```
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMMMMM    MMMMMM        TTT         HHH          HHH  RRR         RRR       TTT         LLL
MMMMMM    MMMMMM        TTT         HHH          HHH  RRR         RRR       TTT         LLL
MMMMMM    MMMMMM        TTT         HHH          HHH  RRR         RRR       TTT         LLL
MMM   MMM   MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLL
MMM   MMM   MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLL
MMM   MMM   MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLL
MMM         MMM         TTT         HHHHHHHHHHHHHHHH  RRRRRRRRRRR           TTT         LLL
MMM         MMM         TTT         HHHHHHHHHHHHHHHH  RRRRRRRRRRRR          TTT         LLL
MMM         MMM         TTT         HHHHHHHHHHHHHHHH  RRRRRRRRRRRR          TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR   RRR             TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR   RRR             TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR   RRR             TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR      RRR          TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR      RRR          TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR      RRR          TTT         LLL
MMM         MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLLLLLLLLLLLLLLL
MMM         MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLLLLLLLLLLLLLLL
MMM         MMM         TTT         HHH          HHH  RRR         RRR       TTT         LLLLLLLLLLLLLLLL
```

FILEID**MTHDACOS

```
MM       MM  TTTTTTTTTT  HH       HH  DDDDDDDD      AAAAAA      CCCCCCCC    OOOOOO      SSSSSSSS
MM       MM  TTTTTTTTTT  HH       HH  DDDDDDDD      AAAAAA      CCCCCCCC    OOOOOO      SSSSSSSS
MMMM   MMMM      TT      HH       HH  DD      DD   AA      AA  CC          OO      OO  SS
MMMM   MMMM      TT      HH       HH  DD      DD   AA      AA  CC          OO      OO  SS
MM  MM  MM       TT      HH       HH  DD      DD   AA      AA  CC          OO      OO  SS
MM  MM  MM       TT      HH       HH  DD      DD   AA      AA  CC          OO      OO  SS
MM      MM       TT      HHHHHHHHHHH  DD      DD   AA      AA  CC          OO      OO    SSSSSS
MM      MM       TT      HHHHHHHHHHH  DD      DD   AA      AA  CC          OO      OO    SSSSSS
MM      MM       TT      HH       HH  DD      DD   AAAAAAAAAA  CC          OO      OO          SS
MM      MM       TT      HH       HH  DD      DD   AAAAAAAAAA  CC          OO      OO          SS
MM      MM       TT      HH       HH  DD      DD   AA      AA  CC          OO      OO          SS      ....
MM      MM       TT      HH       HH  DD      DD   AA      AA  CC          OO      OO          SS      ....
MM      MM       TT      HH       HH  DDDDDDDD    AA      AA    CCCCCCCC    OOOOOO    SSSSSSSS      ....
MM      MM       TT      HH       HH  DDDDDDDD    AA      AA    CCCCCCCC    OOOOOO    SSSSSSSS      ....

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
```

MTH
Sym

D_
D_C
D_C
D_F
D_F
D_T
D_T
ERR
GEC
LOM
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
SD_
TES
TES
VAL

PSE

._MT

Pha

Int
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
623
The
479
1 p

Table of contents

```
0000     1         .TITLE  MTH$DACOS       ; Double Precision Floating Point Arc-cosine routine
0000     2                                 ; (DACOS,DACOSD)
0000     3         .IDENT /1-008/          ; File: MTHDACOS.MAR  Edit: JCW1008
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$DACOS is a function which returns the double precision floating point
0000    34 ; arc-cosine in radians of its double precision floating point argument. The
0000    35 ; call is standard call-by-reference.
0000    36 ;
0000    37 ; MTH$DACOSD is a function which returns the double precision floating point
0000    38 ; arc-cosine in degrees of its double precision floating point argument. The
0000    39 ; call is standard call-by-reference.
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 ; Edit History for Version 01 of MTH$DACOS
0000     60 ;
0000     61 ; 0-2   MTH$$ERROR changed to MTH$$SIGNAL.
0000     62 ;       MTH$_... changed to MTH__....
0000     63 ;       Changed error handling mechanism. Put error result in R0:R1 before
0000     64 ;       calling MTH$$SIGNAL in order to allow user modify error result.
0000     65 ; 0-3   - Changed MOVD #0, to CLRD.  TNH 10-Nov-77
0000     66 ; 1-001 - Updated version number and copyright notice.  JBS 16-NOV-78
0000     67 ; 1-002 - Change MTH_INVARG to MTH$K_INVARGMAT.  JBS 07-DEC-78
0000     68 ; 1-003 - Add "_" to PSECT directive.  JBS 22-DEC-78
0000     69 ; 1-004 - Declare externals.  SBL 17-May-1979
0000     70 ; 1-005 - Change JSB entry to MTH$DACOS_R7.  RBG 28-Sept-1979
0000     71 ; 1-006 - Added degree entry points. RNH 22-MAR-1981
0000     72 ; 1-007 - Modified computation of 1 - X^2 to avoid loss of significance for
0000     73 ;            !X!>=1/2  RNH 02-Sept-1981
0000     74 ; 1-008 - Corrected the floating representation of .5 from #4000 to #^X4000
0000     75 ;            JCW 14-Jul-1982
```

```
                     0000     77          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                     0000     78
                     0000     79 ;
                     0000     80 ; INCLUDE FILES:
                     0000     81 ;
                     0000     82
                     0000     83 ;
                     0000     84 ; EXTERNAL SYMBOLS:
                     0000     85 ;
                     0000     86          .DSABL  GBL
                     0000     87          .EXTRN  MTH$DSQRT_R5
                     0000     88          .EXTRN  MTH$DATAN_R7
                     0000     89          .EXTRN  MTH$DATAND_R7
                     0000     90          .EXTRN  MTH$K_INVARGMAT
                     0000     91          .EXTRN  MTH$$SIGNAL
                     0000     92
                     0000     93 ;
                     0000     94 ; EQUATED SYMBOLS:
                     0000     95
00004080             0000     96          SD_1.0  = ^F1.0                 ; 1.0
00000004             0000     97          value   = 4                     ; value.rd.r
                     0000     98
                     0000     99 ;
                     0000    100 ; MACROS:       none
                     0000    101 ;
                     0000    102 ; PSECT DECLARATIONS:
                     0000    103
00000000             0000    104          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                     0000    105                                          ; program section for math routines
                     0000    106 ;
                     0000    107 ; OWN STORAGE:  none
                     0000    108 ;
                     0000    109 ;
                     0000    110 ; CONSTANTS:
                     0000    111 ;
                     0000    112
                     0000    113 D_PI_OVER_2:
68C2 A221 0FDA 40C9  0000    114          .WORD   ^O040311, ^O007732, ^O121041, ^O064302
                     0008    115                                          ; PI/2
                     0008    116 D_PI:
68C2 A221 0FDA 4149  0008    117          .WORD   ^O040511, ^O007732, ^O121041, ^O064302
                     0010    118                                          ; PI
                     0010    119 D_90:
  00000000 000043B4  0010    120          .LONG   ^X000043B4, ^X0         ; 90
                     0018    121 D_180:
  00000000 00004434  0018    122          .LONG   ^X00004434, ^X0         ; 180
```

```
                                0020   124          .SBTTL  MTH$DACOS  - Standard Double Precision Floating DACOS
                                0020   125
                                0020   126
                                0020   127 ;++
                                0020   128 ; FUNCTIONAL DESCRIPTION:
                                0020   129 ;
                                0020   130 ; DACOS  - double precision floating point function
                                0020   131 ;
                                0020   132 ; DACOS(X) is computed as:
                                0020   133 ;
                                0020   134 ;       If X = 0, then DACOS(X) = PI/2.
                                0020   135 ;       If X = 1, then DACOS(X) = 0.
                                0020   136 ;       If X = -1, then DACOS(X) = PI.
                                0020   137 ;       If 0 < X < 1/2, then DACOS(X) = ATAN(SQRT(1-X**2)/X).
                                0020   138 ;       If 1/2 < X < 1, then DACOS(X) = ATAN(SQRT((1-X)*(1+x))/X).
                                0020   139 ;       If -1/2 < X < 0, then DACOS(X) = ATAN(SQRT(1-X**2)/X) + PI.
                                0020   140 ;       If -1 < X =< -1/2, then DACOS(X) = ATAN(SQRT((1-X)*(1+x))/X) + PI.
                                0020   141 ;       If 1 < |X|, error.
                                0020   142 ;
                                0020   143 ; CALLING SEQUENCE:
                                0020   144 ;
                                0020   145 ;       DACOS.wd.v = MTH$DACOS(x.rd.r)
                                0020   146 ;
                                0020   147 ; INPUT PARAMETERS:
                                0020   148
                       00000004 0020   149          LONG = 4                        ; define longword multiplier
                       00000004 0020   150          x = 1 * LONG                    ; Contents of x is the argument
                                0020   151
                                0020   152 ; IMPLICIT INPUTS:      none
                                0020   153 ;
                                0020   154 ; OUTPUT PARAMETERS:
                                0020   155 ;
                                0020   156 ;       VALUE:  double precision floating arc-cosine of the argument
                                0020   157 ;
                                0020   158 ; IMPLICIT OUTPUTS:     none
                                0020   159 ;
                                0020   160 ; COMPLETION CODES:     none
                                0020   161 ;
                                0020   162 ; SIDE EFFECTS:
                                0020   163 ;
                                0020   164 ; Signals: MTH$_INVARG if |X| > 1 with reserved operand in R0/R1 (copied
                                0020   165 ; to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                                0020   166 ; Associated message is: "INVALID ARGUMENT". Result is reserved operand -0.0
                                0020   167 ; unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1.
                                0020   168 ;
                                0020   169 ; NOTE: This procedure disables floating point underflow, enables integer
                                0020   170 ; overflow.
                                0020   171 ;
                                0020   172 ;---
                                0020   173
                                0020   174
                          40FC  0020   175          .ENTRY  MTH$DACOS, ^M<IV, R2, R3, R4, R5, R6, R7>
                                0022   176                                          ; standard call-by-reference entry
                                0022   177                                          ; disable DV (and FU), enable IV
                                0022   178          MTH$FLAG_JACKET                 ; flag that this is a jacket procedure in
                                0022
                6D  00000000'GF  9E   0022          MOVAB   G^MTH$$JACKET_HND, (FP)
```

```
                                0029
                                0029                                    ; set handler address to jacket
                                0029                                    ; handler
                                0029
                                0029    179                             ; case of an error in routine
                                0029    180                             ; If an error, convert signal to user PC
                                0029    181                             ; and resignal
          50    04 BC    70     0029    182             MOVD    @value(AP), R0          ; R0/R1 = !X! = @value(AP)
                   01    10     002D    183             BSBB    MTH$DACOS_R7            ; call special DACOS routine
                         04     002F    184             RET                             ; return with result in R0/R1
                                0030    185
```

```
                                0030   187          .SBTTL  MTH$DACOS_R7 - Special DACOS routine
                                0030   188
                                0030   189 ; Special DACOS - used by the standard routine and direct JSB call.
                                0030   190 ;
                                0030   191 ; CALLING SEQUENCE:
                                0030   192 ;       save anything needed in R0:R7
                                0030   193 ;       MOVD    ...,R0                  ; input in R0/R1
                                0030   194 ;       JSB     MTH$DACOS_R7
                                0030   195 ;       RSB                             ; return with result in R0/R1
                                0030   196 ;
                                0030   197
                                0030   198 MTH$DACOS_R7::                          ; special DACOS routine
                                0030   199 MTH$DACOS_R9::                          ; Release 1 name
                          56    0030   200          MOVD    R0,R6                   ; save X in R6/R7
                          05    0033   201          BNEQ    TEST_FOR_1.0            ; branch if !X! > 0
                                0035   202
                                0035   203 ;
                                0035   204 ; X = 0
                                0035   205 ;
                                0035   206
                     50   C8 AF 0035   207          MOVD    D_PI_OVER_2,R0          ; R0/R1 = PI/2
                          05    0039   208          RSB                             ; return PI/2 if !X! = 0
                                003A   209
                                003A   210 ;
                                003A   211 ; 0 < !X!
                                003A   212 ;
                                003A   213
                                003A   214 TEST_FOR_1.0:
                     50   8000 8F AA 003A   215          BICW    #^X8000,R0              ; R0/R1 = !X!
                          08    50 71 003F   216          CMPD    R0,S^#SD_1.0            ; compare !X! with 1.0
                                30 18 0042   217          BGEQ    GEQ_TO_1.0              ; branch if .X! >= 1.0
                                0044   218
                                0044   219 ;
                                0044   220 ; 0 < !X! < 1.0
                                0044   221 ;
                                0044   222
                50   4000 8F    B1 0044   223          CMPW    #^X4000,R0              ; Check for possible loss of
                               0C 14 0049   224          BGTR    1$                      ;   significance
                52   08    50 63 004B   225          SUBD3   R0,#1,R2                ; R2/R3 = 1 - X
                     50    08 60 004F   226          ADDD    #1,R0                   ; R0/R1 = 1 + X
                     50    52 64 0052   227          MULD    R2,R0                   ; R0/R1 = 1 - X^2
                           07 11 0055   228          BRB     2$                      ; Join main flow
                     50    50 64 0057   229 1$:      MULD2   R0,R0                   ; R0/R1 = X**2
                50   08    50 63 005A   230          SUBD3   R0,S^#SD_1.0,R0         ; R0/R1 = 1.0 - X**2
                 00000000'EF    16 005E   231 2$:      JSB     MTH$DSQRT_R5            ; R0/R1 = DSQRT(1-X**2)
                     50    56 66 0064   232          DIVD    R6,R0                   ; R0/R1 = DSQRT(1-X**2)/X
                           56 DD 0067   233          PUSHL   R6                      ; save sign of X for sign test
                 00000000'EF    16 0069   234          JSB     MTH$DATAN_R7            ; R0/R1 = DATAN(DSQRT(1-X**2)/X)
                     56    8E DO 006F   235          MOVL    (SP)+,R6                ; restore sign of X
                           04 11 0072   236          BRB     TEST_SIGN               ; branch to TEST_SIGN
                                0074   237
                                0074   238 ;
                                0074   239 ; 1 =< !X!
                                0074   240 ;
                                0074   241
                                0074   242 GEQ_TO_1.0:
                           0B 14 0074   243          BGTR    ERROR                   ; branch to ERROR if !X! > 1.0
```

```
                                0076   244
                                0076   245
                                0076   246 ;
                                0076   247 ; |X| = 1.0
                                0076   248 ;
                                0076   249
                     50   7C    0076   250          CLRD    R0                          ; R0/R1 = 0
                                0078   251
                                0078   252
                                0078   253 ;
                                0078   254 ; Test the sign of X in order to decide if add PI to the result
                                0078   255 ;
                                0078   256
                                0078   257 TEST_SIGN:
                     56   73    0078   258          TSTD    R6                          ; test the sign of X
                     04   18    007A   259          BGEQ    10$                         ; branch if X > 0
                50   89 AF     007C   260          ADDD    D_PI, R0                    ; add PI to R0/R1 if X < 0
                                0080   261 10$:     RSB                                 ; return with result in R0/R1
                                0081   262
                                0081   263 ;
                                0081   264 ; 1 < |X|, error
                                0081   265 ;
                                0081   266
                     6E   DD    0081   267 ERROR:  PUSHL   (SP)                        ; return PC from JSB routine
              7E   00'8F   9A    0083   268          MOVZBL  #MTH$K_INVARGMAT, -(SP)     ; condition value
           50   01   0F   79    0087   269          ASHQ    #15, #1, R0                 ; R0 = result = reserved operand -0.0
                                008B   270                                              ; goes to signal mechanism vector
                                008B   271                                              ; (CHF$L_MCH_R0/R1) so error handler
                                008B   272                                              ; can modify the result.
00000000'GF     02   FB    008B   273          CALLS   #2, G^MTH$$SIGNAL               ; signal error and use real user's PC
                                0092   274                                              ; independent of CALL vs JSB
                     05    0092   275          RSB                                     ; return - R0 restored from CHF$L_MCH_R0/R1
                                0093   276
                                0093   277
```

```
                     0093   279          .SBTTL  MTH$DACOSD  - Standard Double Precision Floating DACOSD
                     0093   280
                     0093   281
                     0093   282 ;++
                     0093   283 ; FUNCTIONAL DESCRIPTION:
                     0093   284 ;
                     0093   285 ; DACOSD  - double precision floating point function
                     0093   286 ;
                     0093   287 ; DACOSD(X) is computed as:
                     0093   288 ;
                     0093   289 ;       If X = 0, then DACOSD(X) = 90.
                     0093   290 ;       If X = 1, then DACOSD(X) = 0.
                     0093   291 ;       If X = -1, then DACOSD(X) = 180.
                     0093   292 ;       If 0 < X < 1, then DACOSD(X) = ATAN(SQRT(1-X**2)/X).
                     0093   293 ;       If 1/2 < X < 1, then DACOS(X) = ATAN(SQRT((1-X)*(1+x))/X).
                     0093   294 ;       If -1/2 < X < 0, then DACOSD(X) = ATAN(SQRT(1-X**2)/X) + 180.
                     0093   295 ;       If -1 < X =< -1/2, then DACOS(X) = ATAN(SQRT((1-X)*(1+x))/X) + 180.
                     0093   296 ;       If 1 < !X!, error.
                     0093   297 ;
                     0093   298 ; CALLING SEQUENCE:
                     0093   299 ;
                     0093   300 ;       DACOSD.wd.v = MTH$DACOSD(x.rd.r)
                     0093   301 ;
                     0093   302 ; INPUT PARAMETERS:
                     0093   303
           00000004  0093   304          LONG = 4                        ; define longword multiplier
           00000004  0093   305          x = 1 * LONG                    ; Contents of x is the argument
                     0093   306
                     0093   307 ; IMPLICIT INPUTS:      none
                     0093   308 ;
                     0093   309 ; OUTPUT PARAMETERS:
                     0093   310 ;
                     0093   311 ;       VALUE:  double precision floating arc-cosine of the argument
                     0093   312 ;
                     0093   313 ; IMPLICIT OUTPUTS:     none
                     0093   314 ;
                     0093   315 ; COMPLETION CODES:     none
                     0093   316 ;
                     0093   317 ; SIDE EFFECTS:
                     0093   318 ;
                     0093   319 ; Signals: MTH$_INVARG if !X! > 1 with reserved operand in R0/R1 (co180ed
                     0093   320 ; to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                     0093   321 ; Associated message is: "INVALID ARGUMENT". Result is reserved operand -0.0
                     0093   322 ; unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1.
                     0093   323 ;
                     0093   324 ; NOTE: This procedure disables floating point underflow, enables integer
                     0093   325 ; overflow.
                     0093   326 ;
                     0093   327 ;--
                     0093   328
                     0093   329
                 40FC 0093   330          .ENTRY  MTH$DACOSD, ^M<IV, R2, R3, R4, R5, R6, R7>
                     0095   331                                          ; standard call-by-reference entry
                     0095   332                                          ; disable DV (and FU), enable IV
                     0095   333          MTH$FLAG_JACKET                 ; flag that this is a jacket procedure in
                     0095
     6D  00000000'GF  9E  0095          MOVAB   G^MTH$$JACKET_HND, (FP)
```

```
                                009C                                    ; set handler address to jacket
                                009C                                    ; handler
                                009C
                                009C    334                             ; case of an error in routine
                                009C    335                             ; If an error, convert signal to user PC
                                009C    336                             ; and resignal
                50    04 BC     009C    337             MOVD    @value(AP), R0          ; R0/R1 = !X! = @value(AP)
                      01 10     00A0    338             BSBB    MTH$DACOSD_R7           ; call special DACOSD routine
                         04     00A2    339             RET                             ; return with result in R0/R1
                                00A3    340
```

```
                                00A3    342          .SBTTL  MTH$DACOSD_R7 - Special DACOSD routine
                                00A3    343
                                00A3    344 ; Special DACOSD - used by the standard routine and direct JSB call.
                                00A3    345 ;
                                00A3    346 ; CALLING SEQUENCE:
                                00A3    347 ;       save anything needed in R0:R7
                                00A3    348 ;       MOVD    ...,R0                  ; input in R0/R1
                                00A3    349 ;       JSB     MTH$DACOSD_R7
                                00A3    350 ;       RSB                             ; return with result in R0/R1
                                00A3    351 ;
                                00A3    352
                                00A3    353 MTH$DACOSD_R7::                         ; special DACOSD routine
          56    50    70        00A3    354          MOVD    R0, R6                 ; save X in R6/R7
                06    12        00A6    355          BNEQ    D_TEST_FOR_1.0         ; branch if |X| > 0
                                00A8    356
                                00A8    357 ;
                                00A8    358 ; X = 0
                                00A8    359 ;
                                00A8    360
          50    FF64  CF  70    00A8    361          MOVD    D_90, R0               ; R0/R1 = 90
                          05    00AD    362          RSB                            ; return 90 if |X| = 0
                                00AE    363
                                00AE    364 ;
                                00AE    365 ; 0 < |X|
                                00AE    366 ;
                                00AE    367
                                00AE    368 D_TEST_FOR_1.0:
          50    8000  8F    AA  00AE    369          BICW    #^X8000, R0            ; R0/R1 = |X|
                08    50    71  00B3    370          CMPD    R0, S^#SD_1.0          ; compare |X| with 1.0
                      30    18  00B6    371          BGEQ    D_GEQ_TO_1.0           ; branch if |X| >= 1.0
                                00B8    372
                                00B8    373 ;
                                00B8    374 ; 0 < |X| < 1.0
                                00B8    375 ;
                                00B8    376
     50   4000  8F    B1        00B8    377          CMPW    #^X4000, R0            ; Check for possible loss of
                0C    14        00BD    378          BGTR    1$                     ;   significance
     52   08    50    63        00BF    379          SUBD3   R0, #1, R2             ; R2/R3 = 1 - X
          50    08    60        00C3    380          ADDD    #1, R0                 ; R0/R1 = 1 + X
          50    52    64        00C6    381          MULD    R2, R0                 ; R0/R1 = 1 - X^2
                07    11        00C9    382          BRB     2$                     ; Join main flow
          50    50    64        00CB    383 1$:      MULD2   R0, R0                 ; R0/R1 = X**2
     50   08    50    63        00CE    384          SUBD3   R0, S^#SD_1.0, R0      ; R0/R1 = 1.0 - X**2
 00000000'EF                    00D2    385 2$:      JSB     MTH$DSQRT_R5           ; R0/R1 = DSQRT(1-X**2)
          50    56    66        00D8    386          DIVD    R6, R0                 ; R0/R1 = DSQRT(1-X**2)/X
                56    DD        00DB    387          PUSHL   R6                     ; save sign of X for sign test
 00000000'EF                    00DD    388          JSB     MTH$DATAND_R7          ; R0/R1 = DATAND(DSQRT(1-X**2)/X)
          56    8E    D0        00E3    389          MOVL    (SP)+, R6              ; restore sign of X
                07    11        00E6    390          BRB     D_TEST_SIGN            ; branch to D_TEST_SIGN
                                00E8    391
                                00E8    392 ;
                                00E8    393 ; 1 =< |X|
                                00E8    394 ;
                                00E8    395
                                00E8    396 D_GEQ_TO_1.0:
                03    13        00E8    397          BEQL    10$
              FF94    31        00EA    398          BRW     ERROR                  ; branch to ERROR if |X| > 1.0
```

```
                     00ED    399
                     00ED    400
                     00ED    401 ;
                     00ED    402 ; |X| = 1.0
                     00ED    403 ;
                     00ED    404
                50   7C  00ED    405 10$:    CLRD    R0                      ; R0/R1 = 0
                     00EF    406
                     00EF    407
                     00EF    408 ;
                     00EF    409 ; Test the sign of X in order to decide if add 180 to the result
                     00EF    410 ;
                     00EF    411
                     00EF    412 D_TEST_SIGN:
                56   73  00EF    413         TSTD    R6                      ; test the sign of X
                05   18  00F1    414         BGEQ    10$                     ; branch if X > 0
          50    FF21 CF  00F3    415         ADDD    D_180, R0               ; add 180 to R0/R1 if X < 0
                05   00F8    416 10$:    RSB                             ; return with result in R0/R1
                     00F9    417
                     00F9    418
                     00F9    419         .END
```

```
D_180                   00000018 R      01
D_90                    00000010 R      01
D_GEQ_TO_1.0            000000E8 R      01
D_PI                    00000008 R      01
D_PI_OVER_2             00000000 R      01
D_TEST_FOR_1.0          000000AE R      01
D_TEST_SIGN             000000EF R      01
ERROR                   00000081 R      01
GEQ_TO_1.0              00000074 R      01
LONG                =   00000004
MTH$$JACKET_HND         ********    X   01
MTH$$SIGNAL             ********    X   00
MTH$DACOS               00000020 RG     01
MTH$DACOSD              00000093 RG     01
MTH$DACOSD_R7           000000A3 RG     01
MTH$DACOS_R7            00000030 RG     01
MTH$DACOS_R9            00000030 RG     01
MTH$DATAND_R7           ********    X   00
MTH$DATAN_R7            ********    X   00
MTH$DSQRT_R5            ********    X   00
MTH$K_INVARGMAT         ********    X   00
SD_1.0              =   00004080
TEST_FOR_1.0            0000003A R      01
TEST_SIGN               00000078 R      01
VALUE               =   00000004
```

+----------------+
! Psect synopsis !
+----------------+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_MTH$CODE	000000F9 (249.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.11	00:00:00.93
Command processing	126	00:00:00.65	00:00:04.14
Pass 1	96	00:00:01.08	00:00:04.57
Symbol table sort	0	00:00:00.01	00:00:00.17
Pass 2	92	00:00:00.97	00:00:04.20
Symbol table output	4	00:00:00.03	00:00:00.11
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	353	00:00:02.87	00:00:14.14

The working set limit was 900 pages.
6232 bytes (13 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 26 non-local and 7 local symbols.
479 source lines were read in Pass 1, producing 14 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

D 14

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

Macro library name                   Macros defined
------------------                   --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

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

MTHCVTDG
LIS

MTHDACOS
LIS

MTHCGABS
LIS

MTHCDSINC
LIS

MTHCLOG
LIS

MTHDASIN
LIS

MTHCGLOG
LIS

MTHCONVER
LIS

MTHDATAN
LIS

MTHCSQRT
LIS

MTHCEXP
LIS

MTHCGSQRT
LIS

MTHCGEXP
LIS

MTHCSINCO
LIS

MTHCONJG
LIS

MTHCDSQRT
LIS

MTHCGSINC
LIS

MTHCOSH
LIS