


```

LL                      IIIII
LL                      IIIII
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LLLLLLLLLLLL           IIIII
LLLLLLLLLLLL           IIIII
SSSSSSSS              SSSSSSS
SSSSSSSS              SSSSSSS
SS                      SS
SS                      SS
SS                      SS
SS                      SS
SSSSSS                SSSSSS
SSSSSS                SSSSSS
SS                      SS
SS                      SS
SS                      SS
SS                      SS
SSSSSSSS              SSSSSSS
SSSSSSSS              SSSSSSS

```


MTH\$DATANH
Table of contents

H 1
; Double Precision Hyperbolic Arctangent 16-SEP-1984 01:15:06 VAX/VMS Macro V04-00

Page 0

(2) 47
(3) 61
(5) 158

HISTORY ; Detailed Current Edit History
DECLARATIONS ; Declarative Part of Module
MTH\$DATANH - Double Precision Hyperbolic Arctangent

```

0000 1      .TITLE  MTH$DATANH      ; Double Precision Hyperbolic Arctangent
0000 2      .IDENT /2-003/          ; File: MTH$DATANH.MAR Edit: PDG2003
0000 3
0000 4      :*****
0000 5      :
0000 6      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      :*  ALL RIGHTS RESERVED.
0000 9      :
0000 10     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     :*  TRANSFERRED.
0000 16     :
0000 17     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     :*  CORPORATION.
0000 20     :
0000 21     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     :
0000 24     :*****
0000 25     :
0000 26     :
0000 27     :
0000 28     : FACILITY: MATH LIBRARY
0000 29     :++
0000 30     : ABSTRACT:
0000 31     :
0000 32     : MTH$DATANH returns the double precision hyperbolic arctangent of the
0000 33     : double precision argument. The call is standard call-by-reference.
0000 34     :
0000 35     :--
0000 36     :
0000 37     : VERSION: 2
0000 38     :
0000 39     : HISTORY:
0000 40     : AUTHOR:
0000 41     :      Peter D Gilbert, 23-Jul-81: Version 2
0000 42     :
0000 43     : MODIFIED BY:
0000 44     :
0000 45     :

```


MTH\$DATANH
2-003

J 1
; Double Precision Hyperbolic Arctangent 16-SEP-1984 01:15:06 VAX/VMS Macro V04-00 Page 2
HISTORY ; Detailed Current Edit History 6-SEP-1984 11:21:51 [MTHRTL.SRC]MTH\$DATANH.MAR;1 (2)

```
0000 47      .SBTTL HISTORY ; Detailed Current Edit History
0000 48
0000 49 : VERSION 1
0000 50 :
0000 51 : 1-001 - Original from PL/I math library
0000 52 :
0000 53 : Edit History for Version 02 of MTH$DATANH
0000 54 :
0000 55 : 2-000 Original July 1981
0000 56 : 2-001 - Change MOVZBL to CVTBL when accessing MTH$$AB ALOG V. PDG 2-Dec-1981
0000 57 : 2-002 - Change RSB to RET after error exit. PDG 6-Jan-1981
0000 58 : 2-003 - Repair problem with POLY instruction. PDG 19-Mar-1982
0000 59 :
```

```
0000 61 .SBTTL DECLARATIONS ; Declarative Part of Module
0000 62
0000 63 :
0000 64 : INCLUDE FILES: MTHJACKET.MAR
0000 65 :
0000 66 :
0000 67 :
0000 68 : EXTERNAL SYMBOLS:
0000 69 :
0000 70 .DSABL GLOBAL
0000 71 .SHOW BINARY,CALLS,CONDITIONALS,DEFINITIONS,EXPANSIONS
0000 72 .EXTRN MTH$K INVARGMAT
0000 73 .EXTRN MTH$$SIGNAL
0000 74 .EXTRN MTH$$AB ALOG_V
0000 75 .EXTRN MTH$$AB_D_FHT
0000 76
0000 77 :
0000 78 : EQUATED SYMBOLS:
0000 79 :
0000 80 :
0000 81 :
0000 82 : MACROS:
0000 83 :
0000 84 :
0000 85 .MACRO OPDEF X, OP, SH
0000 86 .OPDEF ADDX ^X00aSH+OP,R'X,M'X
0000 87 .OPDEF ADDX3 ^X01aSH+OP,R'X,R'X,W'X
0000 88 .OPDEF SUBX ^X02aSH+OP,R'X,M'X
0000 89 .OPDEF SUBX3 ^X03aSH+OP,R'X,R'X,W'X
0000 90 .OPDEF MULX ^X04aSH+OP,R'X,M'X
0000 91 .OPDEF MULX3 ^X05aSH+OP,R'X,R'X,W'X
0000 92 .OPDEF DIVX ^X06aSH+OP,R'X,M'X
0000 93 .OPDEF DIVX3 ^X07aSH+OP,R'X,R'X,W'X
0000 94 .OPDEF CVTWX ^X08aSH+OP,RW,W'X
0000 95 .OPDEF POLYX ^X15aSH+OP,R'X,RW,AB
0000 96 .OPDEF MOVX ^X007D,RQ,WQ ; MOVQ
0000 97 .OPDEF MOVAX ^X007E,AQ,WL ; MOVAQ
0000 98 .ENDM
0000 99
0000 100 OPDEF D, <^X0060>, 0
0000 .OPDEF ADDX ^X00a0+^X0060,RD,MD
0000 .OPDEF ADDX3 ^X01a0+^X0060,RD,RD,WD
0000 .OPDEF SUBX ^X02a0+^X0060,RD,MD
0000 .OPDEF SUBX3 ^X03a0+^X0060,RD,RD,WD
0000 .OPDEF MULX ^X04a0+^X0060,RD,MD
0000 .OPDEF MULX3 ^X05a0+^X0060,RD,RD,WD
0000 .OPDEF DIVX ^X06a0+^X0060,RD,MD
0000 .OPDEF DIVX3 ^X07a0+^X0060,RD,RD,WD
0000 .OPDEF CVTWX ^X08a0+^X0060,RW,WD
0000 .OPDEF POLYX ^X15a0+^X0060,RD,RW,AB
0000 .OPDEF MOVX ^X007D,RQ,WQ ; MOVQ
0000 .OPDEF MOVAX ^X007E,AQ,WL ; MOVAQ
0000
00000007 0000 101
00000007 0000 102 F_EXP = 7 ; Bit offset to exponent
00000007 0000 103 X_EXP = 7 ; Bit offset to exponent
0000 104
```


MTHSDATANH
2-003

L 1

: Double Precision Hyperbolic Arctangent 16-SEP-1984 01:15:06 VAX/VMS Macro V04-00
DECLARATIONS ; Declarative Part of Modul 6-SEP-1984 11:21:51 [MTHRTL.SRC]MTHDATANH.MAR;1 Page 4
(3)

```
0000 105 ;  
0000 106 ; PSECT DECLARATIONS:  
0000 107 ;  
00000000 108 .PSECT _MTH$CODE PIC,SHR,LONG,EXE,NOWRT  
0000 109 ; program section for math routines  
0000 110 ;  
0000 111 ; OWN STORAGE: none  
0000 112 ;
```

```
0000 114 ; CONSTANTS:
0000 115 ;
0000 116 ;
000001FC 0000 117 ACMASK = ^M<R2,R3,R4,R5,R6,R7,R8>
0000 118 ; register entry mask and integer
0000 119 ; overflow enable
0000 120 ;
0000 121 ;+ The "128" in the constants is used to shift the unbiased exponent
0000 122 ; right 7 places so that the rightmost bit is at bit 0.
0000 123 ; -
0000 124 ;
7200F7B1 72173CB1 0000 125 LN2_HI: .QUAD ^X7200F7B172173CB1 ; (Hi 48 bits of ln2)/128
8ECEAF27 75E62B81 0008 126 LN2_LO: .QUAD ^X8ECEAF2775E62B81 ; (Low bits of ln2)/128
0010 127
0010 128 LOGTAB1: ; Constants for q(z). Generated using
0010 129 ; eq. 6.3.10 of Hart et. al. (sin(2a)
0010 130 ; = 1/32)
1E51DE52 4D00BECD 0010 131 .QUAD ^X1E51DE524D00BECD ; C9 = -0.10024452856511271
44C1F2BD 0E683EE4 0018 132 .QUAD ^X44C1F2BD0E683EE4 ; C8 = 0.11135560980588577
9B9BEC78 FFDBBEFF 0020 133 .QUAD ^X9B9BEC78FFDBBEFF ; C7 = -0.12499973121073342
4BB28A46 49143F12 0028 134 .QUAD ^X4BB28A4649143F12 ; C6 = 0.14285690397225510
6C93AD01 AAAABF2A 0030 135 .QUAD ^X6C93AD01AAAABF2A ; C5 = -0.166666666680280835
C92CCE8D CCCC3F4C 0038 136 .QUAD ^XC92CCE8DCCCC3F4C ; C4 = 0.20000000010208756
DCE9FFFF FFFFBF7F 0040 137 .QUAD ^XDCE9FFFFFFFFFFBF7F ; C3 = -0.24999999999996883
AOA5AAAA AAAA3FAA 0048 138 .QUAD ^XAOA5AAAAAAA3FAA ; C2 = 0.33333333333331553
00000000 0000C000 0050 139 .QUAD ^X000000000000C000 ; C1 = -0.50000000000000000
0058 140 ;
0058 141 ; Remove this constant, and do another multiply in-line.
0058 142 ;
0058 143 ;:::~::~: .QUAD ^X0000000000000000 ; C0 = 0.00000000000000000
00000008 0058 144 LOGLEN1 = .-LOGTAB1/8 - 1 ; no. of floating point entries
0058 145 ;
0058 146 LOGTAB2: ; Constants for p(z*z). Generated using
0058 147 ; eq. 6.3.11 of Hart et. al. (sin(2a) =
0058 148 ; (b - 1)/(b + 1) where b = 2**(1/7))
0058 149 ;
88B900ED 70B23F3B 0058 150 .QUAD ^X88B900ED70B23F3B ; C5 = 0.183047086054451500
5D2C3E00 8D3C3F63 0060 151 .QUAD ^X5D2C3E008D3C3F63 ; C4 = 0.222218457493082486
185BC1CE 49243F92 0068 152 .QUAD ^X185BC1CE49243F92 ; C3 = 0.285714291246265538
1CEFCCC4 CCCC3FCC 0070 153 .QUAD ^X1CEFCCC4CCCC3FCC ; C2 = 0.3999999999996049620
AB02AAAA AAAA402A 0078 154 .QUAD ^XAB02AAAAAAA402A ; C1 = 0.666666666666667879
00000000 00004100 0080 155 .QUAD ^X0000000000004100 ; C0 = 2.00000000000000000
00000005 0088 156 LOGLEN2 = .-LOGTAB2/8 - 1
```



```
0088 158 .SBTTL MTH$DATANH - Double Precision Hyperbolic Arctangent
0088 159
0088 160 ;++
0088 161 ; FUNCTIONAL DESCRIPTION:
0088 162 ;
0088 163 ; DATANH - Double precision floating point function
0088 164 ;
0088 165 ; DATANH(X) is computed using the following approximation technique:
0088 166 ;
0088 167 ; If |X| >= 1.0, error. Otherwise
0088 168 ;
0088 169 ; Let (1+X)/(1-X) = f * (2**n), where 1/2 <= f < 1
0088 170 ;
0088 171 ; If n is greater than or equal to 1 then
0088 172 ; set N = n - 1 and F1 = 2*f.
0088 173 ; Else
0088 174 ; set N = n and F = f.
0088 175 ;
0088 176 ; If |F - 1| < 2**-5 then
0088 177 ; 2*atanh(X) = N*ln(2) + W + W*P(W),
0088 178 ; where W = ((1+F)/(1-F))*2**N - 1,
0088 179 ; and P is a polynomial of degree F=5,D=9.
0088 180 ; Else
0088 181 ; Obtain FHI (roughly equal to F) from table lookup.
0088 182 ; 2*atanh(X) = ln((1+X)/(1-X)) = N*ln(2) + ln(FHI) + Z*Q(Z*Z),
0088 183 ; where Q is a polynomial of degree F=2,D=5,
0088 184 ; where Z = (F - FHI)/(F + FHI)
0088 185 ; where F = (2**N)*(1+X)/(1-X)
0088 186 ; Z is computed by:
0088 187 ; Z = (X-D)/(1-X*D)
0088 188 ; where Y = FHI*2**N
0088 189 ; where D = (Y-1)/(Y+1)
0088 190 ; Note that Z may be computed in a variety of ways:
0088 191 ; Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
0088 192 ; Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
0088 193 ; Z = [1 - Y + X + X*Y]/[1 + Y + X - X*Y]
0088 194 ; Z = [(1-Y) + X*(1+Y)]/[(1+Y) + X*(1-Y)]
0088 195 ;
0088 196 ; NOTE: The quantities ln(A=FHI) and ln2 are used in the above
0088 197 ; equations in two parts - a high part (containing the
0088 198 ; high order bits) and a low part (containing the low
0088 199 ; order bits. In the code the high and low parts of the
0088 200 ; constants are indicated by a _HI and _LO suffix respec-
0088 201 ; tively. The values were chosen such that N*LN2_HI +
0088 202 ; LN_FHI_HI is exactly representable.
0088 203 ;
0088 204 ; CALLING SEQUENCE:
0088 205 ;
0088 206 ; datanh.wd.v = MTH$DATANH(x.rd.r)
0088 207 ;
0088 208 ; INPUT PARAMETERS:
0088 209 ;
00000004 0088 210 ; X = 4 ; Contents of x is the argument
0088 211 ;
0088 212 ; IMPLICIT INPUTS: none
0088 213 ;
0088 214 ; OUTPUT PARAMETERS:
```



```
0088 215 :  
0088 216 : VALUE: Double precision hyperbolic arctangent of the argument  
0088 217 :  
0088 218 : IMPLICIT OUTPUTS: none  
0088 219 :  
0088 220 : COMPLETION CODES: none  
0088 221 :  
0088 222 : SIDE EFFECTS:  
0088 223 :  
0088 224 : Signals: MTH$K_INVARGMAT if |X| >= 1.0 with reserved operand in R0 (copied to  
0088 225 : the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).  
0088 226 : Associated message is: "Floating overflow in math library". Result is  
0088 227 : reserved operand -0.0 unless a user supplied (or any) error handler changes  
0088 228 : CHF$MCH_R0/R1.  
0088 229 :  
0088 230 : NOTE: This procedure disables floating point underflow and integer  
0088 231 : overflow, causes no floating overflow or other arithmetic traps, and  
0088 232 : preserves enables across the call.  
0088 233 :  
0088 234 : Note: This routine is written to avoid causing any integer overflows,  
0088 235 : floating overflows, or floating underflows or divide by 0 conditions,  
0088 236 : whether enabled or not.  
0088 237 :  
0088 238 :---  
0088 239 :  
0135 31 0088 240 ERR: BRW ERROR  
0088 241 :  
01FC 0088 242 .ENTRY MTH$DATANH, ACMASK ; standard call-by-reference entry  
0088 243 : disable DV (and FU), enable IV  
50 04 BC 7D 0088 244 MOVX @X(AP), R0 ; R0 = arg  
0091 245 :  
52 08 50 43 0091 246 SUBF3 R0, S^#1.0, R2 ; R2 = 1-X (okay to trunc towards 0)  
0095 247 BLEQ ERR ; ATANH(X) is not defined for X>=1  
56 08 50 41 0097 248 ADDF3 R0, S^#1.0, R6 ; R6 = 1+X (okay to trunc towards 0)  
009B 249 BLEQ ERR ; ATANH(X) is not defined for X<=-1  
56 52 46 009D 250 DIVF2 R2, R6 ; R6 = approximation to (1+X)/(1-X)  
58 00000000'GF 9E 00A0 251 MOVAB G^MTH$AB ALOG_V, R8  
58 68 C0 00A7 252 ADDL2 (R8), R8 ; R8 = address of ALOG table  
55 56 007F 8F AB 00AA 253 BICW3 #10F EXP-1, R6, R5 ; R5 = Biased exponent  
55 4000 8F A2 00B0 254 SUBW #X4000, R5 ; R5 = Unbiased exponent  
64 15 00B5 255 BLEQ NEG_EXP ; Branch to processing for n<0  
00B7 256 :  
55 0080 8F A2 00B7 257 SUBW #10F EXP, R5 ; Exponent is positive, R5 = N = n - 1  
56 55 A2 00BC 258 SUBW R5, R6 ; R6 = F = 2f  
56 56 9A 00BF 259 MOVZBL R6, R6 ; R6 = index into ALOG table  
00000000 00C2 260 .IF NE, F_EXP-X_EXP  
00C2 261 DIVW2 #10<F_EXP-X_EXP>, R5 ; Shift R5 to scale X-floating  
00C2 262 .ENDC  
7E 55 6D 00C2 263 CVTWX R5, -(SP) ; Push N onto the stack  
58 6846 98 00C5 264 CVTBL (R8)[R6], R8 ; R8 = offset into FHI tables  
5D 19 00C9 265 BLSS LN 1 PLUS_W ; Branch to handle F close to 1  
58 00000000'GF48 7E 00CB 266 MOVAX G^MTH$AB_D_FHI[R8], R8 ; R8 = Address of FHI  
56 88 7D 00D3 267 MOVX (R8)+, R6 ; R6 = FHI  
00D6 268 :  
00D6 269 : Compute Z = (F - FHI)/(F + FHI)  
00D6 270 : Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]  
00D6 271 : Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
```



```
00D6 272 ; where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
00D6 273 ;
52 56 55 A0 00D6 274 ADDW R5, R6 ; R6 = FHI * 2**N = SFHI
08 56 63 00D9 275 SUBX3 R6, S^#1.0, R2 ; R2 = 1 - SFHI
56 56 60 00DD 276 ADDX S^#1.0, R6 ; R6 = 1 + SFHI
52 52 66 00E0 277 DIVX R6, R2 ; R2 = (1-SFHI)/(1+SFHI) = D
56 52 50 61 00E3 278 ADDX3 R0, R2, R6 ; R6 = D + X
52 52 64 00E7 279 MULX R0, R2 ; R2 = D * X
52 08 60 00EA 280 ADDX S^#1.0, R2 ; R2 = 1 + D*X
56 52 66 00ED 281 DIVX R2, R6 ; R6 = (D+X)/(1+D*X) = Z
00F0 282 ;
00F0 283 ; Compute Z**2, P(Z**2) and Z*P(Z**2)
00F0 284 ;
FF5E 50 56 56 65 00F0 285 MULX3 R6, R6, R0 ; R0 = Z**2
CF 05 50 75 00F4 286 POLYX R0, #LOGLEN2, LOGTAB2 ; R0 = P(Z**2)
50 56 64 00FA 287 MULX R6, R0 ; R0 = Z*P(Z**2)
00FD 288 ;
00FD 289 ; Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z*Z)
00FD 290 ;
52 FF06 CF 6E 65 00FD 291 MULX3 (SP), LN2_LO, R2 ; R2 = N*LN2_LO
52 52 88 60 0103 292 ADDX (R8)+, R2 ; R2 = N*LN2_LO + LN_FHI_LO
50 52 60 0106 293 ADDX R2, R0 ; R0 = B
0109 294 ;
0109 295 ; Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)
0109 296 ;
52 FEF2 CF 8E 65 0109 297 MULX3 (SP)+, LN2_HI, R2 ; R2 = N*LN2_HI
52 52 68 60 010F 298 ADDX (R8), R2 ; R2 = A = N*LN2_HI + LN_FHI_HI
50 50 52 60 0112 299 ADDX R2, R0 ; R0 = A + B = A[LOG(X)]
50 0080 8F A2 0115 300 SUBW2 #1@X_EXP, R0 ; Divide by 2
04 011A 301 RET
011B 302 ;
011B 303 NEG_EXP:
56 55 A2 011B 304 SUBW R5, R6 ; R6 = F = 2f
56 56 9A 011E 305 MOVZBL R6, R6 ; R6 = index into ALOG table
00000000 0121 306 .IF NE, F_EXP-X_EXP ;
0121 307 DIVW2 #1@<F_EXP-X_EXP>, R5 ; Shift R5 to scale X-floating
0121 308 .ENDC
7E 55 6D 0121 309 CVTWX R5, -(SP) ; Push N onto the stack
58 6846 98 0124 310 CVTBL (R8)[R6], R8 ; R8 = offset into FHI tables
0128 311 LN_1_PLUS_W:
0128 312 BCSS LN_1_PLUS ; Branch to handle F close to 1
58 00000000 GF48 7E 012A 313 MOVAX G^MTH$AB_D_FHI[R8], R8 ; R8 = Address of FHI
56 68 7D 0132 314 MOVX (R8), R6 ; R6 = FHI
0135 315 ;
0135 316 ; Compute Z = (F - FHI)/(F + FHI)
0135 317 ; Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
0135 318 ; Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
0135 319 ; where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
0135 320 ;
52 56 55 A0 0135 321 ADDW R5, R6 ; R6 = FHI * 2**N = SFHI
08 56 63 0138 322 SUBX3 R6, S^#1.0, R2 ; R2 = 1 - SFHI
56 56 60 013C 323 ADDX S^#1.0, R6 ; R6 = 1 + SFHI
52 52 66 013F 324 DIVX R6, R2 ; R2 = (1-SFHI)/(1+SFHI) = D
56 52 50 61 0142 325 ADDX3 R0, R2, R6 ; R6 = D + X
52 52 64 0146 326 MULX R0, R2 ; R2 = D * X
52 08 60 0149 327 ADDX S^#1.0, R2 ; R2 = 1 + D*X
56 52 66 014C 328 DIVX R2, R6 ; R6 = (D+X)/(1+D*X) = Z
```



```
014F 329 ;  
014F 330 ; Compute Z**2, P(Z**2) and Z*P(Z**2)  
014F 331 ;  
FEFF 50 56 56 65 014F 332 MULX3 R6, R6, R0 ; R0 = Z**2  
CF 05 50 75 0153 333 POLYX R0, #LOGLEN2, LOGTAB2 ; R0 = P(Z**2)  
50 56 64 0159 334 MULX R6, R0 ; R0 = Z*P(Z**2)  
015C 335 ;  
015C 336 ; Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z**2)  
015C 337 ;  
52 FEA7 CF 6E 65 015C 338 MULX3 (SP), LN2_LO, R2 ; R2 = N*LN2_LO  
52 52 78 60 0162 339 ADDX -(R8), R2 ; R2 = N*LN2_LO + LN_FHI_LO  
50 52 60 0165 340 ADDX R2, R0 ; R0 = B  
0168 341 ;  
0168 342 ; Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)  
0168 343 ;  
52 FE93 CF 8E 65 0168 344 MULX3 (SP)+, LN2_HI, R2 ; R2 = N*LN2_HI  
52 52 78 62 016E 345 SUBX -(R8), R2 ; R2 = A = N*LN2_HI + LN_FHI_HI  
50 50 52 60 0171 346 ADDX R2, R0 ; R0 = A + B = ALOG(X)  
50 0080 8F A2 0174 347 SUBW2 #10X_EXP, R0 ; Divide by 2  
04 0179 348 RET  
017A 349 ;  
017A 350 ;  
017A 351 ; Special logic for F close to 1  
017A 352 ;  
017A 353 ;  
56 08 50 63 017A 354 LN_1_PLUS:  
55 B5 017E 355 SUBX3 R0, S^#1.0, R6 ; R6 = 1-X  
0F 13 0180 356 TSTW R5 ; Determine which way to calculate W  
56 10 56 67 0182 357 BEQL 10$  
56 56 08 62 0186 358 DIVX3 R6, S^#2.0, R6 ; R6 = 2/(1-X)  
56 55 A2 0189 359 SUBX S^#1.0, R6 ; R6 = (1+X)/(1-X)  
56 56 08 62 018C 360 SUBW R5, R6 ; Scale R6  
56 09 11 018F 361 SUBX S^#1.0, R6 ; R6 = W  
20$ 20$ 362 BRB 20$  
56 50 56 67 0191 363 10$: DIVX3 R6, R0, R6 ; R6 = X / (1-X)  
56 0080 8F A0 0195 364 ADDW #10X_EXP, R6 ; R6 = W = 2*X/(1-X) = (1+X)/(1-X) - 1  
FE70 CF 08 56 75 019A 365 20$: POLYX R6, #LOGLEN1, LOGTAB1 ; R0 = Q(W)  
50 56 64 01A0 366 MULX R6, R0 ; Finish computing Q(W)  
50 56 64 01A3 367 MULX R6, R0 ; R0 = W*Q(W)  
52 FE5D CF 6E 65 01A6 368 MULX3 (SP), LN2_LO, R2 ; R2 = N*LN2_LO  
50 52 60 01AC 369 ADDX R2, R0 ; R0 = N*LN2_LO + W*Q(W)  
50 56 60 01AF 370 ADDX R6, R0 ; R0 = N*LN2_LO + W*Q(W) + W  
6E FE4A CF 64 01B2 371 MULX LN2_HI, (SP) ; (SP) = N*LN2_HI  
50 50 8E 60 01B7 372 ADDX (SP)+, R0 ; R0 = ALOG(X)  
50 0080 8F A2 01BA 373 SUBW2 #10X_EXP, R0 ; Divide by 2  
04 01BF 374 RET  
01C0 375 ;  
01C0 376 ; x <= 0.0, signal error  
01C0 377 ;  
7E 00'8F 9A 01C0 378 ERROR: MOVZBL #MTH$K_INVARGMAT, -(SP) ; condition value  
50 01 0F 79 01C4 379 ASHQ #15, #T, R0 ; R0 = result = reserved operand -0.0  
01C8 380 ; goes to signal mechanism vector  
01C8 381 ; (CHF$MCH_R0/R1) so error handler  
01C8 382 ; can modify the result.  
00000000'GF 01 FB 01C8 383 CALLS #1, G^MTH$$SIGNAL ; signal error and use real user's PC  
01CF 384 ; independent of CALL vs JSB  
04 01CF 385 RET ; return - R0 restored from
```


MTH\$DATANH
2-003

E 2
: Double Precision Hyperbolic Arctangent 16-SEP-1984 01:15:06 VAX/VMS Macro V04-00 Page 10
MTH\$DATANH - Double Precision Hyperbolic 6-SEP-1984 11:21:51 [MTHRTL.SRC]MTH\$DATANH.MAR;1 (5)

01D0 386
01D0 387

.END

; CHF\$\$_MCH_R0/R1

MTH
1-C

MTH\$DATANH
Symbol table

F 2
; Double Precision Hyperbolic Arctangent 16-SEP-1984 01:15:06 VAX/VMS Macro V04-00 Page 11
6-SEP-1984 11:21:51 [MTHRTL.SRC]MTH\$DATANH.MAR;1 (5)

```
ACMASK      = 000001FC
ERR          = 00000088 R    01
ERROR        = 000001C0 R    01
F_EXP        = 00000007
LN2_HI       = 00000000 R    01
LN2_LO       = 00000008 R    01
LN_T_PLUS    = 0000017A R    01
LN_1_PLUS_W  = 00000128 R    01
LOGLEN1      = 00000008
LOGLEN2      = 00000005
LOGTAB1      = 00000010 R    01
LOGTAB2      = 00000058 R    01
MTH$SAB ALOG_V ***** X    00
MTH$SAB D_FHT ***** X    00
MTH$SSIGNAL ***** X    00
MTH$DATANH   = 0000008B RG   01
MTH$K INVARGMAT ***** X    00
NEG_EXP      = 0000011B R    01
X            = 00000004
X_EXP        = 00000007
```

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

PSECT name	Allocation	PSECT No.	Attributes															
ABS	00000000 (0.)	00 (0.)	NOPI	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
MTH\$CODE	000001D0 (464.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG					

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.09	00:00:00.73
Command processing	157	00:00:00.72	00:00:04.94
Pass 1	101	00:00:01.37	00:00:05.19
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	82	00:00:00.93	00:00:03.49
Symbol table output	3	00:00:00.04	00:00:00.12
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	377	00:00:03.20	00:00:14.54

The working set limit was 1050 pages.
6942 bytes (14 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 20 non-local and 2 local symbols.
447 source lines were read in Pass 1, producing 11 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\$:MTH\$DATANH/OBJ=OBJ\$:MTH\$DATANH MSRC\$:MTH\$JACKET/UPDATE=(ENH\$:MTH\$JACKET)+MS

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY