```
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMMMMM   MMMMMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMMMMM   MMMMMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMMMMM   MMMMMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM   MMM   MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM   MMM   MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM   MMM   MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR   RRR              TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR   RRR              TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR   RRR              TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR      RRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR      RRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR      RRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**MTHHATANH

```
MM      MM  TTTTTTTTTT  HH      HH  HH      HH    AAAAAA    TTTTTTTTTT    AAAAAA    NN      NN  HH      HH
MM      MM  TTTTTTTTTT  HH      HH  HH      HH    AAAAAA    TTTTTTTTTT    AAAAAA    NN      NN  HH      HH
MMMM  MMMM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NN      NN  HH      HH
MMMM  MMMM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NN      NN  HH      HH
MM  MM  MM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NNNN    NN  HH      HH
MM  MM  MM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NNNN    NN  HH      HH
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  AA      AA      TT      AA      AA  NN  NN  NN  HHHHHHHHHH
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  AA      AA      TT      AA      AA  NN  NN  NN  HHHHHHHHHH
MM      MM      TT      HH      HH  HH      HH  AAAAAAAAAA      TT      AAAAAAAAAA  NN    NNNN  HH      HH
MM      MM      TT      HH      HH  HH      HH  AAAAAAAAAA      TT      AAAAAAAAAA  NN    NNNN  HH      HH
MM      MM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NN      NN  HH      HH    ....
MM      MM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NN      NN  HH      HH    ....
MM      MM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NN      NN  HH      HH    ....
MM      MM      TT      HH      HH  HH      HH  AA      AA      TT      AA      AA  NN      NN  HH      HH    ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000          .TITLE  MTH$HATANH      ; H-floating Precision Hyperbolic Arctangent
0000          .IDENT /2-002/          ; File: MTHHATANH.MAR  Edit: SBL2002
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;
0000    ; FACILITY: MATH LIBRARY
0000    ;++
0000    ; ABSTRACT:
0000    ;
0000    ; MTH$HATANH returns the H-floating precision hyperbolic arctangent of the
0000    ; H-floating precision argument.  The call is standard call-by-reference.
0000    ;
0000    ;--
0000    ;
0000    ; VERSION: 2
0000    ;
0000    ; HISTORY:
0000    ; AUTHOR:
0000    ;       Peter D Gilbert, 23-Jul-81: Version 2
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;
```

```
0000         .SBTTL  HISTORY ; Detailed Current Edit History
0000
0000  ; VERSION 1
0000  ;
0000  ; 1-001 - Original from PL/I math library.
0000  ;
0000  ; Edit History for Version 02 of MTH$HATANH
0000  ;
0000  ; 2-000   Original       July 1981
0000  ; 2-001 - Change MOVZBL to CVTBL when accessing MTH$$AB_ALOG_V.  PDG 2-Dec-1981
0000  ; 2-002 - Store reserved operand result after error.  SBL 6-Jan-1982
0000  ;
```

```
                    0000          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                    0000
                    0000    ;
                    0000    ; INCLUDE FILES:        MTHJACKET.MAR
                    0000    ;
                    0000
                    0000    ;
                    0000    ; EXTERNAL SYMBOLS:
                    0000    ;
                    0000          .DSABL  GLOBAL
                    0000          .SHOW   BINARY,CALLS,CONDITIONALS,DEFINITIONS,EXPANSIONS
                    0000          .EXTRN  MTH$K_INVARGMAT
                    0000          .EXTRN  MTH$$SIGNAL
                    0000          .EXTRN  MTH$$AB_ALOG_V
                    0000          .EXTRN  MTH$$AB_H_FHI
                    0000
                    0000    ;
                    0000    ; EQUATED SYMBOLS:
                    0000    ;
                    0000
                    0000    ;
                    0000    ; MACROS:
                    0000    ;
                    0000
                    0000          .MACRO  OPDEF   X, OP, SH
                    0000          .OPDEF  ADDX    ^X00@SH+OP,R'X,M'X
                    0000          .OPDEF  ADDX3   ^X01@SH+OP,R'X,R'X,W'X
                    0000          .OPDEF  SUBX    ^X02@SH+OP,R'X,M'X
                    0000          .OPDEF  SUBX3   ^X03@SH+OP,R'X,R'X,W'X
                    0000          .OPDEF  MULX    ^X04@SH+OP,R'X,M'X
                    0000          .OPDEF  MULX3   ^X05@SH+OP,R'X,R'X,W'X
                    0000          .OPDEF  DIVX    ^X06@SH+OP,R'X,M'X
                    0000          .OPDEF  DIVX3   ^X07@SH+OP,R'X,R'X,W'X
                    0000          .OPDEF  CVTWX   ^X0D@SH+OP,RW,W'X
                    0000          .OPDEF  POLYX   ^X15@SH+OP,R'X,RW,AB
                    0000          .OPDEF  MOVX    ^X7DFD,RH,WH              ; MOVO
                    0000          .OPDEF  MOVAX   ^X7EFD,AH,WL              ; MOVAO
                    0000          .ENDM
                    0000
                    0000          OPDEF   H, <^X60FD>, 8
                    0000          .OPDEF  ADDX    ^X00@8+^X60FD,RH,MH
                    0000          .OPDEF  ADDX3   ^X01@8+^X60FD,RH,RH,WH
                    0000          .OPDEF  SUBX    ^X02@8+^X60FD,RH,MH
                    0000          .OPDEF  SUBX3   ^X03@8+^X60FD,RH,RH,WH
                    0000          .OPDEF  MULX    ^X04@8+^X60FD,RH,MH
                    0000          .OPDEF  MULX3   ^X05@8+^X60FD,RH,RH,WH
                    0000          .OPDEF  DIVX    ^X06@8+^X60FD,RH,MH
                    0000          .OPDEF  DIVX3   ^X07@8+^X60FD,RH,RH,WH
                    0000          .OPDEF  CVTWX   ^X0D@8+^X60FD,RW,WH
                    0000          .OPDEF  POLYX   ^X15@8+^X60FD,RH,RW,AB
                    0000          .OPDEF  MOVX    ^X7DFD,RH,WH              ; MOVO
                    0000          .OPDEF  MOVAX   ^X7EFD,AH,WL              ; MOVAO
                    0000
                    0000
          00000007  0000          F_EXP = 7       ; Bit offset to exponent
          00000000  0000          X_EXP = 0       ; Bit offset to exponent
                    0000
```

```
    0000  ;
    0000  ; PSECT DECLARATIONS:
    0000  ;
00000000          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
    0000                                          ; program section for math routines
    0000  ;
    0000  ; OWN STORAGE:  none
    0000  ;
```

```
                                      0000  ; CONSTANTS:
                                      0000  ;
                                      0000
                             000007FC 0000          ACMASK = ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
                                      0000                                          ; register entry mask and integer
                                      0000                                          ; overflow enable
                                      0000
00006730 93C7F357 A39E2FEF 62E44000   0000  LN2_HI: .OCTA   ^X0000673093C7F357A39E2FEF62E44000      ; Hi 98 bits of ln2
069E16C5 4C5B9339 79A157A0 F97B3F9A   0010  LN2_LO: .OCTA   ^X069E16C54C5B933979A157A0F97B3F9A      ; Low bits of ln2
                                      0020
                                      0020
                                      0020  LOGTAB1:         ; Constants for q(z).  Generated using eq. 6.3.10 of Hart et.
                                      0020                   ; al. (sin(2a) = 1/32)
                                      0020
5F95F1B2 A5BAC3D8 F5260A61 9B9BBFFC   0020          .OCTA ^X5F95F1B2A5BAC3D8F5260A619B9BBFFC ;C19 = -.50244827536208487853089580
92DBE7A6 76579059 6C52CC82 B1293FFC   0030          .OCTA ^X92DBE7A6765790596C52CC82B1293FFC ;C18 =  .52876376564549972132965432
7A54AC14 946576FF 14A540A3 C71BBFFC   0040          .OCTA ^X7A54AC14946576FF14A540A3C71BBFFC ;C17 = -.55554987186628930743974171
5E0B06A1 8F25A3D8 4658C0D9 E1E03FFC   0050          .OCTA ^X5E0B06A18F25A3D84658C0D9E1E03FFC ;C16 =  .58822991044688315301145533
AE7E786D A6F3EE3D 371F0033 0000BFFD   0060          .OCTA ^XAE7E786DA6F3EE3D371F00330000BFFD ;C15 = -.62500000745281154707785818
4B4434C7 0327C803 9543113E 11113FFD   0070          .OCTA ^X4B4434C70327C8039543113E11113FFD ;C14 =  .66666667329017444142627970
298C1180 F148E440 879C4924 2492BFFD   0080          .OCTA ^X298C1180F148E440879C49242492BFFD ;C13 = -.71428571427964746924020644
E563A213 835C94C6 0AE7B13B 3B133FFD   0090          .OCTA ^XE563A213835C94C60AE7B13B3B133FFD ;C12 =  .76923076922577400516904957
2CB2BF3F 9152C30A 55565555 5555BFFD   00A0          .OCTA ^X2CB2BF3F9152C30A555655555555BFFD ;C11 = -.83333333333333650534413512
0A06F262 E8796F50 D1751745 745D3FFD   00B0          .OCTA ^X0A06F262E8796F50D1751745745D3FFD ;C10 =  .90909090909091146943301574
62840C66 2AF3997A 99999999 9999BFFD   00C0          .OCTA ^X62840C662AF3997A999999999999BFFD ;C9  = -.99999999999999999893503466
FE968C1F 7E77C707 1C7171C7 C71C3FFD   00D0          .OCTA ^XFE968C1F7E77C7071C7171C7C71C3FFD ;C8  =  .11111111111111111104012828
ED6B8E90 00D70000 00000000 0000BFFE   00E0          .OCTA ^XED6B8E9000D70000000000000000BFFE ;C7  = -.12500000C000000000002228
47B3B871 499F2492 92494924 24923FFE   00F0          .OCTA ^X47B3B871499F24929249492424923FFE ;C6  =  .14285714285714285714286987
BE5E4E98 55555555 55555555 5555BFFE   0100          .OCTA ^XBE5E4E98555555555555555555555BFFE ;C5  = -.16666666666666666666666666
CEED967D 99999999 99999999 99993FFE   0110          .OCTA ^XCEED967D999999999999999999993FFE ;C4  =  .19999999999999999999999999
0DD20000 00000000 00000000 0000BFFF   0120          .OCTA ^X0DD20000000000000000000000000BFFF ;C3  = -.25000000000000000000000000
59F05555 55555555 55555555 55553FFF   0130          .OCTA ^X59F05555555555555555555555553FFF ;C2  =  .33333333333333333333333333
00000000 00000000 00000000 0000C000   0140          .OCTA ^X0000000000000000000000000000C000 ;C1  = -.50000000000000000000000000
00000000 00000000 00000000 00000000   0150          .OCTA ^X00000000000000000000000000000000 ;C0  =  .00000000000000000000000000
                                      0160
                             00000013 0160  LOGLEN1 = .-LOGTAB1/16 - 1       ; no. of floating point entries
                                      0160
                                      0*60
                                      0160  LOGTAB2:         ; Constants for p(z*z). Generated using eq. 6.3.11 of Hart et.
                                      0160                   ; al. (sin(2a) = (b - 1)/(b + 1) where b = 2**(1/7))
                                      0160
8441440A 9DA42272 7A67F044 8B243FFD   0160          .OCTA ^X8441440A9DA422727A67F0448B243FFD ;C10= .9647077421655028896227926448
0019B5C5 3BD4BEC7 61F97E57 AF203FFD   0170          .OCTA ^X0019B5C53BD4BEC761F97E57AF203FFD ;C9 = .1052555976112884972789729915
85B2D526 87082827 EEF2E8F7 E1E13FFD   0180          .OCTA ^X85B2D52687082827EEF2E8F7E1E13FFD ;C8 = .1176470852214462141323328874
C286BAD2 232FAD44 1440110F 11113FFE   0190          .OCTA ^XC286BAD2232FAD441440110F11113FFE ;C7 = .1333333332754878760888250485
90B321D3 AF744625 146BB13B 3B133FFE   01A0          .OCTA ^X90B321D3AF744625146BB13B3B133FFE ;C6 = .1538461538462364659507622054
F8411CEE 61EB3082 D1741745 745D3FFE   01B0          .OCTA ^XF8411CEE61EB3082D1741745745D3FFE ;C5 = .1818181818181817408450657725
73AA312A 4DE1C723 1C7171C7 C71C3FFE   01C0          .OCTA ^X73AA312A4DE1C7231C7171C7C71C3FFE ;C4 = .222222222222222222687058519
5FBA09F6 48D22492 92494924 24923FFF   01D0          .OCTA ^X5FBA09F648D224929249492424923FFF ;C3 = .2857142857142857142856972183
DDA89DE8 99999999 99999999 99993FFF   01E0          .OCTA ^XDDA89DE8999999999999999999993FFF ;C2 = .4000000000000000000000000034
480A5555 55555555 55555555 55554000   01F0          .OCTA ^X480A5555555555555555555555554000 ;C1 = .6666666666666666666666666666
00000000 00000000 00000000 00004002   0200          .OCTA ^X00000000000000000000000000004002 ;C0 = .2000000000000000000000000000
                             0000000A 0210  LOGLEN2 = .-LOGTAB2/16 - 1
```

```
          0210          .SBTTL  MTH$HATANH      - H-floating Precision Hyperbolic Arctangent
          0210
          0210    ;++
          0210    ; FUNCTIONAL DESCRIPTION:
          0210    ;
          0210    ; HATANH  - H-floating precision floating point function
          0210    ;
          0210    ; HATANH(X) is computed using the following approximation technique:
          0210    ;
          0210    ;       If !X! >= 1.0, error.  Otherwise
          0210    ;
          0210    ;       Let (1+X)/(1-X) = f * (2**n), where 1/2 <= f < 1
          0210    ;
          0210    ;       If n is greater than or equal to 1 then
          0210    ;           set N = n - 1 and F1 = 2*f.
          0210    ;       Else
          0210    ;           set N = n and F = f.
          0210    ;
          0210    ;       If !F - 1! < 2**-5 then
          0210    ;           2*atanh(X) =  N*ln(2) + W + W*P(W),
          0210    ;               where W = ((1+F)/(1-F))*2**N - 1,
          0210    ;               and P is a polynomial of degree F=5,D=9.
          0210    ;       Else
          0210    ;           Obtain FHI (roughly equal to F) from table lookup.
          0210    ;           2*atanh(X) = ln((1+X)/(1-X)) = N*ln(2) + ln(FHI) + Z*Q(Z*Z),
          0210    ;               where Q is a polynomial of degree F=2,D=5,
          0210    ;               where Z = (F - FHI)/(F + FHI)
          0210    ;               where F = (2**-N)*(1+X)/(1-X)
          0210    ;           Z is computed by:
          0210    ;               Z = (X-D)/(1-X*D)
          0210    ;               where Y = FHI*2**N
          0210    ;               where D = (Y-1)/(Y+1)
          0210    ;           Note that Z may be computed in a variety of ways:
          0210    ;               Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
          0210    ;               Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
          0210    ;               Z = [1 - Y + X + X*Y]/[1 + Y + X - X*Y]
          0210    ;               Z = [(1-Y) + X*(1+Y)]/[(1+Y) + X*(1-Y)]
          0210    ;
          0210    ;       NOTE:   The quantities ln(A=FHI) and ln2 are used in the above
          0210    ;               equations in two parts - a high part (containing the
          0210    ;               high order bits) and a low part (containing the low
          0210    ;               order bits.  In the code the high and low parts of the
          0210    ;               constants are indicated by a _HI and _LO suffix respec-
          0210    ;               tively.  The values were chosen such that N*LN2_HI +
          0210    ;               LN_FHI_HI is exactly representable.
          0210    ;
          0210    ; CALLING SEQUENCE:
          0210    ;
          0210    ;       MTH$HATANH(hatanh.wh.r, x.rh.r)
          0210    ;
          0210    ; INPUT PARAMETERS:
          0210
00000004  0210          Y = 4                           ; Address to store result
00000008  0210          X = 8                           ; Contents of x is the argument
          0210
          0210    ; IMPLICIT INPUTS:      none
          0210    ;
```

```
                              0210  ; OUTPUT PARAMETERS:
                              0210  ;
                              0210  ;       VALUE:  H-floating precision hyperbolic arctangent of the argument
                              0210  ;
                              0210  ; IMPLICIT OUTPUTS:     none
                              0210  ;
                              0210  ; COMPLETION CODES:     none
                              0210  ;
                              0210  ; SIDE EFFECTS:
                              0210  ;
                              0210  ; Signals: MTH$K_INVARGMAT if !X! >= 1.0 with reserved operand in R0 (copied to
                              0210  ; the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                              0210  ; Associated message is: "Floating overflow in math library". Result is
                              0210  ; reserved operand -0.0 unless a user supplied (or any) error handler changes
                              0210  ; CHF$L_MCH_R0/R1.
                              0210  ;
                              0210  ; NOTE: This procedure disables floating point underflow and integer
                              0210  ; overflow, causes no floating overflow or other arithmetic traps, and
                              0210  ; preserves enables across the call.
                              0210  ;
                              0210  ; Note: This routine is written to avoid causing any integer overflows,
                              0210  ; floating overflows, or floating underflows or divide by 0 conditions,
                              0210  ; whether enabled or not.
                              0210  ;
                              0210  ;---
                              0210
               0183   31      0210 ERR:     BRW     ERROR
                              0213
                      07FC    0213          .ENTRY  MTH$HATANH, ACMASK       ; standard call-by-reference entry
                              0215                                           ; disable DV (and FU), enable IV
        50    08 BC 7DFD      0215          MOVX    @X(AP), R0               ; R0 = arg
                              021A
           56    50 7DFD      021A          MOVX    R0, R6
     7E    08    56 63FD      021E          SUBX3   R6, S^#1.0, -(SP)        ; (SP) = 1-X
                 EB   15      0223          BLEQ    ERR                      ; ATANH(X) is not defined for X>=1
           56    08 60FD      0225          ADDX       S^#1.0, R6
                 E5   15      0229          BLEQ    ERR                      ; ATANH(X) is not defined for X<=-1
           54    8E F6FD      022B          CVTHF   (SP)+, R4
           56    56 F6FD      022F          CVTHF   R6, R6
           56    54   46      0233          DIVF2   R4, R6                   ; R6 = approximation to (1+X)/(1-X)
   5A   00000000'GF   9E      0236          MOVAB   G^MTH$$AB_ALOG_V, R10
           5A    6A   C0      023D          ADDL2   (R10), R10               ; R10 = address of ALOG table
   55   56  007F 8F   AB      0240          BICW3   #1@F_EXP-1, R6, R5       ; R5 = Biased exponent
        55  4000 8F   A2      0246          SUBW    #^X4000, R5              ; R5 = Unbiased exponent
                 7F   15      024B          BLEQ    NEG_EXP                  ; Branch to processing for n=<0
                              024D
        55  0080 8F   A2      024D          SUBW    #1@F_EXP, R5             ; Exponent is positive, R5 = N = n - 1
           56    55   A2      0252          SUBW    R5, R6                   ; R6 = F = 2f
           56    56   9A      0255          MOVZBL  R6, R6                   ; R6 = index into ALOG table
              00000007        0258          .IF     NE, F_EXP-X_EXP
        55  0080 8F   A6      0258          DIVW2   #1@<F_EXP-X_EXP>, R5     ; Shift R5 to scale X-floating
                              025D          .ENDC
           7E    55 6DFD      025D          CVTWX   R5, -(SP)                ; Push N onto the stack
        5A    6A46    98      0261          CVTBL   (R10)[R6], R10           ; R10 = offset into FHI tables
                 78   19      0265          BLSS    LN_1_PLUS_W              ; Branch to handle F close to 1
  5A   00000000'GF4A 7EFD     0267          MOVAX   G^MTH$$AB_H_FHI[R10], R10      ; R10 = Address of FHI
           56    8A 7DFD      0270          MOVX    (R10)+, R6               ; R6 = FHI
```

```
                          0274   ;
                          0274   ; Compute Z = (F - FHI)/(F + FHI)
                          0274   ;                 Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
                          0274   ;                 Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
                          0274   ;       where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
                          0274   ;
         56     55   A0   0274          ADDW    R5, R6                  ; R6 = FHI * 2**N = SFHI
    7E   08   56 63FD     0277          SUBX3   R6, S^#1.0, -(SP)       ; (SP) = 1 - SFHI
         56   08 60FD     027C          ADDX    S^#1.0, R6              ; R6 = 1 + SFHI
         6E   56 66FD     0280          DIVX    R6, (SP)                ; (SP) = (1-SFHI)/(1+SFHI) = D
    56   6E   50 61FD     0284          ADDX3   R0, (SP), R6            ; R6 = D + X
         6E   50 64FD     0289          MULX    R0, (SP)                ; (SP) = D * X
         6E   08 60FD     028D          ADDX    S^#1.0, (SP)            ; (SP) = 1 + D*X
         56   8E 66FD     0291          DIVX    (SP)+, R6               ; R6 = (D+X)/(1+D*X) = Z
                          0295   ;
                          0295   ; Compute Z**2, P(Z**2) and Z*P(Z**2)
                          0295   ;
    50   56   56 65FD     0295          MULX3   R6, R6, R0              ; R0 = Z**2
FEBF CF  0A   50 75FD     029A          POLYX   R0, #LOGLEN2, LOGTAB2   ; R0 = P(Z**2)
         50   56 64FD     02A1          MULX    R6, R0                  ; R0 = Z*P(Z**2)
                          02A5   ;
                          02A5   ; Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z*Z)
                          02A5   ;
 56  FD65 CF  6E 65FD     02A5          MULX3   (SP), LN2_LO, R6        ; R6 = N*LN2_LO
         56   8A 60FD     02AC          ADDX    (R10)+, R6              ; R6 = N*LN2_LO + LN_FHI_LO
         50   56 60FD     02B0          ADDX    R6, R0                  ; R0 = B
                          02B4   ;
                          02B4   ; Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)
                          02B4   ;
 56  FD46 CF  8E 65FD     02B4          MULX3   (SP)+, LN2_HI, R6       ; R6 = N*LN2_HI
         56   6A 60FD     02BB          ADDX    (R10), R6               ; R6 = A = N*LN2_HI + LN_FHI_HI
         50   56 60FD     02BF          ADDX    R6, R0                  ; R0 = A + B = ALOG(X)
         50   01   A2     02C3          SUBW2   #1@X_EXP, R0            ; Divide by 2
    04 BC     50 7DFD     02C6          MOVX    R0, @Y(AP)              ; Store result
                   04     02CB          RET
                          02CC
                          02CC   NEG_EXP:
         56   55   A2     02CC          SUBW    R5, R6                  ; R6 = F = 2f
         56   56   9A     02CF          MOVZBL  R6, R6                  ; R6 = index into ALOG table
              00000007    02D2          .IF     NE, F_EXP-X_EXP
    55   0080 8F   A6     02D2          DIVW2   #1@<F_EXP-X_EXP>, R5    ; Shift R5 to scale X-floating
                          02D7          .ENDC
         7E   55 6DFD     02D7          CVTWX   R5, -(SP)               ; Push N onto the stack
     5A    6A46    98     02DB          CVTBL   (R10)[R6], R10          ; R10 = offset into FHI tables
                          02DF   LN_1_PLUS_W:
              65   19     02DF          BLSS    LN_1_PLUS               ; Branch to handle F close to 1
5A    00000000'GF4A 7EFD  02E1          MOVAX   G^MTH$$AB_H_FHI[R10], R10       ; R10 = Address of FHI
         56   6A 7DFD     02EA          MOVX    (R10), R6               ; R6 = FHI
                          02EE   ;
                          02EE   ; Compute Z = (F - FHI)/(F + FHI)
                          02EE   ;                 Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
                          02EE   ;                 Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
                          02EE   ;       where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
                          02EE   ;
         56     55   A0   02EE          ADDW    R5, R6                  ; R6 = FHI * 2**N = SFHI
    7E   08   56 63FD     02F1          SUBX3   R6, S^#1.0, -(SP)       ; (SP) = 1 - SFHI
         56   08 60FD     02F6          ADDX    S^#1.0, R6              ; R6 = 1 + SFHI
```

```
             6E   56 66FD  02FA          DIVX    R6, (SP)                ; (SP) = (1-SFHI)/(1+SFHI) = D
        56   6E   50 61FD  02FE          ADDX3   R0, (SP), R6            ; R6 = D + X
             6E   50 64FD  0303          MULX    R0, (SP)                ; (SP) = D * X
             6E   08 60FD  0307          ADDX    S^#1.0, (SP)            ; (SP) = 1 + D*X
             56   8E 66FD  030B          DIVX    (SP)+, R6               ; R6 = (D+X)/(1+D*X) = Z
                           030F  ;
                           030F  ; Compute Z**2, P(Z**2) and Z*P(Z**2)
                           030F  ;
        50   56   56 65FD  030F          MULX3   R6, R6, R0              ; R0 = Z**2
   FE45 CF   0A   50 75FD  0314          POLYX   R0, #LOGLEN2, LOGTAB2   ; R0 = P(Z**2)
             50   56 64FD  031B          MULX    R6, R0                  ; R0 = Z*P(Z**2)
                           031F  ;
                           031F  ; Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z*Z)
                           031F  ;
   56   FCEB CF   6E 65FD  031F          MULX3   (SP), LN2_LO, R6        ; R6 = N*LN2_LO
             56   7A 60FD  0326          ADDX    -(R10), R6              ; R6 = N*LN2_LO + LN_FHI_LO
             50   56 60FD  032A          ADDX    R6, R0                  ; R0 = B
                           032E  ;
                           032E  ; Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)
                           032E  ;
   56   FCCC CF   8E 65FD  032E          MULX3   (SP)+, LN2_HI, R6       ; R6 = N*LN2_HI
             56   7A 62FD  0335          SUBX    -(R10), R6              ; R6 = A = N*LN2_HI + LN_FHI_HI
             50   56 60FD  0339          ADDX    R6, R0                  ; R0 = A + B = ALOG(X)
             50   01   A2  033D          SUBW2   #1@X_EXP, R0            ; Divide by 2
          04 BC   50 7DFD  0340          MOVX    R0, @Y(AP)              ; Store result
                       04  0345          RET
                           0346
                           0346  ;
                           0346  ; Special logic for F close to 1
                           0346  ;
                           0346
                           0346  LN_1_PLUS:
        56   08   50 63FD  0346          SUBX3   R0, S^#1.0, R6          ; R6 = 1-X
             55        B5  034B          TSTW    R5                      ; Determine which way to calculate W
             12        13  034D          BEQL    10$
        56   10   56 67FD  034F          DIVX3   R6, S^#2.0, R6          ; R6 = 2/(1-X)
             56   08 62FD  0354          SUBX    S^#1.0, R6              ; R6 = (1+X)/(1-X)
             56   55   A2  0358          SUBW    R5, R6                  ; Scale R6
             56   08 62FD  035B          SUBX    S^#1.0, R6              ; R6 = W
             08        11  035F          BRB     20$
        56   50   56 67FD  0361  10$:    DIVX3   R6, R0, R6              ; R6 = X / (1-X)
             56   01   A0  0366          ADDW    #1@X_EXP, R6            ; R6 = W = 2*X/(1-X) = (1+X)/(1-X) - 1
   FCB0 CF   13   56 75FD  0369  20$:    POLYX   R6, #LOGLEN1, LOGTAB1   ; R0 = Q(W)
             50   56 64FD  0370          MULX    R6, R0                  ; R0 = W*Q(W)
   7E   FC96 CF   6E 65FD  0374          MULX3   (SP), LN2_LO, -(SP)     ; (SP) = N*LN2_LO
             50   8E 60FD  037B          ADDX    (SP)+, R0               ; R0 = N*LN2_LO + W*Q(W)
             50   56 60FD  037F          ADDX    R6, R0                  ; R0 = N*LN2_LO + W*Q(W) + W
        6E   FC78 CF 64FD  0383          MULX    LN2_HI, (SP)            ; (SP) = N*LN2_HI
             50   8E 60FD  0389          ADDX    (SP)+, R0               ; R0 = ALOG(X)
             50   01   A2  038D          SUBW2   #1@X_EXP, R0            ; Divide by 2
          04 BC   50 7DFD  0390          MOVX    R0, @Y(AP)              ; Store result
                       04  0395          RET
                           0396  ;
                           0396  ; X <= 0.0, signal error
                           0396  ;
         7E   00'8F    9A  0396  ERROR:  MOVZBL  #MTH$K_INVARGMAT, -(SP) ; condition value
        50   01   0F   79  039A          ASHQ    #15, #1, R0             ; R0 = result = reserved operand -0.0
```

```
                    52    7C  039E   395         CLRQ    R2
                              03A0   396                                     ; goes to signal mechanism vector
                              03A0   397                                     ; (CHF$L_MCH_R0/R1) so error handler
                              03A0   398                                     ; can modify the result.
     00000000'GF    01    FB  03A0   399         CALLS   #1, G^MTH$$SIGNAL   ; signal error and use real user's PC
                              03A7   400                                     ; independent of CALL vs JSB
           04 BC    50  7DFD  03A7   401         MOVO    R0, @Y(AP)          ; Store result
                          04  03AC   402         RET                         ; return
                              03AD   403
                              03AD   404         .END
```

MTH
VAX
Mac

_$2
0 G
The
MAC

```
ACMASK           = 000007FC
ERR                00000210 R      01
ERROR              00000396 R      01
F_EXP            = 00000007
LN2_HI             00000000 R      01
LN2_LO             00000010 R      01
LN_1_PLUS          00000346 R      01
LN_1_PLUS_W        000002DF R      01
LOGLEN1          = 00000013
LOGLEN2          = 0000000A
LOGTAB1            00000020 R      01
LOGTAB2            00000160 R      01
MTH$$AB_ALOG_V     ********    X   00
MTH$$AB_H_FHI      ********    X   00
MTH$$SIGNAL        ********    X   00
MTH$HATANH         00000213 RG     01
MTH$K_INVARGMAT    ********    X   00
NEG_EXP            000002CC R      01
X                = 00000008
X_EXP            = 00000000
Y                = 00000004
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
_MTH$CODE                       000003AD  (  941.)  01 (  1.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC LONG
```

```
                                     +------------------------+
                                     ! Performance indicators !
                                     +------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                    34    00:00:00.11    00:00:01.13
Command processing               134    00:00:00.66    00:00:04.89
Pass 1                           101    00:00:01.55    00:00:05.27
Symbol table sort                  0    00:00:00.01    00:00:00.01
Pass 2                            90    00:00:00.99    00:00:03.50
Symbol table output                3    00:00:00.02    00:00:00.06
Psect synopsis output              2    00:00:00.02    00:00:00.02
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals             366    00:00:03.38    00:00:14.88

The working set limit was 900 pages.
7868 bytes (16 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 21 non-local and 2 local symbols.
464 source lines were read in Pass 1, producing 13 object records in Pass 2.
3 pages of virtual memory were used to define 2 macros.
```

```
                                        +--------------------------+
                                        ! Macro library statistics !
                                        +--------------------------+

Macro library name                      Macros defined
------------------                      --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:MTHHATANH/OBJ=OBJ$:MTHHATANH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MS
```

MTHGSIGN
LIS

MTHHCOSH
LIS

MTHHACOS
LIS

MTHGPROD
LIS

MTHGSINCO
LIS

MTHHASIN
LIS

MTHHATAN
LIS

MTHGTAN
LIS

MTHGTANH
LIS

MTHGSQRT
LIS

MTHHEXP
LIS

MTHGSINH
LIS

MTHHATANH
LIS