```
MMM          MMM  TTTTTTTTTTTTTTT  HHH          HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM          MMM  TTTTTTTTTTTTTTT  HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
MMM          MMM  TTTTTTTTTTTTTTT  HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
MMMMMM    MMMMMM        TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMMMMM    MMMMMM        TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMMMMM    MMMMMM        TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM  MMM  MMM           TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM  MMM  MMM           TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM  MMM  MMM           TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM       MMM           TTT       HHHHHHHHHHHHHHH  RRRRRRRRRRR         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**MTHDTAN

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

Table of contents

```
0000     1          .TITLE  MTH$DTAN        ; Double Precision Floating Point Tangent routine
0000     2                                  ; (DTAN, DTAND)
0000     3          .IDENT /1-013/          ; File: MTHDTAN.MAR  EDIT:SBL1013
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28 ;
0000    29 ; FACILITY: MATH LIBRARY
0000    30 ;++
0000    31 ; ABSTRACT:
0000    32 ;
0000    33 ; MTH$DTAN is a function  which  returns the double precision floating point
0000    34 ; tangent of its double precision floating point radian argument. The call is
0000    35 ; standard call-by-reference.  It JSB to MTH$DTAN_R7.
0000    36 ;
0000    37 ; MTH$DTAND is a function  which  returns the double precision floating point
0000    38 ; tangent of its double precision floating point degree argument. The call is
0000    39 ; standard call-by-reference. It JSB to MTH$DTAND_R7.
0000    40 ;
0000    41 ;--
0000    42 ;
0000    43 ; VERSION: 01
0000    44 ;
0000    45 ; HISTORY:
0000    46 ; AUTHOR:
0000    47 ;       Peter Yuo, 29-Jun-77: Version 01
0000    48 ;
0000    49 ; MODIFIED BY:
0000    50 ;
0000    51 ;
0000    52 ;
```

```
0000     54          .SBTTL  HISTORY ; Detailed Current Edit History
0000     55
0000     56
0000     57 ; ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000     58 ;
0000     59 ;       The result is reserved operand when DCOS(X) = 0, instead
0000     60 ;       of largest or smallest representable floating number.
0000     61 ;
0000     62 ; Edit History for Version 01 of MTH$DTAN
0000     63 ;
0000     64 ; 01-2  Error handling mechanism changed. Instead of having
0000     65 ;       MTH$FLG_JACKET at the entrance, MTH$$ERROR_CONT is stored on the
0000     66 ;       top of the stack frame so when error happened in MTH$DSIN or MTH$DCOS
0000     67 ;       the message will be hided away, but will get signalled in MTH$DTAN.
0000     68 ; 0-3   MTH$$ERROR changed to MTH$$SIGNAL.
0000     69 ;       MTH$$ERROR_CONT changed to MTH$$SIGNAL_CON
0000     70 ;       MTH$_... changed to MTH__....
0000     71 ;       Changed error handling mechanism. Put error result in R0:R1 before
0000     72 ;       calling MTH$$SIGNAL in order to allow user modify error result.
0000     73 ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000     74 ; 1-002 - Changed MTH__FLOOVEMAT to MTH$K_FLOOVEMAT.  JBS 06-DEC-78
0000     75 ; 1-003 - Removed $SRMDEF macro - not needed.  JBS 16-DEC-78
0000     76 ; 1-004 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000     77 ; 1-005 - Declare externals.  SBL 17-May-1979
0000     78 ; 1-006 - Rearrange code to handle exceptions correctly.  SBL 17-May-1979
0000     79 ; 1-007 - Add JSB entry point.  JBS 16-AUG-1979
0000     80 ; 1-008 - Signal exceptions correctly.  JBS 16-AUG-1979
0000     81 ; 1-009 - Correct a typo in edit 007.  JBS 17-AUG-1979
0000     82 ; 1-010 - Make CALL entry use JSB routine.  Correct error handling.
0000     83 ;         Do COS before SIN.  SBL 31-Oct-1979
0000     84 ; 1-011 - Added degree entry points. RNH 8-MAR-1981
0000     85 ; 1-012 - Modify MTH$DTAN_R7 and MTH$DTAND_R7 to JSB to MTH$DSINCOS_R7 and
0000     86 ;         MTH$DSINCOSD_R7 instead of MTH$DSIN_R7, MTH$DCOS_R7, and
0000     87 ;         MTH$DSIND_R7, MTH$DCOSD_R7.  RNH 27-AUG-81.
0000     88 ; 1-013 - Use general mode addressing.  SBL 30-Nov-1981
0000     89 ;
```

```
                                        .SBTTL  DECLARATIONS    ; Declarative Part of Module

                                ;
                                ; INCLUDE FILES:
                                ;

                                ;
                                ; EXTERNAL SYMBOLS:
                                ;
                                        .DSABL  GBL
                                        .EXTRN  MTH$DSINCOS_R7
                                        .EXTRN  MTH$DSINCOSD_R7
                                        .EXTRN  MTH$K_FLOOVEMAT
                                        .EXTRN  MTH$$SIGNAL
                                        .EXTRN  MTH$K_FLOUNDMAT
                                        .EXTRN  MTH$$JACKET_TST

                                ;
                                ; EQUATED SYMBOLS:      none
                                ;
                                ; MACROS:

                                        $SFDEF                          ; Define SF (Stack Frame) symbols

                                ;
                                ; PSECT DECLARATIONS:

                                        .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                                                                        ; program section for math routines
                                ;
                                ; OWN STORAGE:  none
                                ;
                                ; CONSTANTS:

                                        X = 4                           ;Position of argument from AP.

                                D_SMALLEST_DEG:
2EE04365                                .LONG   ^X2EE04365
0FBED31E                                .LONG   ^X0FBED31E
```

```
                        0008   132          .SBTTL  MTH$DTAN  - Standard Double Precision Floating DTAN
                        0008   133
                        0008   134
                        0008   135 ;++
                        0008   136 ; FUNCTIONAL DESCRIPTION:
                        0008   137 ;
                        0008   138 ; DTAN  - double precision floating point function
                        0008   139 ;
                        0008   140 ;       For algorithm, see MTH$DTAN_R7.
                        0008   141 ;
                        0008   142 ; CALLING SEQUENCE:
                        0008   143 ;
                        0008   144 ;       DTAN.wd.v = MTH$DTAN(x.rd.r)
                        0008   145 ;
                        0008   146 ; INPUT PARAMETERS:
                        0008   147 ;
                        0008   148 ;       X.rd.r                  Address of value of angle in radians.
                        0008   149 ;
                        0008   150 ; IMPLICIT INPUTS:      none
                        0008   151 ;
                        0008   152 ; OUTPUT PARAMETERS:
                        0008   153 ;
                        0008   154 ;       VALUE:  double precision floating tangent of the argument
                        0008   155 ;
                        0008   156 ; IMPLICIT OUTPUTS:     none
                        0008   157 ;
                        0008   158 ; COMPLETION CODES:     none
                        0008   159 ;
                        0008   160 ; SIDE EFFECTS:
                        0008   161 ;
                        0008   162 ;       See MTH$DTAN_R7
                        0008   163 ;
                        0008   164 ;---
                        0008   165
                        0008   166
                 40FC   0008   167          .ENTRY  MTH$DTAN, ^M<IV, R2, R3, R4, R5, R6, R7>
                        000A   168                                                  ; standard call-by-reference entry
                        000A   169                                                  ; disable DV (and FU), enable IV
                        000A   170          MTH$FLAG_JACKET
                        000A
6D  00000000'GF   9E    000A               MOVAB   G^MTH$$JACKET_HND, (FP)
                        0011                                                        ; set handler address to jacket
                        0011                                                        ; handler
                        0011
    50  04 BC   70      0011   171          MOVD    @X(AP), R0                      ; R0/R1 = argument
          01    10      0015   172          BSBB    MTH$DTAN_R7                     ; Get the tangent
                04      0017   173          RET                                     ; With result in R0-R1
```

```
                                        .SBTTL  MTH$DTAN_R7 - JSB entry point
                                ;
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ; DTAN - JSB entry point
                                ;
                                ; Algorithmic steps:
                                ;  1. Compute DSIN and DCOS.
                                ;  2. If DCOS is zero, we have an error.
                                ;  3. Return DSIN / DCOS.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       MOVD    argument, R0
                                ;       JSB     MTH$DTAN_R7
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       R0 / R1 contains X
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       The result is the tangent of X, in double precision.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;--
                                MTH$DTAN_R7::
00000000'GF   16        0018            JSB     G^MTH$DSINCOS_R7        ; Compute DSIN(X) and DCOS(X)
        52    73        001E            TSTD    R2                      ; Is DCOS(X) zero ?
        62    13        0020            BEQL    COSZER                  ; If yes, then go to common error path
   50   52    66        0022            DIVD2   R2, R0                  ; Compute DSIN / DCOS
              05        0025            RSB                             ; Return.
                        0026
                        0026
```

```
                                        .SBTTL  MTH$DTAND  - Standard Double Precision Floating DTAND

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ; DTAND  - double precision floating point function
                                ;
                                ;       For algorithm, see MTH$DTAND_R7.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       DTAND.wd.v = MTH$DTAND(X.rd.r)
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       X.rd.r                          ;Address of value of angle in degrees.
                                ;
                                ; IMPLICIT INPUTS:      none
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       VALUE:  double precision floating tangent of the argument
                                ;
                                ; IMPLICIT OUTPUTS:     none
                                ;
                                ; COMPLETION CODES:     none
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       See MTH$DTAND_R7
                                ;
                                ;---

                                .ENTRY  MTH$DTAND, ^M<IV, R2, R3, R4, R5, R6, R7>
                                                                        ; standard call-by-reference entry
                                                                        ; disable DV (and FU), enable IV
                                MTH$FLAG_JACKET

                                MOVAB   G^MTH$$JACKET_HND, (FP)
                                                                        ; set handler address to jacket
                                                                        ; handler

                                MOVD    @X(AP), R0                      ; R0/R1 = argument
                                BSBB    MTH$DTAND_R7                    ; Get the tangent
                                RET                                     ; With result in R0-R1
```

```
                                0036   264          .SBTTL  MTH$DTAND_R7 - JSB entry point
                                0036   265 ;
                                0036   266 ;++
                                0036   267 ; FUNCTIONAL DESCRIPTION:
                                0036   268 ;
                                0036   269 ; DTAND - JSB entry point
                                0036   270 ;
                                0036   271 ; Algorithmic steps:
                                0036   272 ;  1. Make sure that the absolute value of the argument is greater
                                0036   273 ;     than 180/pi*2*128
                                0036   274 ;  2. Compute DSIND and DCOSD.
                                0036   275 ;  3. If DCOSD is zero, we have an error.
                                0036   276 ;  4. Return DSIND / DCOSD.
                                0036   277 ;
                                0036   278 ; CALLING SEQUENCE:
                                0036   279 ;
                                0036   280 ;       MOVD    argument, R0
                                0036   281 ;       JSB     MTH$DTAND_R7
                                0036   282 ;
                                0036   283 ; INPUT PARAMETERS:
                                0036   284 ;
                                0036   285 ;       R0 / R1 contains x
                                0036   286 ;
                                0036   287 ; IMPLICIT INPUTS:
                                0036   288 ;
                                0036   289 ;       NONE
                                0036   290 ;
                                0036   291 ; OUTPUT PARAMETERS:
                                0036   292 ;
                                0036   293 ;       The result is the tangent of x, in double precision.
                                0036   294 ;
                                0036   295 ; IMPLICIT OUTPUTS:
                                0036   296 ;
                                0036   297 ;       NONE
                                0036   298 ;
                                0036   299 ; SIDE EFFECTS:
                                0036   300 ;
                                0036   301 ;       NONE
                                0036   302 ;--
                                0036   303 MTH$DTAND_R7::
      52    50    70              0036   304          MOVD    R0, R2                  ; Save argument
52    8000  8F    AA              0039   305          BICW    #^X8000, R2             ; R2/R3 = |argument|
52    0300  8F    B1              003E   306          CMPW    #^X0300, R2             ; Compare |arg| with 2**-121
            0D    15              0043   307          BLEQ    20$                     ; No possible underflow, compute DCOSD
  52    B8  AF    71              0045   308          CMPD    D_SMALLEST_DEG, R2      ; Possible underflow, use better check
            07    19              0049   309          BLSS    20$                     ; No possible underflow, compute DCOSD
            52    73              004B   310          TSTD    R2                      ; Check for argument = 0
            11    12              004D   311          BNEQ    UNFL                    ; If arg not 0, go to underflow logic
            50    7C              004F   312          CLRD    R0                      ; Load R0/R1 with zero
                  05              0051   313          RSB                             ; Return with value = 0
                                  0052   314 20$:
00000000'GF       16              0052   315          JSB     G^MTH$DSINCOSD_R7       ; Compute DSIND(X), and DCOSD(X)
            52    73              0058   316          TSTD    R2                      ; Is DCOSD(X) zero ?
            28    13              005A   317          BEQL    COSZER                  ; If yes, then go to common error path
      50    52    66              005C   318          DIVD2   R2, R0                  ; Compute DSIND / DCOSD
                  05              005F   319          RSB                             ; Return.
                                  0060   320
```

0060 321

```
                     0060   323 ;
                     0060   324 ;       COMMON ERROR CODE
                     0060   325 ;
                     0060   326
                     0060   327 ;
                     0060   328 ; Underflow; if user has FU set, signal error.  Always return 0.0
                     0060   329 ;
                     0060   330 UNFL:
              52 DC  0060   331         MOVPSL  R2                              ; R2 = user's or jacket routine's PSL
00000000'GF   00 FB  0062   332         CALLS   #0, G^MTH$$JACKET_TST           ; R0 = TRUE if JSB from jacket routine
           04 50 E9  0069   333         BLBC    R0, 10$                         ; branch if user did JSB
        52 04 AD 3C  006C   334         MOVZWL  SF$W_SAVE_PSW(FP), R2           ; get user PSL saved by CALL
              50 D4  0070   335 10$:    CLRL    R0                              ; R0 = result. LIB$SIGNAL will save in
                     0072   336                                                 ; CHF$L_MCH_R0/R1 so any handler can fixup
        0D 52 06 E1  0072   337         BBC     #6, R2, 20$                     ; has user enabled floating underflow?
              6E DD  0076   338         PUSHL   (SP)                            ; yes, return PC from special routine
        7E 00'8F 9A  0078   339         MOVZBL  #MTH$K_FLOUNDMAT, -(SP)         ; trap code for hardware floating underflow
                     007C   340                                                 ; convert to MTH$_FLOUNDMAT (32-bit VAX-11
                     007C   341                                                 ; exception code)
00000000'GF   02 FB  007C   342         CALLS   #2, G^MTH$$SIGNAL               ; signal (condition, PC)
              05     0083   343 20$:    RSB                                     ; return
                     0084   344
                     0084   345 ;+
                     0084   346 ; Come here if COS value is zero.
                     0084   347 ;-
                     0084   348 COSZER:
              6E DD  0084   349         PUSHL   (SP)                            ; Push "caller" PC
        7E 00'8F 9A  0086   350         MOVZBL  #MTH$K_FLOOVEMAT, -(SP)         ; Condition value
     50 01 0F 79     008A   351         ASHQ    #15, #1, R0                     ; R0/R1 = reserved operand
00000000'GF   02 FB  008E   352         CALLS   #2, G^MTH$$SIGNAL               ; Signal an error
              05     0095   353         RSB                                     ; Return to caller
                     0096   354
                     0096   355         .END
```

```
COSZER              00000084 R      02
D_SMALLEST_DEG      00000000 R      02
MTH$$JACKET_HND     ********    X   02
MTH$$JACKET_TST     ********    X   00
MTH$$SIGNAL         ********    X   00
MTH$DSINCOSD_R7     ********    X   00
MTH$DSINCOS_R7      ********    X   00
MTH$DTAN            00000008 RG     02
MTH$DTAND           00000026 RG     02
MTH$DTAND_R7        00000036 RG     02
MTH$DTAN_R7         00000018 RG     02
MTH$K_FLOOVEMAT     ********    X   00
MTH$K_FLOUNDMAT     ********    X   00
SF$W_SAVE_PSW     = 00000004
UNFL                00000060 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	00000096 (150.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.09	00:00:00.84
Command processing	112	00:00:00.70	00:00:06.99
Pass 1	121	00:00:01.46	00:00:06.06
Symbol table sort	0	00:00:00.03	00:00:00.13
Pass 2	74	00:00:00.90	00:00:03.69
Symbol table output	3	00:00:00.02	00:00:00.25
Psect synopsis output	2	00:00:00.03	00:00:00.33
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	347	00:00:03.23	00:00:18.30

```
The working set limit was 900 pages.
6631 bytes (13 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 44 non-local and 3 local symbols.
415 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
_255DU 28:[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$:MTHDTAN/OBJ=OBJ$:MTHDTAN MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MTHGCONJG
LIS

MTHGINT
LIS

MTHGMOD
LIS

MTHEXP
LIS

MTHFLOOR
LIS

MTHDTAN
LIS

MTHDTANH
LIS

MTHGEXP
LIS

MTHGFLOOR
LIS

MTHGMIN1
LIS

MTHGCOSH
LIS

MTHGLOG
LIS

MTHGACOS
LIS

MTHGASIN
LIS

MTHGNINT
LIS

MTHGATAN
LIS

MTHGATANH
LIS

MTHGMAX1
LIS