```
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**MTHTAN                                        H  1
```

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

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      3S
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

```
0000            .TITLE  MTH$TAN         ; Floating Point Tangent routine
0000                                    ; (TAN, TAND)
0000            .IDENT /1-020/          ; File: MTHTAN.MAR  EDIT:RNH1020
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$TAN is a function  which  returns the floating point tangent
0000    ; of its single precision floating point radian argument. The call is
0000    ; standard call-by-reference.  It does a JSB to MTH$TAN_R5.
0000    ;
0000    ; MTH$TAND is a function  which  returns the floating point tangent
0000    ; of its single precision floating point degree argument. The call is
0000    ; standard call-by-reference.  It does a JSB to MTH$TAND_R5.
0000    ;
0000    ; MTH$TAN_R5, and MTH$TAND_R5 are JSB entry points that JSB to MTH$SINCOS_R5
0000    ; and MTH$SINCOSD_R5 respectively.  MTH$TAN_R4, and MTH$TAND_R4 cannot use
0000    ; the above two routines because they are _R4 routines, so they JSB to
0000    ; MTH$SIN_R4, MTH$COS_R4, and MTH$SIND_R4, MTH$COSD_R4 routines.
0000    ;
0000    ;--
0000    ;
0000    ; VERSION: 1
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    ;       The result is reserved operand when COS(X) = 0, instead
0000    ;       of largest or smallest representable floating number.
0000    ;
0000    ; Edit History
0000    ;
0000    ; 01-2  Error handling mechanism changed. Instead of having
0000    ;       MTH$FLG_JACKET at the entrance, MTH$$ERROR_CONT is stored on the
0000    ;       top of the stack frame so when error happened in MTH$SIN or MTH$COS
0000    ;       the message will be hided away, but will get signalled in MTH$TAN.
0000    ; 01-3  The call to MTH$ERROR_CONT is changed to MTH$SIGNAL_CON
0000    ; 0-4   MTH$$ERROR changed to MTH$$SIGNAL.
0000    ;       MTH$_.... changed to MTH__....
0000    ;       Changed error handling mechanism. Put error result in R0 before
0000    ;       calling MTH$$SIGNAL in order to allow user modify error result.
0000    ; 1-005 - Put version number in standard format (three digits of edit
0000    ;          number) and update copyright notice.  JBS 16-NOV-78
0000    ; 1-006 - Change MTH__FLOOVEMAT to MTH$K_FLOOVEMAT.  JBS 07-DEC-78
0000    ; 1-007 - Remove $SRMDEF macro - not needed.  JBS 16-DEC-78
0000    ; 1-008 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000    ; 1-009 - Fix error handling and detection.  SBL 02-Feb-79
0000    ; 1-010 - Declare externals.  SBL 17-May-1979
0000    ; 1-011 - Add JSB entry point.  JBS 16-AUG-1979
0000    ; 1-012 - Make external references longword, and remove MTH$$SIGNAL_CON
0000    ;          when doing our own signal.  JBS 16-AUG-1979
0000    ; 1-013 - Correct a typo in edit 011.  JBS 17-AUG-1979
0000    ; 1-014 - Have CALL entry JSB to JSB entry.  Use correct signalling
0000    ;         technique for JSB entry.  SBL 31-Oct-1979
0000    ; 1-015 - Reduce argument limit to 2**30 to match SIN/COS.  SBL 2-Nov-1979
0000    ; 1-016 - Added degree entry points. RNH 8-MAR-1981
0000    ; 1-017 - Undo edit 1-015.  SIN/COS can now accept this argument limit. RNH 26-AUG-8
0000    ; 1-018 - Add MTH$TAN_R5, and MTH$TAND_R5.  Rearange the routine for simplicity.
0000    ;         RNH 27-AUG-81.
0000    ; 1-019 - Change external references from W^ to G^. RNH 06-Oct-81
0000    ; 1-020 - Missed a W^.  RNH 08-Oct-81
```

```
                0000            .SBTTL  DECLARATIONS    ; Declarative Part of Module
                0000
                0000    ;
                0000    ; INCLUDE FILES:
                0000    ;
                0000
                0000    ;
                0000    ; EXTERNAL SYMBOLS:
                0000    ;
                0000            .DSABL  GBL
                0000            .EXTRN  MTH$$SINCOS_R5
                0000            .EXTRN  MTH$$SINCOSD_R5
                0000            .EXTRN  MTH$SIN_R4
                0000            .EXTRN  MTH$COS_R4
                0000            .EXTRN  MTH$K_FLOOVEMAT
                0000            .EXTRN  MTH$$SIGNAL
                0000            .EXTRN  MTH$K_FLOUNDMAT
                0000            .EXTRN  MTH$$JACKET_TST
                0000            .EXTRN  MTH$SIND_R4
                0000            .EXTRN  MTH$COSD_R4
                0000
                0000    ;
                0000    ; EQUATED SYMBOLS:
       2EE10365 0000            F_SMALLEST_DEG = ^X2EE10365
                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:
                0000
       00000004 0000            X = 4                           ;Position of argument from AP.
```

```
                              0000          .SBTTL  MTH$TAN  - Standard Single Precision Floating TAN
                              0000
                              0000
                              0000   ;++
                              0000   ; FUNCTIONAL DESCRIPTION:
                              0000   ;
                              0000   ; TAN  - single precision floating point function
                              0000   ;
                              0000   ;      For algorithm, see MTH$TAN_R5.
                              0000   ;
                              0000   ; CALLING SEQUENCE:
                              0000   ;
                              0000   ;      TAN.wf.v = MTH$TAN(X.rf.r)
                              0000   ;
                              0000   ; INPUT PARAMETERS:
                              0000   ;
                              0000   ;      X.rf.r          Address of value of angle in radians.
                              0000   ;
                              0000   ; IMPLICIT INPUTS:      none
                              0000   ;
                              0000   ; OUTPUT PARAMETERS:
                              0000   ;
                              0000   ;      VALUE:  floating tangent of the argument
                              0000   ;
                              0000   ; IMPLICIT OUTPUTS:     none
                              0000   ;
                              0000   ; COMPLETION CODES:     none
                              0000   ;
                              0000   ; SIDE EFFECTS:
                              0000   ;
                              0000   ;      NONE
                              0000   ;
                              0000   ;---
                              0000
                              0000
                        403C  0000          .ENTRY  MTH$TAN, ^M<IV, R2, R3, R4, R5>
                              0002                                          ; standard call-by-reference entry
                              0002                                          ; disable DV (and FU), enable IV
                              0002          MTH$FLAG_JACKET
                              0002
6D    00000000'GF        9E   0002          MOVAB   G^MTH$$JACKET_HND, (FP)
                              0009                                          ; set handler address to jacket
                              0009                                          ; handler
                              0009
        50    04 BC      50   0009          MOVF    @X(AP), R0              ; R0 = argument
                 01      10   000D          BSBB    MTH$TAN_R5              ; Get the tangent
                         04   000F          RET                             ; Return with result in R0
```

```
                     0010         .SBTTL  MTH$TAN_R5 - JSB entry point
                     0010
                     0010  ;++
                     0010  ; FUNCTIONAL DESCRIPTION
                     0010  ;
                     0010  ; TAN - single precision floating point function
                     0010  ;
                     0010  ; Algorithmic steps:
                     0010  ;
                     0010  ; 1.      Compute SIN and COS in one JSB.  Neither computation should fail.
                     0010  ; 2.      If COS is zero, error MTH$_FLOOVEMAT and return with reserved operand.
                     0010  ; 3.      Return SIN / COS.
                     0010  ;
                     0010  ; CALLING SEQUENCE:
                     0010  ;
                     0010  ;         MOVF    argument, R0
                     0010  ;         JSB     MTH$TAN_R5
                     0010  ;
                     0010  ; INPUT PARAMETERS:
                     0010  ;
                     0010  ;         R0 contains x
                     0010  ;
                     0010  ; OUTPUT PARAMETERS:
                     0010  ;
                     0010  ;         NONE
                     0010  ;
                     0010  ; IMPLICIT OUTPUTS:
                     0010  ;
                     0010  ;         NONE
                     0010  ;
                     0010  ; RESULT VALUE:
                     0010  ;
                     0010  ;         The tangent of x
                     0010  ;
                     0010  ; SIDE EFFECTS:
                     0010  ;
                     0010  ;         NONE
                     0010  ;
                     0010  ;--
                     0010 MTH$TAN_R5::
00000000'EF    16    0010         JSB     MTH$SINCOS_R5           ; Compute SIN(X), and COS(x)
         51    53    0016         TSTF    R1                      ; Is COS(X) EQL 0 ?
         04    13    0018         BEQL    20$                     ; If so, error
    50   51    46    001A         DIVF2   R1, R0                  ; Compute SIN(x) / COS(x)
               05    001D         RSB                             ; Return to caller
                     001E  ;
                     001E  ; Branch to common error code
                     001E  ;
                     001E 20$:
       00C4    31    001E         BRW     COSZER                  ;
```

```
                          0021          .SBTTL  MTH$TAN_R4 - JSB entry point
                          0021
                          0021  ;++
                          0021  ; FUNCTIONAL DESCRIPTION
                          0021  ;
                          0021  ; TAN - single precision floating point function
                          0021  ;
                          0021  ; Algorithmic steps:
                          0021  ;
                          0021  ; 1.    Compute SIN, and then COS.  Neither computation should fail.
                          0021  ; 2.    If COS is zero, error MTH$_FLOOVEMAT and return with reserved operand.
                          0021  ; 3.    Return SIN / COS.
                          0021  ;
                          0021  ; CALLING SEQUENCE:
                          0021  ;
                          0021  ;       MOVF    argument, R0
                          0021  ;       JSB     MTH$TAN_R4
                          0021  ;
                          0021  ; INPUT PARAMETERS:
                          0021  ;
                          0021  ;       R0 contains x
                          0021  ;
                          0021  ; OUTPUT PARAMETERS:
                          0021  ;
                          0021  ;       NONE
                          0021  ;
                          0021  ; IMPLICIT OUTPUTS:
                          0021  ;
                          0021  ;       NONE
                          0021  ;
                          0021  ; RESULT VALUE:
                          0021  ;
                          0021  ;       The tangent of x
                          0021  ;
                          0021  ; SIDE EFFECTS:
                          0021  ;
                          0021  ;       NONE
                          0021  ;
                          0021  ;--
                          0021  MTH$TAN_R4::                                   ; entry point
                  50  DD  0021          PUSHL   R0                             ; Save argument
         00000000'EF  16  0023          JSB     MTH$COS_R4                     ; Compute COS(x)
              7E  50  50  0029          MOVF    R0, -(SP)                      ; Put on stack and test for zero
                  11  13  002C          BEQL    20$                            ; If so, error
           50  04 AE  D0  002E          MOVL    4(SP), R0                      ; Get argument back
         00000000'EF  16  0032          JSB     MTH$SIN_R4                     ; Compute SIN(x)
              50  8E  46  0038          DIVF2   (SP)+, R0                      ; Compute SIN(x) / COS(x)
              5E  04  C0  003B          ADDL2   #4, SP                         ; Remove argument from stack
                      05  003E          RSB                                    ; Return to caller
                          003F  ;+
                          003F  ; Restore stack, and go to common error code.
                          003F  ;-
                          003F  20$:
              5E  08  C0  003F          ADDL2   #8, SP                         ; Discard COS and argument
                00A0  31  0042          BRW     COSZER                         ; Go to common error code
```

```
                      0045        .SBTTL  MTH$TAND  - Standard Single Precision Floating TAN
                      0045
                      0045
                      0045   ;++
                      0045   ; FUNCTIONAL DESCRIPTION:
                      0045   ;
                      0045   ; TAND  - Single precision floating point function
                      0045   ;
                      0045   ;          For algorithm, see MTH$TAND_R5.
                      0045   ;
                      0045   ; CALLING SEQUENCE:
                      0045   ;
                      0045   ;        TAND.wf.v = MTH$TAND(X.rf.r)
                      0045   ;
                      0045   ; INPUT PARAMETERS:
                      0045   ;
                      0045   ;        X.rf.r                          ;Address of value of angle in degrees.
                      0045   ;
                      0045   ; IMPLICIT INPUTS:      none
                      0045   ;
                      0045   ; OUTPUT PARAMETERS:
                      0045   ;
                      0045   ;        VALUE:  floating tangent of the argument
                      0045   ;
                      0045   ; IMPLICIT OUTPUTS:     none
                      0045   ;
                      0045   ; COMPLETION CODES:     none
                      0045   ;
                      0045   ; SIDE EFFECTS:
                      0045   ;
                      0045   ;        NONE
                      0045   ;
                      0045   ;---
                      0045
                      0045
                403C  0045        .ENTRY  MTH$TAND, ^M<IV, R2, R3, R4, R5>
                      0047                                        ; standard call-by-reference entry
                      0047                                        ; disable DV (and FU), enable IV
                      0047        MTH$FLAG_JACKET
                      0047
6D   00000000'GF   9E 0047        MOVAB   G^MTH$$JACKET_HND, (FP)
                      004E                                        ; set handler address to jacket
                      004E                                        ; handler
                      004E
     50    04 BC   50 004E        MOVF    @X(AP), R0              ; R0 = argument
              01   10 0052        BSBB    MTH$TAND_R5             ; Get the tangent
                   04 0054        RET                             ; Return with result in R0
```

```
                                  0055                .SBTTL  MTH$TAND_R5 - JSB entry point
                                  0055
                                  0055       ;++
                                  0055       ; FUNCTIONAL DESCRIPTION
                                  0055       ;
                                  0055       ; TAND - Single precision floating point function
                                  0055       ;
                                  0055       ; Algorithmic steps:
                                  0055       ;
                                  0055       ; 1.     Check for argument too small for SIND.  If so, return zero, and
                                  0055       ;        signal floating point underflow if enabled.
                                  0055       ; 2.     Compute SIND and COSD in one JSB.  Neither computation should fail.
                                  0055       ; 3.     If COSD is zero, error MTH$_FLOOVEMAT and return with reserved operand.
                                  0055       ; 4.     Return SIND / COSD.
                                  0055       ;
                                  0055       ; CALLING SEQUENCE:
                                  0055       ;
                                  0055       ;        MOVF    argument, R0
                                  0055       ;        JSB     MTH$TAND_R5
                                  0055       ;
                                  0055       ; INPUT PARAMETERS:
                                  0055       ;
                                  0055       ;        R0 contains x
                                  0055       ;
                                  0055       ; OUTPUT PARAMETERS:
                                  0055       ;
                                  0055       ;        NONE
                                  0055       ;
                                  0055       ; IMPLICIT OUTPUTS:
                                  0055       ;
                                  0055       ;        NONE
                                  0055       ;
                                  0055       ; RESULT VALUE:
                                  0055       ;
                                  0055       ;        The tangent of x
                                  0055       ;
                                  0055       ; SIDE EFFECTS:
                                  0055       ;
                                  0055       ;
                                  0055       ;
                                  0055       ;        Signal MTH$_FLOUNDMAT if |x| < 180/pi*2**-128
                                  0055       ;--
                                  0055
                                  0055
                                  0055       MTH$TAND_R5::
51   50   00008000 8F   CB        0055               BICL3   #^X8000, R0, R1               ; R1 = |X|
       51     0380 8F   B1        005D               CMPW    #^X380, R1                    ; Compare with 2**-121
                    0E  18        0062               BGEQ    30$                           ; No underflow possible
     51   2EE10365 8F   51        0064               CMPF    #F_SMALLEST_DEG, R1           ; Better test.  Compare
                    05  19        006B               BLSS    30$                           ; |X| with 180/pi*2**-128
                    50  D5        006D               TSTL    R0                            ; Check for zero
                    50  12        006F               BNEQ    UNFL                          ; ARG too small and not 0
                        05        0071               RSB                                   ; Return R0 = 0
                                  0072       ;+
                                  0072       ; We now know that MTH$SINCOSD_R5 routine will not fail, and that the
                                  0072       ; divide following them will not fail.
                                  0072       ;-
```

```
                         0072    389 30$:
     00000000'EF     16  0072    390          JSB     MTH$SINCOSD_R5          ; Compute SIND(X), and COSD(X)
              51     53  0078    391          TSTF    R1                      ; Is COSD(X) EQL 0 ?
              69     13  007A    392          BEQL    COSZER                  ; If so, error
        50    51     46  007C    393          DIVF2   R1, R0                  ; Compute SIND(x) / COSD(x)
                     05  007F    394          RSB                             ; Return to caller
                         0080    395
```

```
                              0080         .SBTTL  MTHSTAND_R4 - JSB entry point
                              0080
                              0080   ;++
                              0080   ; FUNCTIONAL DESCRIPTION
                              0080   ;
                              0080   ; TAND - Single precision floating point function
                              0080   ;
                              0080   ; Algorithmic steps:
                              0080   ;
                              0080   ; 1.    Check for argument too small for SIND.  If so, return zero, and
                              0080   ;       signal floating point underflow if enabled.
                              0080   ; 2.    Compute SIND, and then COSD.  Neither computation should fail.
                              0080   ; 3.    If COSD is zero, error MTHS_FLOOVEMAT and return with reserved operand.
                              0080   ; 4.    Return SIND / COSD.
                              0080   ;
                              0080   ; CALLING SEQUENCE:
                              0080   ;
                              0080   ;       MOVF    argument, R0
                              0080   ;       JSB     MTHSTAND_R4
                              0080   ;
                              0080   ; INPUT PARAMETERS:
                              0080   ;
                              0080   ;       R0 contains x
                              0080   ;
                              0080   ; OUTPUT PARAMETERS:
                              0080   ;
                              0080   ;       NONE
                              0080   ;
                              0080   ; IMPLICIT OUTPUTS:
                              0080   ;
                              0080   ;       NONE
                              0080   ;
                              0080   ; RESULT VALUE:
                              0080   ;
                              0080   ;       The tangent of x
                              0080   ;
                              0080   ; SIDE EFFECTS:
                              0080   ;
                              0080   ;
                              0080   ;
                              0080   ;       Signal MTHS_FLOUNDMAT if |x| < 180/pi*2**-128
                              0080   ;--
                              0080
                              0080
                              0080   MTHSTAND_R4::                                       ; entry point
51   50   00008000 8F    CB   0080           BICL3   #^X8000, R0, R1                     ;   R1 = |X|
          51   0380 8F   B1   0088           CMPW    #^X380, R1                          ; Compare with 2**-121
                    0E   18   008D           BGEQ    30$                                 ; No underflow possible
     51   2EE10365 8F    51   008F           CMPF    #F_SMALLEST_DEG, R1                 ; Better test.  Compare
                    05   19   0096           BLSS    30$                                 ; |X| with 180/pi*2**-128
                    50   D5   0098           TSTL    R0                                  ; Check for zero
                    25   12   009A           BNEQ    UNFL                                ; ARG too small and not 0
                         05   009C           RSB                                         ; Return R0 = 0
                              009D   ;+
                              009D   ; We now know that the SIND and COSD routines will not fail, and that the
                              009D   ; divide following them will not fail.
                              009D   ;-
```

```
            50  DD  009D   30$:    PUSHL   R0                              ; Save argument
   00000000'EF  16  009F           JSB     MTH$COSD_R4                     ; Compute COSD(x)
        7E  50  50  00A5           MOVF    R0, -(SP)                       ; Put on stack and test for zero
            11  13  00A8           BEQL    20$                             ; If so, error
     50  04 AE  D0  00AA           MOVL    4(SP), R0                       ; Get argument back
   00000000'EF  16  00AE           JSB     MTH$SIND_R4                     ; Compute SIND(x)
        50  8E  46  00B4           DIVF2   (SP)+, R0                       ; Compute SIND(x) / COSD(x)
        5E  04  C0  00B7           ADDL2   #4, SP                          ; Remove argument from stack
                05  00BA           RSB                                     ; Return to caller
                    00BB   ;+
                    00BB   ; Restore stack, and go to common error code.
                    00BB   ;-
                    00BB   20$:
        5E  08  C0  00BB           ADDL2   #8, SP                          ; Discard COSD and argument
          0024  31  00BE           BRW     COSZER                          ; Go to common error code
```

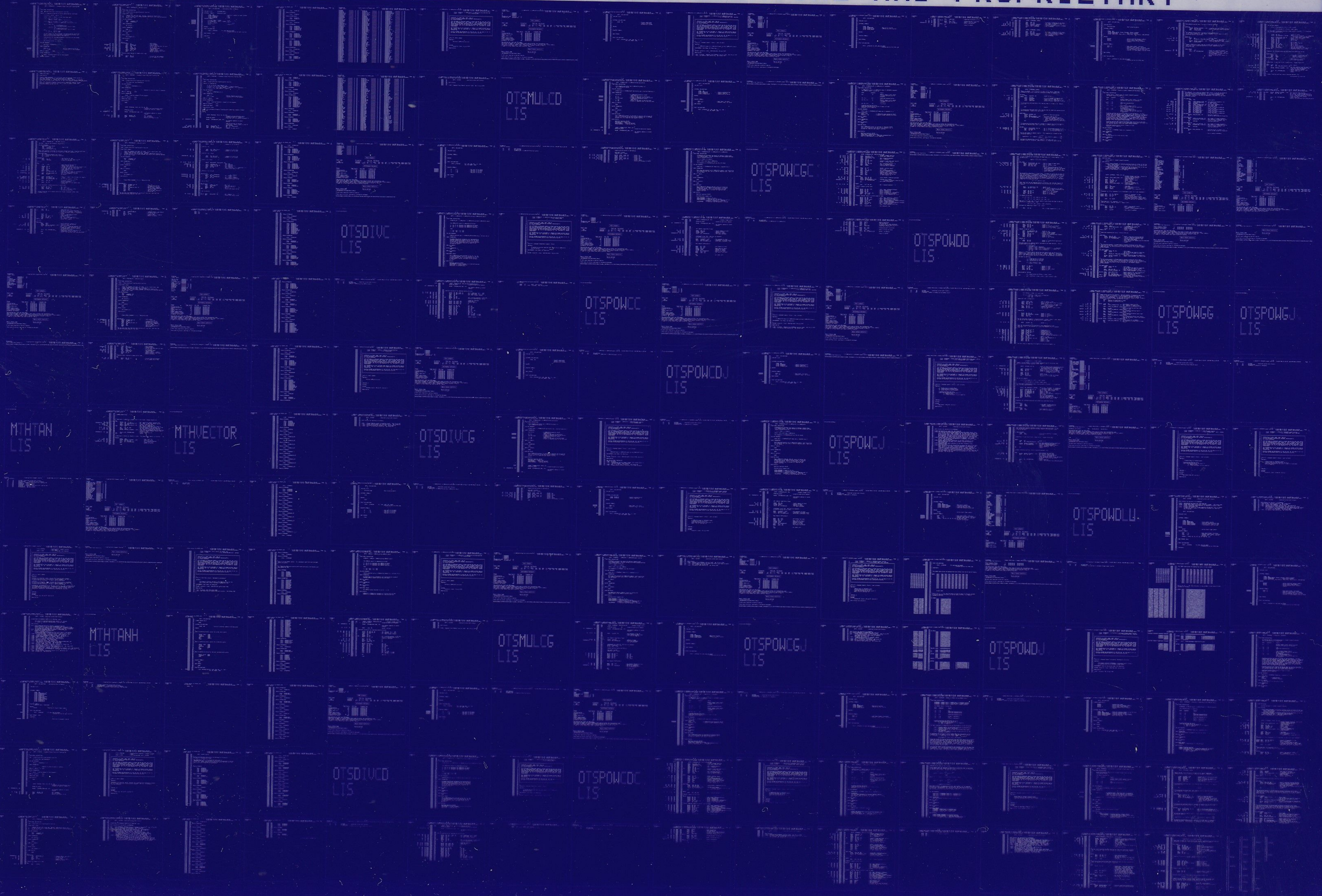
```
                          00C1  ;
                          00C1  ; COMMON ERROR PATHS
                          00C1  ;
                          00C1
                          00C1
                          00C1  ;
                          00C1  ; Come here if underflow; signal error if FU is set.  Always return 0.0
                          00C1  ;
                          00C1  UNFL:
              52    DC    00C1          MOVPSL  R2                        ; R2 = user's or jacket routine's PSL
  00000000'GF   00    FB    00C3          CALLS   #0, G^MTH$$JACKET_TST     ; R0 = TRUE if JSB from jacket routine
           04 50    E9    00CA          BLBC    R0, 10$                   ; branch if user did JSB
     52    04 AD    3C    00CD          MOVZWL  SF$W_SAVE_PSW(FP), R2     ; get user PSL saved by CALL
              50    D4    00D1  10$:    CLRL    R0                        ; R0 = result. LIB$SIGNAL will save in
                          00D3                                            ; CHF$L_MCH_R0/R1 so any handler can fixup
     0D 52    06    F1    00D3          BBC     #6, R2, 20$               ; has user enabled floating underflow?
              6E    DD    00D7          PUSHL   (SP)                      ; yes, return PC from special routine
     7E    00'8F    9A    00D9          MOVZBL  #MTH$K_FLOUNDMAT, -(SP)   ; trap code for hardware floating underflow
                          00DD                                            ; convert to MTH$_FLOUNDMAT (32-bit VAX-11
                          00DD                                            ; exception code)
  00000000'GF   02    FB    00DD          CALLS   #2, G^MTH$$SIGNAL         ; signal (condition, PC)
                    05    00E4  20$:    RSB                               ; return
                          00E5
                          00E5  ;+
                          00E5  ; Come here if COS(X) or COSD(X) is zero.  This means that TAN(X) is infinite.
                          00E5  ;-
                          00E5  COSZER:
              6E    DD    00E5          PUSHL   (SP)                              ; User "call" PC
     7E    00'8F    9A    00E7          MOVZBL  #MTH$K_FLOOVEMAT, -(SP)           ; Condition value
     50    01       0F    78    00EB          ASHL    #15, #1, R0                       ; R0 = reserved operand
  00000000'GF   02    FB    00EF          CALLS   #2, G^MTH$$SIGNAL                 ; Signal the error
                    05    00F6          RSB                                       ; Return to caller.
                          00F7
                          00F7          .END
```

```
COSZER            000000E5 R      02
F_SMALLEST_DEG =  2EE10365
MTH$$JACKET_HND   ********    X   02
MTH$$JACKET_TST   ********    X   00
MTH$$SIGNAL       ********    X   00
MTH$COSD_R4       ********    X   00
MTH$COS_R4        ********    X   00
MTH$K_FLOOVEMAT   ********    X   00
MTH$K_FLOUNDMAT   ********    X   00
MTH$SINCOSD_R5    ********    X   00
MTH$SINCOS_R5     ********    X   00
MTH$SIND_R4       ********    X   00
MTH$SIN_R4        ********    X   00
MTH$TAN           00000000 RG     02
MTH$TAND          00000045 RG     02
MTH$TAND_R4       00000080 RG     02
MTH$TAND_R5       00000055 RG     02
MTH$TAN_R4        00000021 RG     02
MTH$TAN_R5        00000010 RG     02
SF$W_SAVE_PSW   = 00000004
UNFL              000000C1 R      02
X               = 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
$ABS$                      00000000  (     0.)   01 (   1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_MTH$CODE                  000000F7  (   247.)   02 (   2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   29    00:00:00.06    00:00:00.72
Command processing              112    00:00:00.71    00:00:03.39
Pass 1                          126    00:00:01.60    00:00:05.85
Symbol table sort                 0    00:00:00.04    00:00:00.06
Pass 2                           94    00:00:01.15    00:00:05.70
Symbol table output               3    00:00:00.03    00:00:00.03
Psect synopsis output             3    00:00:00.02    00:00:00.02
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            369    00:00:03.63    00:00:15.79

The working set limit was 900 pages.
8464 bytes (17 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 50 non-local and 7 local symbols.
563 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$:MTHTAN/OBJ=OBJ$:MTHTAN MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:MT
```

0264 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



OTSMULCD
LIS

OTSPOWCGC
LIS

OTSDIVC
LIS

OTSPOWDD
LIS

OTSPOWCC
LIS

OTSPOWGG
LIS

OTSPOWGJ
LIS

OTSPOWCDJ
LIS

MTHTAN
LIS

MTHVECTOR
LIS

OTSDIVCG
LIS

OTSPOWCJ
LIS

OTSPOWDLU
LIS

MTHTANH
LIS

OTSMULCG
LIS

OTSPOWCGJ
LIS

OTSPOWDJ
LIS

OTSDIVCD
LIS

OTSPOWCDC
LIS