F 13

```
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     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        HHHHHHHHHHHHHHH  RRRRRRRRRRRR          TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR          TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  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        LLLLLLLLLLLLLLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**MTHDSINH

```
MM      MM  TTTTTTTTTT  HH      HH  DDDDDDDD      SSSSSSSS    IIIIII    NN      NN  HH      HH
MM      MM  TTTTTTTTTT  HH      HH  DDDDDDDD      SSSSSSSS    IIIIII    NN      NN  HH      HH
MMMM  MMMM      TT      HH      HH  DD      DD  SS              II      NN      NN  HH      HH
MMMM  MMMM      TT      HH      HH  DD      DD  SS              II      NN      NN  HH      HH
MM  MM  MM      TT      HH      HH  DD      DD  SS              II      NNNN    NN  HH      HH
MM  MM  MM      TT      HH      HH  DD      DD  SS              II      NNNN    NN  HH      HH
MM      MM      TT      HHHHHHHHHH  DD      DD    SSSSSS        II      NN  NN  NN  HHHHHHHHHH
MM      MM      TT      HHHHHHHHHH  DD      DD    SSSSSS        II      NN  NN  NN  HHHHHHHHHH
MM      MM      TT      HH      HH  DD      DD          SS      II      NN    NNNN  HH      HH
MM      MM      TT      HH      HH  DD      DD          SS      II      NN    NNNN  HH      HH
MM      MM      TT      HH      HH  DD      DD          SS      II      NN      NN  HH      HH       ....
MM      MM      TT      HH      HH  DD      DD          SS      II      NN      NN  HH      HH       ....
MM      MM      TT      HH      HH  DDDDDDDD    SSSSSSSS      IIIIII    NN      NN  HH      HH       ....
MM      MM      TT      HH      HH  DDDDDDDD    SSSSSSSS      IIIIII    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
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000          .TITLE  MTH$DSINH        ; Double Floating Hyperbolic Sine routine
0000                                   ; (DSINH)
0000          .IDENT /1-008/           ; File: MTHDSINH.MAR  Edit: RNH1008
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
000C    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;
0000    ; FACILITY: MATH LIBRARY
0000    ;++
0000    ; ABSTRACT:
0000    ;
0000    ; MTH$DSINH is a function which returns the double floating hyperbolic sine
0000    ; of its double precision floating point argument. The call is standard
0000    ; call-by-reference.
0000    ;
0000    ;--
0000    ;
0000    ; VERSION: 01
0000    ;
0000    ; HISTORY:
0000    ; AUTHOR:
0000    ;       Peter Yuo, 29-Jun-77: Version 01
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;
0000    ;
```

```
0000          .SBTTL  HISTORY ; Detailed Current Edit History
0000
0000
0000    ; ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000    ;
0000    ; Edit History for Version 01 of MTH$DSINH
0000    ;
0000    ; 0-2   MTH$$ERROR changed to MTH$$SIGNAL.
0000    ;       MTH$_.... changed to MTH__....
0000    ;       Changed error handling mechanism. Put error result in R0:R1 before
0000    ;       calling MTH$$SIGNAL in order to allow user modify error result.
0000    ;
0000    ; 0-3   Six term Taylor series, in powers of argument, replaced
0000    ;       by six term Chebyshev series, in powers of ARG**2,
0000    ;       with overhang provided. This improves accuracy. 18-May-1978;
0000    ;       Mary Payne.
0000    ; 0-4   Fix LOG(2) constant.  TNH 16-June-78
0000    ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000    ; 1-002 - Change MTH_FLOOVEMAT to MTH$K_FLOOVEMAT.  JBS 07-DEC-78
0000    ; 1-003 - Remove $SRMDEF - not needed.  JBS 16-DEC-78
0000    ; 1-004 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000    ; 1-005 - Declare externals.  SBL 17-May-1979
0000    ; 1-006 - Use MTH$DEXP_R6.  SBL 27-Sept-1979
0000    ; 1-007 - Changed lower limit for Chebyshev approximation from 2**-27 to
0000    ;         2**-28.
0000    ;       - Eliminated second call to EXP for input values between 28.5*ln2
0000    ;         and 127*ln2.
0000    ;       - Changed all final floating point divisions by 2 to interger
0000    ;         subrtracts of 1 from the exponent field.
0000    ;       - Extended maximum range from 87.69 to 128*ln2=88.72.
0000    ;       - Chaneged logic for computing EXP(|x|-ln2) to reduce error.
0000    ;       - RNH 10-FEB-81
0000    ; 1-008 - Change W^ to G^ on call to MTH$$SIGNAL RNH 09-Sept-1981
```

```
                       0000          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                       0000
                       0000   ;
                       0000   ; INCLUDE FILES:        MTHJACKET.MAR
                       0000   ;
                       0000
                       0000   ;
                       0000   ; EXTERNAL SYMBOLS:
                       0000   ;
                       0000          .DSABL  GBL
                       0000          .EXTRN  MTH$DEXP_R6
                       0000          .EXTRN  MTH$K_FLOOVEMAT
                       0000          .EXTRN  MTH$$SIGNAL
                       0000
                       0000   ;
                       0000   ; EQUATED SYMBOLS:
                       0000
          00004080     0000          SD_1.0  = ^F1.0                    ; 1.0
          00000004     0000          value   = 4                        ; value.rd.r
                       0000
                       0000   ;
                       0000   ; MACROS:        none
                       0000   ;
                       0000   ;
                       0000   ; PSECT DECLARATIONS:
                       0000
                   00000000          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                       0000                                         ; program section for math routines
                       0000   ;
                       0000   ; OWN STORAGE:  none
                       0000   ;
                       0000   ;
                       0000   ; CONSTANTS:
                       0000   ;
                       0000
                       0000   D_127_LOG_2:
 2BDAC7E2 0F3343B0     0000          .QUAD   ^X2BDAC7E20F3343B0         ; 127*ln2
                       0008   D_128_LOG_2:
 CF78F7D1 721743B1     0008          .QUAD   ^XCF78F7D1721743B1         ; 128*ln2-2**-49
                       0010   D_28.5_LOG_2:
 DCC858B6 0990429E     0010          .QUAD   ^XDCC858B60990429E         ; 28.5*ln2
                       0018   D_LOG_2_HI:
 CF80F7D1 72174031     0018          .QUAD   ^XCF80F7D172174031         ; (high 49 bits of ln2)+2**-49
                       0020   D_LOG_2_LO:
 FF81898C 86C3A5CA     0020          .QUAD   ^XFF81898C86C3A5CA         ; ln2 - D_LOG_2_HI
                       0028
                       0028   DSINHTAB:
                       0028
         746B 33D7     0028          .WORD   ^0031727,^0072153
         760F A336     002C          .WORD   ^0121466,^0073017          ; DECIMAL:   0.2508223608819151D-07
         EF13 3738     0030          .WORD   ^0033470,^0167423
         64E0 D935     0034          .WORD   ^0154465,^0062340          ; DECIMAL: 0.2755729803646086D-05
         0D00 3A50     0038          .WORD   ^0035120,^0006400
         43A1 D142     003C          .WORD   ^0150502,^0041641          ; DECIMAL: 0.1984126984813681D-03
         8888 3D08     0040          .WORD   ^0036410,^0104210
         766A 8888     0044          .WORD   ^0104210,^0073152          ; DECIMAL: 0.8333333333332327D-02
         AAAA 3F2A     0048          .WORD   ^0037452,^0125252
```

```
                         AAAD AAAA  004C    141           .WORD    ^O125252,^O125255        ; DECIMAL: 0.1666666666666667D+00
                         0000 0000  0050    142           .WORD    0,0
                         0000 0000  0054    143           .WORD    0,0                      ; DECIMAL: 0.D0
                                    0058    144
                          00000006  0058    145 DSINHLEN          = .- DSINHTAB/8
                                    0058    146
                          0058            .SBTTL  MTH$DSINH  - Standard Double Precision Floating DSINH
                          0058
                          0058
                          0058     ;++
                          0058     ; FUNCTIONAL DESCRIPTION:
                          0058     ;
                          0058     ; DSINH  - double precision floating point function
                          0058     ;
                          0058     ; DSINH(X) is computed as:
                          0058     ;
                          0058     ;        If !X! < 2**-28, DSINH(X) = X.
                          0058     ;        If 2**-28 =< !X! < 0.25, DSINH(X) = Chebyshev Series.
                          0058     ;        If 0.25 =< !X! < 28.5*ln2, DSINH(X) = (DEXP(X) - DEXP(-X))/2.
                          0058     ;        If 28.5 =< !X! < 127*ln2, DSINH(X) = sign(X`*EXP(!X!)/2.
                          0058     ;        If 127*ln2 =< !X! < 128*ln2, then DSINH(X) = sign(X)* DEXP( X -ln2).
                          0058     ;        If 128*ln2 < !X!, then overflow.
                          0058     ;
                          0058     ; CALLING SEQUENCE:
                          0058     ;
                          0058     ;        DSINH.wd.v = MTH$DSINH(x.rd.r)
                          0058     ;
                          0058     ; INPUT PARAMETERS:
                          0058
              00000004    0058            LONG = 4                        ; define longword multiplier
              00000004    0058            x = 1 * LONG                    ; Contents of x is the argument
                          0058
                          0058     ; IMPLICIT INPUTS:      none
                          0058     ;
                          0058     ; OUTPUT PARAMETERS:
                          0058     ;
                          0058     ;        VALUE:  double precision floating hyperbolic sine of the argument
                          0058     ;
                          0058     ; IMPLICIT OUTPUTS:     none
                          0058     ;
                          0058     ; COMPLETION CODES:     none
                          0058     ;
                          0058     ; SIDE EFFECTS:
                          0058     ;
                          0058     ; Signal: MTH$_FLOOVEMAT if 128*ln2 =< !X! with reserved operand in R0/R1
                          0058     ; (copied to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                          0058     ; Associated message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved
                          0058     ; operand -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R
                          0058     ;
                          0058     ; NOTE: This procedure disables floating point underflow, enables integer
                          0058     ; overflow.
                          0058     ;
                          0058     ;---
                          0058
                          0058
                  40FC    0058            .ENTRY  MTH$DSINH, ^M<IV, R2, R3, R4, R5, R6, R7>
                          005A                                            ; standard call-by-reference entry
                          005A                                            ; disable DV (and FU), enable IV
                          005A            MTH$FLAG_JACKET                 ; flag that this is a jacket procedure in
                          005A
   6D    00000000'GF   9E 005A            MOVAB   G^MTH$$JACKET_HND, (FP)
                          0061                                            ; set handler address to jacket
                          0061                                            ; handler
                          0061
                          0061                                                   ; case of an error in routine
                          0061                                                   ; If an error, convert signal to user PC
                          0061                                                   ; and resignal
    56   04 BC       70   0061               MOVD     @value(AP), R6             ; R6/R7 = |X| = @value(AP)
         50    56    7D   0065               MOVQ     R6, R0
    50   8000 8F     AA   0068               BICW2    #^X8000, R0                ; R0/R1 = |X|
   3F80 8F     50    B1   006D               CMPW     R0, #^X3F80                ; compare |X| with 0.25
               1A    18   0072               BGEQ     GEQ_TO_0.25                ; branch if |X| >= 0.25
                          0074
                          0074      ;
                          0074      ; |X| < 0.25
                          0074      ;
                          0074
   3280 8F     50    B1   0074               CMPW     R0, #^X3280                ; compare |X| with 2**-28
               04    18   0079               BGEQ     GEQ_TO_2M28                ; branch if |X| >= 2**-28
                          007B
                          007B      ;
                          007B      ; |X| < 2**-28
                          007B      ;
                          007B
         50    56    7D   007B               MOVQ     R6, R0                     ; R0/R1 = X
                     04   007E               RET                                 ; return with result = argument
                          007F
                          007F      ;
                          007F      ; 2**-28 =< |X| < 0.25
                          007F      ;
                          007F
                          007F      GEQ_TO_2M28:
         50    50    64   007F               MULD     R0,R0                      ; Get ARG**2 for POLYD
A1 AF    05    50    75   0082               POLYD    R0, #DSINHLEN-1, DSINHTAB
                          0087                                                   ; R0/R1 = SUM(Ci*X**i), with
                          0087                                                   ; Last coefficient zero
         50    56    64   0087               MULD     R6,R0                      ; MULD by ARG, and then
         50    56    60   008A               ADDD     R6,R0                      ; Add in ARG with overhang.
                     04   008D               RET                                 ; return with result in R0/R1
                          008E
                          008E      ;
                          008E      ; 0.25 =< |X|
                          008E      ;
                          008E
                          008E      GEQ_TO_0.25:
   FF6D CF     50    71   008E               CMPD     R0, D_127_LOG_2            ; compare |X| with 127*ln2
               3B    14   0093               BGTR     GTR_THAN_127_LOG_2         ; branch if |X| > 127*ln2
                          0095
                          0095      ;
                          0095      ; 0.25 =< |X| < 127*ln2
                          0095      ;
                          0095
   FF76 CF     50    71   0095               CMPD     R0, D_28.5_LOG_2           ; Compare |X| with 28.5*ln2, if
               20    14   009A               BGTR     ONLY_ONE_TERM              ; greater, only one of EXP(X) and
                          009C                                                   ; EXP(-X) is significant.
                          009C      ;
                          009C      ; 0.25 =< |X| < 28.5*ln2
                          009C      ;
         50    56    7D   009C               MOVQ     R6, R0                     ; R0/R1 = X
   00000000'EF       16   009F               JSB      MTH$DEXP_R6                ; R0/R1 = DEXP(X)
```

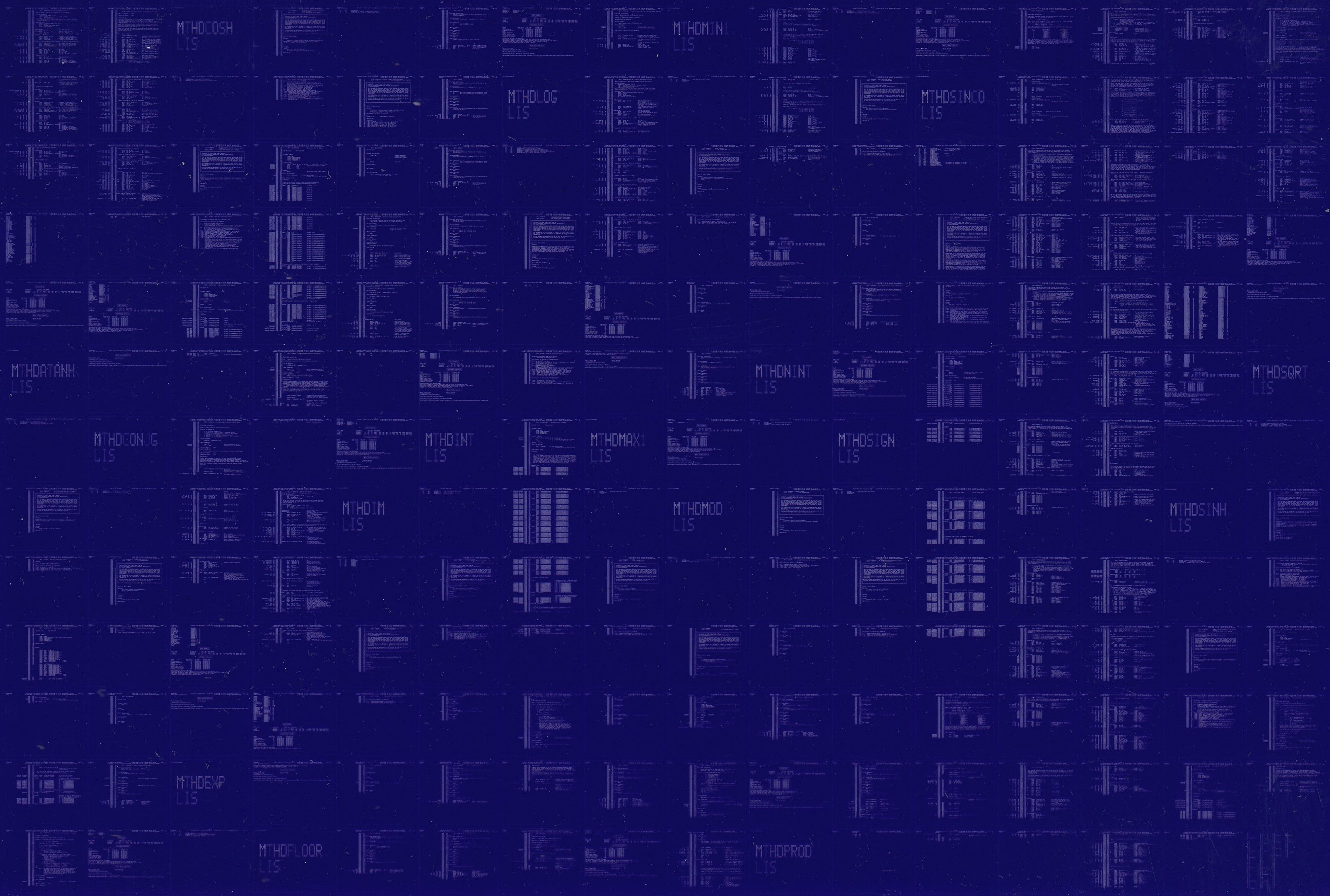
```
      7E    50    7D  00A5            MOVQ    R0, -(SP)                ; push DEXP(X) on stack
   50   04 BC     72  00A8            MNEGD   @value(AP), R0           ; R0/R1 = -X
   00000000'EF    16  00AC            JSB     MTH$DEXP_R6              ; R0/R1 = DEXP(-X)
  50   8E   50    63  00B2            SUBD3   R0, (SP)+, R0            ; R0/R1 = DEXP(X) - DEXP(-X)
  50   0080 8F    A2  00B6            SUBW    #^X0080, R0              ; R0/R1 = (DEXP(X)-DEXP(-X))/2
                  04  00BB            RET                              ; return with result in R0/R1
                      00BC
                      00BC  ;
                      00BC  ; 28.5*ln2 =< !X! <127*ln2
                      00BC  ;
                      00BC
                      00BC  ONLY_ONE_TERM:
   00000000'EF    16  00BC            JSB     MTH$DEXP_R6              ; R0/R1 = DEXP(!X!)
         04 BC    73  00C2            TSTD    @value(AP)               ; Check sign of X
            03    14  00C5            BGTR    POSITIVE                 ; If negative change sign of
      50    50    72  00C7            MNEGD   R0, R0                   ; DEXP(!X!)
                      00CA  POSITIVE:
  50   0080 8F    A2  00CA            SUBW    #^X0080, R0              ; R0/R1 = sign(X)*DEXP(!X!)/2
                  04  00CF            RET
                      00D0  ;
                      00D0  ; 127*ln2 < .X!
                      00D0  ;
                      00D0
                      00D0  GTR_THAN_127_LOG_2:
  FF33 CF   50    71  00D0            CMPD    R0, D_128_LOG_2          ; Compare !X' to 128*ln2
            1D    18  00D5            BGEQ    ERROR                    ; if 128*ln2 =<.X!, overflow occurs
                      00D7
                      00D7
                      00D7  ;
                      00D7  ; 127*ln2 < !X! < 128*ln2
                      00D7  ;
                      00D7
  50   FF3D CF    62  00D7            SUBD    D_LOG_2_HI, R0           ; R0/R1=!X!-D_LOG_2_HI
   00000000'EF    16  00DC            JSB     MTH$DEXP_R6              ; R0/R1 = DEXP(!X!-D_LOG_2_HI)
52   50  FF3A CF  65  00E2            MULD3   D_LOG_2_LO, R0, R2       ; R2/R3=DEXP(!X!-D_LOG_2_HI)*D_LOG_2_LO
      50    52    62  00E8            SUBD    R2, R0                   ; R0/R1=DEXP(!X!-ln2)
         04 BC    73  00EB            TSTD    @value(AP)               ; test the sign of X
            03    18  00EE            BGEQ    10$                      ; branch if X >= 0
      50    50    72  00F0            MNEGD   R0, R0                   ; R0/R1 = sign(X) * DEXP(!X!-ln2)
                  04  00F3  10$:      RET                              ; return with result in R0/R1
                      00F4
                      00F4  ;
                      00F4  ; 128*ln2 =< !X!, error
                      00F4  ;
                      00F4
     7E   00'8F   9A  00F4  ERROR:    MOVZBL  #MTH$K_FLOOVEMAT, -(SP)  ; condition value
  50   01   0F    79  00F8            ASHQ    #15, #1, R0              ; R0/R1 = result = reserved operand -0.0
                      00FC                                             ; goes to signal mechanism vector
                      00FC                                             ; (CHF$L_MCH_R0/R1) so error handler
                      00FC                                             ; can modify the result.
00000000'GF   01  FB  00FC            CALLS   #1, G^MTH$$SIGNAL        ; signal error and use real user's PC
                      0103                                             ; independent of CALL vs JSB
                  04  0103            RET                              ; return - R0/R1 restored from CHF$L_MCH_R0/
                      0104
                      0104
                      0104
                      0104            .END
```

```
DSINHLEN                        = 00000006
DSINHTAB                          00000028 R      01
D_127_LOG_2                       00000000 R      01
D_128_LOG_2                       00000008 R      01
D_28.5_LOG_2                      00000010 R      01
D_LOG_2_HI                        00000018 R      01
D_LOG_2_LO                        00000020 R      01
ERROR                             000000F4 R      01
GEQ_TO_0.25                       0000008E R      01
GEQ_TO_2M28                       0000007F R      01
GTR_THAN_127_LOG_2                000000D0 R      01
LONG                            = 00000004
MTH$$JACKET_HND                   ********    X   01
MTH$$SIGNAL                       ********    X   00
MTH$DEXP_R6                       ********    X   00
MTH$DSINH                         00000058 RG     01
MTH$K_FLOOVEMAT                   ********    X   00
ONLY_ONE_TERM                     000000BC R      01
POSITIVE                          000000CA 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                         00000104  (  260.)  01 (  1.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                       35    00:00:00.12    00:00:01.77
Command processing                  120    00:00:00.67    00:00:05.67
Pass 1                               88    00:00:00.90    00:00:05.83
Symbol table sort                     0    00:00:00.01    00:00:00.01
Pass 2                               68    00:00:00.71    00:00:03.50
Symbol table output                   4    00:00:00.03    00:00:00.03
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                319    00:00:02.46    00:00:16.83

The working set limit was 900 pages.
4682 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 22 non-local and 1 local symbols.
373 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
------------------                           --------------
_$255$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$:MTHDSINH/OBJ=OBJ$:MTHDSINH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```

0259 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


MTHDCOSH
LIS

MTHDMIN1
LIS

MTHDLOG
LIS

MTHDSINCO
LIS

MTHDATANH
LIS

MTHDNINT
LIS

MTHDSQRT
LIS

MTHDCONJG
LIS

MTHDINT
LIS

MTHDMAX1
LIS

MTHDSIGN
LIS

MTHDIM
LIS

MTHDMOD
LIS

MTHDSINH
LIS

MTHDEXP
LIS

MTHDFLOOR
LIS

MTHDPROD
LIS