```
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRR     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         HHHHHHHHHHHHHHHH  RRRRRRRRRRR           TTT         LLL
MMM         MMM         TTT         HHHHHHHHHHHHHHHH  RRRRRRRRRRRR          TTT         LLL
MMM         MMM         TTT         HHHHHHHHHHHHHHHH  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         LLLLLLLLLLLLLLLL
MMM         MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLLLLLLLLLLLLLLL
MMM         MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLLLLLLLLLLLLLLL
```

```
MM      MM  TTTTTTTTTT  HH      HH    CCCCCCCC    000000      SSSSSSSS  HH      HH
MM      MM  TTTTTTTTTT  HH      HH    CCCCCCCC    000000      SSSSSSSS  HH      HH
MMMM  MMMM      TT      HH      HH  CC          00      00  SS            HH      HH
MMMM  MMMM      TT      HH      HH  CC          00      00  SS            HH      HH
MM  MM  MM      TT      HH      HH  CC          00      00  SS            HH      HH
MM  MM  MM      TT      HH      HH  CC          00      00  SS            HH      HH
MM      MM      TT      HHHHHHHHHH  CC          00      00    SSSSSS    HHHHHHHHHH
MM      MM      TT      HHHHHHHHHH  CC          00      00    SSSSSS    HHHHHHHHHH
MM      MM      TT      HH      HH  CC          00      00          SS  HH      HH
MM      MM      TT      HH      HH  CC          00      00          SS  HH      HH
MM      MM      TT      HH      HH  CC          00      00          SS  HH      HH      ....
MM      MM      TT      HH      HH  CC          00      00          SS  HH      HH      ....
MM      MM      TT      HH      HH    CCCCCCCC    000000    SSSSSSSS    HH      HH      ....
MM      MM      TT      HH      HH    CCCCCCCC    000000    SSSSSSSS    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     1          .TITLE  MTH$COSH        ; Floating Point Hyperbolic Cosine routine
0000     2                                  ; (COSH)
0000     3          .IDENT /1-006/          ; File: MTHCOSH.MAR  Edit: RNH1006
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$COSH is a function  which  returns the floating point hyperbolic cosine
0000    34 ; of its single precision floating point argument. The call is standard
0000    35 ; call-by-reference.
0000    36 ;
0000    37 ;--
0000    38 ;
0000    39 ; VERSION: 01
0000    40 ;
0000    41 ; HISTORY:
0000    42 ; AUTHOR:
0000    43 ;       Peter Yuo, 29-Jun-77: Version 01
0000    44 ;
0000    45 ; MODIFIED BY:
0000    46 ;
0000    47 ;
0000    48 ;
```

```
0000     50          .SBTTL  HISTORY ; Detailed Current Edit History
0000     51
0000     52
0000     53 ; ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000     54 ;
0000     55 ; Edit History for Version 01 of MTH$COSH
0000     56 ;
0000     57 ; 0-2    MTH$$ERROR changed to MTH$$SIGNAL.
0000     58 ;        MTH$_... changed to MTH__....
0000     59 ;        Changed error handling mechanism. Put error result in R0 before
0000     60 ;        calling MTH$$SIGNAL in order to allow user modify error result.
0000     61 ;
0000     62 ;0-3     Series changed from four term Taylor series to four term
0000     63 ;        Chebyshev series, to improve accuracy. 18-May-1978; Mary Payne
0000     64 ; 1-001 - Update copyright notice and version number.  JBS 16-NOV-78
0000     65 ; 1-002 - Change MTH_FLOOVEMAT to MTH$K_FLOOVEMAT.  JBS 07-DEC-78
0000     66 ; 1-003 - Remove $SRMDEF macro -- not needed.  JBS 16-DEC-78
0000     67 ; 1-004 - Add "_" to the PSECT directive.  JBS 21-DEC-78
0000     68 ; 1-005 - Declare externals.  SBL 17-May-1979
0000     69 ; 1-006 - Changed lower limit for Chebyshev approximation from 2**-11 to
0000     70 ;          2**-12.
0000     71 ;        - Eliminated second call to EXP for input values between .25 and
0000     72 ;          12.5*ln2 by computing COSH(x) = (Z + 1/Z)/2, with Z = EXP(|x|).
0000     73 ;        - Eliminated second call to EXP for input values between 12.5*ln2
0000     74 ;          and 127*ln2.
0000     75 ;        - Changed all final floating point divisions by 2 to interger
0000     76 ;          subrtracts of 1 from the exponent field.
0000     77 ;        - Changed entry mask to excluded R5 - no longer needed.
0000     78 ;        - Extended maximum range from 87.69 to 128*ln2=88.72.
0000     79 ;        - Chaneged logic for computing EXP(|x|-ln2) to reduce error.
0000     80 ;        - RNH 10-FEB-81
```

```
                     0000     82          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                     0000     83
                     0000     84 ;
                     0000     85 ; INCLUDE FILES:
                     0000     86 ;
                     0000     87
                     0000     88 ;
                     0000     89 ; EXTERNAL SYMBOLS:
                     0000     90 ;
                     0000     91          .DSABL  GBL
                     0000     92          .EXTRN  MTH$EXP_R4
                     0000     93          .EXTRN  MTH$K_FLOOVEMAT
                     0000     94          .EXTRN  MTH$$SIGNAL
                     0000     95
                     0000     96 ;
                     0000     97 ; EQUATED SYMBOLS:
                     0000     98
72804031             0000     99          LF_LOG_2_HI     = ^X72804031    ; high 17 bit of ln2 + 2**-17
105CB7D0             0000    100          LF_LOG_2_LO     = ^X105CB7D0    ; ln2 - LF_LOG_2_HI
04F346B5             0000    101          LF_2_POWER_12.5 = ^X04F346B5    ; 2**12.5 = 5792.6188
0F3343B0             0000    102          LF_127_LOG_2    = ^X0F3343B0    ; 127*ln2 = 88.029613
721743B1             0000    103          LF_128_LOG_2    = ^X721743B1    ; 128*ln2 = 88.722839
00004080             0000    104          SF_1.0  = %F1.0                 ; 1.0
00000004             0000    105          value   = 4                     ; value.rf.r
                     0000    106
                     0000    107 ;
                     0000    108 ; MACROS:
                     0000    109 ;
                     0000    110
                     0000    111
                     0000    112
                     0000    113 ;
                     0000    114 ;
                     0000    115 ; PSECT DECLARATIONS:
                     0000    116
                00000000     117          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                     0000    118                                          ; program section for math routines
                     0000    119 ;
                     0000    120 ; OWN STORAGE:  none
                     0000    121 ;
                     0000    122 ;
                     0000    123 ; CONSTANTS:
                     0000    124 ;
                     0000    125
                     0000    126 COSHTAB:
7381 3BB6            0000    127          .WORD   ^0035666,^0071601       ; DECIMAL: 0.1391992E-02
AA8A 3E2A            0004    128          .WORD   ^037052,^0125212        ; DECIMAL: 0.416654E-01
0000 4000            0008    129          .WORD   ^0040000,0              ; DECIMAL: 0.5000000E+00
0000 4080            000C    130          .WORD   ^0040200,0              ; DECIMAL: 0.1000000E+01
                     0010    131
 00000004            0010    132 COSHLEN = .- COSHTAB/4
                     0010    133
```

```
                     0010   135          .SBTTL  MTH$COSH  - Standard Single Precision Floating COSH
                     0010   136
                     0010   137
                     0010   138 ;++
                     0010   139 ; FUNCTIONAL DESCRIPTION:
                     0010   140 ;
                     0010   141 ; COSH  - single precision floating point function
                     0010   142 ;
                     0010   143 ; COSH(X) is computed as:
                     0010   144 ;
                     0010   145 ;       If |X| < 2**-12, COSH(X) = 1.
                     0010   146 ;       If 2**-12 =< |X| < 0.25, COSH(X) = Chebyshev series
                     0010   147 ;       If 0.25 =< |X| =< 8.6643975, set Z = EXP(|X|), and compute
                     0010   148 ;           COSH(X) = (Z + 1/Z)/2
                     0010   149 ;       If 8.6643975 < |X| < 88.029613 then COSH(X) = EXP(|X|)/2
                     0010   150 ;       If 88.029613 < |X| < 88.722839 then COSH(X) = EXP(|X|-ln2)
                     0010   151 ;       If 88.722839 <= |X| overflow is signaled
                     0010   152 ;
                     0010   153 ; CALLING SEQUENCE:
                     0010   154 ;
                     0010   155 ;       COSH.wf.v = MTH$COSH(x.rf.r)
                     0010   156 ;
                     0010   157 ; INPUT PARAMETERS:
                     0010   158
           00000004  0010   159          LONG = 4                        ; define longword multiplier
           00000004  0010   160          x = 1 * LONG                    ; Contents of x is the argument
                     0010   161
                     0010   162 ; IMPLICIT INPUTS:      none
                     0010   163 ;
                     0010   164 ; OUTPUT PARAMETERS:
                     0010   165 ;
                     0010   166 ;       VALUE:  floating hyperbolic cosine of the argument
                     0010   167 ;
                     0010   168 ; IMPLICIT OUTPUTS:     none
                     0010   169 ;
                     0010   170 ; COMPLETION CODES:     none
                     0010   171 ;
                     0010   172 ; SIDE EFFECTS:
                     0010   173 ;
                     0010   174 ; Signal: MTH$_FLOOVEMAT if 88.722839 <= |X| with reserved operand in R0
                     0010   175 ; (copied to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                     0010   176 ; Associated message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved
                     0010   177 ; operand -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1
                     0010   178 ;
                     0010   179 ; NOTE: This procedure disables floating point underflow, enables integer
                     0010   180 ; overflow.
                     0010   181 ;
                     0010   182 ;---
                     0010   183
                     0010   184
                 401C 0010   185          .ENTRY  MTH$COSH, ^M<IV ,R2, R3, R4>
                     0012   186                                          ; standard call-by-reference entry
                     0012   187                                          ; disable DV (and FU), enable IV
                     0012   188          MTH$FLAG_JACKET                 ; flag that this is a jacket procedure in
                     0012
           6D 00000000'GF 9E 0012                MOVAB   G^MTH$$JACKET_HND, (FP)
                     0019                                               ; set handler address to jacket
```

```
                                0019
                                0019
                                0019    189                                     ; case of an error in routine
                                0019    190                                     ; If an error, convert signal to user PC
                                0019    191                                     ; and resignal
   50    04 BC    50           0019    192             MOVF    @value(AP), R0  ; R0 = X = @value(AP)
50    8000 8F    AA           001D    193             BICW2   #^X8000, R0     ; R0 = |X|
3F80 8F    50    B1           0022    194             CMPW    R0, #^X3F80     ; compare |X| with 0.25
           14    18           0027    195             BGEQ    GEQ_TO_0.25     ; branch if |X| >= 0.25
                                0029    196
                                0029    197 ;
                                0029    198 ; |X| < 0.25
                                0029    199 ;
                                0029    200
3A80 8F    50    B1           0029    201             CMPW    R0, #^X3A80     ; compare |X| with 2**-12
           04    18           002E    202             BGEQ    GEQ_TO_2M12     ; branch if |X| >= 2**-12
                                0030    203
                                0030    204 ;
                                0030    205 ; |X| < 2**-12
                                0030    206 ;
                                0030    207
     50    08    50           0030    208             MOVF    S^#SF_1.0, R0   ; R0 = 1.0
           04    04           0033    209             RET                     ; return with result = 1.0
                                0034    210
                                0034    211 ;
                                0034    212 ; 2**-12 =< |X| < 0.25
                                0034    213 ;
                                0034    214
                                0034    215 GEQ_TO_2M12:
     50    50    44           0034    216             MULF    R0,R0           ; Get ARG**2 for POLYF
C4 AF    03    50    55       0037    217             POLYF   R0, #COSHLEN-1, COSHTAB ; R0 = SUM(Ci*X**(2*i))
                                003C    218                                     ; Last coefficient of 1.0
                                003C    219                                     ; provides overhang.
           04                   003C    220             RET                     ; return with result in R0
                                003D    221
                                003D    222 ;
                                003D    223 ; 0.25 =< |X|
                                003D    224 ;
                                003D    225
                                003D    226 GEQ_TO_0.25:
0F3343B0 8F    50    51       003D    227             CMPF    R0, #LF_127_LOG_2       ; compare |X| with 127*ln2
               1C    14       0044    228             BGTR    GTR_THAN_127_LOG_2      ; branch if |X| > 127*ln2
                                0046    229
                                0046    230 ;
                                0046    231 ; 0.25 =< |X| =< 127*ln2
                                0046    232 ;
                                0046    233
     00000000'EF               16       0046    234             JSB     MTH$EXP_R4              ; R0 = EXP(|X|)
04F346B5 8F    50    51       004C    235             CMPF    R0, #LF_2_POWER_12.5    ; compare EXP(|X|) with 2**12.5
               07    14       0053    236             BGTR    ONE_TERM_ONLY           ; If smaller EXP(-|X|) is significant
                                0055    237                                     ; and must be added in
     51    08    50    47       0055    238             DIVF3   R0, #SF_1.0, R1 ; R1 = EXP(-|X|)
           50    51    40       0059    239             ADDF    R1, R0          ; R0 = EXP(X) + EXP(-X)
                                005C    240 ONE_TERM_ONLY:
     50    0080 8F    A2       005C    241             SUBW    #^X0080, R0     ; R0 = (EXP(X) + EXP(-X))/2
                           04    0061    242             RET                     ; return with result in R0
                                0062    243
```

```
                                0062   244
                                0062   245 ;
                                0062   246 ; 127*ln2 < |X|
                                0062   247 ;
                                0062   248
                                0062   249 GTR_THAN_127_LOG_2:
        721743B1 8F    50   51  0062   250         CMPF    R0, #LF_128_LOG_2       ; Check for possible overflow
                        19   14  0069   251         BGTR    ERROR                   ; If greater, COSH(X) will overflow
        50    72804031 8F   42  0068   252         SUBF    #LF_LOG_2_HI, R0        ; R0 = |X| - (high order bits of ln2)
              00000000'EF   16  0072   253         JSB     MTH$EXP_R4              ; R0 = EXP(|X|-LF_LOG_2_HI)
  51  50    105CB7D0 8F   45  0078   254         MULF3   #LF_LOG_2_LO, R0, R1    ;
                 50   51   42  0080   255         SUBF    R1, R0                  ;
                      04   0083   256         RET
                           0084   257
                           0084   258
                           0084   259
                           0084   260 ;
                           0084   261 ; 128*ln2 =< |X|, error
                           0084   262 ;
                           0084   263
           7E    00'8F    9A  0084   264 ERROR:  MOVZBL  #MTH$K_FLOOVEMAT, -(SP) ; condition value
        50   01    0F    78  0088   265         ASHL    #15, #1, R0             ; R0 = result = reserved operand -0.0
                           008C   266                                           ; goes to signal mechanism vector
                           008C   267                                           ; (CHF$L_MCH_R0/R1) so error handler
                           008C   268                                           ; can modify the result.
        00000000'GF    01  FB  008C   269         CALLS   #1, G^MTH$$SIGNAL       ; signal error and use real user's PC
                           0093   270                                           ; independent of CALL vs JSB
                      04   0093   271         RET                             ; return - R0 restored from CHF$L_MCH_R0/R1
                           0094   272
                           0094   273
                           0094   274
                           0094   275         .END
```

```
COSHLEN                         = 00000004
COSHTAB                           00000000 R      01
ERROR                             00000084 R      01
GEQ_TO_0.25                       0000003D R      01
GEQ_TO_2M12                       00000034 R      01
GTR_THAN_127_LOG_2                00000062 R      01
LF_127_LOG_2                    = 0F3343B0
LF_128_LOG_2                    = 721743B1
LF_2_POWER_12.5                 = 04F346B5
LF_LOG_2_HI                     = 72804031
LF_LOG_2_LO                     = 105CB7D0
LONG                            = 00000004
MTH$$JACKET_HND                   ********    X   01
MTH$$SIGNAL                       ********    X   00
MTH$COSH                          00000010 RG     01
MTH$EXP_R4                        ********    X   00
MTH$K_FCOOVEMAT                   ********    X   00
ONE_TERM_ONLY                     0000005C R      01
SF_1.0                          = 00004080
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	00000094 (148.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.62
Command processing	121	00:00:00.63	00:00:04.32
Pass 1	88	00:00:00.93	00:00:03.61
Symbol table sort	0	00:00:00.01	00:00:00.05
Pass 2	61	00:00:00.72	00:00:02.66
Symbol table output	3	00:00:00.03	00:00:00.05
Psect synopsis output	3	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	307	00:00:02.42	00:00:11.35

The working set limit was 900 pages.
3994 bytes (8 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 0 local symbols.
335 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.

J 10

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

Macro library name	Macros defined
_255DUA28:[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$:MTHCOSH/OBJ=OBJ$:MTHCOSH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:
```

MTHCVTDG
LIS

MTHDACOS
LIS

MTHCGABS
LIS

MTHCDSINC
LIS

MTHCLOG
LIS

MTHDASIN
LIS

MTHCGLOG
LIS

MTHCONVER
LIS

MTHDATAN
LIS

MTHCSQRT
LIS

MTHCEXP
LIS

MTHCGSQRT
LIS

MTHCGEXP
LIS

MTHCSINCO
LIS

MTHCONJG
LIS

MTHCDSQRT
LIS

MTHCGSINC
LIS

MTHCOSH
LIS