIN(PIBY2 - F)
5361  0000            FEXT
5362  4200            JMS SINE
5363  5755            JMP I COS

            /FLOATING EXPONENT CONSTANTS

5364  7764  X7,       7764
5365  5542            5542
5366  5227            5227
5367  4775            4775
5370  7767  X6,       7767
5371  2560            2560
5372  3573            3573
5373  7333            7333
5374  7772  X5,       7772
5375  5700            5700
5376  2131            2131
5377  0200            0200
```

```
                /FLOATING ARCTANGENT
                *5400
5400  0000  ARTN,      0
5401  1045              TAD 45
5402  7700              SMA CLA
5403  5206              JMP ATNI
5404  4463              JMS I NEG             /NEGATIVE INPUT
5405  7240              CLA CMA
5406  3274  ATNI,      DCA FLAG3
5407  4407              JMS I 7
5410  6052              FPUT FPAC
5411  2677              FSUB I AONE
5412  0000              FEXT
5413  1045              TAD 45
5414  7710              SPA CLA
5415  5224              JMP ATN2
5416  4407              JMS I /               /INPUT G.T. ONE
5417  5677              FGET I AONE           /TAKE RECIPROCAL
5420  4052              FDIV FPAC
5421  6052              FPUT FPAC
5422  0000              FEXT
5423  7240              CLA CMA
5424  3273  ATN2,      DCA FLAG2
5425  4407              JMS I 7               /EVALUATE SERIES
5426  5052              FGET FPAC
5427  3052              FMPY FPAC
5430  6676              FPUT I ARG2
5431  3300              FMPY K23
5432  1304              FADD K21
5433  3676              FMPY I ARG2
5434  1310              FADD MK19
5435  3676              FMPY I ARG2
5436  1314              FADD K17
5437  3676              FMPY I ARG2
5440  1320              FADD MK15
5441  3676              FMPY I ARG2
5442  1324              FADD K13
5443  3676              FMPY I ARG2
5444  1330              FADD MK11
5445  3676              FMPY I ARG2
5446  1334              FADD K9
5447  3676              FMPY I ARG2
5450  1340              FADD MK7
5451  3676              FMPY I ARG2
5452  1344              FADD K5
5453  3676              FMPY I ARG2
5454  1350              FADD MK3
5455  3676              FMPY I ARG2
5456  1354              FADD K1
5457  3052              FMPY FPAC
5460  0000              FEXT                  /SERIES EVALUATED
5461  2273              ISZ FLAG2
5462  5267              JMP ATN3
5463  4463              JMS I NEG
5464  4407              JMS I /
5465  1675              FADD I PI2              /ARTN(F)=PIBY2 - ARTN(1/F)
5466  0000              FEXT                    /FOR F G.T. ONE.
5467  2274  ATN3,      ISZ FLAG3
5470  5600              JMP I ARTN
5471  4463              JMS I NEG
5472  5600              JMP I ARTN
```

```
5473  0000  FLAG2,    0                    /INPUT G.T. ONE FLAG
5474  0000  FLAG3,    0                    /NEGATION FLAG
5475  5311  PI2,      PIOT
5476  5072  ARG2,     IIER
5477  5157  AONE,     ONE

                     /ARCTANGENT CONSTANTS
5500  7766  K23,      7766
5501  5457             5457
5502  4432             4432
5503  1701             1701
5504  7771  K21,      7771
5505  2145             2145
5506  4241             4241
5507  4605             4605
5510  7772  MK19,     7772
5511  4166             4166
5512  3357             3357
5513  4120             4120
5514  7774  K17,      7774
5515  2040             2040
5516  1626             1626
5517  5457             5457
5520  7774  MK15,     7774
5521  4507             4507
5522  1221             1221
5523  3170             3170
5524  7775  K13,      7775
5525  2222             2222
5526  2557             2557
5527  0167             0167
5530  7775  MK11,     7775
5531  5107             5107
5532  0475             0475
5533  7567             7567
5534  7775  K9,       7775
5535  3427             3427
5536  7472             7472
5537  2175             2175
5540  7776  MK7,      7776
5541  5555             5555
5542  7621             7621
5543  6402             6402
5544  7776  K5,       7776
5545  3146             3146
5546  3041             3041
5547  1767             1767
5550  7777  MK3,      7777
5551  5252             5252
5552  5253             5253
5553  5611             5611
5554  0000  K1,       0000
5555  3777             3777
5556  7777             7777
5557  7755             7755

                     /EXPONENT CONSTANTS
5560  7774  X4,       7774
5561  2524             2524
5562  7613             7613
5563  5106             5106
```

```
5564   7776   X3,        7776
5565   5252              5252
5566   5353              5353
5567   1521              1521
5570   7777   X2,        7777
5571   3777              3777
5572   7775              7775
5573   1652              1652
5574   0000   X1,        0000
5575   4000              4000
5576   0000              0000
5577   0275              0275
```

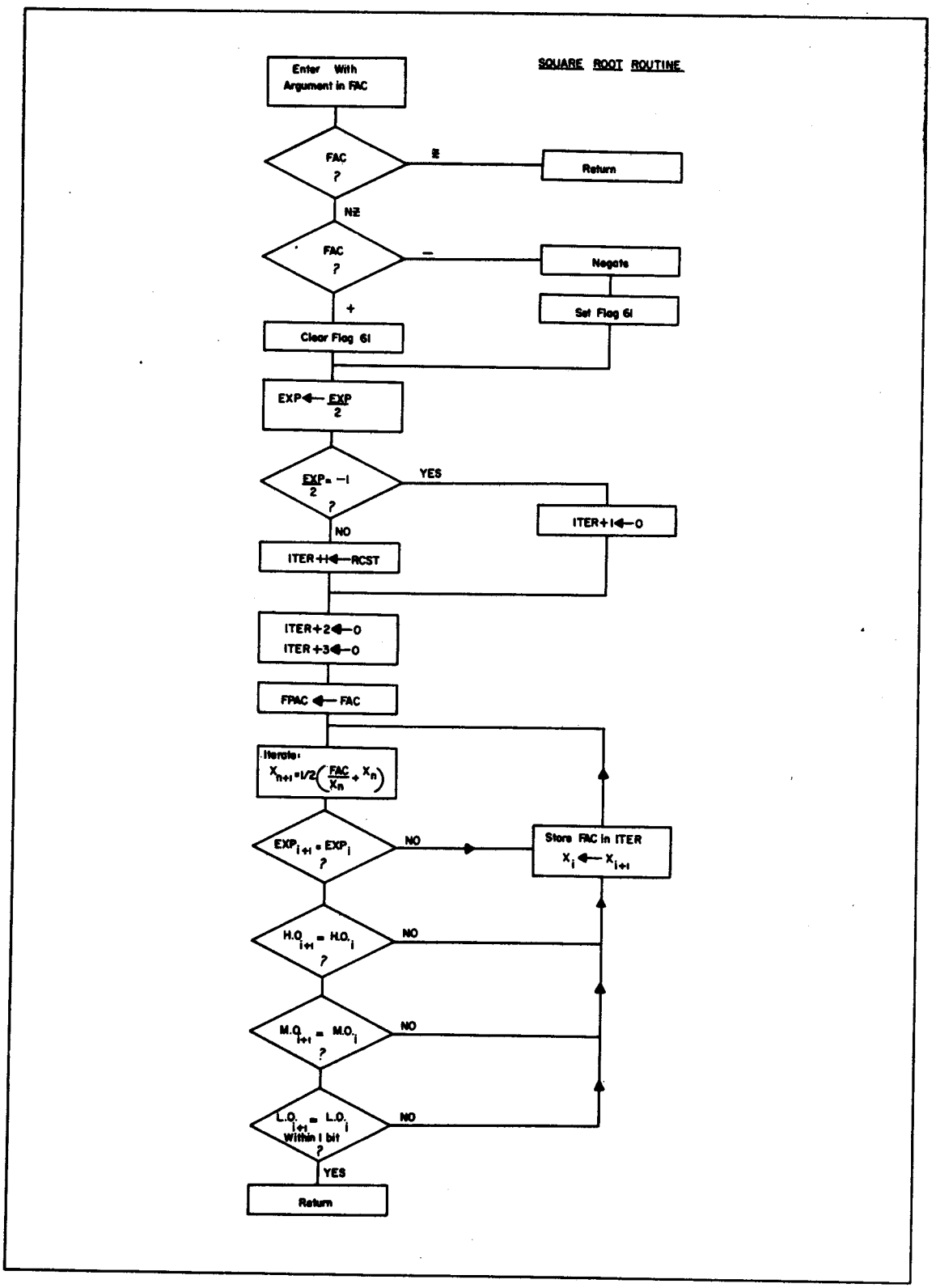
```
                /FLOATING LOGARITHM
                *4600
4600  0000  LOG,       0
4601  1045            TAD 45
4602  7700            SMA CLA
4603  5206            JMP LOG1
4604  4463            JMS I NEG             /NEGATIVE INPUT
4605  7001            IAC
4606  3061  LOG1,      DCA 61
4607  4407            JMS I 7
4610  6052            FPUT FPAC             /STORE POSITIVE ARGUMENT
4611  2705            FSUB I LONE
4612  0000            FEXT
4613  1045            TAD 45
4614  7700            SMA CLA               /ARG L.T. 1
4615  5224            JMP LOG2              /NO
4616  4407            JMS I 7               /INPUT L.T. 1, INVERT
4617  5705            FGET I LONE
4620  4052            FDIV FPAC
4621  6052            FPUT FPAC
4622  0000            FEXT
4623  7240            CLA CMA
4624  3307  LOG2,      DCA FLAG4             /FLAG4 SET IF ARG L.T. 1
4625  4407            JMS I 7               /RESTORE FAC
4626  5052            FGET FPAC
4627  0000            FEXT
4630  7040            CMA                   /GET N
4631  1044            TAD 44
4632  3045            DCA 45
4633  1313            TAD RA                /SET UP RETURN FROM INT
4634  3710            DCA I LINT
4635  5712            JMP I AFLOAT
4636  4407  BACK,      JMS I 7
4637  3314            FMPY LGE2
4640  6711            FPUT I LITER          /STORE N*LGE2
4641  5052            FGET FPAC
4642  0000            FEXT
4643  7001            IAC                   /GET F
4644  3044            DCA 44
4645  4407            JMS I 7
4646  2705            FSUB I LONE           /GET F-1
4647  6052            FPUT FPAC             /EVALUATE SERIES
4650  3320            FMPY L12
4651  1324            FADD L11
4652  3052            FMPY FPAC
4653  1330            FADD L10
4654  3052            FMPY FPAC
4655  1334            FADD L9
4656  3052            FMPY FPAC
4657  1340            FADD L8
4660  3052            FMPY FPAC
4661  1344            FADD L7
4662  3052            FMPY FPAC
4663  1350            FADD L6
4664  3052            FMPY FPAC
4665  1354            FADD L5
4666  3052            FMPY FPAC
4667  1360            FADD L4
4670  3052            FMPY FPAC
4671  1364            FADD L3
4672  3052            FMPY FPAC
```

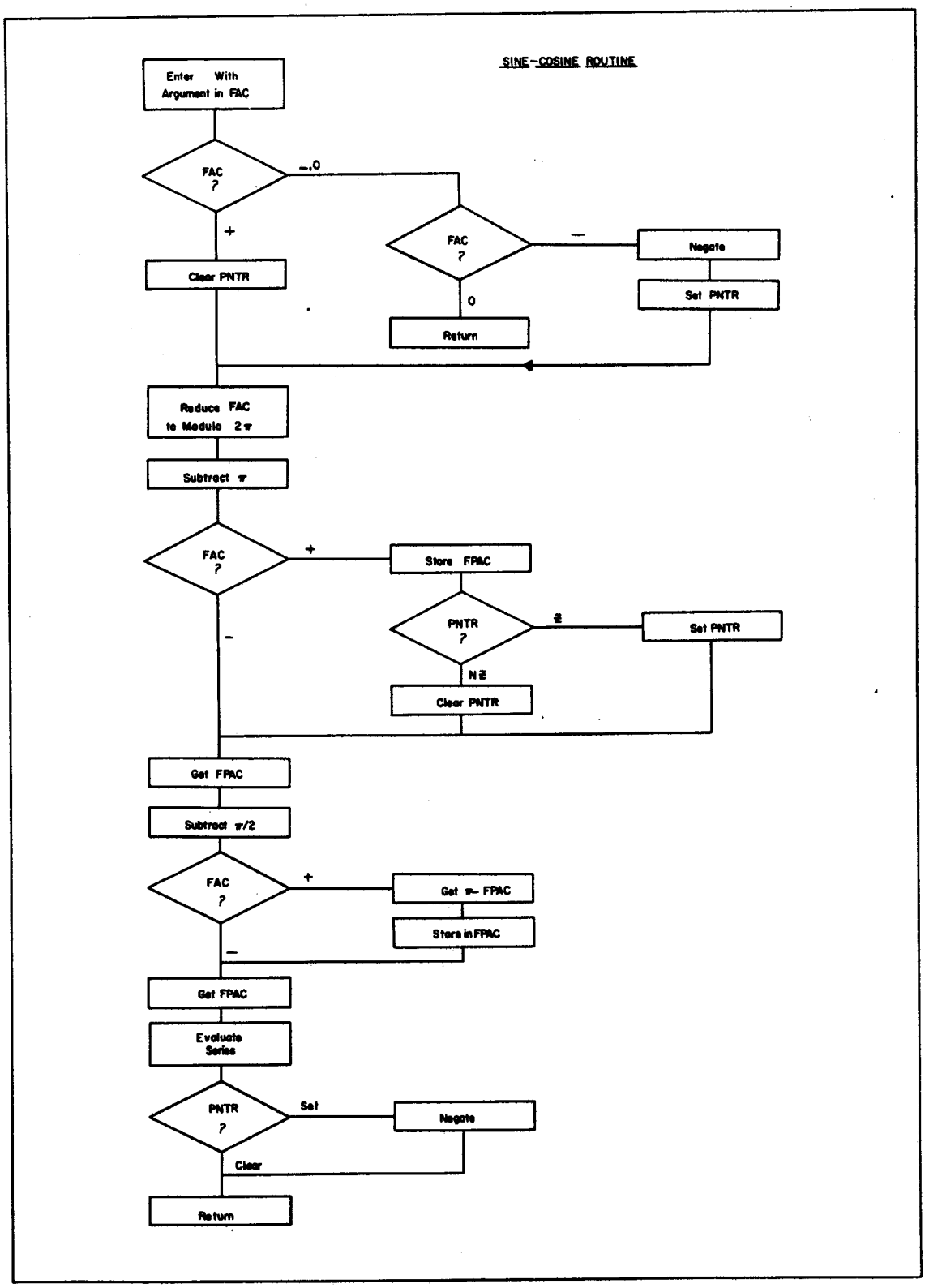
```
4673  1370            FADD L2
4674  3052            FMPY FPAC
4675  1374            FADD L1
4676  3052            FMPY FPAC               /SERIES NOW EVALUATED
4677  1711            FADD I LIIER            /ADD N*LGE2
4700  0000            FEXT
4701  2307            ISZ FLAG4               /WAS ARGUMENT L.T. 1
4702  5600            JMP I LOG
4703  4463            JMS I NEG               /YES, NEGATE SUM
4704  5600            JMP I LOG

4705  5157  LONE,     ONE
4706  5321  LTWO,     TWO
4707  0000  FLAG4,    0                       /NEGATION FLAG
4710  6752  LINT,     INT
4711  5072  LITER,    ITER
4712  6767  AFLOAT,   INT2+2
4713  4636  RA,       BACK

4714  0000  LGE2,     0
4715  2613            2613
4716  4413            4413
4717  7676            7676
                /LOGARITHM CONSTANTS
4720  7766  L12,      7766
4721  4132            4132
4722  5467            5467
4723  5141            5141
4724  7771  L11,      7771
4725  3467            3467
4726  0413            0413
4727  5110            5110
4730  7773  L10,      7773
4731  4633            4633
4732  3721            3721
4733  5500            5500
4734  7774  L9,       7774
4735  3470            3470
4736  0312            0312
4737  3507            3507
4740  7775  L8,       7775
4741  4770            4770
4742  3123            3123
4743  3611            3611
4744  7776  L7,       7776
4745  2050            2050
4746  7523            7523
4747  5173            5173
4750  7776  L6,       7776
4751  5312            5312
4752  1653            1653
4753  0406            0406
4754  7776  L5,       7776
4755  3137            3137
4756  6765            6765
4757  6402            6402
4760  7776  L4,       7776
4761  4000            4000
4762  7041            7041
4763  0031            0031
4764  7777  L3,       7777
4765  2525            2525
```

4766	2301		2301
4767	7431		7431
4770	7777	L2,	7777
4771	4000		4000
4772	0006		0006
4773	2241		2241
4774	0000	L1,	0000
4775	3777		3777
4776	7777		7777
4777	7445		7445

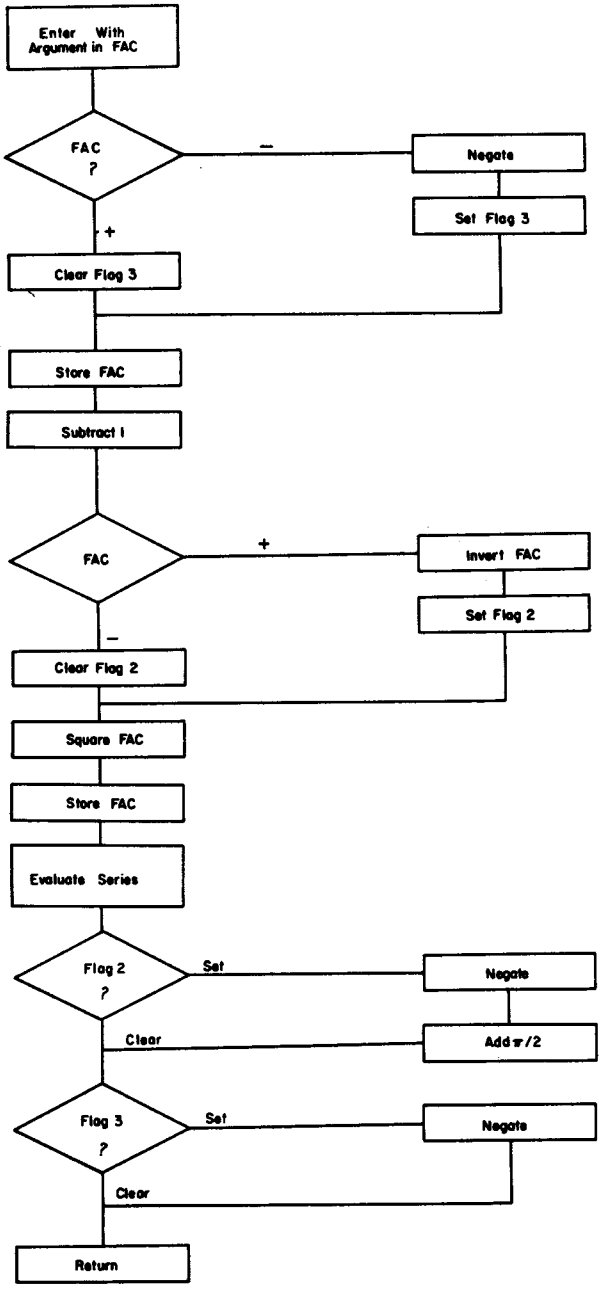
AFLOAT	4712
AONE	5417
ARG2	5416
ARTN	5400
ATN1	5406
ATN2	5424
ATN3	5467
BACK	4636
COS	5355
C11	5331
C13	5325
C3	5351
C5	5345
C7	5341
C9	5335
EINT	5155
EK0	5162
EK1	5174
EK2	5173
EK3	5172
EK4	5171
EK5	5170
EK6	5167
EK7	5166
ELGE2	5156
EXP	5016
EXP1	5104
FATN	0005
FCOS	0004
FEXP	0007
FLAG2	5413
FLAG3	5414
FLAG4	4707
FLOG	0006
FNEG	0016
FPAC	0052
FSIN	0003
FSQ	0001
FSQRT	0002
INT	6752
INT1	6760
INT2	6765
INT3	6767
ITER	5012
K1	5554
K13	5524
K17	5514

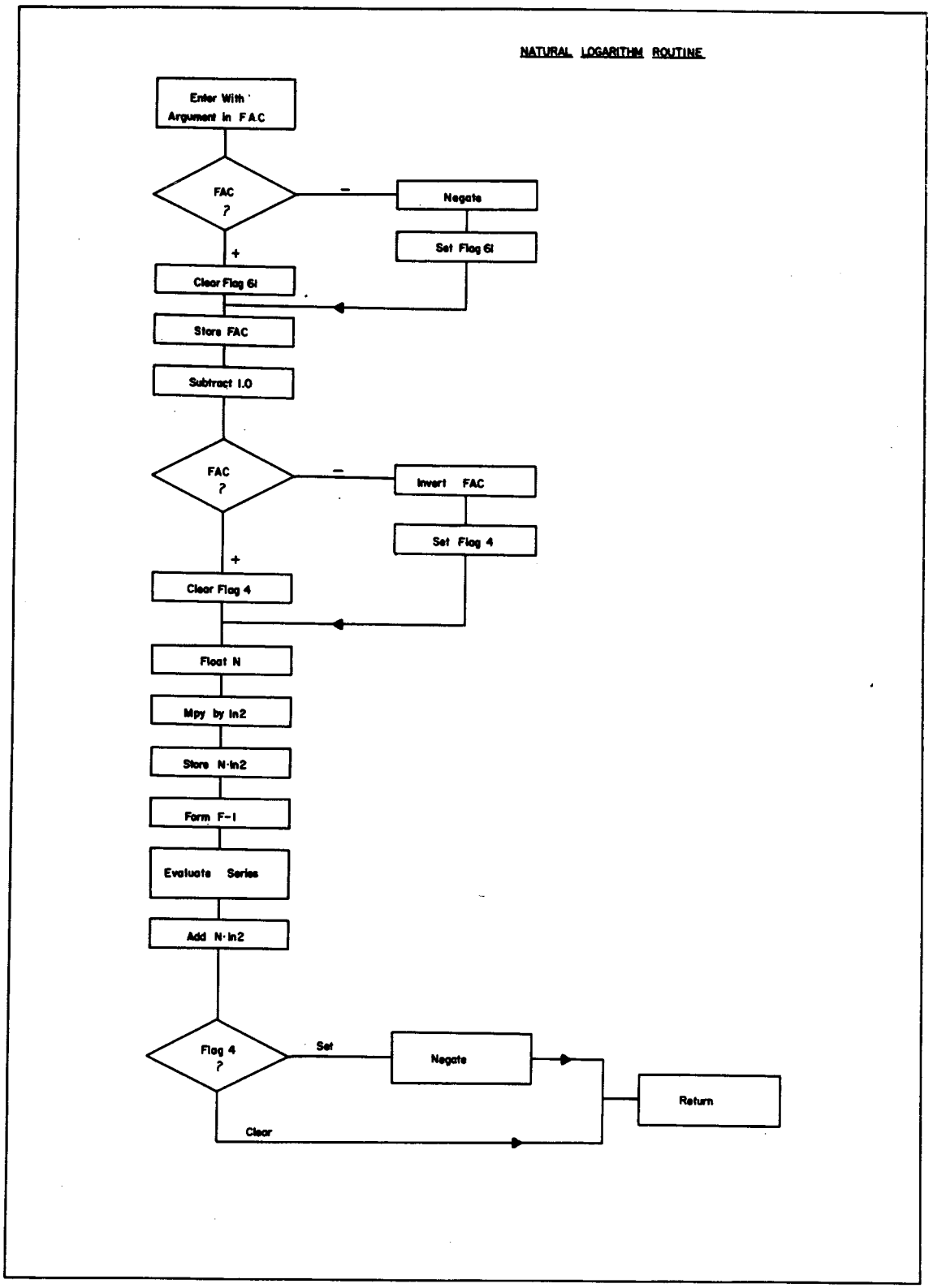
```
K21       5504
K23       5500
K5        5544
K9        5534
LFED      6175
LGE2      4714
LINT      4710
LITER     4711
LOG       4600
LOG1      4606
LOG2      4624
LONE      4705
LTWO      4706
L1        4774
L10       4730
L11       4724
L12       4720
L2        4770
L3        4764
L4        4760
L5        4754
L6        4750
L7        4744
L8        4740
L9        4734
MK11      5530
MK15      5520
MK19      5510
MK3       5550
MK7       5540
M13       6775
NEG       0063
ONE       5157
OPUT      6176
PI        5315
PIOT      5311
PI2       5475
PNTR      5306
PRCHAR    6166
PI3       6776
RA        4713
ROOT      5000
ROOT1     5010
ROOT2     5027
ROOT3     5032
SHFT      6777
SINE      5200
SINT      5307
SIN1      5211
SIN2      5242
SIN3      5256
SQUARE    5772
TWO       5321
XSQR      5310
X1        5574
X2        5570
X3        5564
X4        5560
X5        5374
X6        5370
X7        5364
≠
```

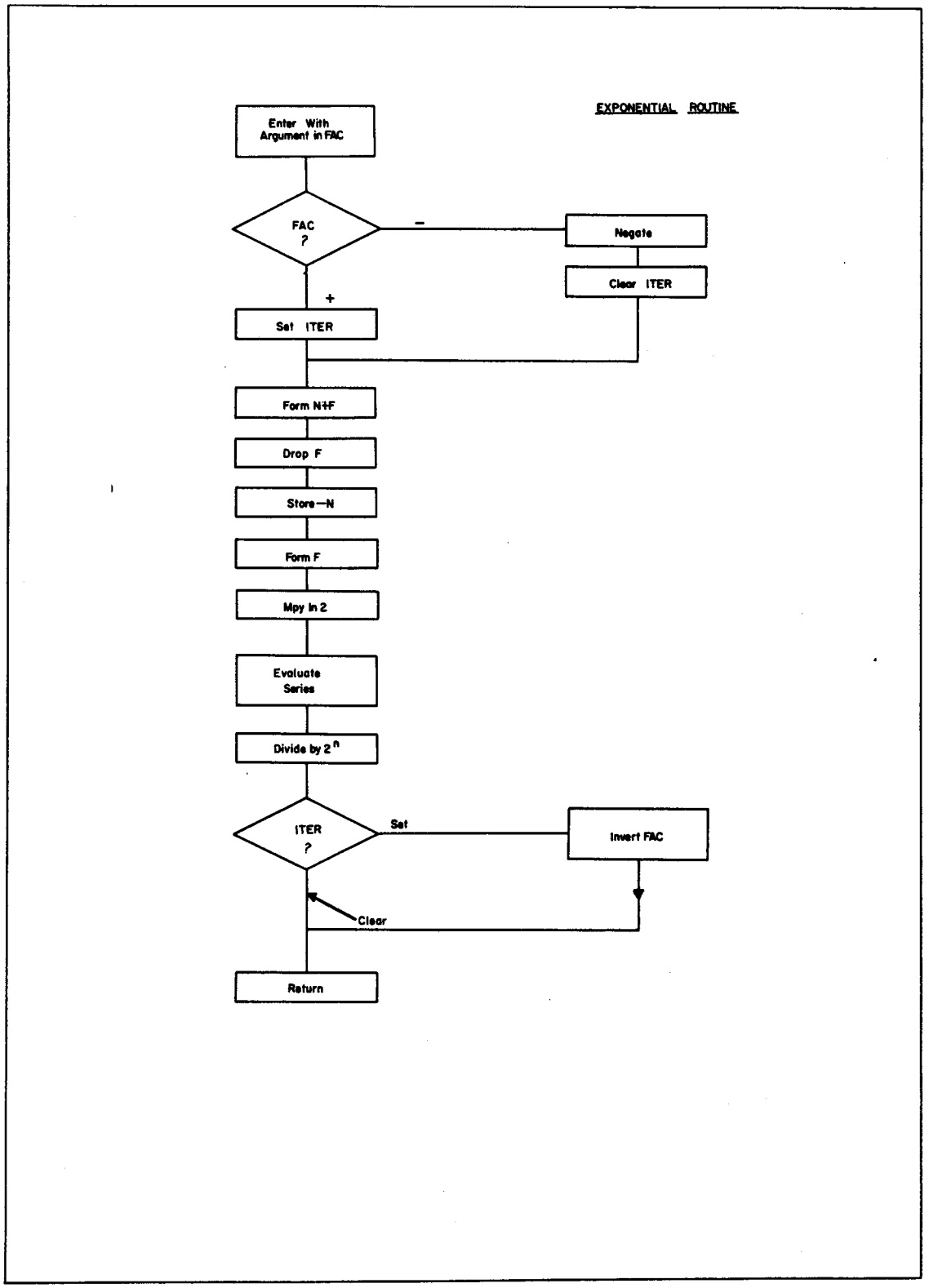

SQUARE ROOT ROUTINE
Enter With Argument in FAC
FAC ?
=
Return
≠0
FAC ?
−
Negate
Set Flag 6I
+
Clear Flag 6I
EXP ← EXP/2
EXP/2 = −1 ?
YES
ITER+1 ← 0
NO
ITER+1 ← RCST
ITER+2 ← 0
ITER+3 ← 0
FPAC ← FAC
Iterate: $X_{n+1}=1/2\left(\frac{FAC}{X_n}+X_n\right)$
$EXP_{i+1} = EXP_i$?
NO
Store FAC in ITER
$X_i \leftarrow X_{i+1}$
$H.O._{i+1} = H.O._i$?
NO
$M.O._{i+1} = M.O._i$?
NO
$L.O._{i+1} = L.O._i$ Within 1 bit ?
NO
YES
Return


SINE-COSINE ROUTINE
Enter With Argument in FAC
FAC ?
-,0
+
Clear PNTR
FAC ?
-
Negate
Set PNTR
0
Return
Reduce FAC to Modulo 2π
Subtract π
FAC ?
+
Store FPAC
PNTR ?
≥
Set PNTR
N≥
Clear PNTR
-
Get FPAC
Subtract π/2
FAC ?
+
Get π- FPAC
Store in FPAC
-
Get FPAC
Evaluate Series
PNTR ?
Set
Negate
Clear
Return

ARCTANGENT ROUTINE




NATURAL LOGARITHM ROUTINE
Enter With Argument in FAC
FAC ?
−
Negate
Set Flag 6I
+
Clear Flag 6I
Store FAC
Subtract 1.0
FAC ?
−
Invert FAC
Set Flag 4
+
Clear Flag 4
Float N
Mpy by ln2
Store N·ln2
Form F-1
Evaluate Series
Add N·ln2
Flag 4 ?
Set
Negate
Return
Clear


EXPONENTIAL ROUTINE
Enter With Argument in FAC
FAC ?
−
Negate
Clear ITER
+
Set ITER
Form N+F
Drop F
Store—N
Form F
Mpy In 2
Evaluate Series
Divide by 2^n
ITER ?
Set
Invert FAC
Clear
Return

Addendum to DECUS No.'s 8-103A, B, C, & D

Users of the Four Word Floating Point Package (Digital 8-20-F) will have noticed the lengthy execution times of the FMPY and FDIV operations in this package (approximately 3.4 msec. in each case).

Since there is not an EAE version of this package available a short patch which utilizes the EAE to speed up the FMPY operation to approximately 0.9 msec is given. This patch is of particular use with programs DECUS No.'s 8-103A, B, C, & D, since the extended functions frequently use the FMPY operation.

LOCATION	CONTENTS	MNEMONIC
		* 6333
6333	0000	MULTIP, 0
6334	3337	DCA . + 3
6335	1362	TAD MP2CON
6336	7425	MQL MUY
6337	0000	0
6340	3364	DCA MPSCON
6341	7501	MQA
6342	5733	JMP 1 MULTIP
		MP2 CON = 6362
		MPSCON = 6364