

[illegible]

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

MM	MM	TTTTTTTTTT	HH	HH	GGGGGGGG	AAAAAA	TTTTTTTTTT	AAAAAA	NN	NN	HH	HH
MM	MM	TTTTTTTTTT	HH	HH	GGGGGGGG	AAAAAA	TTTTTTTTTT	AAAAAA	NN	NN	HH	HH
MMM	MMM	TT	HH	HH	GG	AA	TT	AA	NN	NN	HH	HH
MMM	MMM	TT	HH	HH	GG	AA	TT	AA	NN	NN	HH	HH
MM	MM	TT	HH	HH	GG	AA	TT	AA	NNNN	NN	HH	HH
MM	MM	TT	HH	HH	GG	AA	TT	AA	NNNN	NN	HH	HH
MM	MM	TT	HHHHHHHHHH	HH	GG	AA	TT	AA	NN	NN	HHHHHHHHHH	HH
MM	MM	TT	HHHHHHHHHH	HH	GG	AA	TT	AA	NN	NN	HHHHHHHHHH	HH
MM	MM	TT	HH	HH	GG	GGGGGG	TT	AAAAAAAAAA	NN	NNNN	HH	HH
MM	MM	TT	HH	HH	GG	GGGGGG	TT	AAAAAAAAAA	NN	NNNN	HH	HH
MM	MM	TT	HH	HH	GG	AA	TT	AA	NN	NN	HH	HH
MM	MM	TT	HH	HH	GG	AA	TT	AA	NN	NN	HH	HH
MM	MM	TT	HH	HH	GG	AA	TT	AA	NN	NN	HH	HH
MM	MM	TT	HH	HH	GGGGGG	AA	TT	AA	NN	NN	HH	HH
MM	MM	TT	HH	HH	GGGGGG	AA	TT	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
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```


(2) 47
(3) 61
(5) 154

HISTORY : Detailed Current Edit History
DECLARATIONS : Declarative Part of Module
MTH\$GATANH - G-floating Precision Hyperbolic Arctangent

```
0000 1 .TITLE MTH$GATANH ; G-floating Precision Hyperbolic Arctangent
0000 2 .IDENT /2-002/ ; File: MTHGATANH.MAR Edit: PDG2002
0000 3 :
0000 4 :*****
0000 5 :
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 8 :* ALL RIGHTS RESERVED. *
0000 9 :
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 15 :* TRANSFERRED. *
0000 16 :
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 19 :* CORPORATION. *
0000 20 :
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 23 :
0000 24 :
0000 25 :*****
0000 26 :
0000 27 :
0000 28 : FACILITY: MATH LIBRARY
0000 29 :++
0000 30 : ABSTRACT:
0000 31 :
0000 32 : MTH$GATANH returns the G-floating precision hyperbolic arctangent of the
0000 33 : G-floating 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\$GATANH
2-002

D 9

; G-floating Precision Hyperbolic Arctan 16-SEP-1984 01:25:48 VAX/VMS Macro V04-00 Page 2
HISTORY ; Detailed Current Edit History 6-SEP-1984 11:23:27 [MTHRTL.SRC]MTHGATANH.MAR;1 (2)

```
0000 47      .SBTTL HISTORY ; Detailed Current Edit History
0000 48
0000 49 : VERSION 1
0000 50 :
0000 51 : Edit History for Version 01 of MTH$GATANH
0000 52 :
0000 53 : 01-0 Original      July 1981
0000 54 :
0000 55 : Edit History for Version 02 of MTH$HATANH
0000 56 :
0000 57 : 2-000 Original      July 1981
0000 58 : 2-001 - Change MOVZBL to CVTBL when accessing MTH$$AB ALOG V. PDG 2-Dec-1981
0000 59 : 2-002 - Change RSB to RET after error exit. PDG 6-Jan-1981
```

```
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_G_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  ^X0DaSH+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  G, <^X40FD>, 8
0000      .OPDEF  ADDX   ^X00a8+^X40FD,RG,MG
0000      .OPDEF  ADDX3  ^X01a8+^X40FD,RG,RG,WG
0000      .OPDEF  SUBX   ^X02a8+^X40FD,RG,MG
0000      .OPDEF  SUBX3  ^X03a8+^X40FD,RG,RG,WG
0000      .OPDEF  MULX   ^X04a8+^X40FD,RG,MG
0000      .OPDEF  MULX3  ^X05a8+^X40FD,RG,RG,WG
0000      .OPDEF  DIVX   ^X06a8+^X40FD,RG,MG
0000      .OPDEF  DIVX3  ^X07a8+^X40FD,RG,RG,WG
0000      .OPDEF  CVTWX  ^X0Da8+^X40FD,RW,WG
0000      .OPDEF  POLYX  ^X15a8+^X40FD,RG,RW,AB
0000      .OPDEF  MOVX   ^X007D,RQ,WQ      : MOVQ
0000      .OPDEF  MOVAX  ^X007E,AQ,WL      : MOVAQ
0000
00000007 0000 101
00000004 0000 102      F_EXP = 7      ; Bit offset to exponent
0000      0000 103      X_EXP = 4      ; Bit offset to exponent
0000      0000 104
```


MTH\$GATANH
2-002

F 9
: G-floating Precision Hyperbolic Arctan 16-SEP-1984 01:25:48 VAX/VMS Macro V04-00
DECLARATIONS : Declarative Part of Modul 6-SEP-1984 11:23:27 [MTHRTL.SRC]MTHGATANH.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 "16" in the constants is used to shift the unbiased exponent
0000 122 : right 4 places so that the rightmost bit is at bit 0.
0000 123 :-
0000 124
2800FEF6 2E423FC6 0000 125 LN2_HI: .QUAD ^X2800FEF62E423FC6 : (Hi 42 bits of ln2)/16
F1DAD5E4 47BC3DA0 0008 126 LN2_LO: .QUAD ^XF1DAD5E447BC3DA0 : (Low bits of ln2)/16
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)
A8981E57 81CD3FDC 0010 131          .QUAD ^XA8981E5781CD3FDC : C8 = 0.11135560980588577
38EFC0D0 0802BFE0 0018 132          .QUAD ^X38EFC0D00802BFE0 : C7 = -0.12524446882930060
C9769148 49223FE2 0020 133          .QUAD ^XC976914849223FE2 : C6 = 0.14285690397225509
BBAC9487 5553BFE5 0028 134          .QUAD ^XBBAC94875553BFE5 : C5 = -0.16666645767642529
B92699D1 99993FE9 0030 135          .QUAD ^XB92699D199993FE9 : C4 = 0.20000000010208757
0A540014 0000BFF0 0038 136          .QUAD ^X0A5400140000BFF0 : C3 = -0.25000000007290635
54155555 55553FF5 0040 137          .QUAD ^X5415555555553FF5 : C2 = 0.333333333333331555
FF60FFFF FFFFBFFF 0048 138          .QUAD ^XFF60FFFFFBFFF : C1 = -0.49999999999999112
00000000 00000000 0050 139          .QUAD ^X0000000000000000 : C0 = 0.00000000000000000
00000008 0058 140 LOGLEN1 = .-LOGTAB1/8 - 1 : no. of floating point entries
0058 141
0058 142
0058 143 LOGTAB2: : Constants for p(z*z). Generated using
0058 144 : eq. 6.3.11 of Hart et. al. (sin(2a) =
0058 145 : (b - 1)/(b + 1) where b = 2**(1/7))
B117401D 6E163FE7 0058 146          .QUAD ^XB117401D6E163FE7 : C5 = 0.183047086054451497
0BA587C0 71A73FEC 0060 147          .QUAD ^X0BA587C071A73FEC : C4 = 0.222218457493082472
C30B9839 49243FF2 0068 148          .QUAD ^XC30B983949243FF2 : C3 = 0.285714291246265517
839E9998 99993FF9 0070 149          .QUAD ^X839E999899993FF9 : C2 = 0.3999999999996049627
55605555 55554005 0078 150          .QUAD ^X5560555555554005 : C1 = 0.666666666666667851
00000000 00004020 0080 151          .QUAD ^X0000000000004020 : C0 = 2.000000000000000000
00000005 0088 152 LOGLEN2 = .-LOGTAB2/8 - 1
```



```
0088 154 .SBTTL MTH$GATANH - G-floating Precision Hyperbolic Arctangent
0088 155
0088 156 :++
0088 157 : FUNCTIONAL DESCRIPTION:
0088 158
0088 159 : GATANH - G-floating precision floating point function
0088 160
0088 161 : GATANH(X) is computed using the following approximation technique:
0088 162
0088 163 : If |X| >= 1.0, error. Otherwise
0088 164
0088 165 : Let (1+X)/(1-X) = f * (2**n), where 1/2 <= f < 1
0088 166
0088 167 : If n is greater than or equal to 1 then
0088 168 :   set N = n - 1 and F1 = 2*f.
0088 169 : Else
0088 170 :   set N = n and F = f.
0088 171
0088 172 : If |F - 1| < 2**-5 then
0088 173 :   2*atanh(X) = N*ln(2) + W + W*P(W),
0088 174 :   where W = ((1+F)/(1-F))*2**N - 1,
0088 175 :   and P is a polynomial of degree F=5,D=9.
0088 176 : Else
0088 177 :   Obtain FHI (roughly equal to F) from table lookup.
0088 178 :   2*atanh(X) = ln((1+X)/(1-X)) = N*ln(2) + ln(FHI) + Z*Q(Z*Z),
0088 179 :   where Q is a polynomial of degree F=2,D=5,
0088 180 :   where Z = (F - FHI)/(F + FHI)
0088 181 :   where F = (2**N)*(1+X)/(1-X)
0088 182 :   Z is computed by:
0088 183 :   Z = (X-D)/(1-X*D)
0088 184 :   where Y = FHI*2**N
0088 185 :   where D = (Y-1)/(Y+1)
0088 186 :   Note that Z may be computed in a variety of ways:
0088 187 :   Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
0088 188 :   Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
0088 189 :   Z = [1 - Y + X + X*Y]/[1 + Y + X - X*Y]
0088 190 :   Z = [(1-Y) + X*(1+Y)]/[(1+Y) + X*(1-Y)]
0088 191
0088 192 : NOTE: The quantities ln(A=FHI) and ln2 are used in the above
0088 193 : equations in two parts - a high part (containing the
0088 194 : high order bits) and a low part (containing the low
0088 195 : order bits. In the code the high and low parts of the
0088 196 : constants are indicated by a HI and LO suffix respec-
0088 197 : tively. The values were chosen such that N*LN2_HI +
0088 198 : LN_FHI_HI is exactly representable.
0088 199
0088 200 : CALLING SEQUENCE:
0088 201
0088 202 :   gatanh.wg.v = MTH$GATANH(x.rg.r)
0088 203
0088 204 : INPUT PARAMETERS:
0088 205
0088 206 :   X = 4 ; Contents of x is the argument
0088 207
0088 208 : IMPLICIT INPUTS: none
0088 209
0088 210 : OUTPUT PARAMETERS:
```

```
0088 211 :  
0088 212 : VALUE: G-floating precision hyperbolic arctangent of the argument  
0088 213 :  
0088 214 : IMPLICIT OUTPUTS: none  
0088 215 :  
0088 216 : COMPLETION CODES: none  
0088 217 :  
0088 218 : SIDE EFFECTS:  
0088 219 :  
0088 220 : Signals: MTH$K_INVARGMAT if !X! >= 1.0 with reserved operand in R0 (copied to  
0088 221 : the signal mechanism vector CHFSL_MCH_R0/R1 by LIB$SIGNAL).  
0088 222 : Associated message is: "Floating overflow in math library". Result is  
0088 223 : reserved operand -0.0 unless a user supplied (or any) error handler changes  
0088 224 : CHFSL_MCH_R0/R1.  
0088 225 :  
0088 226 : NOTE: This procedure disables floating point underflow and integer  
0088 227 : overflow, causes no floating overflow or other arithmetic traps, and  
0088 228 : preserves enables across the call.  
0088 229 :  
0088 230 : Note: This routine is written to avoid causing any integer overflows,  
0088 231 : floating overflows, or floating underflows or divide by 0 conditions,  
0088 232 : whether enabled or not.  
0088 233 :  
0088 234 :---  
0088 235 :  
016A 31 0088 236 ERR: BRW ERROR  
0088 237 :  
01FC 0088 238 .ENTRY MTH$GATANH, ACMASK ; standard call-by-reference entry  
008D 239 : disable DV (and FU), enable IV  
50 04 BC 7D 008D 240 MOVX @X(AP), R0 ; R0 = arg  
0091 241 :  
52 56 50 7D 0091 242 MOVX R0, R6  
08 56 56 43FD 0094 243 SUBX3 R6, S^#1.0, R2 ; R2 = 1-X  
ED 15 0099 244 BLEQ ERR ; ATANH(X) is not defined for X>=1  
56 08 40FD 009B 245 ADDX S^#1.0, R6  
E7 15 009F 246 BLEQ ERR ; ATANH(X) is not defined for X<=-1  
52 52 33FD 00A1 247 CVTGF R2, R2  
56 56 33FD 00A5 248 CVTGF R6, R6  
56 52 46 00A9 249 DIVF2 R2, R6 ; R6 = approximation to (1+X)/(1-X)  
58 00000000'GF 9E 00AC 250 MOVAB G^MTH$$AB ALOG_V, R8 ; R8 = address of ALOG table  
58 58 68 C0 00B3 251 ADDL2 (R8), R8  
55 56 007F 8F AB 00B6 252 BICW3 #10F EXP-1, R6, R5 ; R5 = Biased exponent  
55 4000 8F A2 00BC 253 SUBW #^X4000, R5 ; R5 = Unbiased exponent  
76 15 00C1 254 BLEQ NEG_EXP ; Branch to processing for n<0  
00C3 255 :  
55 0080 8F A2 00C3 256 SUBW #10F EXP, R5 ; Exponent is positive, R5 = N = n - 1  
56 55 A2 00C8 257 SUBW R5, R6 ; R6 = F = 2f  
56 56 9A 00CB 258 MOVZBL R6, R6 ; R6 = index into ALOG table  
00000003 00CE 259 .IF NE, F_EXP-X_EXP  
55 08 A6 00CE 260 DIVW2 #10<F_EXP-X_EXP>, R5 ; Shift R5 to scale X-floating  
00D1 261 .ENDC  
7E 55 4DFD 00D1 262 CVTWX R5, -(SP) ; Push N onto the stack  
58 6846 98 00D5 263 CVTBL (R8)[R6], R8 ; R8 = offset into FHI tables  
6F 19 00D9 264 BLSS LN 1 PLUS_W ; Branch to handle F close to 1  
58 00000000'GF 48 7E 00DB 265 MOVAX G^MTH$$AB-G_FHI[R8], R8 ; R8 = Address of FHI  
56 88 7D 00E3 266 MOVX (R8)+, R6 ; R6 = FHI  
00E6 267 :
```



```
00E6 268 : Compute Z = (F - FHI)/(F + FHI)
00E6 269 : Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
00E6 270 : Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
00E6 271 : where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
00E6 272 :
52 56 55 A0 00E6 273 : ADDW R5, R6 ; R6 = FHI * 2**N = SFHI
08 56 56 43FD 00E9 274 : SUBX3 R6, S^#1.0, R2 ; R2 = 1 - SFHI
56 56 56 40FD 00EE 275 : ADDX S^#1.0, R6 ; R6 = 1 + SFHI
52 56 56 46FD 00F2 276 : DIVX R6, R2 ; R2 = (1-SFHI)/(1+SFHI) = D
56 52 50 41FD 00F6 277 : ADDX3 R0, R2, R6 ; R6 = D + X
52 52 50 44FD 00FB 278 : MULX R0, R2 ; R2 = D * X
52 08 40FD 00FF 279 : ADDX S^#1.0, R2 ; R2 = 1 + D*X
56 52 46FD 0103 280 : DIVX R2, R6 ; R6 = (D+X)/(1+D*X) = Z
0107 281 :
0107 282 : Compute Z**2, P(Z**2) and Z*P(Z**2)
0107 283 :
FF45 50 56 56 45FD 0107 284 : MULX3 R6, R6, R0 ; R0 = Z**2
CF 05 50 55FD 010C 285 : POLYX R0, #LOGLEN2, LOGTAB2 ; R0 = P(Z**2)
50 56 44FD 0113 286 : MULX R6, R0 ; R0 = Z*P(Z**2)
0117 287 :
0117 288 : Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z**2)
0117 289 :
52 FEED CF 6E 45FD 0117 290 : MULX3 (SP), LN2_LO, R2 ; R2 = N*LN2_LO
52 88 40FD 011E 291 : ADDX (R8)+, R2 ; R2 = N*LN2_LO + LN_FHI_LO
50 52 40FD 0122 292 : ADDX R2, R0 ; R0 = B
0126 293 :
0126 294 : Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)
0126 295 :
52 FED4 CF 8E 45FD 0126 296 : MULX3 (SP)+, LN2_HI, R2 ; R2 = N*LN2_HI
52 68 40FD 012D 297 : ADDX (R8), R2 ; R2 = A = N*LN2_HI + LN_FHI_HI
50 52 40FD 0131 298 : ADDX R2, R0 ; R0 = A + B = ALOG(X)
50 10 A2 0135 299 : SUBW2 #10X_EXP, R0 ; Divide by 2
04 0138 300 : RET
0139 301 :
0139 302 : NEG_EXP:
56 55 A2 0139 303 : SUBW R5, R6 ; R6 = F = 2f
56 56 9A 013C 304 : MOVZBL R6, R6 ; R6 = index into ALOG table
00000003 013F 305 : IF NE, F_EXP-X_EXP
55 08 A6 013F 306 : DIVW2 #10<F_EXP-X_EXP>, R5 ; Shift R5 to scale X-floating
0142 307 : ENDC
7E 55 4DFD 0142 308 : CVTWX R5, -(SP) ; Push N onto the stack
58 6846 98 0146 309 : CVTBL (R8)[R6], R8 ; R8 = offset into FHI tables
014A 310 : LN_1_PLUS_W:
58 00000000'GF48 7E 014A 311 : BCSS LN_1_PLUS ; Branch to handle F close to 1
56 68 7D 014C 312 : MOVAX G^MTH$SAB_G_FHI[R8], R8 ; R8 = Address of FHI
0154 313 : MOVX (R8), R6 ; R6 = FHI
0157 314 :
0157 315 : Compute Z = (F - FHI)/(F + FHI)
0157 316 : Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
0157 317 : Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
0157 318 : where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
0157 319 :
52 56 55 A0 0157 320 : ADDW R5, R6 ; R6 = FHI * 2**N = SFHI
08 56 56 43FD 015A 321 : SUBX3 R6, S^#1.0, R2 ; R2 = 1 - SFHI
56 56 56 40FD 015F 322 : ADDX S^#1.0, R6 ; R6 = 1 + SFHI
52 56 56 46FD 0163 323 : DIVX R6, R2 ; R2 = (1-SFHI)/(1+SFHI) = D
56 52 50 41FD 0167 324 : ADDX3 R0, R2, R6 ; R6 = D + X
```

```
52 50 44FD 016C 325      MULX    R0, R2      ; R2 = D * X
52 08 40FD 0170 326      ADDX    S^#1.0, R2    ; R2 = 1 + D*X
56 52 46FD 0174 327      DIVX    R2, R6      ; R6 = (D+X)/(1+D*X) = Z
                    0178 328      :
                    0178 329      : Compute Z**2, P(Z**2) and Z*P(Z**2)
                    0178 330      :
FED4 50 56 56 45FD 0178 331      MULX3   R6, R6, R0    ; R0 = Z**2
CF 05 50 55FD 017D 332      POLYX   R0, #LOGLEN2, LOGTAB2 ; R0 = P(Z**2)
50 56 44FD 0184 333      MULX    R6, R0      ; R0 = Z*P(Z**2)
                    0188 334      :
                    0188 335      : Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z**2)
                    0188 336      :
52 FE7A CF 6E 45FD 0188 337      MULX3   (SP), LN2_LO, R2    ; R2 = N*LN2_LO
52 78 40FD 018F 338      ADDX    -(R8), R2    ; R2 = N*LN2_LO + LN_FHI_LO
50 52 40FD 0193 339      ADDX    R2, R0      ; R0 = B
                    0197 340      :
                    0197 341      : Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)
                    0197 342      :
52 FE63 CF 8E 45FD 0197 343      MULX3   (SP)+, LN2_HI, R2    ; R2 = N*LN2_HI
52 78 42FD 019E 344      SUBX    -(R8), R2    ; R2 = A = N*LN2_HI + LN_FHI_HI
50 52 40FD 01A2 345      ADDX    R2, R0      ; R0 = A + B = ALOG(X)
50 10 A2 01A6 346      SUBW2   #10X_EXP, R0    ; Divide by 2
                    01A9 347      RET
                    01AA 348      :
                    01AA 349      : Special logic for F close to 1
                    01AA 350      :
                    01AA 351      :
                    01AA 352      :
                    01AA 353      LN_1_PLUS:
56 08 50 43FD 01AA 354      SUBX3   R0, S^#1.0, R6    ; R6 = 1-X
55 B5 01AF 355      TSTW    R5      ; Determine which way to calculate W
12 13 01B1 356      BEQL    10$
56 10 56 47FD 01B3 357      DIVX3   R6, S^#2.0, R6    ; R6 = 2/(1-X)
56 08 42FD 01B8 358      SUBX    S^#1.0, R6    ; R6 = (1+X)/(1-X)
56 55 A2 01BC 359      SUBW    R5, R6      ; Scale R6
56 08 42FD 01BF 360      SUBX    S^#1.0, R6    ; R6 = W
08 11 01C3 361      BRB     20$
56 50 56 47FD 01C5 362 10$: DIVX3   R6, R0, R6      ; R6 = X / (1-X)
56 10 A0 01CA 363      ADDW    #10X_EXP, R6    ; R6 = W = 2*X/(1-X) = (1+X)/(1-X) - 1
FE3C CF 08 56 55FD 01CD 364 20$: POLYX   R6, #LOGLEN1, LOGTAB1 ; R0 = Q(W)
50 56 44FD 01D4 365      MULX    R6, R0      ; R0 = W*Q(W)
52 FE2A CF 6E 45FD 01D8 366      MULX3   (SP), LN2_LO, R2    ; R2 = N*LN2_LO
50 52 40FD 01DF 367      ADDX    R2, R0      ; R0 = N*LN2_LO + W*Q(W)
50 56 40FD 01E3 368      ADDX    R6, R0      ; R0 = N*LN2_LO + W*Q(W) + W
6E FE14 CF 44FD 01E7 369      MULX    LN2_HI, (SP)    ; (SP) = N*LN2_HI
50 8E 40FD 01ED 370      ADDX    (SP)+, R0      ; R0 = ALOG(X)
50 10 A2 01F1 371      SUBW2   #10X_EXP, R0    ; Divide by 2
                    01F4 372      RET
                    01F5 373      :
                    01F5 374      : x <= 0.0, signal error
                    01F5 375      :
                    01F5 376      ERROR: MOVZBL #MTH$K_INVARGMAT, -(SP) ; condition value
50 7E 00'8F 9A 01F5 377      ASHQ    #15, #T, R0    ; R0 = result = reserved operand -0.0
01 0F 79 01F9 378      : goes to signal mechanism vector
                    01FD 379      : (CHFSL_MCH_R0/R1) so error handler
                    01FD 380      : can modify the result.
00000000'GF 01 FB 01FD 381      CALLS   #1, G^MTH$$SIGNAL ; signal error and use real user's PC
```


MTHSGATANH
2-002

L 9
; G-floating Precision Hyperbolic Arctan 16-SEP-1984 01:25:48 VAX/VMS Macro V04-00
MTHSGATANH - G-floating Precision Hyperb 6-SEP-1984 11:23:27 [MTHRTL.SRC]MTHGATANH.MAR;1 Page 10
(5)

04 0204 382
0204 383 RET
0205 384
0205 385 .END

; independent of CALL vs JSB
; return - R0 restored from
; CHFSL_MCH_R0/R1

MTHSGATANH
Symbol table

M 9
; G-floating Precision Hyperbolic Arctan 16-SEP-1984 01:25:48 VAX/VMS Macro V04-00 Page 11
6-SEP-1984 11:23:27 [MTHRTL.SRC]MTHGATANH.MAR;1 (5)

```
ACMASK      = 000001FC
ERR          = 00000088 R    01
ERROR       = 000001F5 R    01
F_EXP       = 00000007
LN2_HI      = 00000000 R    01
LN2_LO      = 00000008 R    01
LN_T_PLUS   = 000001AA R    01
LN-1-PLUS_W = 0000014A R    01
LOGLEN1     = 00000008
LOGLEN2     = 00000005
LOGTAB1     = 00000010 R    01
LOGTAB2     = 00000058 R    01
MTHSSAB ALOG V ***** X    00
MTHSSAB G_FHT ***** X    00
MTHSSIGNAL ***** X    00
MTHSGATANH  = 0000008B RG   01
MTHSK INVARGMAT ***** X    00
NEG_EXP     = 00000139 R    01
X           = 00000004
X_EXP       = 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			
_MTHSCODE	00000205 (517.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG			

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.85
Command processing	114	00:00:00.71	00:00:04.64
Pass 1	102	00:00:01.37	00:00:04.02
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	81	00:00:00.99	00:00:02.32
Symbol table output	3	00:00:00.00	00:00:00.17
Psect synopsis output	2	00:00:00.01	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	333	00:00:03.16	00:00:12.03

The working set limit was 900 pages.

7026 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.

445 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

_S255SDUA28:[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=LISS:MTHGATANH/OBJ=OBJ\$:MTHGATANH MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MS

0260 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

