```
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  HH      HH    CCCCCCCC    000000      SSSSSSSS  HH      HH
MM      MM  TTTTTTTTTT  HH      HH  HH      HH    CCCCCCCC    000000      SSSSSSSS  HH      HH
MMMM  MMMM      TT      HH      HH  HH      HH  CC          00      00  SS            HH      HH
MMMM  MMMM      TT      HH      HH  HH      HH  CC          00      00  SS            HH      HH
MM  MM  MM      TT      HH      HH  HH      HH  CC          00      00  SS            HH      HH
MM  MM  MM      TT      HH      HH  HH      HH  CC          00      00  SS            HH      HH
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  CC          00      00    SSSSSS    HHHHHHHHHH
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  CC          00      00    SSSSSS    HHHHHHHHHH
MM      MM      TT      HH      HH  HH      HH  CC          00      00          SS  HH      HH
MM      MM      TT      HH      HH  HH      HH  CC          00      00          SS  HH      HH
MM      MM      TT      HH      HH  HH      HH  CC          00      00          SS  HH      HH      ....
MM      MM      TT      HH      HH  HH      HH  CC          00      00          SS  HH      HH      ....
MM      MM      TT      HH      HH  HH      HH    CCCCCCCC    000000    SSSSSSSS    HH      HH      ....
MM      MM      TT      HH      HH  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
```

```
0000     1          .TITLE  MTH$HCOSH       ; H floating Hyperbolic Cosine routine
0000     2                                  ; (HCOSH)
0000     3          .IDENT /1-006/          ; File: MTHHCOSH.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$HCOSH is a function  which  returns the H floating hyperbolic cosine
0000    34 ; of its H 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 ;       John A. Wheeler, 12-Sep-1979: 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
0000     53 ; Edit History for Version 1 of MTH$HCOSH
0000     54 ;
0000     55 ; 1-001 - Adapted from MTH$GCOSH.  JW 12-Sep-1979
0000     56 ; 1-002 - Use MTH$HEXP_R6.  SBL 4-Oct-1979
0000     57 ; 1-003 - H_0.25 should be literal rather than own storage.  SBL 7-Oct-1979
0000     58 ; 1-004 - Don't store reserved operand before signal.  SBL 7-Feb-1980
0000     59 ; 1-005 - Changed lower limit for Chebyshev approximation from 2**-56 to
0000     60 ;         2**-57.
0000     61 ;       - Eliminated second call to EXP for input values between .25 and
0000     62 ;         57*ln2 by computing COSH(x) = (Z + 1/Z)/2, with Z = EXP(|x|)
0000     63 ;       - Eliminated second call to EXP for input values between 57*ln2
0000     64 ;         and 16383*ln2.
0000     65 ;       - Changed all final floating point divisions by 2 to interger
0000     66 ;         subrtracts of 1 from the exponent field.
0000     67 ;       - Extended maximum range to 16384*ln2.
0000     68 ;       - Chaneged logic for computing EXP(|x|-ln2) to reduce error.
0000     69 ;       - RNH 10-FEB-81
0000     70 ; 1-006 - Eliminated symbolic short literals.  RNH 15-Oct-81
```

```
                                0000     72          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                                0000     73
                                0000     74 ;
                                0000     75 ; INCLUDE FILES:
                                0000     76 ;
                                0000     77 ; EXTERNAL SYMBOLS:
                                0000     78          .DSABL  GBL
                                0000     79          .EXTRN  MTH$$SIGNAL
                                0000     80          .EXTRN  MTH$K_FLOOVEMAT
                                0000     81          .EXTRN  MTH$HEXP_R6
                                0000     82 ;
                                0000     83 ; EQUATED SYMBOLS:
                                0000     84
                                0000     85
                                0000     86 ;
                                0000     87 ; MACROS:        none
                                0000     88 ;
                                0000     89
                                0000     90
                                0000     91 ;
                                0000     92 ; PSECT DECLARATIONS:
                                0000     93
                           00000000     94          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                                0000     95                                          ; program section for math routines
                                0000     96 ;
                                0000     97 ; OWN STORAGE:  none
                                0000     98 ;
                                0000     99 ;
                                0000    100 ; CONSTANTS:
                                0000    101 ;
                                0000    102
                                0000    103 H_16383_LOG_2:
E3E0A45E 62DE400E               0000    104          .QUAD   ^XE3E0A45E62DE400E
6B251812 C66964DB               0008    105          .QUAD   ^X6B251812C66964DB      ; 16383*ln2
                                0010    106 H_16384_LOG_2:
A39E2FEF 62E4400E               0010    107          .QUAD   ^XA39E2FEF62E4400E
07E66730 93C7F357               0018    108          .QUAD   ^X07E6673093C7F357      ; 16384*ln2
                                0020    109 H_LOG_2_HI:
A39E2FEF 62E44000               0020    110          .QUAD   ^XA39E2FEF62E44000
40006730 93C7F357               0028    111          .QUAD   ^X4000673093C7F357      ; (high 109 bits of ln2)*2**-109
                                0030    112 H_LOG_2_LO:
F0CB950B C0D0BF9D               0030    113          .QUAD   ^XF0CB950BC0D0BF9D
5F359D27 D674CD98               0038    114          .QUAD   ^X5F359D27D674CD98      ; ln2 - H_LOG_2_HI
                                0040    115 H_2_POWER_57:
00000000 0000403A               0040    116          .QUAD   ^X000000000000403A
00000000 00000000               0048    117          .QUAD   0                       ; 2**57
                                0050    118
                                0050    119 HCOSHTAB:
2B3EE172 8C8F3FA8               0050    120          .LONG   ^X8C8F3FA8,^X2B3EE172   ;  2.50265827956782399016245815625471lE -27
ADEB4F57 5BD2E986               0058    121          .LONG   ^X5BD2E986,^XADEB4F57
D01949AF F2C93FB1               0060    122          .LONG   ^XF2C93FB1,^XD01949AF   ;  1.61166539506646248033071976723655 9E -24
C60346B9 1DF63B8C               0068    123          .LONG   ^X1DF63B8C,^XC60346B9
64FF997D 0CE33FBB               0070    124          .LONG   ^X0CE33FBB,^X64FF997D   ;  8.89679272147687223764798237825356 8E -22
7CB38AA4 0B3F40F3               0078    125          .LONG   ^X0B3F40F3,^X7CB38AA4
F4FDBA3C E5423FC3               0080    126          .LONG   ^XE5423FC3,^XF4FDBA3C   ;  4.11031762171246693744590271725186 0E -19
60105113 6DEFA6F1               0088    127          .LONG   ^X6DEFA6F1,^X60105113
9928863B 68273FCC               0090    128          .LONG   ^X68273FCC,^X9928863B   ;  1.56192069685994452872180519113086 5E -16
```

```
E533A92E 54A49ACF  0098   129          .LONG   ^X54A49ACF,^XE533A92E
3B813E73 AE7F3FD4  00A0   130          .LONG   ^XAE7F3FD4,^X3B813E73   ;  4.77947733238737763490125517480847 1E -14
4B6B3FDC D10B2ED8  00A8   131          .LONG   ^XD10B2ED8,^X4B6B3FDC
07C94A8C 93973FDC  00B0   132          .LONG   ^X93973FDC,^X07C94A8C   ;  1.14707455977297247451323408376859 0E -11
7DE98CBB FC3DD25A  00B8   133          .LONG   ^XFC3DD25A,^X7DE98CBB
8D898EFF 1EED3FE4  00C0   134          .LONG   ^X1EED3FE4,^X8D898EFF   ;  2.08767569878680989791211249721242 3E  -9
85215944 371B7B54  00C8   135          .LONG   ^X371B7B54,^X85215944
89F5FB77 27E43FEB  00D0   136          .LONG   ^X27E43FEB,^X89F5FB77   ;  2.75573192239858906525574909285191 1E  -7
94EA88C4 F01FC72E  00D8   137          .LONG   ^XF01FC72E,^X94EA88C4
1A0101A0 A01A3FF1  00E0   138          .LONG   ^XA01A3FF1,^X1A0101A0   ;  2.48015873015873015873015870869392 2E  -5
1BE015A7 01A0A01A  00E8   139          .LONG   ^X01A0A01A,^X1BE015A7
16C1C16C 6C163FF7  00F0   140          .LONG   ^X6C163FF7,^X16C1C16C   ;  1.38888888888888888888888888890495 9E  -3
B9DF16C2 C16C6C16  00F8   141          .LONG   ^XC16C6C16,^XB9DF16C2
55555555 55553FFC  0100   142          .LONG   ^X55553FFC,^X55555555   ;  4.16666666666666666666666666666060 3E  -2
54ED5555 55555555  0108   143          .LONG   ^X55555555,^X54ED5555
00000000 00004000  0110   144          .LONG   ^X00004000,^X00000000   ;  5.00000000000000000000000000000000 0E  -1
00000000 00000000  0118   145          .LONG   ^X00000000,^X00000000
00000000 00004001  0120   146          .LONG   ^X00004001,^X00000000   ;  9.99999999999999999999999999999999 9E  -1
00000000 00000000  0128   147          .LONG   ^X00000000,^X00000000
                   0130   148
          0000000E 0130   149 HCOSHLEN        = .- HCOSHTAB/16
                   0130   150
```

```
                     0130   152          .SBTTL  MTH$HCOSH  - H floating point COSH function
                     0130   153
                     0130   154
                     0130   155 ;++
                     0130   156 ; FUNCTIONAL DESCRIPTION:
                     0130   157 ;
                     0130   158 ; HCOSH  - double precision H floating point function
                     0130   159 ;
                     0130   160 ; HCOSH(X) is computed as:
                     0130   161 ;
                     0130   162 ;       If |X| < 2**-57, HCOSH(X) = 1.
                     0130   163 ;       IF 2**-57 =< |X| < 0.25, HCOSH(X) = Chebyshev series
                     0130   164 ;       If 0.25 =< |X| < 57*ln2, let Z = HEXP(|X|) and set HCOSH(X) = (Z+1/Z)/2
                     0130   165 ;       If 57*ln2 =< |X| 16383*ln2, then HCOSH(X) = HEXP(|X|)/2.
                     0130   166 ;       If 16383*ln2 =< |X| < 16384*ln2, then HCOSH(X) = HEXP(|X|-ln2).
                     0130   167 ;       If 16384*ln2 =< |X|, then overflow.
                     0130   168 ;
                     0130   169 ; CALLING SEQUENCE:
                     0130   170 ;
                     0130   171 ;       hcosh.wh.v = MTH$HCOSH(x.rh.r)
                     0130   172 ;
                     0130   173 ;               -or-
                     0130   174 ;
                     0130   175 ;       CALL MTH$HCOSH(hcosh.wh.r, x.rh.r)
                     0130   176 ;
                     0130   177 ;       Because an H-floating result cannot be expressed in 64 bits, it is
                     0130   178 ;       returned as the first argument, with the input parameter displaced
                     0130   179 ;       to the second argument, in accordance with the Procedure Calling
                     0130   180
                     0130   181 ;
                     0130   182 ; INPUT PARAMETERS:
                     0130   183
00000004             0130   184          LONG = 4                        ; define longword multiplier
00000008             0130   185          x = 2 * LONG                    ; Contents of x is the argument
                     0130   186
                     0130   187 ; IMPLICIT INPUTS:      none
                     0130   188 ;
                     0130   189 ; OUTPUT PARAMETERS:
                     0130   190 ;
00000004             0130   191          hcosh = 1 * LONG
                     0130   192 ;
                     0130   193 ; IMPLICIT OUTPUTS:     none
                     0130   194 ;
                     0130   195 ; COMPLETION CODES:     none
                     0130   196 ;
                     0130   197 ; SIDE EFFECTS:
                     0130   198 ;
                     0130   199 ; Signal: MTH$_FLOOVEMAT if 16384*ln2 =< |X| with reserved operand in R0/R3
                     0130   200 ; (copied to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                     0130   201 ; Associated message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved
                     0130   202 ; operand -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1
                     0130   203 ;
                     0130   204 ; NOTE: This procedure disables floating point underflow, enables integer
                     0130   205 ; overflow.
                     0130   206 ;
                     0130   207 ;---
                     0130   208
```

```
                                0130   209
                         40FC   0130   210          .ENTRY  MTH$HCOSH, ^M<IV, R2, R3, R4, R5, R6, R7 >
                                0132   211                                              ; standard call-by-reference entry
                                0132   212                                              ; disable DV (and FU), enable IV
                                0132   213          MTH$FLAG_JACKET                     ; flag that this is a jacket procedure in
                                0132                                                    ;
    6D   00000000'GF     9E    0132                MOVAB   G^MTH$$JACKET_HND, (FP)
                                0139                                                    ; set handler address to jacket
                                0139                                                    ; handler
                                0139
                                0139   214                                              ; case of an error in routine
    6D   00000000'GF     9E    0139   215          MOVAB   G^MTH$$JACKET_HND, (FP)     ; Set handler address to jacket handler
                                0140   216                                              ; If an error, convert signal to user PC
                                0140   217                                              ; and resignal
      50    08 BC   70FD    0140   218          MOVH    @x(AP), R0                      ; R0/R3 = |X| = @value(AP)
    50    8000 8F    AA     0145   219          BICW2   #^X8000, R0                     ; R0/R3 = |X|
    3FFF 8F    50    B1     014A   220          CMPW    R0, #^X3FFF                     ; compare |X| with 0.25
               1E    18     014F   221          BGEQ    GEQ_TO_0.25                     ; branch if |X| >= 0.25
                             0151   222
                             0151   223 ;
                             0151   224 ; |X| < 0.25
                             0151   225 ;
                             0151   226
    3FC8 8F    50    B1     0151   227          CMPW    R0, #^X3FC8                     ; compare |X| with 2**-57
               06    18     0156   228          BGEQ    GEQ_TO_2M57                     ; branch if |X| >= 2**-57
                             0158   229
                             0158   230 ;
                             0158   231 ; |X| < 2**-57
                             0158   232 ;
                             0158   233
      04 BC    08 70FD     0158   234          MOVH    #1, @hcosh(AP)                  ; Store 1.0 as result
                  04       015D   235          RET                                     ; return
                             015E   236
                             015E   237 ;
                             015E   238 ; 2**-57 =< |X| < 0.25
                             015E   239 ;
                             015E   240
                             015E   241 GEQ_TO_2M57:
      50    50 64FD     015E   242          MULH2   R0,R0                           ;Get ARG*2 for POLYH.
FEE7 CF    0D    50 75FD     0162   243          POLYH   R0, #HCOSHLEN-1, HCOSHTAB
                             0169   244                                              ; R0/R3 = SUM(Ci*X**i)
      04 BC    50 70FD     0169   245          MOVH    R0, @hcosh(AP)                  ; Store result in first argument
                  04       016E   246          RET                                     ; return
                             016F   247
                             016F   248 ;
                             016F   249 ; 0.25 =< |X|
                             016F   250 ;
                             016F   251
                             016F   252 GEQ_TO_0.25:
    FE8B CF    50 71FD     016F   253          CMPH    R0, H_16383_LOG_2               ; compare |X| with 16383*ln2
               20    14     0175   254          BGTR    GTR_THAN_16383_LOG_2            ; branch if |X| > 16383*ln2
                             0177   255
                             0177   256 ;
                             0177   257 ; 0.25 =< |X| =< 16383*ln2
                             0177   258 ;
                             0177   259
 00000000'EF     16    0177   260          JSB     MTH$HEXP_R6                     ; R0/R3 = HEXP(|X|)
```

```
                FEBD CF    50 71FD  017D   261          CMPH    R0, H_2_POWER_57        ; Compare HEXP(|X|) with 2*57, if
                           09   14  0183   262          BGTR    ONE_TERM_ONLY           ; larger, only one term is needed.
                54   08    50 67FD  0185   263          DIVH3   R0, #1, R4              ; R4/R7 = HEXP(-|X|)
                      50    54 60FD  018A   264          ADDH2   R4, R0                  ; R0/R3 = HEXP(X) + HEXP(-X)
                                     018E   265 ONE_TERM_ONLY:
                      50    01   A2  018E   266          SUBW2   #^X0001, R0             ; R0/R3 = (HEXP(X) + HEXP(-X))/2
                  04 BC    50 7DFD  0191   267          MOVO    R0, @hcosh(AP)          ; move COSH(x) to first argument
                                04  0196   268          RET                             ; and return
                                     0197   269
                                     0197   270 ;
                                     0197   271 ; 16383*ln2 < |X|
                                     0197   272 ;
                                     0197   273
                                     0197   274 GTR_THAN_16383_LOG_2:
                FE73 CF    50 71FD  0197   275          CMPH    R0, H_16384_LOG_2       ; compare |X| with 16384*ln2
                           1A   18  019D   276          BGEQ    ERROR                   ; branch to ERROR if 16384*ln2 =<|X|
                                     019F   277
                                     019F   278
                                     019F   279 ;
                                     019F   280 ; 16383*ln2 =< |X| < 16384*ln2
                                     019F   281 ;
                                     019F   282
                50    FE7C CF 62FD  019F   283          SUBH2   H_LOG_2_HI, R0          ; R0/R3=|X|-(hi 109 bits of ln2+2*-109)
                 00000000'EF   16  01A5   284          JSB     MTH$HEXP_R6             ; R0/R3=HEXP(|X|-H_LOG_2_HI)
          54    50    FE80 CF 65FD  01AB   285          MULH3   H_LOG_2_LO, R0, R4      ; R4/R7=H_LOG_2_LO*HEXP(|X|-H_LOG_2_LO)
            04 BC    50   54 61FD  01B2   286          ADDH3   R4, R0, @hcosh(AP)      ; Store HEXP(|X|-ln2) in 1st argument
                                04  01B8   287          RET                             ; return
                                     01B9   288
                                     01B9   289 ;
                                     01B9   290 ; 16383*ln2 + LOG(2) =< |X|, error
                                     01B9   291 ;
                                     01B9   292
                  7E    00'8F    9A  01B9   293 ERROR:  MOVZBL  #MTH$K_FLOOVEMAT, -(SP) ; condition value
                50   01    0F   79  01BD   294          ASHQ    #15, #T, R0             ; R0 = result = reserved operand -0.0
                                     01C1   295                                          ; goes to R0/R3 so error handler
                                     01C1   296                                          ; can modify the result.
                           52   7C  01C1   297          CLRQ    R2                      ; ...
00000000'EF    01   FB  01C3   298          CALLS   #1, MTH$$SIGNAL         ; signal error and use real user's PC
                  04 BC    50 7DFD  01CA   299          MOVO    R0, @hcosh(AP)          ; Restore result
                                04  01CF   300          RET                             ; return
                                     01D0   301
                                     01D0   302
                                     01D0   303
                                     01D0   304          .END
```

```
ERROR                           000001B9 R      01
GEQ_TO_0.25                     0000016F R      01
GEQ_TO_2M57                     0000015E R      01
GTR_THAN_16383_LOG_2            00000197 R      01
HCOSH                         = 00000004
HCOSHLEN                      = 0000000E
HCOSHTAB                        00000050 R      01
H_16383_LOG_2                   00000000 R      01
H_16384_LOG_2                   00000010 R      01
H_2_POWER_57                    00000040 R      01
H_LOG_2_HI                      00000020 R      01
H_LOG_2_LO                      00000030 R      01
LONG                          = 00000004
MTH$$JACKET_HND                 ********    X   01
MTH$$SIGNAL                     ********    X   00
MTH$HCOSH                       00000130 RG     01
MTH$HEXP_R6                     ********    X   00
MTH$K_FLOOVEMAT                 ********    X   00
ONE_TERM_ONLY                   0000018E R      01
X                             = 00000008
```

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	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	32	00:00:00.07	00:00:01.11
Command processing	126	00:00:00.68	00:00:03.06
Pass 1	91	00:00:01.06	00:00:04.74
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	66	00:00:00.78	00:00:02.77
Symbol table output	4	00:00:00.02	00:00:00.02
Psect synopsis output	3	00:00:00.02	00:00:00.12
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	324	00:00:02.64	00:00:11.85

The working set limit was 900 pages.
4622 bytes (10 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.
364 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
_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$:MTHHCOSH/OBJ=OBJ$:MTHHCOSH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```