RAYTHEON

QUALITY SOFTWARE

700 PROGRAMMING SYSTEMS

SQUARE ROOT - MP FLOATING POINT

DATE: April 1968

ID CODE: BSH

DRAWING: 390006

LABEL: MSQR

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To extract the square root of a mid-precision floating point number and leave the result in the same format in the software registers MNT1, MNT2, MNT3.

USAGE

Calling Sequence

The routine returns at L+2.

```
L-1     SMB     MSQR
L       JSX     MSQR
L+1     D       ARG
L+2     Return
```

Argument Description

The argument is a mid-precision floating point number occupying three consecutive words in memory, beginning at address "ARG", as pointed out in the calling sequence.

METHOD

Let the argument X and its square root $\sqrt{X}$ be written in the floating point format:

$$X = Xm \cdot 2^{Xe}$$

$$\sqrt{X} = \sqrt{M} \cdot 2^{E}$$

Since the exponent E must be an integer:

$$M = Xm \; ; \; E = \frac{Xe}{2} \quad \text{(for Xe even)}$$

$$M = \frac{Xm}{2} \; ; \; E = \frac{Xe+1}{2} \quad \text{(for Xe odd)}$$

Except for zero, the argument is assumed to be normalized. An unnormalized argument is treated as zero. Therefore, M is within the range (1/4, 1):

$$.25 \leq M < 1$$

$\sqrt{M}$ is extracted by the Newton-Raphson method. To limit the process to two iterations, a first linear approximation is made to yield 8 bits of accuracy. This is done by dividing the range of M into six equal intervals which are allocated distinct linear coefficients.

$$A_0 = p \cdot M + q = \sqrt{M} \pm 2^{-9}$$

$$A_1 = 1/2 \left(\frac{M}{A_0} + A_0\right) = \sqrt{M} \pm 2^{-15}$$

$$A_2 = 1/2 \left(\frac{M}{A_1} + A_1\right) = \sqrt{M} \pm 2^{-30}$$

ERROR CONDITIONS

A negative argument is treated as zero. No error message is released.

ACCURACY

29 bits

RESTRICTIONS

Loading

This subroutine locally references storage words and constants located in the "MATH POOL" module, and must therefore be loaded in the same 2k block as the pool.

Other Routines

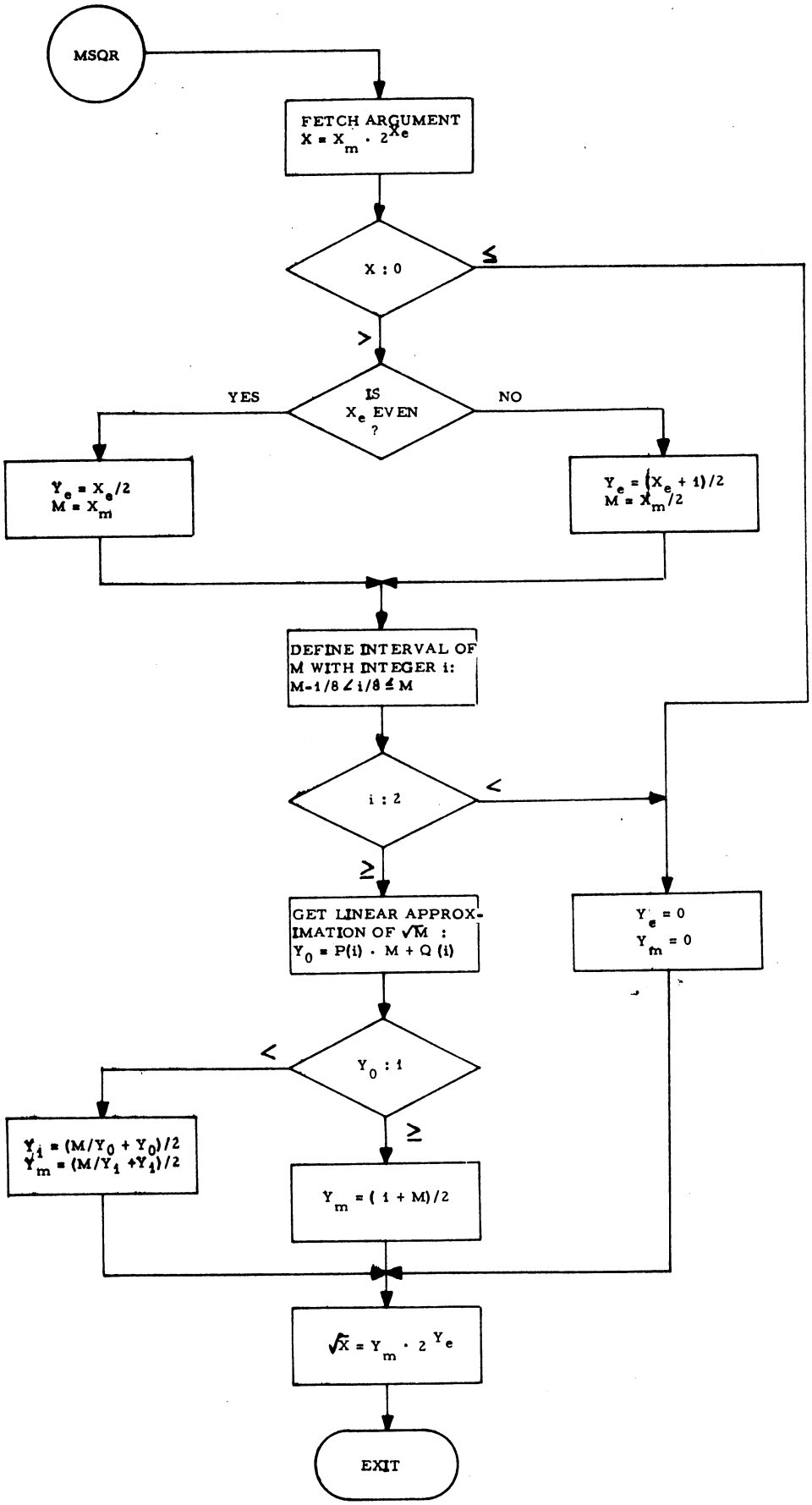
DIVS, M.ZE, MPYS

Space Used

82 words

Timing (in cycles)

	Standard	Hardware Multiply & Divide
Minimum	31	31
Maximum	730	232
Average	676	215


MSQR
FETCH ARGUMENT
X = X_m · 2^X_e
X : 0
≤
>
IS
X_e EVEN
?
YES
NO
Y_e = X_e/2
M = X_m
Y_e = (X_e + 1)/2
M = X_m/2
DEFINE INTERVAL OF
M WITH INTEGER i:
M-1/8 < i/8 ≤ M
i : 2
<
≥
GET LINEAR APPROX-
IMATION OF √M :
Y_0 = P(i) · M + Q (i)
Y_e = 0
Y_m = 0
Y_0 : 1
<
≥
Y_i = (M/Y_0 + Y_0)/2
Y_m = (M/Y_i + Y_i)/2
Y_m = (1 + M)/2
√X = Y_m · 2^Y_e
EXIT

APPENDIX A

ASSEMBLY LISTING

of

SQUARE ROOT - MP FLOATING POINT

Drawing No.

390006

ID Code

BSH

```
                           2   'SQUARE ROOT - MP FLOATING POINT 390006'                  BSH 0005
                           3   *                                                         BSH 0006
                           4   *                                                         BSH 0007
                           5             BLK   MATH                                      BSH 0008
                           6             LIBR  MSQR                                      BSH 0009
                           7   ***********************************************          BSH 0010
  0 000 0  7312  7312      8   K         DATA  29458     2(V6-2)                         BSH 0011
  0 001 0  6102  6102      9             DATA  24834     2(2V2-V6)                       BSH 0012
  0 002 0  515E  515E     10             DATA  20830     2(V3-V2)                        BSH 0013
  0 003 0  515F  515E     11             D     20830                                     BSH 0014
  0 004 0  4498  4498     12             DATA  17560     2(2-V3)                         BSH 0015
  0 005 0  4498  4498     13             D     17560                                     BSH 0016
  0 006 0  236A  236A     14   H         DATA  9066      (26-7V6)/32                     BSH 0017
  0 007 0  2A1D  2A1D     15             DATA  10781     (26V2-15V6)/32                  BSH 0018
  0 008 0  3215  3215     16             DATA  12821     (13V2-7V3)/16                   BSH 0019
  0 009 0  3215  3215     17             D     12821                                     BSH 0020
  0 00A 0  3B8F  3B8F     18             D     15247     1 - 2(2-V3)                     BSH 0021
  0 00B 0  3B69  3B69     19             DATA  15209     1 - 2(2-V3)(1-2**(-15))         BSH 0022
                          20   ***********************************************          BSH 0023
           000C  000C     21   MSQR      EQU   $                                         BSH 0024
* 0 00C 0  67FF  6 0 7FF  22             STX   RET2      SAVE RETURN                     BSH 0025
  0 00D 0  9800  9 1 000  23             LDX * 0                                         BSH 0026
  0 00E 0  8800  8 1 000  24             LDW * 0                                         BSH 0027
* 0 00F 0  A7FF  A 0 7FF  25             ADD   D129                                      BSH 0028
  0 010 0  0A41  0A4 1    26             SRC   1                                         BSH 0029
* 0 011 0  A7FF  A 0 7FF  27             ADD   B0                                        BSH 0030
* 0 012 0  E7FF  E 0 7FF  28             AND   M15R                                      BSH 0031
* 0 013 0  77FF  7 0 7FF  29             STW   MNT1      EXPONENT OF SQUARE ROOT         BSH 0032
  0 014 0  8801  8 1 001  30             LDW * 1                                         BSH 0033
  0 015 0  0810  0810     31             SAP                                             BSH 0034
  0 016 0  1026  1 0 026  32             JMP   SQRZ      NEGATIVE ARGUMENT               BSH 0035
  0 017 0  9802  9 1 002  33             LDX * 2                                         BSH 0036
  0 018 0  08A0  08A0     34             SNO             TEST PARITY OF ARGUMENT EXP,    BSH 0037
  0 019 0  101F  1 0 01F  35             JMP   EVEN                                      BSH 0038
  0 01A 0  0A31  0A3 1    36   ODD       SLL D 1                                         BSH 0039
  0 01B 0  0A01  0A0 1    37             SRL   1                                         BSH 0040
  0 01C 0  0A21  0A2 1    38             SRL D 1                                         BSH 0041
  0 01D 0  0A11  0A1 1    39             SLL   1                                         BSH 0042
  0 01E 0  0A21  0A2 1    40             SRL D 1                                         BSH 0043
* 0 01F 0  77FF  7 0 7FF  41   EVEN      STW   TMP4                                      BSH 0044
* 0 020 0  67FF  6 0 7FF  42             STX   TMP5                                      BSH 0045
* 0 021 0  77FF  7 0 7FF  43             STW   MNT3                                      BSH 0046
* 0 022 0  9011  9 0 011  44             LDX   B0                                        BSH 0047
  0 023 0  0A74  0A7 4    45             SLC D 4                                         BSH 0048
  0 024 0  0502  05 02    46             DXS   2                                         BSH 0049
  0 025 0  1028  1 0 028  47             JMP   SQR1      NORMALIZED ARGUMENT             BSH 0050
           0026  0026     48   SQRZ      EQU   $                                         BSH 0051
* 0 026 0  27FF  2 0 7FF  49             JSX   M,ZE      ZERO OR NON-NORMALIZED          BSH 0052
  0 027 0  104D  1 0 04D  50             JMP   EXIT      OR NEGATIVE ARGUMENT            BSH 0053
  0 028 0  0040  0040     51   SQR1      SLM                                             BSH 0054
  0 029 0  8806  8 1 006  52             LDW * H                                         BSH 0055
* 0 02A 0  77FF  7 0 7FF  53             STW   TMP3                                      BSH 0056
  0 02B 0  8800  8 1 000  54             LDW * K                                         BSH 0057
```

```
*     0 02C 0  27FF  2 0 7FF     55              JSX   MPYS                                          BSH 0058
*     0 02D 0  87FF  8 0 7FF     56              LDW   MNT2                                          BSH 0059
*     0 02E 0  A02A  A 0 02A     57              ADD   TMP3      Y1 = K * X + H                      BSH 0060
*     0 02F 0  901F  9 0 01F     58              LDX   TMP4            X **(1/2) *ORG,0024           BSH 0061
*     0 030 0  602D  6 0 02D     59              STX   MNT2                                          BSH 0062
*     0 031 0  9020  9 0 020     60              LDX   TMP5                                          BSH 0063
*     0 032 0  6021  6 0 021     61              STX   MNT3                                          BSH 0064
      0 033 0  08A0  08A0        62              SNC                                                 BSH 0065
      0 034 0  104F  1 0 04F     63              JMP   BIG                                           BSH 0066
*     0 035 0  702E  7 0 02E     64              STW   TMP3                                          BSH 0067
*     0 036 0  27FF  2 0 7FF     65              JSX   DIVS      X / Y1                              BSH 0068
*     0 037 0  A035  A 0 035     66              ADD   TMP3                                          BSH 0069
      0 038 0  0A01  0A0 1       67              SRL   1         Y2 =  1/2 (X/Y1 + Y1)               BSH 0070
*     0 039 0  7037  7 0 037     68              STW   TMP3                                          BSH 0071
*     0 03A 0  902F  9 0 02F     69              LDX   TMP4                                          BSH 0072
*     0 03B 0  6030  6 0 030     70              STX   MNT2                                          BSH 0073
*     0 03C 0  9031  9 0 031     71              LDX   TMP5                                          BSH 0074
*     0 03D 0  6032  6 0 032     72              STX   MNT3                                          BSH 0075
*     0 03E 0  2036  2 0 036     73              JSX   DIVS      X / Y2 HIGHER WORD                  BSH 0076
*     0 03F 0  703A  7 0 03A     74              STW   TMP4                                          BSH 0077
      0 040 0  0100  0100        75              CLR                                                 BSH 0078
*     0 041 0  703D  7 0 03D     76              STW   MNT3                                          BSH 0079
*     0 042 0  8039  8 0 039     77              LDW   TMP3                                          BSH 0080
*     0 043 0  203E  2 0 03E     78              JSX   DIVS      X / Y2 LOWER WORD                   BSH 0081
*     0 044 0  8042  8 0 042     79              LDW   TMP3                                          BSH 0082
*     0 045 0  A03F  A 0 03F     80    DONE      ADD   TMP4      Y = 1/2 (X/Y2 + Y2)                 BSH 0083
*     0 046 0  9041  9 0 041     81              LDX   MNT3                                          BSH 0084
      0 047 0  0A21  0A2 1       82              SRL D 1                                             BSH 0085
*     0 048 0  703B  7 0 03B     83              STW   MNT2                                          BSH 0086
      0 049 0  0140  0140        84              CXA                                                 BSH 0087
      0 04A 0  0810  0810        85              SAP                                                 BSH 0088
*     0 04B 0  B7FF  B 0 7FF     86              SUB   B1                                            BSH 0089
*     0 04C 0  7046  7 0 046     87              STW   MNT3                                          BSH 0090
*     0 04D 0  900C  9 0 00C     88    EXIT      LDX   RET2      RETURN TO CALLER                    BSH 0091
      0 04E 0  1801  1 1 001     89              JMP * 1                                             BSH 0092
      0 04F 0  0401  04 01       90    BIG       IXS   1                                             BSH 0093
      0 050 0  104D  1 0 04D     91              JMP   EXIT                                          BSH 0094
      0 051 0  1045  1 0 045     92              JMP   DONE                                          BSH 0095
                                 93    ********************************************************    BSH 0096
                                 94              NTRY  MSQR                                          BSH 0097
      0 051 0*******81           95
```

X-REF

```
    0 04F 0  BIG        0 034 0
EXT 0022     B0         0 011 0    0 022 0
EXT 004B     B1         0 048 0
EXT 0043     DIVS       0 036 0    0 03E 0    0 043 0
    0 045 0  DONE       0 051 0
EXT 000F     D129       0 00F 0
    0 01F 0  EVEN       0 019 0
    0 04D 0  EXIT       0 027 0    0 050 0
    0 006 0  H          0 029 0
    0 000 0  K          0 028 0
EXT 0026     M,ZE       0 026 0
EXT 0013     MNT1       0 013 0
EXT 0048     MNT2       0 02D 0    0 030 0    0 038 0    0 048 0
EXT 004C     MNT3       0 021 0    0 032 0    0 03D 0    0 041 0    0 046 0    0 04C 0
EXT 002C     MPYS       0 02C 0
LIB 0 00C 0  MSQR       0 00C 0
EXT 0012     M15R       0 012 0
    0 01A 0  ODD
EXT 004D     RET2       0 00C 0    0 04D 0
    0 026 0  SQRZ       0 016 0    0 026 0
    0 028 0  SQR1       0 025 0
EXT 0044     TMP3       0 02A 0    0 02E 0    0 035 0    0 037 0    0 039 0    0 042 0    0 044 0
EXT 0045     TMP4       0 01F 0    0 02F 0    0 03A 0    0 03F 0    0 045 0
EXT 003C     TMP5       0 020 0    0 031 0    0 03C 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
95	24	622	0	2

SIN AND COS - MP FLOATING POINT

DATE: April 1968

ID CODE: BSJ

DRAWING: 390007

LABEL: MSIN, MCOS

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To find the sine or the cosine of a mid-precision floating point number and leave the result in the same format in the software registers MNT1, MNT2, MNT3.

USAGE

Calling Sequence

The routine returns at L+2.

```
L-1     SMB     MSIN (MCOS)
L       JSX     MSIN (MCOS)
L+1     D       ARG
L+2     Return
```

Argument Description

The argument is given in radians, and is a mid-precision floating point number occupying three consecutive words in memory, beginning at address "ARG" as pointed out in the calling sequence.

METHOD

Let

$$F(X) = \sin X$$

or $F(X) = \cos X$

Through transformation of the given argument X, the sine function of a positive arc x is substituted to the given function F(X). The new arc x is defined in terms of two positive numbers, an integer N, and a fraction S.

$\cos X = \sin(\pi/2 + |X|)$
$\sin(-|X|) = \sin(\pi - X)$

$|X| = Q \cdot \pi/4 + S$
$N = Q - 8K$ for sin X and $X \geq 0$
$N = Q+4-8K$ for sin X and $X < 0$
$N = Q+2-8K$ for cos X

$x = N \cdot \pi/4 + S$
$F(X) = \sin x$
$0 \leq N < 8$
$0 \leq S < \pi/4$

N defines a half-quadrant, from 0 to 7. S is replaced by its complement if N is odd.

$Z = S$ if N is even
$Z = \pi/2 - S$ if N is odd
$f(Z) = \sin Z$ if N = 0, 3, 4, 7
$f(Z) = \cos Z$ if N = 1, 2, 5, 6

The function f(Z) is approximated by means of a 5-term Chebyshev polynomial:

$\sin Z = S_o \cdot Z + S_1 \cdot Z^3 \ldots + S_4 \cdot Z^9 \pm Se$
$\cos Z = C_o + C_1 \cdot Z^2 \ldots + C_4 \cdot Z^8 \pm Ce$

Finally the result is set negative, if N points to one of the last two quadrants:

$F(X) = f(Z)$ if $N < 4$
$F(X) = -f(Z)$ if $N \geq 4$

The coefficients of the polynomials are:

Sine		Cosine	
S_0	1.	C_0	.99999 99995
S_1	-.16666 6664	C_1	-.49999 9973
S_2	.83333 1376 E-2	C_2	.41666 4514 E-1
S_3	-.19835 7897 E-3	C_3	-.13882 8607 E-2
S_4	.26931 0165 E-5	C_4	.24112 6543 E-4
$\frac{Se}{\sin Z} < 2^{-31}$		$Ce \leq 2^{-31}$	

ERROR CONDITIONS

For argument values greater than or equal to 2^{13} the function is set to zero, not computed.

To form the function for arguments of this size, extra instructions would be required that would unfavorably affect the timing of the routine for all argument values. Since the results for large arguments would be of such poor accuracy (see following section) the function is arbitrarily set to zero.

ACCURACY

29 bits for arguments whose magnitude is under 1. Deduct 1 bit for each positive unit in the argument exponent, down to 16 bits. Beyond that arbitrary limit, the function is replaced by zero.

RESTRICTIONS

Loading

This subroutine locally references storage words and constants located in the "MATH POOL" module, and must therefore be loaded in the same 2K block as the pool.

Other Routines

DAD, DIVS, DSM, D2C, FLD, FSB, M.P, M.ZE, MPYS.

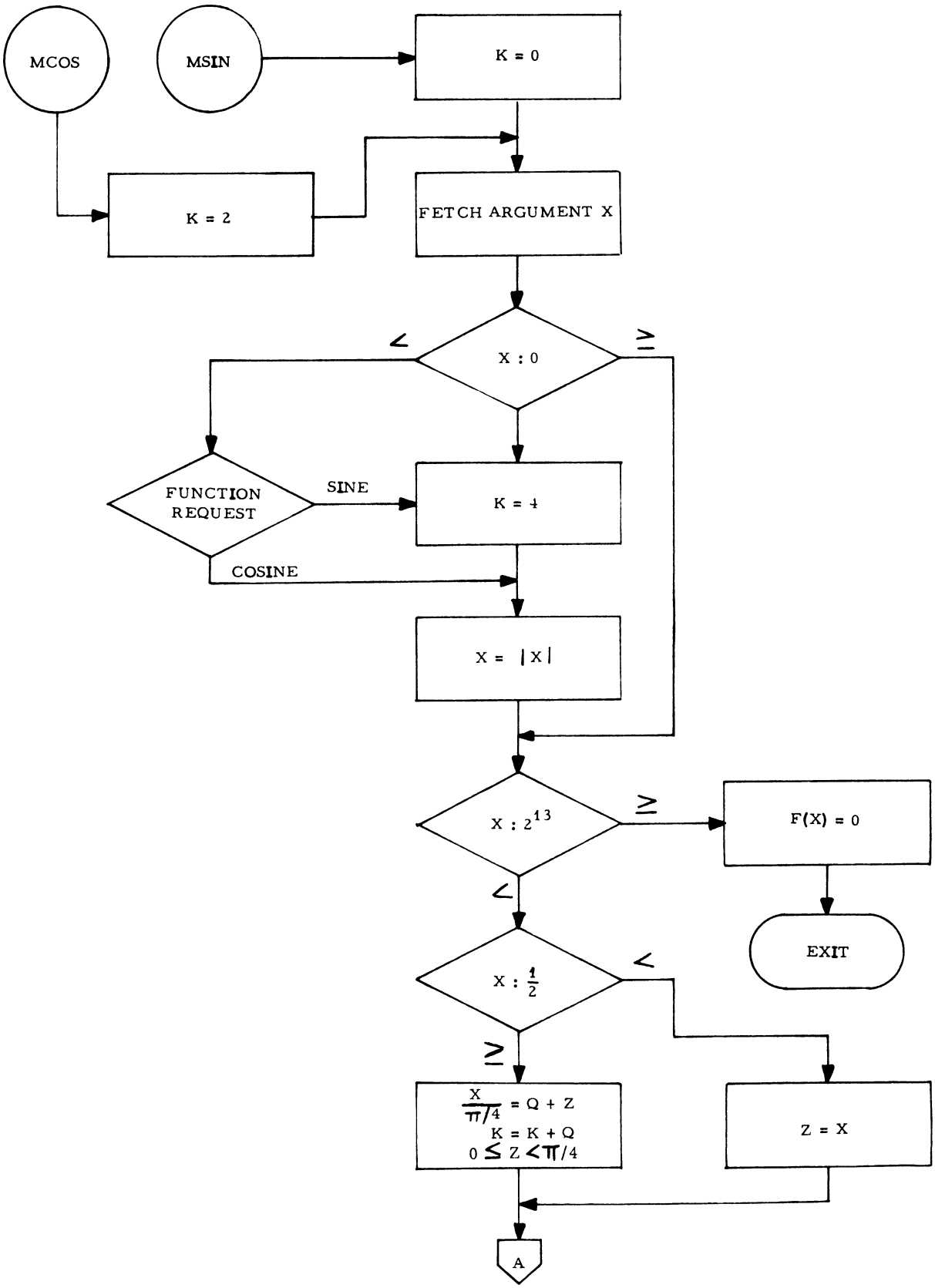
Space Used

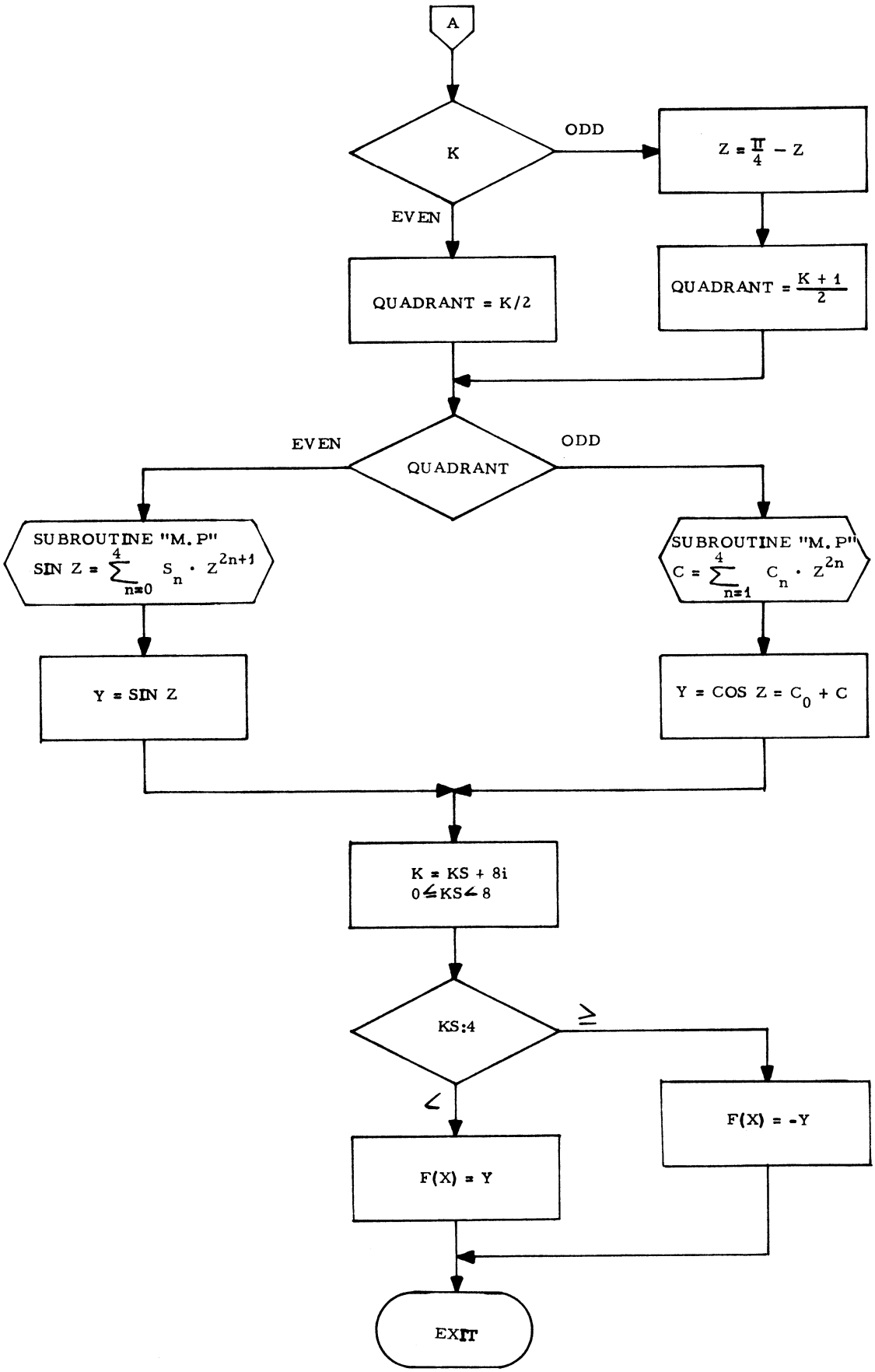
119 words.

Timing

	Standard	Hardware Multiply & Divide
Minimum (MSIN)	2980	1990
Minimum (MCOS)	2730	1910
Maximum	3800	2350
Average	3110	2110

SINE AND COSINE - MP FLOATING POINT




A
K
ODD
$Z = \frac{\pi}{4} - Z$
EVEN
QUADRANT = K/2
QUADRANT = $\frac{K+1}{2}$
EVEN
QUADRANT
ODD
SUBROUTINE "M. P"
SIN Z = $\sum_{n=0}^{4} S_n \cdot Z^{2n+1}$
SUBROUTINE "M. P"
C = $\sum_{n=1}^{4} C_n \cdot Z^{2n}$
Y = SIN Z
Y = COS Z = C_0 + C
K = KS + 8i
$0 \leq KS < 8$
KS:4
$\geq$
$<$
F(X) = -Y
F(X) = Y
EXIT

APPENDIX A

ASSEMBLY LISTING

of

SIN AND COS - MP FLOATING POINT

Drawing No.	ID Code
390007	BSJ

```
                          'SIN AND COS - MP FLOATING POINT 390007'                BSJ 0005
                          *                                                       BSJ 0006
                          *                                                       BSJ 0007
                                   BLK    MATH                                    BSJ 0008
                                   LIBR   MSIN,MCOS                               BSJ 0009
                          * COMPUTE       Y = SIN X                               BSJ 0010
                          * OR            Y = COS X                               BSJ 0011
                          * X GIVEN IN RADIAN                                     BSJ 0012
                          ************************************************        BSJ 0013
0 000 0  0080  0080       PI4      D      128                                     BSJ 0014
0 001 0  6487  6487       PI       D      25735                                   BSJ 0015
0 002 0  76A9  76A9                D      303 7                                   BSJ 0016
         0003  0003       STAB     EQU    $                                       BSJ 0017
                          *               S =      1/(512*11!)                    BSJ 0018
0 003 0  006E  006E                D      110      1/9! = 1280*S                  BSJ 0019
0 004 0  5A5D  5A5D                D      23133                                   BSJ 0020
0 005 0  4895  4895                D      18581                                   BSJ 0021
0 006 0  0074  0074                D      116      1/7! = 1120*S                  BSJ 0022
0 007 0  9800  9800                D      -26624                                  BSJ 0023
0 008 0  6D48  6D48                D      27976                                   BSJ 0024
0 009 0  007A  007A                D      122      1/5! =  400*S                  BSJ 0025
0 00A 0  4444  4444                D      17476                                   BSJ 0026
0 00B 0  1CE1  1CE1                D      07393                                   BSJ 0027
0 00C 0  FFFF  FFFF                D      -1       FLAG FOR EXTRA COEFFICIENT     BSJ 0028
0 00D 0  007E  007E                D      126      1/3! =   50*S                  BSJ 0029
0 00E 0  AAAA  AAAA                D      -21846                                  BSJ 0030
0 00F 0  5560  5560                D      21856                                   BSJ 0031
         0010  0010       CTAB     EQU    $                                       BSJ 0032
                          *               C =      1/(512*10!)                    BSJ 0033
0 010 0  0071  0071                D      113      1/8! = 1280*C                  BSJ 0034
0 011 0  6522  6522                D      25890                                   BSJ 0035
0 012 0  61FA  61FA                D      25082                                   BSJ 0036
0 013 0  0077  0077                D      119      1/6! = 1120*C                  BSJ 0037
0 014 0  A504  A504                D      -23292                                  BSJ 0038
0 015 0  365C  365C                D      13916                                   BSJ 0039
0 016 0  007C  007C                D      124      1/4! =  400*C                  BSJ 0040
0 017 0  5555  5555                D      21845                                   BSJ 0041
0 018 0  1C38  1C38                D      07224                                   BSJ 0042
0 019 0  007F  007F                D      127      1/2  =   50*C                  BSJ 0043
0 01A 0  8000  8000                D      X'8000'                                 BSJ 0044
0 01B 0  003A  003A                D      00058                                   BSJ 0045
0 01C 0  8957  8957       NPI2     D      -30377   HANDY NEGATIVE FOR END FLAG    BSJ 0046
                          *               C0 =     1 - 2**-31                     BSJ 0047
0 01D 0  0080  0080       C01      D      128      -1.                            BSJ 0048
0 01E 0  8000  8000                D      X'8000'  (ILLEGAL FORMAT)               BSJ 0049
0 01F 0  0000  0000                D      0                                       BSJ 0050
0 020 0  007F  007F       C02      D      127      2**-31                         BSJ 0051
0 021 0  0000  0000                D      0                                       BSJ 0052
0 022 0  0001  0001                D      1                                       BSJ 0053
                          ************************************************        BSJ 0054
         0023  0023       MCOS     EQU    $                                       BSJ 0055
0 023 0  0602  06 02               LLB    2        FLAG COSINE FUNCTION           BSJ 0056
0 024 0  1026  1 0 026             JMP    BOTH                                    BSJ 0057
```

```
                                 55   * NOTE=  2 MEANS THE ARGUMENT X WILL BE TURNED               BSJ 0058
                                 56   *        BY +2*PI/4                                          BSJ 0059
              0025   0025        57   MSIN     EQU    $                                            BSJ 0060
    0 025 0   0100   0100        58            CLR              FLAG SINE FUNCTION                 BSJ 0061
              0026   0026        59   BOTH     EQU    $                                            BSJ 0062
*   0 026 0   67FF   6 0 7FF     60            STX    RET5      SAVE RETURN                        BSJ 0063
*   0 027 0   77FF   7 0 7FF     61            STW    TMP6      SAVE FLAG OF FUNCTION              BSJ 0064
    0 028 0   9800   9 1 000     62            LDX *  0                                            BSJ 0065
    0 029 0   602B   6 0 02B     63            STX    ARG                                          BSJ 0066
*   0 02A 0   27FF   2 0 7FF     64            JSX    FLD       FETCH ARGUMENT                     BSJ 0067
    0 02B 0   0000   0000        65   ARG      D      0       **                                   BSJ 0068
    0 02C 0   0820   0820        66            SAM              TEST SIGN                          BSJ 0069
    0 02D 0   1034   1 0 034     67            JMP    SCAL                                         BSJ 0070
*   0 02E 0   8027   8 0 027     68            LDW    TMP6                                         BSJ 0071
    0 02F 0   0800   0800        69            SAZ                                                 BSJ 0072
    0 030 0   1033   1 0 033     70            JMP    ABSO                                         BSJ 0073
    0 031 0   0604   06 04       71            LLB    4         SIN(-X) =SIN(X+PI)                 BSJ 0074
*   0 032 0   702E   7 0 02E     72            STW    TMP6                                         BSJ 0075
              0033   0033        73   ABSO     EQU    $                                            BSJ 0076
*   0 033 0   27FF   2 0 7FF     74            JSX    D2C       X = ABS.(X)                        BSJ 0077
                                 75   * SCALE ARGUMENT                                             BSJ 0078
              0034   0034        76   SCAL     EQU    $                                            BSJ 0079
*   0 034 0   97FF   9 0 7FF     77            LDX    MNT1                                         BSJ 0080
    0 035 0   058E   05 8E       78            DXS    142                                          BSJ 0081
    0 036 0   1075   1 0 075     79            JMP    BIG                                          BSJ 0082
    0 037 0   040E   04 0E       80            IXS    14                                           BSJ 0083
    0 038 0   105B   1 0 05B     81            JMP    QUAD      UNDER,5                            BSJ 0084
                                 82   * ARGUMENT IS .5 OR OVER                                     BSJ 0085
                                 83   * DIVIDE BY PI/4 TO LOCATE THE HALF QUADRANT                 BSJ 0086
                                 84   * AND WORK WITH AN ARGUMENT UNDER PI/4.                      BSJ 0087
                                 85   * THIS DIVISION YIELDS A ONE-WORD INTEGER QUOTIENT Q         BSJ 0088
                                 86   * AND A TWO-WORD FIXED POINT REMAINDER X                     BSJ 0089
                                 87   * DIVIDEND       N = 2**15*N1 + N2 + 2**-15*N3               BSJ 0090
                                 88   * DIVISOR        D = D1 + 2**-15D2                           BSJ 0091
              0039   0039        89   HIGH     EQU    $         EXPAND MANTISSA OF X               BSJ 0092
    0 039 0   0130   0130        90            CAX              IN A THREE-WORD DIVIDEND N         BSJ 0093
    0 03A 0   06B1   06 B1       91            LLB    X'B1'                                        BSJ 0094
*   0 03B 0   A034   A 0 034     92            ADD    MNT1                                         BSJ 0095
    0 03C 0   307D   3 0 03E 1   93            STB    HSFT+1                                       BSJ 0096
    0 03D 0   0100   0100        94            CLR                                                 BSJ 0097
    0 03E 0   0A30   0A3 0       95   HSFT     SLL D  0       **COUNT SET ABOVE                    BSJ 0098
*   0 03F 0   77FF   7 0 7FF     96            STW    TMP5      PART LEFT OF BINARY POINT          BSJ 0099
    0 040 0   06B0   06 B0       97            LLB    X'B0'                                        BSJ 0100
*   0 041 0   A03B   A 0 03B     98            ADD    MNT1                                         BSJ 0101
*   0 042 0   27FF   2 0 7FF     99            JSX    DSM                                          BSJ 0102
*   0 043 0   97FF   9 0 7FF    100            LDX    MNT3                                         BSJ 0103
*   0 044 0   67FF   6 0 7FF    101            STX    TMP4      THIRD WORD OF DIVIDEND, N3         BSJ 0104
*   0 045 0   7043   7 0 043    102            STW    MNT3      SECOND WORD             N2         BSJ 0105
*   0 046 0   803F   8 0 03F    103            LDW    TMP5      FIRST WORD              N1         BSJ 0106
*   0 047 0   77FF   7 0 7FF    104            STW    MNT2                                         BSJ 0107
    0 048 0   8001   8 0 001    105            LDW    PI        PI/4 = D1 + 2**-15*D2              BSJ 0108
*   0 049 0   27FF   2 0 7FF    106            JSX    DIVS                                         BSJ 0109
*   0 04A 0   A032   A 0 032    107            ADD    TMP6      ADD Q TO QUADRANT BITS             BSJ 0110
```

```
*    0 04B 0  704A  7 0 04A   108          STW   TMP6                                    BSJ 0111
*    0 04C 0  8047  8 0 047   109          LDW   MNT2                                   BSJ 0112
*    0 04D 0  77FF  7 0 7FF   110          STW   TMP3     SAVE REMAINDER R1             BSJ 0113
     0 04E 0  801C  8 0 01C   111          LDW   NPI2     NEGATIVE 2ND WORD OF DIVISOR  BSJ 0114
*    0 04F 0  27FF  2 0 7FF   112          JSX   MPYS     X = - Q * D2,,,               BSJ 0115
*    0 050 0  27FF  2 0 7FF   113          JSX   DAD      ,,, + (R1 + 2**-15*N3)        BSJ 0116
*    0 051 0  304D  304D      114          D     TMP3                                   BSJ 0117
*    0 052 0  97FF  9 0 7FF   115          LDX   D128     SET EXPONENT OF REMAINDER X   BSJ 0118
*    0 053 0  6041  6 0 041   116          STX   MNT1                                   BSJ 0119
     0 054 0  0820  0820      117          SAM            SIGN OF X                     BSJ 0120
     0 055 0  105B  1 0 05B   118          JMP   QUAD                                   BSJ 0121
*    0 056 0  2050  2 0 050   119          JSX   DAD      MINUS, ADD AN EXTRA PI/4      BSJ 0122
     0 057 0  0001  0001      120          D     PI                                     BSJ 0123
*    0 058 0  804B  8 0 04B   121          LDW   TMP6     AND DECREMENT QUADRANT COUNT  BSJ 0124
*    0 059 0  B7FF  B 0 7FF   122          SUB   D1                                     BSJ 0125
*    0 05A 0  7058  7 0 058   123          STW   TMP6                                   BSJ 0126
                              124  *                                                   BSJ 0127
              005B  005B      125  QUAD     EQU   $                                    BSJ 0128
*    0 05B 0  805A  8 0 05A   126          LDW   TMP6     GET QUADRANT BITS             BSJ 0129
     0 05C 0  0830  0830      127          SAC                                          BSJ 0130
     0 05D 0  1063  1 0 063   128          JMP   POLY                                   BSJ 0131
*    0 05E 0  27FF  2 0 7FF   129          JSX   FSB      HIGHER HALF OF A QUADRANT     BSJ 0132
     0 05F 0  0000  0000      130          D     PI4      X = PI/4 - X                  BSJ 0133
*    0 060 0  2033  2 0 033   131          JSX   D2C                                    BSJ 0134
*    0 061 0  805B  8 0 05B   132          LDW   TMP6                                   BSJ 0135
*    0 062 0  A059  A 0 059   133          ADD   D1                                     BSJ 0136
              0063  0063      134  POLY     EQU   $        GIVE POSITIVE ACR FOR EVEN    BSJ 0137
     0 063 0  0A22  0A2 2     135          SRL D 2        TEST FUNCTION TO USE          BSJ 0138
     0 064 0  0400  04 00     136          IXS   0        (NOT FUNCTION REQUEST)        BSJ 0139
     0 065 0  1069  1 0 069   137          JMP   FCOS                                   BSJ 0140
              0066  0066      138  FSIN     EQU   $                                    BSJ 0141
*    0 066 0  27FF  2 0 7FF   139          JSX   M,P      SIN X = X * P(X**2)           BSJ 0142
     0 067 0  8003  8003      140          D     STAB+X'8000'                           BSJ 0143
     0 068 0  106F  1 0 06F   141          JMP   SIGN                                   BSJ 0144
              0069  0069      142  FCOS     EQU   $                                    BSJ 0145
*    0 069 0  2066  2 0 066   143          JSX   M,P      COS X = P(X**2) ,,,           BSJ 0146
     0 06A 0  8010  8010      144          D     CTAB+X'8000'                           BSJ 0147
*    0 06B 0  205E  2 0 05E   145          JSX   FSB      ,,, + C0                      BSJ 0148
     0 06C 0  001D  001D      146          D     C01                                    BSJ 0149
*    0 06D 0  206B  2 0 06B   147          JSX   FSB      C0 + - C01 - C02              BSJ 0150
     0 06E 0  0020  0020      148          D     C02                                    BSJ 0151
              006F  006F      149  SIGN     EQU   $        TEST SIGN TO GIVE FUNCTION    BSJ 0152
*    0 06F 0  8061  8 0 061   150          LDW   TMP6                                   BSJ 0153
     0 070 0  0A43  0A4 3     151          SRC   3                                      BSJ 0154
     0 071 0  0810  0810      152          SAP                                          BSJ 0155
*    0 072 0  2060  2 0 060   153          JSX   D2C                                    BSJ 0156
*    0 073 0  9026  9 0 026   154  EXIT     LDX   RET5     RETURN TO CALLER              BSJ 0157
     0 074 0  1801  1 1 001   155          JMP * 1                                      BSJ 0158
                              156  * ARGUMENT REFUSED IF 2**13 OR MORE,               BSJ 0159
                              157  * BECAUSE ACCURACY OF FUNCTION WOULD                BSJ 0160
                              158  * BE LIMITED TO 16 BITS OR LESS,                    BSJ 0161
                              159  * (WOULD ALSO REQUIRE A FOUR-WORD                   BSJ 0162
                              160  * DIVIDEND IN THE 'HIGH' SEQUENCE,)                 BSJ 0163
```

```
              0075  0075      161   BIG     EQU   $                                       BSJ 0164
*     0 075 0  27FF  2 0 7FF   162           JSX   M.ZE      GIVE ZERO                     BSJ 0165
      0 076 0  1073  1 0 073   163           JMP   EXIT                                    BSJ 0166
                               164   *************************************************    BSJ 0167
                               165           NTRY  MSIN,MCOS                               BSJ 0168
      0 076 0*******118        166
```

X-REF

```
    0 033 0  ABSO    0 030 0   0 033 0
    0 02B 0  ARG     0 029 0
    0 075 0  BIG     0 036 0   0 075 0
    0 026 0  BOTH    0 024 0   0 026 0
    0 010 0  CTAB    0 010 0   0 06A 0
    0 01D 0  C01     0 06C 0
    0 020 0  C02     0 06E 0
EXT 0056     DAD     0 050 0   0 056 0
EXT 0049     DIVS    0 049 0
EXT 0042     DSM     0 042 0
EXT 0062     D1      0 059 0   0 062 0
EXT 0052     D128    0 052 0
EXT 0072     D2C     0 033 0   0 060 0   0 072 0
    0 073 0  EXIT    0 076 0
    0 069 0  FCOS    0 065 0   0 069 0
EXT 002A     FLD     0 02A 0
EXT 006D     FSB     0 05E 0   0 06B 0   0 06D 0
    0 066 0  FSIN    0 066 0
    0 039 0  HIGH    0 039 0
    0 03E 0  HSFT    0 03C 0
EXT 0069     M.P     0 066 0   0 069 0
EXT 0075     M.ZE    0 075 0
LIB 0 023 0  MCOS    0 023 0
EXT 0053     MNT1    0 034 0   0 03B 0   0 041 0   0 053 0
EXT 004C     MNT2    0 047 0   0 04C 0
EXT 0045     MNT3    0 043 0   0 045 0
EXT 004F     MPYS    0 04F 0
LIB 0 025 0  MSIN    0 025 0
    0 01C 0  NPI2    0 04E 0
    0 001 0  PI      0 048 0   0 057 0
    0 000 0  PI4     0 05F 0
    0 063 0  POLY    0 05D 0   0 063 0
    0 05B 0  QUAD    0 038 0   0 055 0   0 05B 0
EXT 0073     RET5    0 026 0   0 073 0
    0 034 0  SCAL    0 02D 0   0 034 0
    0 06F 0  SIGN    0 068 0   0 06F 0
    0 003 0  STAB    0 003 0   0 067 0
EXT 0051     TMP3    0 04D 0   0 051 0
EXT 0044     TMP4    0 044 0
EXT 0046     TMP5    0 03F 0   0 046 0
EXT 006F     TMP6    0 027 0   0 02E 0   0 032 0   0 04A 0   0 04B 0   0 058 0   0 05A 0   0 05B 0
                     0 061 0   0 06F 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
166	41 618	0	6

ARC TANGENT - MP FLOATING POINT

DATE: April 1968

ID CODE: BSM

DRAWING: 390010 (Rev B)

LABEL: MATN

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To obtain the arc tangent of an argument given in the mid-precision floating point format. The result is set in the same format in the software registers MNT1, MNT2, MNT3.

USAGE

Calling Sequence

```
L-1     SMB     MATN
L       JSX     MATN
L+1     D       ARG
L+2     Return
```

Argument Description

The argument is a mid-precision floating point number occupying three consecutive words in memory, beginning at address "ARG" as pointed out in the calling sequence.

METHOD

To improve the timing with negative tangents, all computations are performed on the absolute value of the arguments, then the final arc is given the sign of the original argument. Let

$Y = \arctan X$

$x = |X|$ if $|X| < 1$

$x = \frac{1}{|X|}$ if $|X| \geq 1$

$Z = x$ if $x < 1/8$

$Z = \frac{x - \tan Ax}{1 + x \cdot \tan Ax}$ if $x \geq 1/8$

Where tan Ax is a multiple of 1/8, such that:

$$x - 1/8 < \tan Ax \leq x$$

Then

$$0 \leq Z < 1/8$$

$$\arctan x = \arctan Z + Ax$$

To obtain arctan Z, the Taylor series is expanded in terms of Z, truncated to the first five terms because of the reduced range of Z.

The value of Ax is found in radian in a seven-entry table.

Finally:

$y = \arctan x$	if	$\lvert X \rvert < 1$
$y = \pi/2 - \arctan x$	if	$\lvert X \rvert \geq 1$
$Y = y$	if	$X \geq 0$
$Y = -y$	if	$X < 0$

ERROR CONDITIONS

None

RANGE AND ACCURACY

The arc is computed in the interval $-\pi/2, +\pi/2$.
The accuracy is 29 bits.

RESTRICTIONS

Loading

This subroutine locally references storage words and constants located in the "MATH POOL" module, and must therefore be loaded in the same 2K block as the pool.

Other Routines

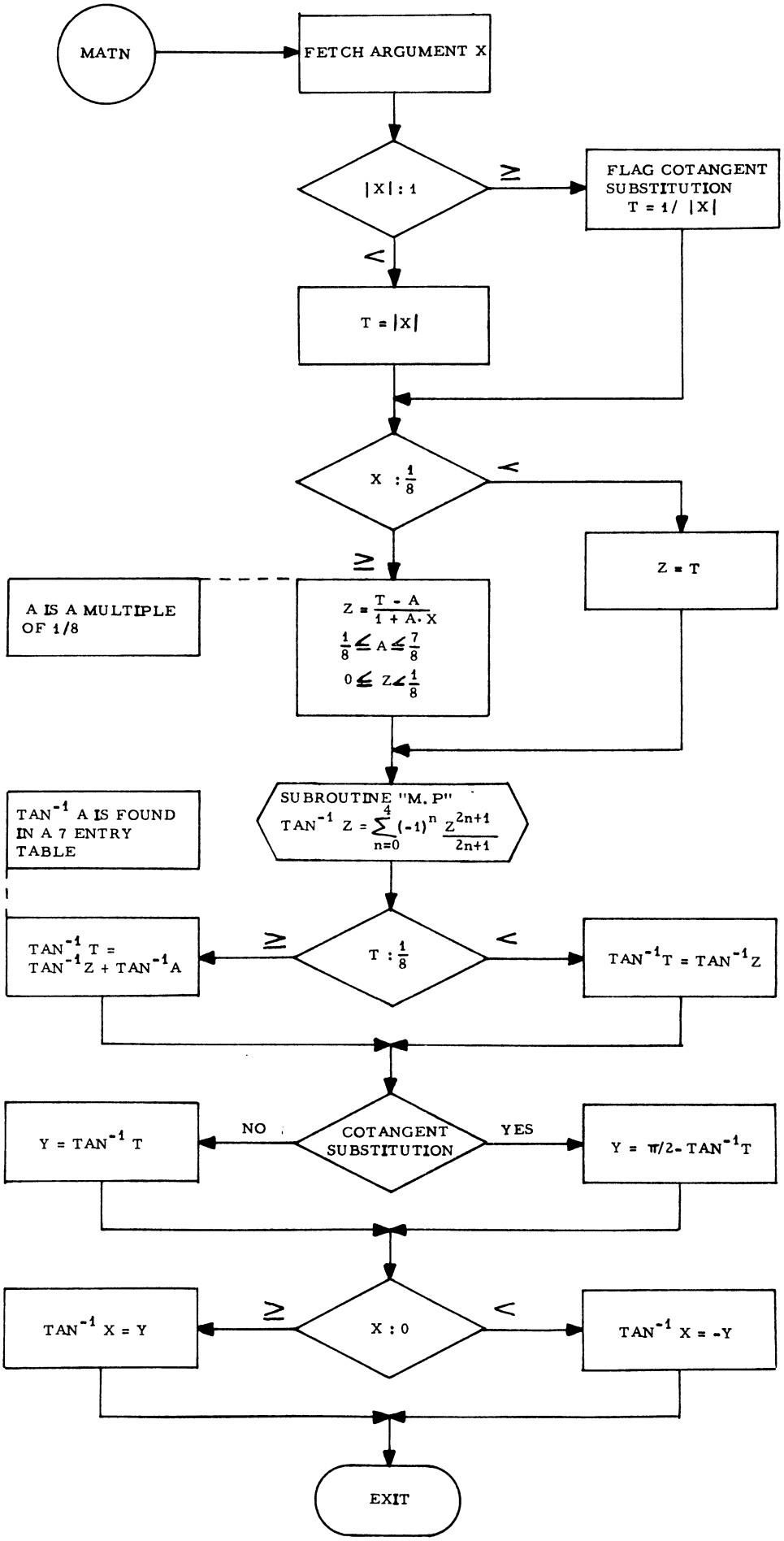
ACMY, DLD, DSM, DST, D2C, FAD, FDV, FLD, FNRM, M.P, MPYS

Space Used

128 words

Timing (in cycles)

	Standard	Hardware Multiply & Divide
Minimum	2700	2000
Maximum	4800	2900
Average	4120	2690
tangent under 1:		
Average	3820	2470


MATN
FETCH ARGUMENT X
|X| : 1
≥
FLAG COTANGENT SUBSTITUTION
T = 1/ |X|
<
T = |X|
X : 1/8
<
Z = T
≥
A IS A MULTIPLE OF 1/8
Z = (T - A)/(1 + A·X)
1/8 ≤ A ≤ 7/8
0 ≤ Z < 1/8
SUBROUTINE "M. P"
TAN⁻¹ Z = Σ (n=0 to 4) (-1)^n Z^(2n+1)/(2n+1)
TAN⁻¹ A IS FOUND IN A 7 ENTRY TABLE
TAN⁻¹ T = TAN⁻¹ Z + TAN⁻¹ A
≥
T : 1/8
<
TAN⁻¹ T = TAN⁻¹ Z
Y = TAN⁻¹ T
NO
COTANGENT SUBSTITUTION
YES
Y = π/2 - TAN⁻¹ T
TAN⁻¹ X = Y
≥
X : 0
<
TAN⁻¹ X = -Y
EXIT

APPENDIX A

ASSEMBLY LISTING

of

ARC TANGENT - MP FLOATING POINT

Drawing No.	ID Code
390010 B	BSM

```
                                2   'ARC TANGENT - MP FLOATING POINT 390010 B'          BSM 0005
                                3   *                                                   BSM 0006
                                4   *                                                   BSM 0007
                                5             BLK   MATH                                BSM 0008
                                6             LIBR  MATN                                BSM 0009
                                7   * COMPUTE       Y = ATAN(X)                         BSM 0010
                                8   ************************************************** BSM 0011
0 000 0  0081  0081             9   PI2       D     129      PI/2                       BSM 0012
0 001 0  6487  6487            10             D     25735                               BSM 0013
0 002 0  76A9  76A9            11             D     30377                               BSM 0014
0 003 0  0080  0080            12   PI4       D     128      PI/4                       BSM 0015
0 004 0  6487  6487            13             D     25735                               BSM 0016
0 005 0  76A9  76A9            14             D     30377                               BSM 0017
0 006 0  007D  007D            15   TABL      D     125      .12435 49945  ATAN(.125)   BSM 0018
0 007 0  7F56  7F56            16             D     32598                               BSM 0019
0 008 0  7535  7535            17             D     30005                               BSM 0020
0 009 0  007E  007E            18             D     126      .24497 86631  ATAN(.250)   BSM 0021
0 00A 0  7D6D  7D6D            19             D     32109                               BSM 0022
0 00B 0  6BF2  6BF2            20             D     27634                               BSM 0023
0 00C 0  007F  007F            21             D     127      .35877 06703  ATAN(.375)   BSM 0024
0 00D 0  5BD8  5BD8            22             D     23512                               BSM 0025
0 00E 0  3284  3284            23             D     12932                               BSM 0026
0 00F 0  007F  007F            24             D     127      .46364 76090  ATAN(.500)   BSM 0027
0 010 0  76B1  76B1            25             D     30385                               BSM 0028
0 011 0  4E0B  4E0B            26             D     19979                               BSM 0029
0 012 0  0080  0080            27             D     128      .55859 93153  ATAN(.625)   BSM 0030
0 013 0  4780  4780            28             D     18304                               BSM 0031
0 014 0  1757  1757            29             D     05975                               BSM 0032
0 015 0  0080  0080            30             D     128      .64350 11088  ATAN(.750)   BSM 0033
0 016 0  525E  525E            31             D     21086                               BSM 0034
0 017 0  1F46  1F46            32             D     08006                               BSM 0035
0 018 0  0080  0080            33             D     128      .71882 99996  ATAN(.875)   BSM 0036
0 019 0  5C02  5C02            34             D     23554                               BSM 0037
0 01A 0  4F8B  4F8B            35             D     20363                               BSM 0038
0 01B 0  8003  8003            36   TAB       D     TABL-3+X'8000'                      BSM 0039
         001C  001C            37   TAYL      EQU   $                                   BSM 0040
0 01C 0  007D  007D            38             D     125      1/9                        BSM 0041
0 01D 0  71C7  71C7            39             D     29127                               BSM 0042
0 01E 0  0E39  0E39            40             D     03641                               BSM 0043
0 01F 0  007E  007E            41             D     126     -1/7                        BSM 0044
0 020 0  B6DB  B6DB            42             D     -18725                              BSM 0045
0 021 0  36DB  36DB            43             D     14043                               BSM 0046
0 022 0  007E  007E            44             D     126      1/5                        BSM 0047
0 023 0  6666  6666            45             D     26214                               BSM 0048
0 024 0  3333  3333            46             D     13107                               BSM 0049
0 025 0  FFFF  FFFF            47             D     -1       FLAG EXTRA COEFFICIENT     BSM 0050
0 026 0  007F  007F            48             D     127     -1/3                        BSM 0051
0 027 0  AAAA  AAAA            49             D     -21846                              BSM 0052
0 028 0  5555  5555            50             D     21845                               BSM 0053
                               51   *                                                   BSM 0054
0 029 0  F000  F000            52   M4L       D     X'F000'                             BSM 0055
0 02A 0  0000  0000            53   FSIG      D     0                                   BSM 0056
0 02B 0  0000  0000            54   FLOT      D     0                                   BSM 0057
```

```
                                 ************************************************  BSM 0058
          002C  002C       MATN  EQU   $                                          BSM 0059
*  0 02C 0  67FF  6 0 7FF        STX   RET5     SAVE RETURN                       BSM 0060
   0 02D 0  9800  9 1 000        LDX * 0                                          BSM 0061
   0 02E 0  8800  8 1 000        LDW * 0        EXPONENT OF ARGUMENT              BSM 0062
*  0 02F 0  B7FF  B 0 7FF        SUB   D129                                       BSM 0063
   0 030 0  702B  7 0 02B        STW   FCOT     FLAG FOR COTANGENT                BSM 0064
   0 031 0  0810  0810           SAP                                              BSM 0065
   0 032 0  103A  1 0 03A        JMP   TAN                                        BSM 0066
          0033  0033       COTG  EQU   $        ARGUMENT NOT UNDER 1,             BSM 0067
   0 033 0  6037  6 0 037        STX   CALC     SAVE ARGUMENT ADDRESS             BSM 0068
*  0 034 0  27FF  2 0 7FF        JSX   FLD                                        BSM 0069
*  0 035 0  37FF  37FF           D     F,1                                        BSM 0070
*  0 036 0  27FF  2 0 7FF        JSX   FDV      X = 1 / X                         BSM 0071
   0 037 0  0000  0000     CALC  D     0        **                                BSM 0072
*  0 038 0  87FF  8 0 7FF        LDW   MNT2                                       BSM 0073
   0 039 0  103D  1 0 03D        JMP   BOTH                                       BSM 0074
          003A  003A       TAN   EQU   $        ARGUMENT UNDER 1,                 BSM 0075
   0 03A 0  603C  6 0 03C        STX   CALT     SAVE ARGUMENT ADDRESS             BSM 0076
*  0 03B 0  2034  2 0 034        JSX   FLD      FETCH ARGUMENT                    BSM 0077
   0 03C 0  0000  0000     CALT  D     0        **                                BSM 0078
          003D  003D       BOTH  EQU   $                                          BSM 0079
   0 03D 0  702A  7 0 02A        STW   FSIG     SAVE SIGN OF ARGUMENT             BSM 0080
   0 03E 0  0810  0810           SAP                                              BSM 0081
*  0 03F 0  27FF  2 0 7FF        JSX   D2C      X = ABS,(X)                       BSM 0082
                           * SCALE ARGUMENT                                       BSM 0083
*  0 040 0  97FF  9 0 7FF        LDX   MNT1                                       BSM 0084
   0 041 0  0040  0040           SLM                                              BSM 0085
   0 042 0  057E  05 7E          DXS   126                                        BSM 0086
   0 043 0  1846  1 1 046        JMP * CLIP                                       BSM 0087
   0 044 0  7074  7 0 074        STW   CALA     UNDER ,125 - FLAG FULL ARC        BSM 0088
   0 045 0  106E  1 0 06E        JMP   POLY                                       BSM 0089
          0046  0046       CLIP  EQU   $                                          BSM 0090
   0 046 0  0A01  0A0 1          SRL   1        EXPONENT = 126                    BSM 0091
   0 047 0  0A01  0A0 1          SRL   1                 =127                     BSM 0092
   0 048 0  104C  1 0 04C        JMP   TRAN              = 128                    BSM 0093
*  0 049 0  203B  2 0 03B        JSX   FLD               = 129                    BSM 0094
   0 04A 0  0003  0003           D     PI4                                        BSM 0095
   0 04B 0  107B  1 0 07B        JMP   ASIG                                       BSM 0096
                           * TRANSFORM ARGUMENT TO STAY UNDER ,125                BSM 0097
                           *     A = TAN(I/8) , I AN INTEGER                      BSM 0098
                           *     X = (X - A)/(1 + X * A)                          BSM 0099
          004C  004C       TRAN  EQU   $                                          BSM 0100
   0 04C 0  E029  E 0 029        AND   M4L      GET HIGHEST MULTIPLE OF 1/8       BSM 0101
*  0 04D 0  77FF  7 0 7FF        STW   TMP1     INSIDE X AS A                     BSM 0102
   0 04E 0  0A01  0A0 1          SRL   1                                          BSM 0103
*  0 04F 0  A040  A 0 040        ADD   TMP1                                       BSM 0104
   0 050 0  0A55  0A5 5          SLC   5                                          BSM 0105
   0 051 0  A01B  A 0 01B        ADD   TAB      ADDRESS OF ATAN(A)                BSM 0106
   0 052 0  7074  7 0 074        STW   CALA     SAVE AND FLAG EXTRA ARC           BSM 0107
*  0 053 0  27FF  2 0 7FF        JSX   DST      SAVE X                            BSM 0108
*  0 054 0  37FF  37FF           D     TMP2                                       BSM 0109
*  0 055 0  804F  8 0 04F        LDW   TMP1                                       BSM 0110
```

```
*     0 056 0  27FF  2 0 7FF   108            JSX   MPYS      A * X ...                    BSM 0111
*     0 057 0  8054  8 0 054   109            LDW   TMP2                                 BSM 0112
*     0 058 0  77FF  7 0 7FF   110            STW   MNT3                                 BSM 0113
*     0 059 0  8055  8 0 055   111            LDW   TMP1                                 BSM 0114
*     0 05A 0  27FF  2 0 7FF   112            JSX   ACMY                                 BSM 0115
      0 05B 0  06A1  06 A1     113            LLB   X'A1'                                BSM 0116
*     0 05C 0  8040  8 0 040   114            SUB   MNT1                                 BSM 0117
*     0 05D 0  27FF  2 0 7FF   115            JSX   DSM                                  BSM 0118
*     0 05E 0  A7FF  A 0 7FF   116            ADD   B1        ... + 1                    BSM 0119
*     0 05F 0  9058  9 0 058   117            LDX   MNT3                                 BSM 0120
*     0 060 0  77FF  7 0 7FF   118            STW   TMP8                                 BSM 0121
*     0 061 0  67FF  6 0 7FF   119            STX   TMP9                                 BSM 0122
*     0 062 0  27FF  2 0 7FF   120            JSX   DLD       GET X BACK                 BSM 0123
*     0 063 0  3057  3057      121            D     TMP2                                 BSM 0124
      0 064 0  06B3  06 B3     122            LLB   X'B3'                                BSM 0125
*     0 065 0  A05C  A 0 05C   123            ADD   MNT1                                 BSM 0126
*     0 066 0  205D  2 0 05D   124            JSX   DSM       X = A                      BSM 0127
*     0 067 0  97FF  9 0 7FF   125            LDX   D128                                 BSM 0128
*     0 068 0  67FF  6 0 7FF   126            STX   TMP7                                 BSM 0129
      0 069 0  0504  05 04     127            DXS   4         SCALE X-A AND 1+A*X        BSM 0130
*     0 06A 0  6065  6 0 065   128            STX   MNT1                                 BSM 0131
*     0 06B 0  27FF  2 0 7FF   129            JSX   FNRM      NORMALIZE (X - A)          BSM 0132
*     0 06C 0  2036  2 0 036   130            JSX   FDV       X = (X - A)/(1 + A * X)    BSM 0133
*     0 06D 0  3068  3068      131            D     TMP7                                 BSM 0134
                               132  *                                                   BSM 0135
               006E  006E      133  POLY     EQU   $         X IS WITHIN RANGE (0, 1/8)  BSM 0136
                               134  * NOTE - BE SURE ACR IS POSITIVE FOR P(X**2)       BSM 0137
*     0 06E 0  27FF  2 0 7FF   135            JSX   M,P       GET ARCTAN THRU TAYLOR     BSM 0138
      0 06F 0  801C  801C      136            D     TAYL+X'8000'                         BSM 0139
                               137  * FIX ANSWER IN CASE OF ARGUMENT TRANSFORMATION    BSM 0140
               0070  0070      138  ARC      EQU   $                                    BSM 0141
      0 070 0  8074  8 0 074   139            LDW   CALA      CHECK ARGUMENT ALTERATION  BSM 0142
      0 071 0  0820  0820      140            SAM                                        BSM 0143
      0 072 0  1075  1 0 075   141            JMP   ACOT                                 BSM 0144
*     0 073 0  27FF  2 0 7FF   142            JSX   FAD       Y = Y + ATAN(A)            BSM 0145
      0 074 0  0000  0000      143  CALA     D     0       **                           BSM 0146
               0075  0075      144  ACOT     EQU   $                                    BSM 0147
      0 075 0  802B  8 0 02B   145            LDW   FCOT      CHECK COTANGENT SUBSTITUTION BSM 0148
      0 076 0  0810  0810      146            SAP                                        BSM 0149
      0 077 0  107B  1 0 07B   147            JMP   ASIG                                 BSM 0150
*     0 078 0  203F  2 0 03F   148            JSX   D2C                                  BSM 0151
*     0 079 0  2073  2 0 073   149            JSX   FAD       Y = PI/4 - Y               BSM 0152
      0 07A 0  0000  0000      150            D     PI2                                  BSM 0153
               007B  007B      151  ASIG     EQU   $                                    BSM 0154
      0 07B 0  802A  8 0 02A   152            LDW   FSIG      CHECK SIGN OF ARGUMENT     BSM 0155
      0 07C 0  0810  0810      153            SAP                                        BSM 0156
*     0 07D 0  207B  2 0 07B   154            JSX   D2C                                  BSM 0157
*     0 07E 0  902C  9 0 02C   155  EXIT     LDX   RET5      RETURN TO CALLER           BSM 0158
      0 07F 0  1801  1 1 001   156            JMP * 1                                    BSM 0159
                               157  ********************************************************BSM 0160
                               158            NTRY  MATN                                 BSM 0161
      0 07F 0********127       159            END                                        BSM 0162
```

X-REF

```
EXT 005A      ACMY      0 05A 0
    0 075 0   ACOT      0 072 0     0 075 0
    0 070 0   ARC       0 070 0
    0 078 0   ASIG      0 048 0     0 077 0     0 078 0
    0 03D 0   BOTH      0 039 0     0 03D 0
EXT 005E      B1        0 05E 0
    0 074 0   CALA      0 044 0     0 052 0     0 070 0
    0 037 0   CALC      0 033 0
    0 03C 0   CALT      0 03A 0
    0 046 0   CLIP      0 043 0     0 046 0
    0 033 0   COTG      0 033 0
EXT 0062      DLD       0 062 0
EXT 0066      DSM       0 05D 0     0 066 0
EXT 0053      DST       0 053 0
EXT 0067      D128      0 067 0
EXT 002F      D1E9      0 02F 0
EXT 007D      D2C       0 03F 0     0 078 0     0 07D 0
    0 07E 0   EXIT
EXT 0035      F.1       0 035 0
EXT 0079      FAD       0 073 0     0 079 0
    0 028 0   FCOT      0 030 0     0 075 0
EXT 006C      FDV       0 036 0     0 06C 0
EXT 0049      FLD       0 034 0     0 03B 0     0 049 0
EXT 0068      FNRM      0 068 0
    0 02A 0   FSIG      0 03D 0     0 078 0
EXT 006E      M.P       0 06E 0
LIB 0 02C 0   MATN      0 02C 0
EXT 006A      MNT1      0 040 0     0 05C 0     0 065 0     0 06A 0
EXT 0038      MNT2      0 038 0
EXT 005F      MNT3      0 058 0     0 05F 0
EXT 0056      MPYS      0 056 0
    0 029 0   M4L       0 04C 0
    0 000 0   PI2       0 07A 0
    0 003 0   PI4       0 04A 0
    0 06E 0   POLY      0 045 0     0 06E 0
EXT 007E      RETS      0 02C 0     0 07E 0
    0 01B 0   TAB       0 051 0
    0 006 0   TABL      0 018 0
    0 03A 0   TAN       0 032 0     0 03A 0
    0 01C 0   TAYL      0 01C 0     0 06F 0
EXT 0059      TMP1      0 04D 0     0 04F 0     0 055 0     0 059 0
EXT 0063      TMP2      0 054 0     0 057 0     0 063 0
EXT 006D      TMP7      0 068 0     0 06D 0
EXT 0060      TMP8      0 060 0
EXT 0061      TMP9      0 061 0
    0 04C 0   TRAN      0 048 0     0 04C 0
```

NO ERRORS

CARDS SYMBOLS LITR STACK

159 46 615 0 6

EXPONENTIAL - MP FLOATING POINT

DATE: April 1968

ID CODE: BSL

DRAWING: 390009 (Rev D)

LABEL: MEXP

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To find the exponential value of a mid-precision floating point number and leave the result in the same format in the software registers MNT1, MNT2, MNT3.

USAGE

Calling Sequence

The routine returns at L+2.

```
L-1     SMB     MEXP
L       JSX     MEXP
L+1     D       ARG
L+2     Return
```

Argument Description

The argument is a mid-precision floating point number occupying three consecutive words in memory, beginning at address "ARG" as pointed out in the calling sequence.

METHOD

Let the magnitude x of the argument X be redefined in terms of Ln 2 by means of a positive integer N and a positive fraction Z.

$$x = |X| = N \cdot \text{Ln } 2 + Z$$

$$0 \leq N$$

$$0 \leq Z < \text{Ln } 2$$

$$e^x = e^{N.\text{Ln } 2} . e^Z = 2^N . e^Z$$

Let Z be split in terms of Az, a multiple of 1/8, and whatever remains in Z.

$$Z = Az + z$$

$$Az \leq Z < Az + 1/8$$

$$0 \leq z < 1/8$$

$$e^{Z} = e^{Az} \cdot e^{z}$$

The value of e^{Az} is found in a five-entry table. The value of e^{z} is approximated by means of the Taylor series truncated to the first six terms.

ERROR CONDITIONS

The subroutine M.OV is called when the argument is greater than +88.0296918.

RANGE AND ACCURACY

The argument should be within the range -89.4159863, +88.0296918.

The accuracy is 29 bits.

RESTRICTIONS

Loading

This subroutine locally references storage words and constants located in the "MATH POOL" module, and must therefore be loaded in the same 2K block as the pool.

Other Routines

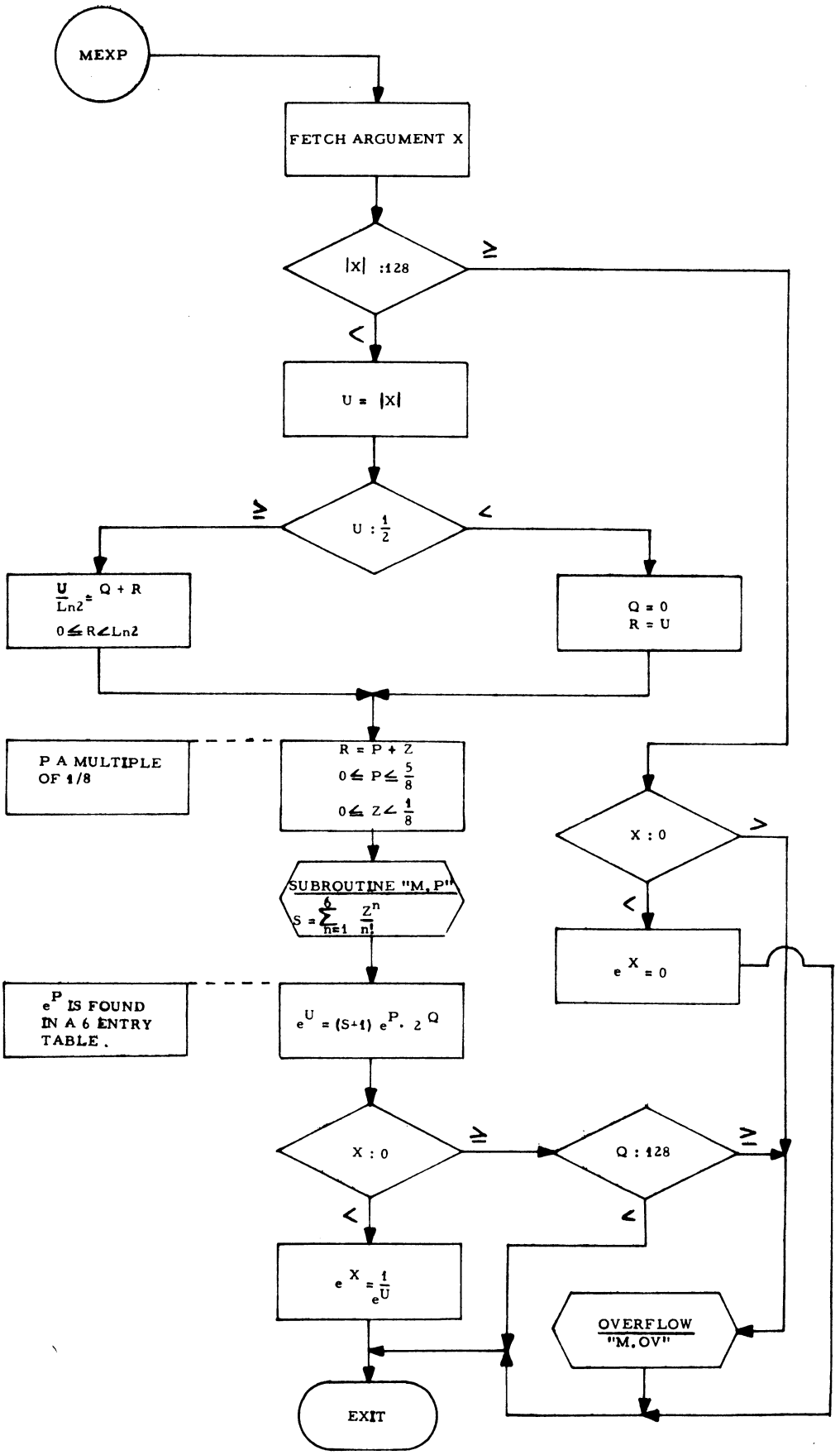
DAD, DIVS, DSM, D2C, FAD, FDV, FLD, FMP, FNRM, FST, M.OV, M.P, M.ZE, MPYS.

Space Used

142 words

Timing (in cycles)

	Standard	Hardware Multiply & Divide
Minimum	50	50
Maximum	4750	2760
Average	3950	2530


MEXP
FETCH ARGUMENT X
|X| :128
≥
<
U = |X|
U : 1/2
≥
<
U/Ln2 = Q + R
0 ≤ R < Ln2
Q = 0
R = U
P A MULTIPLE OF 1/8
R = P + Z
0 ≤ P ≤ 5/8
0 ≤ Z < 1/8
X : 0
>
<
SUBROUTINE "M.P"
S = Σ (n=1 to 6) Z^n/n!
e^X = 0
e^P IS FOUND IN A 6 ENTRY TABLE.
e^U = (S+1) e^P . 2^Q
X : 0
≥
Q : 128
≥
<
<
e^X = 1/e^U
OVERFLOW "M.OV"
EXIT

APPENDIX A

ASSEMBLY LISTING

of

EXPONENTIAL - MP FLOATING POINT

Drawing No.

390009

ID Code

BSL (Rev. D)

```
                               2     'EXPONENTIAL-MP FLOATING PT DN390009D
                               3     *                                                                SL 00050
                               4     *                                                                SL 00060
                               5                BLK    MATH                                           SL 00070
                               6                LIBR   MEXP                                           SL 00080
                               7     * COMPUTE         Y  =  E ** X                                   SL 00090
                               8     * ARGUMENT        X  =  2 ** X1 * X2                             SL 00100
                               9     * LET             U  =  ABS,(X)                                  SL 00110
                              10     *                 V  =  E ** U                                   SL 00120
                              11     * THEN            U  =  Q * LN(2) + R                            SL 00130
                              12     *                 V  =  2 ** Q * E ** R                          SL 00140
                              13     *                 Y  =  V         FOR  X  POSITIVE               SL 00150
                              14     *                 Y  =  1 / V     FOR  X  NEGATIVE               SL 00160
                              15     ***********************************************                  SL 00170
            0000   0000       16     TAYL       EQU    $                                              SL 00180
0 000 0     0077   0077       17                D      119       1/6!                                 SL 00190
0 001 0     5B05   5B05       18                D      23301                                          SL 00200
0 002 0     582E   582E       19                D      22574                                          SL 00210
0 003 0     007A   007A       20                D      122       1/5!                                 SL 00220
0 004 0     4444   4444       21                D      17476                                          SL 00230
0 005 0     2222   2222       22                D      8738                                           SL 00240
0 006 0     007C   007C       23                D      124       1/4!                                 SL 00250
0 007 0     5555   5555       24                D      21845                                          SL 00260
0 008 0     2AAB   2AAB       25                D      10923                                          SL 00270
0 009 0     007E   007E       26                D      126       1/3!
0 00A 0     5555   5555       27                D      21845                                          SL 00290
0 00B 0     2AAB   2AAB       28                D      10923                                          SL 00300
0 00C 0     0080   0080       29                D      128       1/2                                  SL 00310
0 00D 0     4000   4000       30                D      16384                                          SL 00320
0 00E 0     0000   0000       31                D      0                                              SL 00330
0 00F 0     0081   0081       32                D      129       1                                    SL 00340
0 010 0     4000   4000       33                D      16384                                          SL 00350
0 011 0     0000   0000       34                D      0                                              SL 00360
0 012 0     8014   8014       35     TAB        D      TABL+X'8000'  END FLAG IN SIGN BIT             SL 00370
0 013 0     007D   007D       36     D125       D      125
0 014 0     0081   0081       37     TABL       D      129       1,             E ** 0                SL 00400
0 015 0     4000   4000       38                D      16384                                          SL 00410
0 016 0     0000   0000       39                D      0                                              SL 00420
0 017 0     0081   0081       40                D      129       1,133148453    E ** ,125
0 018 0     4885   4885       41                D      18565                                          SL 00450
0 019 0     408B   408B       42                D      16523                                          SL 00460
0 01A 0     0081   0081       43                D      129       1,284025417    E ** ,25
0 01B 0     522D   522D       44                D      21037                                          SL 00490
0 01C 0     3C79   3C79       45                D      15481                                          SL 00500
0 01D 0     0081   0081       46                D      129       1,454991415    E ** ,375             SL 00510
0 01E 0     5D1E   5D1E       47                D      23838                                          SL 00520
0 01F 0     4A28   4A28       48                D      18984                                          SL 00530
0 020 0     0081   0081       49                D      129       1,648721271    E ** ,5               SL 00540
0 021 0     6984   6984       50                D      27012                                          SL 00550
0 022 0     531C   531C       51                D      21276                                          SL 00560
0 023 0     0081   0081       52                D      129       1,868245957    E ** ,625             SL 00570
0 024 0     7791   7791       53                D      30609                                          SL 00580
0 025 0     2BBF   2BBF       54                D      11199                                          SL 00590
```

```
                                   *                                                          SL 00600
  0 026 0   5889   5889            LN2      D      22713      LN(2)              .693147181   SL 00610
  0 027 0   05FE   05FE                     D      1534                                       SL 00620
  0 028 0   FA02   FA02            NL22     D      -1534                                      SL 00630
                                   **************************************************         SL 00640
* 0 029 0   27FF   2 0 7FF         CFAD     JSX    FAD
            002A   002A            MEXP     EQU    $                                          SL 00650
  0 02A 0   8029   8 0 029                  LDW    CFAD
  0 02B 0   7076   7 0 076                  STW    ALTE       SECURE ORIGINAL INSTRUCTION
* 0 02C 0   67FF   6 0 7FF                  STX    RET5       SAVE RETURN                     SL 00660
  0 02D 0   9800   9 1 000                  LDX *  0                                          SL 00670
  0 02E 0   6030   6 0 030                  STX    ARG                                        SL 00680
* 0 02F 0   27FF   2 0 7FF                  JSX    FLD        FETCH ARGUMENT 'X'              SL 00690
  0 030 0   0000   0000            ARG      D      0        **                                SL 00700
  0 031 0   7030   7 0 030                  STW    ARG        SAVE ARGUMENT SIGN              SL 00710
                                   * SCALE ARGUMENT                                           SL 00720
* 0 032 0   97FF   9 0 7FF                  LDX    MNT1       GET EXPONENT  X1                SL 00730
  0 033 0   0588   05 88                    DXS    136                                        SL 00740
  0 034 0   108A   1 0 08A                  JMP    BIG        ARGUMENT AT LEAST 128           SL 00750
  0 035 0   0810   0810                     SAP                                               SL 00760
* 0 036 0   27FF   2 0 7FF                  JSX    D2C        U  =  ABS,(X)                   SL 00770
* 0 037 0   9032   9 0 032                  LDX    MNT1                                       SL 00780
  0 038 0   0580   05 80                    DXS    128                                        SL 00790
  0 039 0   1042   1 0 042                  JMP    HIGH                                       SL 00800
                                   * ARGUMENT IS UNDER  1/2                                   SL 00810
                                   *                Q  =  0                                   SL 00820
                                   *                R  =  X2                                  SL 00830
            003A   003A            LOW      EQU    $                                          SL 00840
  0 03A 0   0100   0100                     CLR               SET EXPONENT  Q = 0
* 0 03B 0   77FF   7 0 7FF                  STW    TMP6                                       SL 00860
  0 03C 0   060F   06 0F                    LLB    15                                         SL 00870
  0 03D 0   0402   04 02                    IXS    2                                          SL 00880
  0 03E 0   1064   1 0 064                  JMP    SFTS       X1 IS UNDER -3                  SL 00890
  0 03F 0   068C   06 8C                    LLB    140                                        SL 00900
* 0 040 0   B037   B 0 037                  SUB    MNT1       SHIFT COUNT = 12 + X1           SL 00910
  0 041 0   1062   1 0 062                  JMP    SFTL                                       SL 00920
                                   * ARGUMENT IS AT LEAST  1/2                                SL 00930
                                   * POSITION X  AT B.0 IN A 3-WORD DIVIDEND                  SL 00940
                                   * AND DIVIDE BY LN(2)                                      SL 00950
                                   *                U  =  Q * LN(2) + R                       SL 00960
                                   *                Q  IS THE ONE-WORD QUOTIENT               SL 00970
                                   *                R IS THE TWO-WORD REMAINDER               SL 00980
  0 042 0   0130   0130            HIGH     CAX                                               SL 00990
  0 043 0   06B1   06 B1                    LLB    X'B1'      (SLL D 1) - 128                 SL 01000
* 0 044 0   A040   A 0 040                  ADD    MNT1       SHIFT COUNT = X1 + 1 (SIGN)     SL 01010
  0 045 0   306F   3 0 047 1                STB    HSFT+1                                     SL 01020
  0 046 0   0100   0100                     CLR                                               SL 01030
  0 047 0   0A30   0A3 0           HSFT     SLL D  0          **SHIFT LEFT UP TO B.0          SL 01040
* 0 048 0   703B   7 0 03B                  STW    TMP6       1ST WORD HAS BITS ABOVE B,0     SL 01050
  0 049 0   06B0   06 B0                    LLB    X'B0'                                      SL 01060
* 0 04A 0   A044   A 0 044                  ADD    MNT1       SHIFT COUNT = X1                SL 01070
* 0 04B 0   27FF   2 0 7FF                  JSX    DSM        GET 2ND AND 3RD WORDS           SL 01080
* 0 04C 0   97FF   9 0 7FF                  LDX    MNT3                                       SL 01090
```

```
*     0 04D 0  67FF  6 0 7FF     108            STX   TMP4      SAVE 3RD WORD                          SL 01100
*     0 04E 0  704C  7 0 04C     109            STW   MNT3      GET THE TWO HIGHER WORDS               SL 01110
*     0 04F 0  8048  8 0 048     110            LDW   TMP6      AS DIVIDEND                            SL 01120
*     0 050 0  77FF  7 0 7FF     111            STW   MNT2                                             SL 01130
      0 051 0  8026  8 0 026     112            LDW   LN2       LN(2) = D1 + 2**-15*D2                 SL 01140
*     0 052 0  27FF  2 0 7FF     113            JSX   DIVS      DIVIDE BY 1ST WORD OF LN(2)            SL 01150
*     0 053 0  704F  7 0 04F     114            STW   TMP6      SAVE Q                                 SL 01170
*     0 054 0  8050  8 0 050     115            LDW   MNT2      PLACE HIGHER WORD OF R IN              SL 01180
*     0 055 0  77FF  7 0 7FF     116            STW   TMP3      FRONT OF WORD 3 OF DIVIDEND            SL 01190
      0 056 0  8028  8 0 028     117            LDW   NL22      NEGATIVE 2ND WORD OF DIVISOR           SL 01200
*     0 057 0  27FF  2 0 7FF     118            JSX   MPYS      R  =  - Q * D2                         SL 01210
*     0 058 0  27FF  2 0 7FF     119            JSX   DAD       ...   + R                              SL 01220
*     0 059 0  3055  3055        120            D     TMP3                                             SL 01230
      0 05A 0  0820  0820        121            SAM             NEED ADJUSTING                         SL 01240
      0 05B 0  1061  1 0 061     122            JMP   SFTH      NO                                     SL 01250
*     0 05C 0  2058  2 0 058     123            JSX   DAD       YES  ...+ LN(2)                        SL 01260
      0 05D 0  0026  0026        124            D     LN2                                              SL 01270
*     0 05E 0  8053  8 0 053     125            LDW   TMP6                                             SL 01280
*     0 05F 0  B7FF  B 0 7FF     126            SUB   D1        Q  =  Q  -  1                          SL 01290
*     0 060 0  705E  7 0 05E     127            STW   TMP6                                             SL 01300
                                 128  * COMPUTE             E ** R                                   SL 01310
                                 129  *                     R  IS WITHIN THE RANGE (0, LN(2))        SL 01320
                                 130  * LET                 R = P + Z                                SL 01330
      0 061 0  060C  06 0C       131  SFTH      LLB   12                                             SL 01340
      0 062 0  9013  9 0 013     132  SFTL      LDX   D125                                           SL 01350
*     0 063 0  604A  6 0 04A     133            STX   MNT1                                             SL 01360
      0 064 0  30CD  3 0 066 1   134  SFTS      STB   SFT+1                                          SL 01370
*     0 065 0  8054  8 0 054     135            LDW   MNT2      ISOLATE LEFTMOST BITS AS P             SL 01380
      0 066 0  0A00  0A0 0       136  SFT       SRL   0                                              SL 01390
*     0 067 0  77FF  7 0 7FF     137            STW   TMP1                                             SL 01400
      0 068 0  0A11  0A1 1       138            SLL   1                                              SL 01410
*     0 069 0  A067  A 0 067     139            ADD   TMP1                                             SL 01420
      0 06A 0  A012  A 0 012     140            ADD   TAB       GET TO FACTOR TABLE                    SL 01430
      0 06B 0  7075  7 0 075     141            STW   CAL1      ADDRESS OF E**P                        SL 01440
      0 06C 0  7077  7 0 077     142            STW   CAL2                                             SL 01450
      0 06D 0  063F  06 3F       143            LLB   X'3F'                                            SL 01460
      0 06E 0  D066  D 0 066     144            ORE   SFT       SLL D (15-SFT)                         SL 01470
*     0 06F 0  2048  2 0 048     145            JSX   DSM       Z = R - P                              SL 01480
                                 146  * COMPUTE             E ** Z THROUGH TAYLOR'S SERIES           SL 01490
                                 147  *                     Z  IS WITHIN THE RANGE (0, 1/8)          SL 01500
                0070  0070       148  POLY      EQU   $                                              SL 01510
*     0 070 0  27FF  2 0 7FF     149            JSX   FNRM                                             SL 01520
      0 071 0  8028  8 0 028     150            LDW   NL22      SET ACR NEGATIVE FOR P(X)              SL 01530
*     0 072 0  27FF  2 0 7FF     151            JSX   M,P       (Z**6/6!+...Z**2/2+Z)...               SL 01540
      0 073 0  8000  8000        152            D     TAYL+X'8000'                                     SL 01550
*     0 074 0  27FF  2 0 7FF     153            JSX   FMP       ... * E**P                             SL 01560
      0 075 0  0000  0000        154  CAL1      D     0       **                                       SL 01570
                0076  0076       155  ALTE      EQU   $         NEXT MAY BE ALTERED BY MTNH
*     0 076 0  2029  2 0 029     156            JSX   FAD       ... + E**P                             SL 01580
      0 077 0  0000  0000        157  CAL2      D     0       **                                       SL 01590
                                 158  *  Y  =  E ** X                                                SL 01600
*     0 078 0  8060  8 0 060     159            LDW   TMP6      GIVE Q AS EXPONENT FOR V               SL 01610
*     0 079 0  A7FF  A 0 7FF     160            ADD   D129
```

```
*  0 07A 0  7063  7 0 063   161             STW   MNT1                                SL 01620
   0 07B 0  9030  9 0 030   162   CHKS      LDX   ARG       ARGUMENT SIGN             SL 01630
   0 07C 0  0500  05 00     163             DXS   0                                   SL 01640
   0 07D 0  1085  1 0 085   164             JMP   CHKE                                SL 01650
*  0 07E 0  27FF  2 0 7FF   165   NEGE      JSX   FST       NEGATIVE, Y = 1 / V       SL 01660
*  0 07F 0  37FF  37FF      166             D     TMP7                                SL 01670
*  0 080 0  202F  2 0 02F   167             JSX   FLD                                 SL 01680
*  0 081 0  37FF  37FF      168             D     F,1                                 SL 01690
*  0 082 0  27FF  2 0 7FF   169             JSX   FDV                                 SL 01700
*  0 083 0  307F  307F      170             D     TMP7                                SL 01710
   0 084 0  1088  1 0 088   171             JMP   EXIT                                SL 01720
   0 085 0  0600  06 00     172   CHKE      LLB   0         POSITIVE, CHECK EXPONENT  SL 01730
   0 086 0  0800  0800      173             SAZ                                       SL 01740
*  0 087 0  27FF  2 0 7FF   174   OVER      JSX   M,OV      OVERFLOW                  SL 01750
*  0 088 0  902C  9 0 02C   175   EXIT      LDX   RET5      RETURN TO CALLER          SL 01760
   0 089 0  1801  1 1 001   176             JMP   * 1                                 SL 01770
                            177   * ARGUMENT OUTSIDE RANGE (-2**7, 2**7)              SL 01780
            008A  008A      178   BIG       EQU   $                                   SL 01790
   0 08A 0  0820  0820      179             SAM             TEST ARGUMENT SIGN        SL 01800
   0 08B 0  1087  1 0 087   180             JMP   OVER      OVERFLOW                  SL 01810
*  0 08C 0  27FF  2 0 7FF   181             JSX   M,ZE      UNDERFLOW, GIVE ZERO      SL 01820
   0 08D 0  1088  1 0 088   182             JMP   EXIT                                SL 01830
                            183   ***********************************************     SL 01840
                            184             NTRY  MEXP                                SL 01850
   0 08D 0*******141        185             END                                       SL 01860
```

X-REF

```
    0 076 0   ALTE    0 028 0   0 076 0
    0 030 0   ARG     0 02E 0   0 031 0   0 07B 0
    0 08A 0   BIG     0 034 0   0 08A 0
    0 075 0   CAL1    0 06B 0
    0 077 0   CAL2    0 06C 0
    0 029 0   CFAD    0 02A 0
    0 085 0   CHKE    0 07D 0
    0 07B 0   CHKS
EXT 005C      DAD     0 058 0   0 05C 0
EXT 0052      DIVS    0 052 0
EXT 006F      DSM     0 04B 0   0 06F 0
EXT 005F      D1      0 05F 0
    0 013 0   D125    0 062 0
EXT 0079      D129    0 079 0
EXT 0036      D2C     0 036 0
    0 088 0   EXIT    0 084 0   0 08D 0
EXT 0081      F,1     0 081 0
EXT 0076      FAD     0 029 0   0 076 0
EXT 0082      FDV     0 082 0
EXT 0080      FLD     0 02F 0   0 080 0
EXT 0074      FMP     0 074 0
EXT 0070      FNRM    0 070 0
EXT 007E      FST     0 07E 0
    0 042 0   HIGH    0 039 0
    0 047 0   HSFT    0 045 0
```

```
    0 026 0  LN2        0 051 0    0 05D 0
    0 03A 0  LOW        0 03A 0
EXT 0087     M,OV       0 087 0
EXT 0072     M,P        0 072 0
EXT 008C     M,ΣE       0 08C 0
LIB 0 02A 0  MEXP       0 02A 0
EXT 007A     MNT1       0 032 0    0 037 0    0 040 0    0 044 0    0 04A 0    0 063 0    0 07A 0
EXT 0065     MNT2       0 050 0    0 054 0    0 065 0
EXT 004E     MNT3       0 04C 0    0 04E 0
EXT 0057     MPYS       0 057 0
    0 07E 0  NEGE
    0 028 0  NL22       0 056 0    0 071 0
    0 087 0  OVER       0 088 0
    0 070 0  POLY       0 070 0
EXT 0088     RET5       0 02C 0    0 088 0
    0 066 0  SFT        0 064 0    0 06F 0
    0 061 0  SFTH       0 05B 0
    0 062 0  SFTL       0 041 0
    0 064 0  SFTS       0 03E 0
    0 012 0  TAB        0 06A 0
    0 014 0  TABL       0 012 0
    0 000 0  TAYL       0 000 0    0 073 0
EXT 0069     TMP1       0 067 0    0 069 0
EXT 0059     TMP3       0 055 0    0 059 0
EXT 004D     TMP4       0 04D 0
EXT 0078     TMP6       0 03B 0    0 048 0    0 04F 0    0 053 0    0 05E 0    0 060 0    0 078 0
EXT 0083     TMP7       0 07F 0    0 083 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
185	52	613	0	6

NATURAL LOGARITHM - MP FLOATING POINT

DATE: April 1968

ID CODE: BSK

DRAWING: 390008

LABEL: MLOG

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To find the natural logarithm of a mid-precision floating point number and leave the result in the same format in the software registers MNT1, MNT2, MNT3.

USAGE

Calling Sequence

The routine returns at L+2.

```
L-1     SMB     MLOG
L       JSX     MLOG
L+1     D       ARG
L+2     Return
```

Argument Description

The argument is a mid-precision floating point number occupying three consecutive words in memory, beginning at address "ARG" as pointed out in the calling sequence.

METHOD

Let the argument be

$$X = Xm \cdot 2^{Xe}$$

$$.5 \leq Xm < 1$$

$$Ln\ X = Ln\ Xm + Xe \cdot Ln\ 2$$

Let Xm be redefined as the product of two factors, Ax and x, Ax a multiple of 1/8:

$$Xm = Ax \cdot x$$

$$1/2 \leq Ax \leq 7/8$$

$$0 \leq Xm - Ax < 1/8$$

$$Ln\ Xm = Ln\ Ax + Ln\ X$$

Ln Ax is obtained from a 4-entry table. Let

$$z = \frac{Xm - Ax}{Xm + Ax} \qquad 0 \leq z < 1/8$$

$$x = \frac{1 + z}{1 - z}$$

Subtracting the two Taylor series of Ln(1+z) and Ln(1-z):

$$Ln\ x = 2(z+z^3/3 \ldots +z^9/9) + E$$

$$E < 2^{-31}$$

ERROR CONDITIONS

A negative or a null argument will result in a call to the subroutine M.OV.

RANGE AND ACCURACY

Any positive argument is acceptable. The result is accurate to 29 bits.

RESTRICTIONS

Loading

This subroutine locally references storage words and constants located in the "MATH POOL" module, and must therefore be loaded in same 2K block as the pool.

Other Routines

DDV, DSM, FAD, FLD, FLT, FMP, FSB, FST, M.OV, M.P

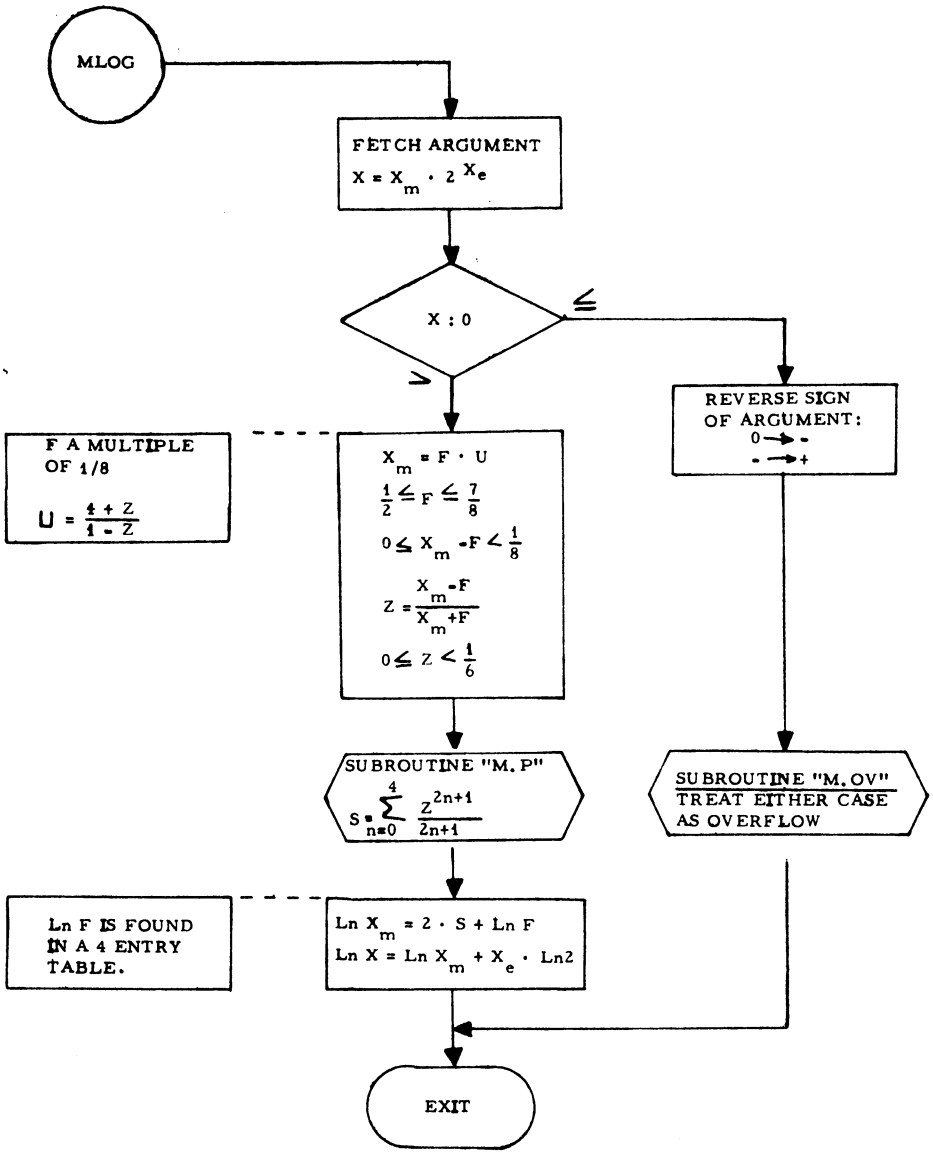
Space Used

83 words

Time (in cycles)

	Standard	Hardware Multiply & Divide
Minimum	4000	2700
Maximum	4600	2850
Average	4230	2740

NATURAL LOGARITHM - MP FLOATING POINT



APPENDIX A

ASSEMBLY LISTING

of

NATURAL LOGARITHM - MP FLOATING POINT

Drawing No.

390008

ID Code

BSK

```
                                 'NATURAL LOGARITHM - MP FLOATING POINT 390008'          BSK 0005
                                 *                                                       BSK 0006
                                 *                                                       BSK 0007
                                           BLK   MATH                                    BSK 0008
                                           LIBR  MLOG                                    BSK 0009
                                 * ARGUMENT   Y = X * 2 ** XP                            BSK 0010
                                 * LET     X     = F * U                                 BSK 0011
                                 *         F     = (N + I) / 2N, WITH I LESS THAN N      BSK 0012
                                 * AND     Z     = (X - F) / (X + F)                     BSK 0013
                                 * THEN    U     = (1 + Z) / (1 - Z)                     BSK 0014
                                 * AND     LN(U) = 2* ( Z + Z**3/3 + ,,, )               BSK 0015
                                 *               WITH Z LESS THAN 1/N                    BSK 0016
                                 *         LN(Y) = LN(U) + LN(F) + XP * LN(2)            BSK 0017
                                 *               N IS CHOSEN EQUAL TO 4                  BSK 0018
                                 *               TO LIMIT THE LN(F) TABLE TO 4 ENTRIES   BSK 0019
                                 ******************************************************  BSK 0020
              0000  0000         TABL      EQU   $                                       BSK 0021
              0000  0000         LN2       EQU   ,                                       BSK 0022
     0 000 0  0080  0080                   D     128      ,69314 71806  = -LN(,5)        BSK 0023
     0 001 0  58B9  58B9                   D     22713                  =  LN(2,)        BSK 0024
     0 002 0  05FE  05FE                   D     1534                                    BSK 0025
     0 003 0  7FF0  7FF0         TAB       D     TABL+X'7FF0'                            BSK 0026
     0 004 0  007F  007F                   D     127      ,47000 36292  = -LN(,625)      BSK 0027
     0 005 0  7852  7852                   D     30802                                   BSK 0028
     0 006 0  1434  1434                   D     5172                                    BSK 0029
     0 007 0  F000  F000         M4L       D     X'F000'                                 BSK 0030
     0 008 0  007F  007F                   D     127      ,28768 20724  = -LN(,750)      BSK 0031
     0 009 0  49A5  49A5                   D     18853                                   BSK 0032
     0 00A 0  4422  4422                   D     17442                                   BSK 0033
     0 00B 0  0000  0000         SAVE      D     0                                       BSK 0034
     0 00C 0  007E  007E                   D     126      ,13353 13926  = -LN(,875)      BSK 0035
     0 00D 0  445E  445E                   D     17502                                   BSK 0036
     0 00E 0  1D04  1D04                   D     7428                                    BSK 0037
              000F  000F         TAYL      EQU   $                                       BSK 0038
              000F  000F         D125      EQU   $                                       BSK 0039
     0 00F 0  007D  007D                   D     125      1/9                            BSK 0040
     0 010 0  71C7  71C7                   D     29127                                   BSK 0041
     0 011 0  0E39  0E39                   D     3641                                    BSK 0042
     0 012 0  007E  007E                   D     126      1/7                            BSK 0043
     0 013 0  4924  4924                   D     18724                                   BSK 0044
     0 014 0  4925  4925                   D     18725                                   BSK 0045
     0 015 0  007E  007E                   D     126      1/5                            BSK 0046
     0 016 0  6666  6666                   D     26214                                   BSK 0047
     0 017 0  3333  3333                   D     13107                                   BSK 0048
     0 018 0  FFFF  FFFF                   D     -1       FLAG EXTRA COEFFICIENT         BSK 0049
     0 019 0  007F  007F                   D     127      1/3                            BSK 0050
     0 01A 0  5555  5555                   D     21845                                   BSK 0051
     0 01B 0  2AAB  2AAB                   D     10923                                   BSK 0052
                                 ******************************************************  BSK 0053
              001C  001C         MLOG      EQU   $                                       BSK 0054
*    0 01C 0  67FF  6 0 7FF                STX   RET5     SAVE RETURN                    BSK 0055
     0 01D 0  9800  9 1 000                LDX * 0                                       BSK 0056
     0 01E 0  6020  6 0 020                STX   ARG                                     BSK 0057
```

```
*   0 01F 0  27FF  2 0 7FF    55                JSX   FLD                                        BSK 0058
    0 020 0  0000  0000       56   ARG          D     0        **                                BSK 0059
    0 021 0  0800  0800       57                SAZ                                              BSK 0060
    0 022 0  0810  0810       58                SAP                                              BSK 0061
    0 023 0  104F  1 0 04F    59                JMP   BAD      NULL (OVERFLOW) OR NEGATIVE       BSK 0062
    0 024 0  E007  E 0 007    60                AND   M4L                                        BSK 0063
*   0 025 0  77FF  7 0 7FF    61                STW   TMP6     SAVE FIRST 3 BITS OF X AS F       BSK 0064
    0 026 0  0A56  0A5 6      62                SLC   6        GET TO LN(F)                      BSK 0065
    0 027 0  A003  A 0 003    63                ADD   TAB                                        BSK 0066
    0 028 0  7044  7 0 044    64                STW   CALF     SET ADDR, IN 'FSB' CALL           BSK 0067
(   0 029 0  8025  8 0 025    65                LDW   TMP6                                       BSK 0068
*   0 02A 0  D7FF  D 0 7FF    66                ORE   MNT2                                       BSK 0069
*   0 02B 0  702A  7 0 02A    67                STW   MNT2     X - F                             BSK 0070
*   0 02C 0  8029  8 0 029    68                LDW   TMP6                                       BSK 0071
*   0 02D 0  97FF  9 0 7FF    69                LDX   MNT3                                       BSK 0072
    0 02E 0  0A31  0A3 1      70                SLL D 1                                          BSK 0073
*   0 02F 0  C02B  C 0 02B    71                ORI   MNT2     X + F                             BSK 0074
    0 030 0  0A21  0A2 1      72                SRL D 1                                          BSK 0075
*   0 031 0  77FF  7 0 7FF    73                STW   TMP3                                       BSK 0076
    0 032 0  0140  0140       74                CXA                                              BSK 0077
    0 033 0  0A01  0A0 1      75                SRL   1                                          BSK 0078
*   0 034 0  77FF  7 0 7FF    76                STW   TMP4                                       BSK 0079
    0 035 0  0632  06 32      77                LLB   X'32'    SLL D 2                           BSK 0080
*   0 036 0  27FF  2 0 7FF    78                JSX   DSM      SHIFT APPLIES TO (X-F)            BSK 0081
*   0 037 0  27FF  2 0 7FF    79                JSX   DDV      Z=(X-F) / (X+F)                   BSK 0082
*   0 038 0  3031  3031       80                D     TMP3                                       BSK 0083
*   0 039 0  87FF  8 0 7FF    81                LDW   MNT1     ARGUMENT EXPONENT                 BSK 0084
*   0 03A 0  B7FF  B 0 7FF    82                SUB   D128     GET RID OF BIAS                   BSK 0085
*   0 03B 0  702C  7 0 02C    83                STW   TMP6     SAVE AS INTEGER 'XP'              BSK 0086
    0 03C 0  800F  8 0 00F    84                LDW   D125     EXPONENT FOR Z                    BSK 0087
*   0 03D 0  7039  7 0 039    85                STW   MNT1                                       BSK 0088
             003E  003E       86   POLY         EQU   $                                          BSK 0089
                              87   * NOTE - BE SURE ACR IS POSITIVE FOR P(X**2)                  BSK 0090
*   0 03E 0  27FF  2 0 7FF    88                JSX   M,P      (Z + Z**3/3 +,,,+ Z**9/9),,,      BSK 0091
    0 03F 0  800F  800F       89                D     TAYL+X'8000'                               BSK 0092
*   0 040 0  803D  8 0 03D    90                LDW   MNT1                                       BSK 0093
*   0 041 0  A7FF  A 0 7FF    91                ADD   D1       ,,, * 2 = LN(U)                   BSK 0094
*   0 042 0  7040  7 0 040    92                STW   MNT1                                       BSK 0095
*   0 043 0  27FF  2 0 7FF    93                JSX   FSB      LN(X) = LN(U) + LN(F)             BSK 0096
    0 044 0  0000  0000       94   CALF         D     0        **                                BSK 0097
*   0 045 0  27FF  2 0 7FF    95                JSX   FST      SAVE LN(X)                        BSK 0098
*   0 046 0  37FF  37FF       96                D     TMP7                                       BSK 0099
*   0 047 0  27FF  2 0 7FF    97                JSX   FLT      FLOAT 'XP'                        BSK 0100
*   0 048 0  303B  303B       98                D     TMP6                                       BSK 0101
*   0 049 0  27FF  2 0 7FF    99                JSX   FMP      XP * LN(2) ,,,                    BSK 0102
    0 04A 0  0000  0000      100                D     LN2                                        BSK 0103
*   0 04B 0  27FF  2 0 7FF   101                JSX   FAD      ,,, + LN(X) = LN(Y)               BSK 0104
*   0 04C 0  3046  3046      102                D     TMP7                                       BSK 0105
*   0 04D 0  901C  9 0 01C   103   EXIT         LDX   RET5                                       BSK 0106
    0 04E 0  1801  1 1 001   104                JMP * 1                                          BSK 0107
             004F  004F      105   BAD          EQU   $        ERROR - NULL OR NEGATIVE          BSK 0108
    0 04F 0  0120  0120      106                INV            REVERSE SIGN FOR OVERFLOW         BSK 0109
*   0 050 0  702F  7 0 02F   107                STW   MNT2                                       BSK 0110
```

```
   *     0 051 0  27FF  2 0 7FF     108            JSX   M,8V                                    BSK 0111
         0 052 0  104D  1 0 04D     109            JMP   EXIT                                    BSK 0112
                                    110  ************************************************         BSK 0113
                                    111            NTRY  MLOG                                    BSK 0114
         0 052 0*******82           112
```

X-REF

```
     0 020 0  ARG
     0 04F 0  BAD
     0 044 0  CALF
EXT  0037     DDV
EXT  0036     DSM
EXT  0041     D1
     0 00F 0  D125
EXT  003A     D128
     0 04D 0  EXIT
EXT  004B     FAD
EXT  001F     FLD
EXT  0047     FLT
EXT  0049     FMP
EXT  0043     FSB
EXT  0045     FST
     0 000 0  LN2
EXT  0051     M.8V
EXT  003E     M.P
LIB  0 01C 0  MLOG
EXT  0042     MNT1
EXT  0050     MNT2
EXT  002D     MNT3
     0 007 0  M4L
     0 03E 0  POLY
EXT  004D     RET5
     0 00B 0  SAVE
     0 003 0  TAB
     0 000 0  TABL       0 000 0
     0 00F 0  TAYL
EXT  0038     TMP3
EXT  0034     TMP4
EXT  0048     TMP6
EXT  004C     TMP7
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
112	33	620	0	6

DATE: May 1968

ID CODE: BSS

DRAWING: 390016

LABEL: M.P

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To assist the mathematical function subroutines in the expansion of series in the mid-precision floating point format.

This subroutine can sum up:

$$P_1 = X + A2 \cdot X^2 \ldots + An \cdot X^n$$
$$\text{or } P_2^1 = A1 \cdot X^2 + A2 \cdot X^4 \ldots + An \cdot X^{2n}$$
$$\text{or } P_3^2 = X + A1 \cdot X^3 \ldots + An \cdot X^{2n+1}$$

USAGE

The use of M.P is not encouraged outside the library. The reader is referred to the assembly listings of MEXP, MLOG, MSIN, MATN for examples of calling sequences and coefficient tables.

Error Conditions

Overflow conditions are flagged by FAD and FMP, routines called by M.P.

RESTRICTIONS

Loading

M.P must be loaded in the same 2K block as the "MATH POOL".

Other Routines

FAD, FMP, FNRM, FST

Space Used

40 words

Timing (in cycles)

	Standard Hardware	Hardware Multiply
P1	398 + 507 (n-1)	233 + 342 (n-1)
P2	307 + 507n	142 + 342n
P3	291 + 507n	126 + 342n

Deduct or add 20 cycles per term positive or negative.

APPENDIX A

ASSEMBLY LISTING

of

POLYNOMIAL - MP FLOATING POINT

Drawing No.

390016

ID Code

BSS

```
                                2   'POLYNOMIAL - MP FLOATING POINT 390016'                         BSS 0005
                                3   *                                                               BSS 0006
                                4   *                                                               BSS 0007
                                5           BLK   MATH                                              BSS 0008
                                6           LIBR  M.P                                               BSS 0009
                                7   *       COMPUTE     AN*X**N + ,,, + A2*X**2 + X                 BSS 0010
                                8   *       OR          AN*X**2N + ,,, + A1*X**2                    BSS 0011
                                9   *       OR          AN*X**(2N+1) + .,, + A1*X**3 + X            BSS 0012
               0000  0000      10   M.P     EQU   $                                                 BSS 0013
*    0 000 0   67FF  6 0 7FF   11           STX   RET4      SAVE RETURN                             BSS 0014
     0 001 0   9800  9 1 000   12           LDX * 0         TABLE POINTER (SIGN BIT ON)             BSS 0015
     0 002 0   600D  6 0 00D   13           STX   CAL1                                              BSS 0016
     0 003 0   0810  0810      14           SAP             KIND OF POLYNOMIAL                      BSS 0017
                               15   *                       TERM FOR EACH POWER                     BSS 0018
     0 004 0   100A  1 0 00A   16           JMP   POLY      (NORMALIZED ARGUMENT ONLY)              BSS 0019
                               17   *                       TERM FOR EVERY OTHER POWER              BSS 0020
*    0 005 0   27FF  2 0 7FF   18           JSX   FNRM      NORMALIZE ARGUMENT                      BSS 0021
*    0 006 0   27FF  2 0 7FF   19           JSX   FST       STORE ARGUMENT AS 'Z'                   BSS 0022
*    0 007 0   37FF  37FF      20           D     TM10                                              BSS 0023
*    0 008 0   27FF  2 0 7FF   21           JSX   FMP       GET SQUARE OF ARGUMENT                  BSS 0024
*    0 009 0   3007  3007      22           D     TM10                                              BSS 0025
               000A  000A      23   POLY    EQU   $                                                 BSS 0026
*    0 00A 0   2006  2 0 006   24           JSX   FST       SAVE PROPER POWER                       BSS 0027
*    0 00B 0   37FF  37FF      25           D     TMP7      OF ARGUMENT X                           BSS 0028
*    0 00C 0   2008  2 0 008   26           JSX   FMP                                               BSS 0029
     0 00D 0   0000  0000      27   CAL1    D     0     **                                          BSS 0030
     0 00E 0   9800  9 1 000   28           LDX * 0                                                 BSS 0031
     0 00F 0   0403  04 03     29   NEXT    IXS   3                                                 BSS 0032
     0 010 0   6012  6 0 012   30           STX   CALL                                              BSS 0033
*    0 011 0   27FF  2 0 7FF   31           JSX   FAD                                               BSS 0034
     0 012 0   0000  0000      32   CALL    D     0     **                                          BSS 0035
*    0 013 0   200C  2 0 00C   33           JSX   FMP                                               BSS 0036
*    0 014 0   300B  300B      34           D     TMP7                                              BSS 0037
     0 015 0   9012  9 0 012   35           LDX   CALL                                              BSS 0038
     0 016 0   8803  8 1 003   36           LDW * 3                                                 BSS 0039
     0 017 0   0820  0820      37           SAM                                                     BSS 0040
     0 018 0   100F  1 0 00F   38           JMP   NEXT                                              BSS 0041
*    0 019 0   A7FF  A 0 7FF   39           ADD   D1                                                BSS 0042
     0 01A 0   0810  0810      40           SAP             ANY EXTRA                               BSS 0043
     0 01B 0   1026  1 0 026   41           JMP   EXIT      NO                                      BSS 0044
     0 01C 0   0404  04 04     42           IXS   4         YES, GET TO LAST COEFFICIENT            BSS 0045
     0 01D 0   601F  6 0 01F   43           STX   CALN                                              BSS 0046
*    0 01E 0   2011  2 0 011   44           JSX   FAD                                               BSS 0047
     0 01F 0   0000  0000      45   CALN    D     0     **                                          BSS 0048
*    0 020 0   2013  2 0 013   46           JSX   FMP                                               BSS 0049
*    0 021 0   3014  3014      47           D     TMP7                                              BSS 0050
*    0 022 0   2020  2 0 020   48           JSX   FMP                                               BSS 0051
*    0 023 0   3009  3009      49           D     TM10                                              BSS 0052
*    0 024 0   201E  2 0 01E   50           JSX   FAD       I,E  A1 = 1                             BSS 0053
*    0 025 0   3023  3023      51           D     TM10                                              BSS 0054
*    0 026 0   9000  9 0 000   52   EXIT    LDX   RET4                                              BSS 0055
     0 027 0   1801  1 1 001   53           JMP * 1                                                 BSS 0056
                               54   ************************************************                BSS 0057
```

```
                                55          NTRY  M.P                                BSS 0058
        0 027 0*******39        56
```

X-REF

```
    0 012 0  CALL
    0 01F 0  CALN
    0 00D 0  CAL1
EXT 0019     D1
    0 026 0  EXIT
EXT 0024     FAD
EXT 0022     FMP
EXT 0005     FNRM
EXT 000A     FST
LIB 0 000 0  M.P        0 000 0
    0 00F 0  NEXT
    0 00A 0  POLY
EXT 0026     RET4       0 000 0
EXT 0021     TMP7
EXT 0025     TM10
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
56	15	624	0	2

DOUBLE SHIFT MAGNITUDE

DATE: May 1968
ID CODE: BST
DRAWING: 390017
LABEL: DSM
AUTHOR: JACQ
SOURCE: SYM I
OBJECT: Relocatable in Block "MATH"

PURPOSE

To assist the mathematical function subroutines in the double arithmetic shift operations.

USAGE

The use of DSM is not encouraged outside the library.

The shift is performed on the contents of the software registers MNT2, MNT3. The sign bit of MNT3 is left out of the shift. On return both sign bits are cleaned off and MNT2 is duplicated in the accumulator register.

Calling Sequence

Place the right byte of the shift instruction (left or right, double, shift count up to 15) in the accumulator register, before calling the routine:

```
L-1     SMB     DSM
L       JSX     DSM
L+1     Return
```

Error Conditions

No check is made on the right half of the shift instruction provided by the caller.

RESTRICTIONS

Loading

This subroutine locally references storage words located in the "MATH POOL" module, and must therefore be loaded in the same 2K block as the pool.

Other Routines

None

Space Used

12 words

Timing (in cycles)

Maximum 26
Minimum 23
Average 24

APPENDIX A

ASSEMBLY LISTING

of

DOUBLE SHIFT MAGNITUDE

Drawing No.

390017

ID Code

BST

```
                                      2   'DOUBLE SHIFT MAGNITUDE 390017'                        BST 0005
                                      3   *                                                      BST 0006
                                      4   *                                                      BST 0007
                                      5            BLK   MATH                                    BST 0008
                                      6            LIBR  DSM                                     BST 0009
                                      7   * SHIFT CODE AND COUNT IN ACCUMULATOR                  BST 0010
                 0000  0000           8   DSM      EQU   $                                       BST 0011
*   0 000 0  67FF  6 0 7FF            9            STX   RET1                                    BST 0012
    0 001 0  300B  3 0 005 1         10            STB   SFTC+1                                  BST 0013
*   0 002 0  97FF  9 0 7FF           11            LDX   MNT3                                    BST 0014
    0 003 0  0A31  0A3 1             12            SLL D 1                                       BST 0015
*   0 004 0  87FF  8 0 7FF           13            LDW   MNT2                                    BST 0016
    0 005 0  0A00  0A00              14   SFTC     D     X'A00' **COUNT AND MODE SET FROM ACCR   BST 0017
    0 006 0  0A11  0A1 1             15            SLL   1                                       BST 0018
    0 007 0  0A21  0A2 1             16            SRL D 1                                       BST 0019
*   0 008 0  7004  7 0 004           17            STW   MNT2                                    BST 0020
*   0 009 0  6002  6 0 002           18            STX   MNT3                                    BST 0021
*   0 00A 0  9000  9 0 000           19            LDX   RET1                                    BST 0022
    0 00B 0  1800  1 1 000           20            JMP * 0                                       BST 0023
                                     21   ***********************************************        BST 0024
                                     22            NTRY  DSM                                     BST 0025

    0 00B 0*******11                 23

X-REF

LIB 0 000 0  DSM     0 000 0
EXT 0008     MNT2
EXT 0009     MNT3
EXT 000A     RET1    0 000 0
    0 005 0  SFTC    0 001 0

NO ERRORS

CARDS  SYMBOLS    LITR  STACK
23     5     627  0     6
```

RAYTHEON

QUALITY SOFTWARE

700 PROGRAMMING SYSTEMS

MP HYPERBOLIC TANGENT

DATE: April 1969

ID CODE: BYB

DRAWING: 392279

LABEL: MTNH

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in block "Math"

PURPOSE

To find the hyperbolic tangent of a mid-precision floating point number and leave the result in the same format in the software registers MNT1, MNT2, MNT3.

USAGE

Calling Sequence

```
L-1     SMB     MTNH
L       JSX     MTNH
L+1     D       ARG
L+2     Return
```

Argument Description

The argument is a mid-precision floating point variable occupying three contiguous words in memory, beginning at address "ARG" as indicated in the calling sequence.

METHOD

Operations are carried on the absolute value X of the variable x, then the function is given the sign of x.

The range of $X = |x|$ is divided into four intervals:

$0 \leq X < 2^{-29}$ $\qquad \tanh x = x$

$2^{-29} \leq X < 1$ $\qquad \tanh x = \dfrac{e^{2x} - 1}{e^{2x} + 1}$

$1 \leq X < 16$ $\qquad \tanh x = \dfrac{e^{2x}-1}{e^{2x}+1}$

$16 \leq X$ $\qquad \tanh x = \dfrac{x}{X}$

In order to keep the accuracy of the function when X is within the interval $(2^{-29}, 1)$, it is necessary to obtain the difference $e^{2X}-1$ directly, rather than subtracting 1 from e^{2x}.

The exponential routine "MEXP" works as follows:

$$2X = Q \cdot \text{Ln } 2 + [F + Z \pmod{1/8}] \pmod{\text{Ln } 2}$$
$$e^{2X} = e^{F} \cdot e^{Z} \cdot 2^{Q}$$

A special exit is provided in "MEXP" to give the partial result Y:

$$Y = (e^{Z}-1)\, e^{F}$$

Upon return from MEXP the expression $e^{2x}-1$ is obtained as follows:

$$e^{2X}-1 = [Y + (e^{F}-1)]\ 2^{Q} + (2^{Q}-1)$$

ERROR CONDITIONS

None

ACCURACY

28 magnitude bits

RESTRICTIONS

Loading

This subroutine locally references storage words and constants located in the "Math Pool" module, and must therefore be loaded in the same 2K block as the pool.

MEXP References

Three non-entry words of the subroutine MEXP are referenced in MTNH as relocatable addresses. If these words are displaced in MEXP, their addresses should be redefined in MTNH.

Other Routines

D2C, FAD, FDV, FLD, FSB, FST, MEXP

Space Used

91 words

Timing (Machine Cycles)

	Without Hardware Multiply and Divide	With Hardware Multiply and Divide
Minimum ($X \angle 2^{-29}$)	35	35
Maximum	4800	3100
Average	4550	2950

APPENDIX A

ASSEMBLY LISTING

of

MP HYPERBOLIC TANGENT

Drawing No.	ID Code
392279	BYB

```
                    1 'MP HYPERBOLIC TANGENT DN392279 A'                                    YB 00000
                    2 * Y = (E**2X-1)/(E**2X+1)                                             YB 00010
                    3           BLK   MATH                                                  YB 00020
                    4           LIBR  MTNH                                                  YB 00030
                    5 ******************************************************              YB 00040
0000 0000           6 TEMP      D     0,0,0                                                 YB 00050
0001 0000
0002 0000
0003 0082           7 F2        D     130,X'4000'                                           YB 00060
0004 4000
0005 0000           8 TABL      D     0                                                     YB 00070
                    9 ******************************************************              YB 00080
                   10 * NEXT THREE CARDS TO AGREE WITH MEXP                                 YB 00090
0006 001B          11 TAB       D     TABL+X'16'      X'2A'-X'14'                           YB 00100
     004C          12 ALTE      EQU   X'4C'           X'76'-X'2A'                           YB 00110
     000D          13 ENTE      EQU   X'D'            X'37'-X'2A'                           YB 00120
                   14 ******************************************************              YB 00130
0007 0000          15 RET       D     0                                                     YB 00140
0008 007E          16           D     X'7E'     .1331484531     E**.125 = 1                 YB 00150
0009 442C          17           D     X'442C',X'458'                                        YB 00160
000A 0458
000B 007F          18           D     X'7F'     .2840254167     E**.250 = 1                 YB 00170
000C 4885          19           D     X'4885',X'71E1'                                       YB 00180
000D 71E1
000E 007F          20           D     X'7F'     .4549914146     E**.375 = 1                 YB 00190
000F 747A          21           D     X'747A',X'289E'                                       YB 00200
0010 289E
0011 0080          22           D     X'80'     .6487212707     E**.500 = 1                 YB 00210
0012 5309          23           D     X'5309',X'2638'                                       YB 00220
0013 2638
0014 0080          24           D     X'80'     .8682459574     E**.625 = 1                 YB 00230
0015 6F22          25           D     X'6F22',X'577D'                                       YB 00240
0016 577D
                   26 ******************************************************              YB 00250
     0017          27 MTNH      EQU   $                                                     YB 00260
0017 6007          28           STX   RET                                                   YB 00270
0018 8800          29           LDW * 0                                                     YB 00280
0019 701B          30           STW   LOAD                                                  YB 00290
*    001A 27FF     31           JSX   FLD             FETCH ARGUMENT X                      YB 00300
     001B 0000     32 LOAD      D     0                                                     YB 00310
*    001C 97FF     33           LDX   MNT1            SCALE MAGNITUDE                       YB 00320
     001D 0585     34           DXS   133                                                   YB 00330
     001E 1043     35           JMP   BIG             EQUAL OR OVER 16                      YB 00340
     001F 0423     36           IXS   35                                                    YB 00350
     0020 1059     37           JMP   EXIT            UNDER 2**(-29), Y = X                 YB 00360
     0021 0463     38           IXS   99                                                    YB 00370
*    0022 37FF     39 CALE      D     MEXP                                                  YB 00380
*    0023 601C     40           STX   MNT1            2 X                                   YB 00390
     0024 0810     41           SAP                                                         YB 00400
*    0025 27FF     42           JSX   D2C             WORK ON ABSOLUTE VALUE                YB 00410
*    0026 9023     43           LDX   MNT1                                                  YB 00420
     0027 0582     44           DXS   130                                                   YB 00430
     0028 1046     45           JMP   HIGH            X EQUAL OR OVER 1                     YB 00440
                   46 * ARGUMENT IS LESS THAN 1 - SECURE E**2X-1                           YB 00450
     0029 9022     47           LDX   CALE            ALTER INSTRUCTION IN                  YB 00460
     002A 802E     48           LDW   QUIT            MEXP ROUTINE TO JUMP                  YB 00470
```

```
    002B 784C    49          STW * ALTE        OUT WITH (E**Z-1)*E**F                YB 00480
*   002C 87FF    50          LDW   MNT2                                              YB 00490
    002D 180D    51          JMP * ENTE                                              YB 00500
    002E 202F    52 QUIT     JSX   BACK                                              YB 00510
         002F    53 BACK     EQU   $                                                 YB 00520
    002F 8800    54          LDW * 0           ADDRESS OF E**F                       YB 00530
    0030 B022    55          SUB   CALE                                              YB 00540
    0031 A006    56          ADD   TAB                                               YB 00550
    0032 7034    57          STW   CALF        ADDRESS OF E**F-1                     YB 00560
*   0033 27FF    58          JSX   FAD         (E**Z-1)*E**F*E**F-1                  YB 00570
    0034 0000    59 CALF     D     0                                                 YB 00580
         0035    60 FIXQ     EQU   $                                                 YB 00590
*   0035 87FF    61          LDW   TMP6                                              YB 00600
    0036 0800    62          SAZ               WAS 2X LESS THAN LN 2                 YB 00610
    0037 1039    63          JMP   $+2                                               YB 00620
    0038 104A    64          JMP   ALL         YES                                   YB 00630
    0039 0130    65          CAX               NO - FIX E**2X-1                      YB 00640
*   003A A026    66          ADD   MNT1        (E**2X-1)*2**Q ,..                    YB 00650
*   003B 703A    67          STW   MNT1                                              YB 00660
    003C 0850    68          SXE               ,..*2**Q-1                            YB 00670
    003D 1040    69          JMP   $+3         Q = 1                                 YB 00680
*   003E 2033    70          JSX   FAD         Q = 2                                 YB 00690
    003F 0003    71          D     F2                                                YB 00700
*   0040 203E    72          JSX   FAD                                               YB 00710
*   0041 37FF    73          D     F.1                                               YB 00720
    0042 104A    74          JMP   ALL                                               YB 00730
         0043    75 BIG      EQU   $           BIG ARGUMENT, Y = 1                   YB 00740
*   0043 201A    76          JSX   FLD                                               YB 00750
*   0044 3041    77          D     F.1                                               YB 00760
    0045 1054    78          JMP   SIGN                                              YB 00770
         0046    79 HIGH     EQU   $           X IS 1 OR OVER                        YB 00780
*   0046 2022    80          JSX   MEXP        E**2X                                 YB 00790
*   0047 303B    81          D     MNT1                                              YB 00800
*   0048 27FF    82          JSX   FSB         E**2X-1                               YB 00810
*   0049 3044    83          D     F.1                                               YB 00820
         004A    84 ALL      EQU   $           Y = (E**2X-1)/(E**2X+1)                YB 00830
*   004A 27FF    85          JSX   FST         E**2X-1                               YB 00840
    004B 0000    86          D     TEMP                                              YB 00850
*   004C 2040    87          JSX   FAD                                               YB 00860
    004D 0003    88          D     F2                                                YB 00870
*   004E 204A    89          JSX   FST         E**2X+1                               YB 00880
*   004F 37FF    90          D     TMP1                                              YB 00890
*   0050 2043    91          JSX   FLD                                               YB 00900
    0051 0000    92          D     TEMP                                              YB 00910
*   0052 27FF    93          JSX   FDV                                               YB 00920
*   0053 304F    94          D     TMP1                                              YB 00930
         0054    95 SIGN     EQU   $                                                 YB 00940
    0054 9007    96          LDX   RET                                               YB 00950
    0055 9800    97          LDX * 0                                                 YB 00960
    0056 8801    98          LDW * 1                                                 YB 00970
    0057 0810    99          SAP                                                     YB 00980
*   0058 2025   100          JSX   D2C                                               YB 00990
         0059   101 EXIT     EQU   $                                                 YB 01000
    0059 9007   102          LDX   RET                                               YB 01010
    005A 1801   103          JMP * 1                                                 YB 01020
                104          NTRY  MTNH                                              YB 01030
```

```
      105          END                                                    YB 01040

0058  D2C
0049  F.1
004C  FAD
0052  FDV
0050  FLD
0048  FSB
004E  FST
0046  MEXP
0047  MNT1
002C  MNT2
0053  TMP1
0035  TMP6

NO ERRORS
```

ALL	004A	ALTE	004C	BACK	002F	BIG	0043
CALE	0022	CALF	0034	D2C	0058	ENTE	000D
EXIT	0059	F.1	0049	F2	0003	FAD	004C
FDV	0052	FIXQ	0035	FLD	0050	FSB	0048
FST	004E	HIGH	0046	LOAD	001B	MATH	0000
MEXP	0046	MNT1	0047	MNT2	002C	MTNH	0017
QUIT	002E	RET	0007	SIGN	0054	TAB	0006
TABL	0005	TEMP	0000	TMP1	0053	TMP6	0035