

[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

```

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
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

(2)	50
(3)	69
(4)	128

HISTORY ; Detailed Current Edit History
DECLARATIONS ; Declarative Part of Module
MTH\$GCOSH - G Double Precision Floating GCOSH


```
0000 1      .TITLE  MTH$GCOSH      : G Floating Hyperbolic Cosine routine
0000 2      : (GCOSH)
0000 3      .IDENT /1-005/      : File: MTHGCOSH.MAR  EDIT: RNH1005
0000 4      :
0000 5      :*****
0000 6      :
0000 7      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :*  ALL RIGHTS RESERVED.
0000 10     :
0000 11     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :*  TRANSFERRED.
0000 17     :
0000 18     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :*  CORPORATION.
0000 21     :
0000 22     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :
0000 25     :*****
0000 26     :
0000 27     :
0000 28     :
0000 29     : FACILITY: MATH LIBRARY
0000 30     :++
0000 31     : ABSTRACT:
0000 32     :
0000 33     : MTH$GCOSH is a function which returns the G floating hyperbolic cosine
0000 34     : of its G floating point argument. The call is standard
0000 35     : call-by-reference.
0000 36     :
0000 37     :--
0000 38     :
0000 39     : VERSION: 1
0000 40     :
0000 41     : HISTORY:
0000 42     : AUTHOR:
0000 43     :      Steven B. Lionel, 26-Jan-79: Version 1
0000 44     :
0000 45     : MODIFIED BY:
0000 46     :
0000 47     :
0000 48     :
```

```

0000 50      .SBTTL  HISTORY ; Detailed Current Edit History
0000 51
0000 52 : Edit History for Version 1 of MTH$GCOSH
0000 53 :
0000 54 : 1-001 - Adapted from MTH$DCOSH.  SBL
0000 55 : 1-002 - Use MTH$GEXP_R6.  SBL 27-Sept-1979
0000 56 : 1-003 - Eliminated second call to EXP for input values between .25 and
0000 57 :           27*ln2 by computing COSH(x) = (Z + 1/Z)/2, with Z = EXP(ixi).
0000 58 :           - Eliminated second call to EXP for input values between 27*ln2
0000 59 :           and 1023*ln2.
0000 60 :           - Changed all final floating point divisions by 2 to interger
0000 61 :           subtracts of 1 from the exponent field.
0000 62 :           - Changed entry mask to excluded R7 - no longer needed.
0000 63 :           - Extended maximum range to 1024*ln2.
0000 64 :           - Changed logic for computing EXP(ixi-ln2) to reduce error.
0000 65 :           - RNH 10-FEB-81
0000 66 : 1-004 - Changed W^ to G^ in call to MTH$$SIGNAL RNH 09-Sept-1981
0000 67 : 1-005 - Eliminated symbolic short literals.  RNH 15-Oct-81

```

```
0000 69 .SBTTL DECLARATIONS ; Declarative Part of Module
0000 70
0000 71 :
0000 72 : INCLUDE FILES:
0000 73 :
0000 74 : EXTERNAL SYMBOLS:
0000 75 .DSABL GBL
0000 76 .EXTRN MTH$$SIGNAL
0000 77 .EXTRN MTH$K_FLOOVEMAT
0000 78 .EXTRN MTH$GEXP_R6
0000 79 :
0000 80 : EQUATED SYMBOLS:
0000 81
00000004 0000 82 value = 4 ; value.rg.r
0000 83
0000 84 :
0000 85 : MACROS: none
0000 86 :
0000 87 :
0000 88 :
0000 89 :
0000 90 : PSECT DECLARATIONS:
0000 91
00000000 0000 92 .PSECT _MTH$CODE PIC,SHR,LONG,EXE,NOWRT
0000 93 ; program section for math routines
0000 94 :
0000 95 : OWN STORAGE: none
0000 96 :
0000 97 :
0000 98 : CONSTANTS:
0000 99 :
0000 100 :
0000 101 G_1023_LOG_2:
7B606E3A 28B740A6 0000 102 .QUAD ^X7B606E3A28B740A6 ; 1023*ln2
0000 103 G_1024_LOG_2:
39EEFEFA 2E4240A6 0008 104 .QUAD ^X39EEFEFA2E4240A6 ; 1024*ln2
0010 105 G_LOG_2_HI:
3C00FEFA 2E424006 0010 106 .QUAD ^X3C00FEFA2E424006 ; (high 43 bits of ln2)+2** -43
0018 107 G_LOG_2_LO:
4C67361C 8654BD50 0018 108 .QUAD ^X4C67361C8654BD50 ; ln2 - G_LOG_2_HI
0020 109 G_2_POWER_27:
00000000 000041C0 0020 110 .QUAD ^X0000000000000041C0 ; 2**27
0028 111
0028 112 GCOSHTAB:
850A 3EB2 0028 113 .WORD ^0037262,^0102412
9B7A 67DD 002C 114 .WORD ^0063735,^0115572 ; DECIMAL: 0.2759648863787355D-06
019E 3F1A 0030 115 .WORD ^0037432,^0000636
194A 3565 0034 116 .WORD ^0032545,^0014512 ; DECIMAL: 0.2480155975461668D-04
C16C 3F76 0038 117 .WORD ^0037566,^0140554
3FC1 1700 003C 118 .WORD ^0013400,^0037701 ; DECIMAL: 0.13888888889781712D-02
5555 3FC5 0040 119 .WORD ^0037705,^0052525
4DF9 5555 0044 120 .WORD ^0052525,^0046771 ; DECIMAL: 0.4166666666665359D-01
0000 4000 0048 121 .WORD ^0040000,0
0001 0000 004C 122 .WORD 0,^0000001 ; DECIMAL: 0.5000000000000000D0
0000 4010 0050 123 .WORD ^0040020,0
0000 0000 0054 124 .WORD 0,0 ; DECIMAL: 0.1000000000000000D+01
00000006 0058 125 GCOSHTAB/8 = .- GCOSHTAB/8
```


MTHSGCOSH
1-005

N 10
; G Floating Hyperbolic Cosine routine 16-SEP-1984 01:26:35 VAX/VMS Macro V04-00
DECLARATIONS ; Declarative Part of Modul 6-SEP-1984 11:23:34 [MTHRTL.SRC]MTHGCOSH.MAR;1 Page 4
0058 126 (3)

```
0058 128 .SBTTL MTH$GCOSH - G Double Precision Floating GCOSH
0058 129
0058 130
0058 131 :++
0058 132 : FUNCTIONAL DESCRIPTION:
0058 133 :
0058 134 : GCOSH - double precision G floating point function
0058 135 :
0058 136 : GCOSH(X) is computed as:
0058 137 :
0058 138 : If |X| < 2**(-27), GCOSH(X) = 1.
0058 139 : If 2**(-27) <= |X| < 0.25, GCOSH(X) = Chebyshev series
0058 140 : If 0.25 <= |X| < 27*ln2, let Z = GEXP(|X|) and set GCOSH(X) = (Z+1/Z)/2
0058 141 : If 27*ln2 <= |X| < 1023*ln2, then GCOSH(X) = GEXP(|X|)/2.
0058 142 : If 1023*ln2 <= |X| < 1024*ln2, then GCOSH(X) = GEXP(|X|-ln2).
0058 143 : If 1024*ln2 <= |X|, then overflow.
0058 144 :
0058 145 : CALLING SEQUENCE:
0058 146 :
0058 147 : GCOSH.wg.v = MTH$GCOSH(x.rg.r)
0058 148 :
0058 149 : INPUT PARAMETERS:
0058 150 :
00000004 0058 151 : LONG = 4 ; define longword multiplier
00000004 0058 152 : x = 1 * LONG ; Contents of x is the argument
0058 153 :
0058 154 : IMPLICIT INPUTS: none
0058 155 :
0058 156 : OUTPUT PARAMETERS:
0058 157 :
0058 158 : VALUE: G floating hyperbolic cosine of the argument
0058 159 :
0058 160 : IMPLICIT OUTPUTS: none
0058 161 :
0058 162 : COMPLETION CODES: none
0058 163 :
0058 164 : SIDE EFFECTS:
0058 165 :
0058 166 : Signal: MTH$ FLOOVEMAT if 1024*ln2 <= |X| with reserved operand in R0/R1
0058 167 : (copied to the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
0058 168 : Associated message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved
0058 169 : operand -0.0 unless a user supplied (or any) error handler changes CHF$MCH_R0/R1
0058 170 :
0058 171 : NOTE: This procedure disables floating point underflow, enables integer
0058 172 : overflow.
0058 173 :
0058 174 : ---
0058 175 :
0058 176 :
0058 177 : .ENTRY MTH$GCOSH, ^M<IV, R2, R3, R4, R5, R6 >
005A 178 : ; standard call-by-reference entry
005A 179 : ; disable DV (and FU), enable IV
005A 180 : MTH$FLAG_JACKET ; flag that this is a jacket procedure in
005A
005A MOVAB G^MTH$$JACKET_HND, (FP)
0061 : ; set handler address to jacket
0061 : handler
```



```
0061 181
0061 182
0061 183
0061 184      MOVG    @value(AP), R0
50 04 BC 50FD 0066 185      BICW2   #^X8000, R0
50 8000 8F 50 AA 0068 186      CMPW    R0, #^X3FF0
3FF0 8F 50 B1 0070 187      BGEC    GEQ_TO_0.25
17 18 0072 188
0072 189
0072 190      : |X| < 0.25
0072 191
0072 192
0072 193      CMPW    R0, #^X3E60
3E60 8F 50 B1 0077 194      BGEC    GEQ_TO_2M27
05 18 0079 195
0079 196
0079 197      : |X| < 2**-27
0079 198
0079 199
0079 200      MOVG    #1, R0
50 08 50FD 007D 201      RET
04 007E 202
007E 203
007E 204      : 2**-27 =< |X| < 0.25
007E 205
007E 206
007E 207      GEQ_TO_2M27:
A0 AF 50 50 44FD 007E 208      MULG2   R0,R0
05 50 55FD 0082 209      POLYG    R0, #GCOSHLN-1, GCOSHTAB
04 0088 210
0088 211      RET
0089 212
0089 213
0089 214      : 0.25 =< |X|
0089 215
0089 216
0089 217      GEQ_TO_0.25:
FF71 CF 50 51FD 0089 218      CMPG    R0, G 1023 LOG 2
1A 14 008F 219      BGTR    GTR_TRAN_1023_LOG_2
0091 220
0091 221
0091 222      : 0.25 =< |X| =< 1023*ln2
0091 223
0091 224
0091 225      JSB      MTH$GEXP R6
00000000'EF 16 0097 226      CMPG    R0, G 2 POWER_27
84 AF 50 51FD 009C 227      BGTR    ONE_TERM_ONLY-27
09 14 009E 228      DIVG3   R0, #1, R2
52 08 50 47FD 00A3 229      ADDG2   R2, R0
50 52 40FD 00A7 230      ONE_TERM_ONLY:
50 10 A2 00A7 231      SUBW    #^X0010, R0
04 00AA 232
00AB 233
00AB 234
00AB 235      : 1023*ln2 < |X|
00AB 236
```

```
: case of an error in routine
: If an error, convert signal to user PC
: and resignal
: R0/R1 = |X| = @value(AP)
: R0/R1 = |X|
: compare |X| with 0.25
: branch if |X| >= 0.25
```

```
: compare |X| with 2**-27
: branch if |X| >= 2**-27
```

```
: R0/R1 = 1.0
: return with result = 1.0
```

```
: Get ARG**2 for POLYG.
: R0/R1 = SUM(Ci*X**i)
: return with result in R0
```

```
: compare |X| with 1023*ln2
: branch if |X| > 1023*ln2
```

```
: R0/R1 = GEXP(|X|)
: Compare GEXP(|X|) with 2**27, if
: larger, only one term is needed.
: R2/R3 = GEXP(-|X|)
: R0/R1 = GEXP(X) + GEXP(-X)
: R0/R1 = (GEXP(X) + GEXP(-X))/2
: return with result in R0/R1
```

```
00AB 237
00AB 238 GTR_THAN 1023_LOG_2:
FF57 CF 50 51FD 00AB 239 CMPG R0, G_1024_LOG_2 ; compare !X! with 1024*ln2
18 18 00B1 240 BGEQ ERROR ; branch to ERROR if 1024*ln2 <= !X!
00B3 241
00B3 242
00B3 243
00B3 244 : 1023*ln2 <= !X! < 1024*ln2
00B3 245 :
00B3 246
50 FF58 CF 42FD 00B3 247 SUBG2 G_LOG_2_HI, R0 ; R0/R1=!X!-(high 43 bits of ln2+2**-43)
00000000'EF 16 00B9 248 JSB MTH$GEXP_R6 ; R0/R1=GEXP(!X!-G_LOG_2_HI)
52 50 FF54 CF 45FD 00BF 249 MULG3 G_LOG_2_LO, R0, R2 ; R2/R3=G_LOG_2_LO*GEXP(!X!-G_LOG_2_LO)
50 52 40FD 00C6 250 ADDG2 R2, R0 ; R0/R1=GEXP(!X!-ln2)
04 00CA 251 RET ; return with result in R0/R1
00CB 252
00CB 253 : 1023*ln2 + LOG(2) <= !X!, error
00CB 254 :
00CB 255 :
00CB 256
7E 00'8F 9A 00CB 257 ERROR: MOVZBL #MTH$K_FLOOVEMAT, -(SP) ; condition value
50 01 0F 79 00CF 258 ASHQ #15, #T, R0 ; R0 = result = reserved operand -0.0
00D3 259 ; goes to signal mechanism vector
00D3 260 ; (CHF$MCH_R0/R1) so error handler
00D3 261 ; can modify the result.
00D3 262 CALLS #1, G^MTH$$SIGNAL ; signal error and use real user's PC
00DA 263 ; independent of CALL vs JSB
04 00DA 264 RET ; return - R0/R1 restored from CHF$MCH_R0/
00DB 265
00DB 266
00DB 267
00DB 268 .END
```


MTH\$GCOSH
Symbol table

E 11
; G Floating Hyperbolic Cosine routine

16-SEP-1984 01:26:35
6-SEP-1984 11:23:34

VAX/VMS Macro V04-00
[MTHRTL.SRC]MTHGCOSH.MAR;1

Page 8
(4)

```
ERROR          000000CB R      01
GCOSHLEN       = 00000006
GCOSHTAB       00000028 R      01
GEQ_TO_0.25    00000089 R      01
GEQ_TO_2M27    0000007E R      01
GTR_THAN_1023_LOG_2 000000AB R      01
G_1023_LOG_2   00000000 R      01
G_1024_LOG_2   00000008 R      01
G_2_POWER_27   00000020 R      01
G_LOG_2_HI     00000010 R      01
G_LOG_2_LO     00000018 R      01
LONG           = 00000004
MTH$$JACKET_HND ***** X      01
MTH$$SIGNAL     ***** X      00
MTH$GCOSH       00000058 RG     01
MTH$GEXP_R6     ***** X      00
MTH$K_FLOOVMAT  ***** X      00
ONE_TERM_ONLY   000000A7 R      01
VALUE           = 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	000000DB (219.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG				

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.73
Command processing	114	00:00:00.62	00:00:04.31
Pass 1	89	00:00:00.91	00:00:03.70
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	60	00:00:00.71	00:00:01.96
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	300	00:00:02.37	00:00:10.93

The working set limit was 900 pages.
3992 bytes (8 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 0 local symbols.
328 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

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

Macro library name

Macros defined

_S255\$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=LISS:MTHGCOSH/OBJ=OBJ\$:MTHGCOSH MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

0260

**DIGITAL
CONFIDE**

MTHGCONJ
LIS

MTHGINT
LIS

MTHGMOD
LISMTHES
LIS

MTHFLOOR
LIS

MTHDTAN
LISMTHDTANH
LISMTHGEXP
LIS

MTGFLOR
LIS

MTHGMINI
LISMTHGCOSH
LISMTHGLOG
LISMTHGACOS
LISMTHGMAX
LISMTHGNINT
LIS