```
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
```

```
**FILE**ID**MTHHTAN                                   G 12
```

```
MM      MM  TTTTTTTTTT  HH      HH  HH      HH  TTTTTTTTTT    AAAAAA    NN      NN
MM      MM  TTTTTTTTTT  HH      HH  HH      HH  TTTTTTTTTT    AAAAAA    NN      NN
MMMM  MMMM      TT      HH      HH  HH      HH      TT      AA      AA  NN      NN
MMMM  MMMM      TT      HH      HH  HH      HH      TT      AA      AA  NN      NN
MM  MM  MM      TT      HH      HH  HH      HH      TT      AA      AA  NNNN    NN
MM  MM  MM      TT      HH      HH  HH      HH      TT      AA      AA  NNNN    NN
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH      TT      AA      AA  NN  NN  NN
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH      TT      AA      AA  NN  NN  NN
MM      MM      TT      HH      HH  HH      HH      TT      AAAAAAAAAA  NN    NNNN
MM      MM      TT      HH      HH  HH      HH      TT      AAAAAAAAAA  NN    NNNN
MM      MM      TT      HH      HH  HH      HH      TT      AA      AA  NN      NN      ....
MM      MM      TT      HH      HH  HH      HH      TT      AA      AA  NN      NN      ....
MM      MM      TT      HH      HH  HH      HH      TT      AA      AA  NN      NN      ....
MM      MM      TT      HH      HH  HH      HH      TT      AA      AA  NN      NN      ....

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
```

MT
1-

Table of contents

```
MT
Sy

CO
HT
HT
H_
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
SF
UN
X

PS
--

$A
_M

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
84
Th
52
9
```

```
0000         .TITLE  MTH$HTAN         ; H Floating Point Tangent routine
0000                                  ; (HTAN, HTAND)
0000         .IDENT /1-006/           ; File: MTHHTAN.MAR  EDIT:  RNH1006
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  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$HTAN is a function  which  returns the H floating point tangent
0000    ; of its H floating point radian argument. The call is standard
0000    ; call-by-reference.  It JSB to MTH$HTAN_R7.
0000    ;
0000    ; MTH$HTAND is a function  which  returns the H floating point tangent
0000    ; of its H floating point degree argument. The call is standard
0000    ; call-by-reference.  It JSB to MTH$HTAND_R7.
0000    ;
0000    ;--
0000    ;
0000    ; VERSION: 1
0000    ;
0000    ; HISTORY:
0000    ; AUTHOR:
0000    ;       John A. Wheeler, 15-Oct-1979: Version 1
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;
0000    ;
```

```
0000            .SBTTL  HISTORY ; Detailed Current Edit History
0000
0000
0000     ; Edit History for Version 1 of MTH$HTAN
0000     ;
0000     ; 1-001 - Adapted from MTH$GTAN version 1-001.  JAW 15-Oct-1979
0000     ; 1-002 - Call MTH$SIGNAL with user PC arg for JSB.  SBL 31-Oct-1979
0000     ; 1-003 - Added degree entry points. RNH 8-MAR-1981
0000     ; 1-004 - Added MTH$HTAN_R7, and MTH$HTAND_R7.  RNH 27-AUG-81.
0000     ; 1-005 - Change shared external references to G^ RNH 25-Sep-81
0000     ; 1-006 - Change remaining external references to G^.  RNH 06-Oct-81
```

```
                      0000            .SBTTL  DECLARATIONS    ; Declarative Part of Module
                      0000
                      0000    ;
                      0000    ; INCLUDE FILES:          none
                      0000    ;
                      0000    ; EXTERNAL SYMBOLS:
                      0000            .DSABL  GBL                     ; Prevent undefineds from becoming
                      0000                                            ; Global
                      0000            .EXTRN  MTH$HSIN_R5     ; H Floating sine routine (radians)
                      0000            .EXTRN  MTH$HCOS_R5     ; H Floating cosine routine (radians)
                      0000            .EXTRN  MTH$HSINCOS_R7  ; H Floating sine and cosine routine (radians)
                      0000            .EXTRN  MTH$HSINCOSD_R7 ; H Floating sine and cosine routine (degrees)
                      0000            .EXTRN  MTH$$SIGNAL     ; Math error signal routine
                      0000            .EXTRN  MTH$$SIGNAL_CON ; Handler that just returns
                      0000            .EXTRN  MTH$K_FLOOVEMAT ; Error code
                      0000            .EXTRN  MTH$K_FLOUNDMAT ; Error code
                      0000            .EXTRN  MTH$$JACKET_TST ;
                      0000            .EXTRN  MTH$HSIND_R5    ; H Floating sine routine (degrees)
                      0000            .EXTRN  MTH$HCOSD_R5    ; H Floating cosine routine (degrees)
                      0000
                      0000    ;
                      0000    ; EQUATED SYMBOLS:        none
                      0000    ;
                      0000    ; MACROS:
                      0000            $SFDEF                          ; Define SF (Stack Frame) symbols
                      0000    ;
                      0000    ;
                      0000    ; PSECT DECLARATIONS:
                      0000
                  00000000            .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                      0000                                            ; Program section for math routines
                      0000    ;
                      0000    ; OWN STORAGE:  none
                      0000    ;
                      0000    ; CONSTANTS:
             00000004 0000            HTAN = 4                                ; Position of output parameter from AP
             00000004 0000            HTAND = 4                               ; Position of output parameter from AP
             00000008 0000            X = 8                                   ; Position of input parameter from AP
                      0000
                      0000    H_SMALLEST_DEG:
    3C1FC1A6 CA5D0006 0000            .LONG   ^XCA5D0006, ^X3C1FC1A6          ; 180/pi*2**-16384
    81A5A6FE 152E7B86 0008            .LONG   ^X152E7B86, ^X81A5A6FE
                      0010
```

```
                             0010   110          .SBTTL  MTH$HTAN  - Standard H Floating HTAN
                             0010   111
                             0010   112
                             0010   113 ;++
                             0010   114 ; FUNCTIONAL DESCRIPTION:
                             0010   115 ;
                             0010   116 ; HTAN  - H floating point tangent function
                             0010   117 ;
                             0010   118 ;        For algorithm, see MTH$HTAN_R7
                             0010   119 ;
                             0010   120 ; CALLING SEQUENCE:
                             0010   121 ;
                             0010   122 ;        CALL MTH$HTAN(HTAN.wh.r, X.rh.r)
                             0010   123 ;
                             0010   124 ;
                             0010   125 ; INPUT PARAMETERS:
                             0010   126 ;
                             0010   127 ;        X.rh.r                          Address of value of angle in radians.
                             0010   128 ;
                             0010   129 ; IMPLICIT INPUTS:      none
                             0010   130 ;
                             0010   131 ; OUTPUT PARAMETERS:
                             0010   132 ;
                             0010   133 ;
                             0010   134 ;        VALUE:  H floating tangent of the argument.
                             0010   135 ;                Output parameter is the first parameter from the left.
                             0010   136 ;
                             0010   137 ; IMPLICIT OUTPUTS:     none
                             0010   138 ;
                             0010   139 ; COMPLETION CODES:     none
                             0010   140 ;
                             0010   141 ; SIDE EFFECTS:
                             0010   142 ;
                             0010   143 ;        NONE
                             0010   144 ;
                             0010   145 ;---
                             0010   146
                             0010   147
                       40FC  0010   148          .ENTRY  MTH$HTAN, ^M<IV, R2, R3, R4, R5, R6, R7>
                             0012   149                                          ; Standard call-by-reference entry
                             0012   150                                          ; Disable DV (and FU), enable IV
                             0012   151          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                                                ; handler
                             0019
                             0019   152                                          ; case of an error in special routine
        50    08 BC 70FD     0019   153          MOVH    @X(AP), R0              ; R0/R3 = argument
                  06    10   001E   154          BSBB    MTH$HTAN_R7             ; Call special HTAN routine
        04 BC    50 7DFD     0020   155          MOVO    R0, @HTAN(AP)           ; Store result in second argument
                        04   0025   156          RET                             ; Return to caller
                        0026            .SBTTL  MTH$HTAN_R7 - JSB entry point
                        0026      ;
                        0026      ;++
                        0026      ; FUNCTIONAL DESCRIPTION:
                        0026      ;
                        0026      ; HTAN - JSB entry point
                        0026      ;
                        0026      ; Algorithmic steps:
                        0026      ;  1. Compute HSIN and HCOS.
                        0026      ;  2. If HCOS is zero, we have an error.
                        0026      ;  3. Return HSIN / HCOS.
                        0026      ;
                        0026      ; CALLING SEQUENCE:
                        0026      ;
                        0026      ;       MOVH    argument, R0
                        0026      ;       JSB     MTH$HTAN_R7
                        0026      ;
                        0026      ; INPUT PARAMETERS:
                        0026      ;
                        0026      ;       R0 / R3 contains x
                        0026      ;
                        0026      ; IMPLICIT INPUTS:
                        0026      ;
                        0026      ;       NONE
                        0026      ;
                        0026      ; OUTPUT PARAMETERS:
                        0026      ;
                        0026      ;       The result is the H-floating tangent of x.
                        0026      ;
                        0026      ; IMPLICIT OUTPUTS:
                        0026      ;
                        0026      ;       NONE
                        0026      ;
                        0026      ; SIDE EFFECTS:
                        0026      ;
                        0026      ;       NONE
                        0026      ;--
                        0026      MTH$HTAN_R7::
00000000'GF    16       0026            JSB     G^MTH$HSINCOS_R7        ; Compute HSIN, and HCOS of X
        54 73FD         002C            TSTH    R4                      ; Is HCOS zero ?
        05   13         002F            BEQL    30$                     ; If zero, HTAN is infinite
   50   54 66FD         0031            DIVH2   R4, R0                  ; Compute HSIN / HCOS
             05         0035            RSB                             ; Return to caller
                        0036      ;
                        0036      ; HCOS is zero, so HTAN is infinite.  Go to common error code.
                        0036      ;
                        0036      30$:
      00DF   31         0036            BRW     COSZER
```

```
                     0039           .SBTTL  MTH$HTAN_R5 - JSB entry point
                     0039   ;
                     0039   ;++
                     0039   ; FUNCTIONAL DESCRIPTION:
                     0039   ;
                     0039   ; HTAN - JSB entry point
                     0039   ;
                     0039   ; Algorithmic steps:
                     0039   ;  1. Compute HSIN and HCOS.
                     0039   ;  2. If HCOS is zero, we have an error.
                     0039   ;  3. Return HSIN / HCOS.
                     0039   ;
                     0039   ; CALLING SEQUENCE:
                     0039   ;
                     0039   ;       MOVH    argument, R0
                     0039   ;       JSB     MTH$HTAN_R5
                     0039   ;
                     0039   ; INPUT PARAMETERS:
                     0039   ;
                     0039   ;       R0 / R3 contains x
                     0039   ;
                     0039   ; IMPLICIT INPUTS:
                     0039   ;
                     0039   ;       NONE
                     0039   ;
                     0039   ; OUTPUT PARAMETERS:
                     0039   ;
                     0039   ;       The result is the H-floating tangent of x.
                     0039   ;
                     0039   ; IMPLICIT OUTPUTS:
                     0039   ;
                     0039   ;       NONE
                     0039   ;
                     0039   ; SIDE EFFECTS:
                     0039   ;
                     0039   ;       NONE
                     0039   ;--
                     0039   MTH$HTAN_R5::
       7E   50 70FD  0039           MOVH    R0, -(SP)           ; Save argument
00000000'EF      16  003D           JSB     MTH$HCOS_R5         ; Compute HCOS
       7E   50 70FD  0043           MOVH    R0, -(SP)           ; Save HCOS
            13   13  0047           BEQL    20$                 ; If zero, HTAN is infinite
     50  10 AE 7DFD  0049           MOVO    16(SP), R0          ; Get argument back
00000000'GF      16  004E           JSB     G^MTH$HSIN_R5       ; Compute HSIN
       50   8E 66FD  0054           DIVH2   (SP)+, R0           ; Compute HSIN / HCOS
       5E   10   C0  0058           ADDL2   #16, SP             ; Discard saved argument
                 05  005B           RSB                         ; Return to caller
                     005C   ;+
                     005C   ; Come here if HCOS is zero.  This means that HTAN is infinite.
                     005C   ;-
                     005C   20$:
       5E   20   C0  005C           ADDL2   #32, SP             ; Discard saved HCOS and saved argument
          00B6   31  005F           BRW     COSZER              ; Go to common error code
                     0062
```

```
                        0062         .SBTTL  MTH$HTAND  - Standard H Floating HTAND
                        0062
                        0062
                        0062   ;++
                        0062   ; FUNCTIONAL DESCRIPTION:
                        0062   ;
                        0062   ; HTAND  - H floating point tangent function
                        0062   ;
                        0062   ;          For algorithm, see MTH$HTAND_R7
                        0062   ;
                        0062   ; CALLING SEQUENCE:
                        0062   ;
                        0062   ;       CALL MTH$HTAND(HTAND.wh.r, X.rh.r)
                        0062   ;
                        0062   ;
                        0062   ; INPUT PARAMETERS:
                        0062   ;
                        0062   ;       X.rh.r                          address of value o' angle in degrees.
                        0062   ;
                        0062   ; IMPLICIT INPUTS:      none
                        0062   ;
                        0062   ; OUTPUT PARAMETERS:
                        0062   ;
                        0062   ;
                        0062   ;       VALUE:  H floating tangent of the argument.
                        0062   ;               Output parameter is the first argument from the left.
                        0062   ;
                        0062   ; IMPLICIT OUTPUTS:     none
                        0062   ;
                        0062   ; COMPLETION CODES:     none
                        0062   ;
                        0062   ; SIDE EFFECTS:
                        0062   ;
                        0062   ;       NONE
                        0062   ;---
                        0062
                        0062
                  40FC  0062         .ENTRY  MTH$HTAND, ^M<IV, R2, R3, R4, R5, R6, R7>
                        0064                                         ; Standard call-by-reference entry
                        0064                                         ; Disable DV (and FU), enable IV
                        0064         MTH$FLAG_JACKET                 ; Flag that this is a jacket procedure in
                        0064
    6D   00000000'GF   9E  0064         MOVAB   G^MTH$$JACKET_HND, (FP)
                        006B                                         ; set handler address to jacket
                        006B                                         ; handler
                        006B
                        006B                                         ; case of an error in special routine
       50   08 BC 70FD  006B         MOVH    @X(AP), R0              ; R0/R3 = argument
               06   10  0070         BSBB    MTH$HTAND_R7            ; Call special HTAND routine
       04 BC   50 7DFD  0072         MOVO    R0, @HTAND(AP)          ; Store result in second argument
                    04  0077         RET                             ; Return to caller
```

```
                              0078          .SBTTL  MTH$HTAND_R7 - JSB entry point
                              0078     ;
                              0078     ;++
                              0078     ; FUNCTIONAL DESCRIPTION:
                              0078     ;
                              0078     ; HTAND - JSB entry point
                              0078     ;
                              0078     ; Algorithmic steps:
                              0078     ;  1. Make sure that the absolute value of the argument is greater than
                              0078     ;     180/pi*2**-16384 to avoid underflow in HSIND.
                              0078     ;  2. Compute HSIND and HCOSD.
                              0078     ;  3. If HCOSD is zero, we have an error.
                              0078     ;  4. Return HSIND / HCOSD.
                              0078     ;
                              0078     ; CALLING SEQUENCE:
                              0078     ;
                              0078     ;       MOVH    argument, R0
                              0078     ;       JSB     MTH$HTAND_R7
                              0078     ;
                              0078     ; INPUT PARAMETERS
                              0078     ;
                              0078     ;       R0 / R3 contains x
                              0078     ;
                              0078     ; IMPLICIT INPUTS:
                              0078     ;
                              0078     ;       NONE
                              0078     ;
                              0078     ; OUTPUT PARAMETERS:
                              0078     ;
                              0078     ;       The result is the H-floating tangent of x.
                              0078     ;
                              0078     ; IMPLICIT OUTPUTS:
                              0078     ;
                              0078     ;       NONE
                              0078     ;
                              0078     ; SIDE EFFECTS:
                              0078     ;
                              0078     ;       NONE
                              0078     ;--
                              0078 MTH$HTAND_R7::
         7E    50 70FD        0078          MOVH    R0, -(SP)               ; Save argument
   6E    8000 8F    AA        007C          BICW    #^X8000, (SP)           ; (SP) = |argument|
         6E    07    B1       0081          CMPW    #7, (SP)                ; Compare |ARG| with 2**-16377
               14    15       0084          BLEQ    20$                     ; No possible underflow compute HTAND.
   6E    FF75 CF 71FD         0086          CMPH    H_SMALLEST_DEG, (SP)    ; Possible underflow, use better check
               0C    15       008C          BLEQ    20$                     ; No underflow.
               6E 73FD        008E          TSTH    (SP)                    ; If |arg| = 0, no underflow, otherwise
               5E    12       0091          BNEQ    UNFL                    ; HSIND will underflow
         5E    10    C0       0093          ADDL    #16, SP                 ; Remove argument from the stack
               50 7CFD        0096          CLRH    R0                      ; Zero the result
                     05       0099          RSB                             ; Return with value = 0
                              009A 20$:
         5E    10    C0       009A          ADDL2   #16, SP                 ; Discard saved argument
   00000000'GF       16       009D          JSB     G^MTH$HSINCOSD_R7               ; Compute HCOSD
               54 73FD        00A3          TSTH    R4                      ; Is HCOSD zero ?
               70    13       00A6          BEQL    COSZER                  ; If zero, HTAND is infinite
         50    54 66FD        00A8          DIVH2   R4, R0                  ; Compute HSIND / HCOSD
```

```
05  00AC    366         RSB                             ; Return to caller
    00AD    367
```

```
                        00AD              .SBTTL  MTH$HTAND_R5 - JSB entry point
                        00AD      ;
                        00AD      ;++
                        00AD      ; FUNCTIONAL DESCRIPTION:
                        00AD      ;
                        00AD      ; HTAND - JSB entry point
                        00AD      ;
                        00AD      ; Algorithmic steps:
                        00AD      ;  1. Make sure that the absolute value of the argument is greater than
                        00AD      ;      180/pi*2**-16384 to avoid underflow in HSIND.
                        00AD      ;  2. Compute HSIND and HCOSD.
                        00AD      ;  3. If HCOSD is zero, we have an error.
                        00AD      ;  4. Return HSIND / HCOSD.
                        00AD      ;
                        00AD      ; CALLING SEQUENCE:
                        00AD      ;
                        00AD      ;       MOVH    argument, R0
                        00AD      ;       JSB     MTH$HTAND_R5
                        00AD      ;
                        00AD      ; INPUT PARAMETERS:
                        00AD      ;
                        00AD      ;       R0 / R3 contains x
                        00AD      ;
                        00AD      ; IMPLICIT INPUTS:
                        00AD      ;
                        00AD      ;       NONE
                        00AD      ;
                        00AD      ; OUTPUT PARAMETERS:
                        00AD      ;
                        00AD      ;       The result is the H-floating tangent of x.
                        00AD      ;
                        00AD      ; IMPLICIT OUTPUTS:
                        00AD      ;
                        00AD      ;       NONE
                        00AD      ;
                        00AD      ; SIDE EFFECTS:
                        00AD      ;
                        00AD      ;       NONE
                        00AD      ;--
                        00AD      MTH$HTAND_R5::
      7E    50 70FD     00AD              MOVH    R0, -(SP)               ; Save argument
50    8000 8F    AA     00B1              BICW    #^X8000, R0             ; R0/R3 = |argument|
      50    07    B1    00B6              CMPW    #7, R0                  ; Compare |ARG| with 2**-16377
      11    15          00B9              BLEQ    20$                     ; No possible underflow compute HTAND.
50    FF40 CF 71FD      00BB              CMPH    H_SMALLEST_DEG, R0      ; Possible underflow, use better check
      09    15          00C1              BLEQ    20$                     ; No underflow.
            50 73FD     00C3              TSTH    R0                      ; If |arg| = 0, no underflow, otherwise
      29    12          00C6              BNEQ    UNFL                    ; HSIND will underflow
      5E    10    C0    00C8              ADDL    #16, SP                 ; Remove argument from the stack
                  05    00CB              RSB                             ; Return with value = 0
                        00CC      20$:
 00000000'EF      16    00CC              JSB     MTH$HCOSD_R5            ; Compute HCOSD
      7E    50 70FD     00D2              MOVH    R0, -(SP)               ; Save HCOSD
      13    13          00D6              BEQL    30$                     ; If zero, HTAND is infinite
 50    10 AE 7DFD       00D8              MOVO    16(SP), R0              ; Get argument back
 00000000'GF      16    00DD              JSB     G^MTH$HSIND_R5          ; Compute HSIND
      50    8E 66FD     00E3              DIVH2   (SP)+, R0               ; Compute HSIND / HCOSD
```

```
5E    10   C0  00E7           ADDL2   #16, SP                 ; Discard saved argument
           05  00EA           RSB                             ; Return to caller
               00EB      ;+
               00EB      ; Come here if HCOSD is zero.  This means that HTAND is infinite.
               00EB      ;-
5E    20   C0  00EB      30$: ADDL2   #32, SP                 ; Discard saved HCOSD and saved argument
    0027   31  00EE           BRW     COSZER                  ; Go to common error code
               00F1
```

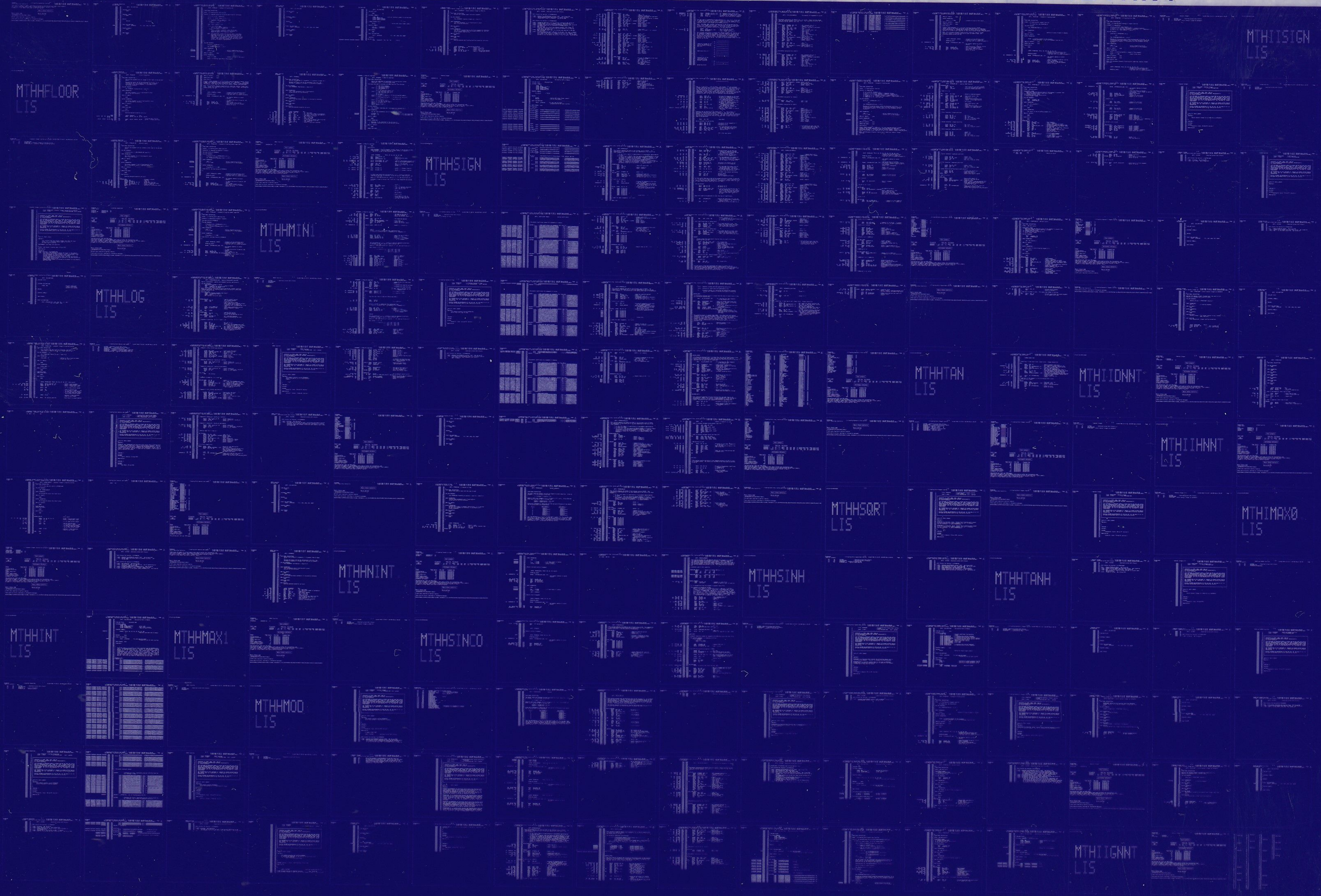
```
                                  00F1  ;
                                  00F1  ;           COMMON ERROR CODE
                                  00F1  ;
                                  00F1
                                  00F1  ;
                                  00F1  ; Underflow; if user has FU set, signal error.  Always return 0.0
                                  00F1  ;
                                  00F1  UNFL:
               5E    10    C0     00F1              ADDL    #16, SP                   ; Remove argument from stack
                     52    DC     00F4              MOVPSL  R2                        ; R2 = user's or jacket routine's PSL
        00000000'GF  00    FB     00F6              CALLS   #0, G^MTH$$JACKET_TST     ; R0 = TRUE if JSB from jacket routine
                  04 50    E9     00FD              BLBC    R0, 10$                   ; branch if user did JSB
            52    04 AD    3C     0100              MOVZWL  SF$W_SAVE_PSW(FP), R2     ; get user PSL saved by CALL
                     50    D4     0104  10$:        CLRL    R0                        ; R0 = result. LIB$SIGNAL will save in
                                  0106                                                ; CHF$L_MCH_R0/R1 so any handler can fixup
            0D 52    06    E1     0106              BBC     #6, R2, 20$               ; has user enabled floating underflow?
                     6E    DD     010A              PUSHL   (SP)                      ; yes, return PC from special routine
            7E    00'8F    9A     010C              MOVZBL  #MTH$K_FLOUNDMAT, -(SP)   ; trap code for hardware floating underflow
                                  0110                                                ; convert to MTH$_FLOUNDMAT (32-bit VAX-11
                                  0110                                                ; exception code)
        00000000'GF  02    FB     0110              CALLS   #2, G^MTH$$SIGNAL         ; signal (condition, PC)
                           05     0117  20$:        RSB                               ; return
                                  0118  ;+
                                  0118  ; Come here if the tangent is infinite because COS is zero.
                                  0118  ; Give an error signal.
                                  0118  ;-
                                  0118  COSZER:
                     6E    DD     0118              PUSHL   (SP)                      ; Push user "call" PC
            7E    00'8F    9A     011A              MOVZBL  #MTH$K_FLOOVEMAT, -(SP)   ; Condition value
          50   01    0F    79     011E              ASHQ    #15, #1, R0               ; R0/R3 = reserved operand
                     52    7C     0122              CLRQ    R2                        ; ...
        00000000'GF  02    FB     0124              CALLS   #2, G^MTH$$SIGNAL         ; Signal an error
                           05     012B              RSB                               ; Return to caller
                                  012C
                                  012C              .END
```

```
COSZER             00000118 R      02
HTAN             = 00000004
HTAND            = 00000004
H_SMALLEST_DEG     00000000 R      02
MTH$$JACKET_HND    ********    X   02
MTH$$JACKET_TST    ********    X   00
MTH$$SIGNAL        ********    X   00
MTH$$SIGNAL_CON    ********    X   00
MTH$HCOSD_R5       ********    X   00
MTH$HCOS_R5        ********    X   00
MTH$HSINCOSD_R7    ********    X   00
MTH$HSINCOS_R7     ********    X   00
MTH$HSIND_R5       ********    X   00
MTH$HSIN_R5        ********    X   00
MTH$HTAN           00000010 RG     02
MTH$HTAND          00000062 RG     02
MTH$HTAND_R5       000000AD RG     02
MTH$HTAND_R7       00000078 RG     02
MTH$HTAN_R5        00000039 RG     02
MTH$HTAN_R7        00000026 RG     02
MTH$K_FLOOVEMAT    ********    X   00
MTH$K_FLOUNDMAT    ********    X   00
SF$W_SAVE_PSW    = 00000004
UNFL               000000F1 R      02
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
$ABS$                           00000000  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_MTH$CODE                       0000012C  (   300.)  02 (  2.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                    Page faults    CPU Time        Elapsed Time
-----                    -----------    --------        ------------
Initialization                   30    00:00:00.08      00:00:00.70
Command processing              106    00:00:00.58      00:00:03.69
Pass 1                          125    00:00:01.69      00:00:06.16
Symbol table sort                 0    00:00:00.03      00:00:00.04
Pass 2                          106    00:00:01.09      00:00:04.82
Symbol table output               5    00:00:00.04      00:00:00.04
Psect synopsis output             5    00:00:00.02      00:00:00.02
Cross-reference output            0    00:00:00.00      00:00:00.00
Assembler run totals            379    00:00:03.55      00:00:15.47

The working set limit was 1050 pages.
8438 bytes (17 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 53 non-local and 7 local symbols.
529 source lines were read in Pass 1, producing 16 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.
```

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

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                  4

88 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:MTHHTAN/OBJ=OBJ$:MTHHTAN MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:
```

0262 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



MTHIISIGN
LIS

MTHHFLOOR
LIS

MTHHSIGN
LIS

MTHHMIN1
LIS

MTHHLOG
LIS

MTHHTAN
LIS

MTHIIDNNT
LIS

MTHIIHNNT
LIS

MTHHSQRT
LIS

MTHIMAX0
LIS

MTHHNINT
LIS

MTHHSINH
LIS

MTHHTANH
LIS

MTHHINT
LIS

MTHHMAX1
LIS

MTHHSINCO
LIS

MTHHMOD
LIS

MTHIIGNNT
LIS