```
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
```

```
MM      MM  TTTTTTTTTT  HH      HH  HH      HH    AAAAAA      SSSSSSSS    IIIIII    NN      NN
MM      MM  TTTTTTTTTT  HH      HH  HH      HH    AAAAAA      SSSSSSSS    IIIIII    NN      NN
MMMM  MMMM      TT      HH      HH  HH      HH  AA      AA  SS              II      NN      NN
MMMM  MMMM      TT      HH      HH  HH      HH  AA      AA  SS              II      NN      NN
MM  MM  MM      TT      HH      HH  HH      HH  AA      AA  SS              II      NNNN    NN
MM  MM  MM      TT      HH      HH  HH      HH  AA      AA  SS              II      NNNN    NN
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  AA      AA    SSSSSS        II      NN  NN  NN
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  AA      AA    SSSSSS        II      NN  NN  NN
MM      MM      TT      HH      HH  HH      HH  AAAAAAAAAA            SS    II      NN    NNNN
MM      MM      TT      HH      HH  HH      HH  AAAAAAAAAA            SS    II      NN    NNNN
MM      MM      TT      HH      HH  HH      HH  AA      AA            SS    II      NN      NN      ....
MM      MM      TT      HH      HH  HH      HH  AA      AA            SS    II      NN      NN      ....
MM      MM      TT      HH      HH  HH      HH  AA      AA  SSSSSSSS      IIIIII    NN      NN      ....
MM      MM      TT      HH      HH  HH      HH  AA      AA  SSSSSSSS      IIIIII    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$HASIN       ; H Floating Point Arcsine routine
0000     2                                  ; (HASIN,HASIND)
0000     3          .IDENT /1-004/          ; File: MTHHASIN.MAR  RNH1004
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$HASIN is a function which returns the H floating point arcsine
0000    34 ; in radians of its H floating point argument. The call is standard call-
0000    35 ; by-reference.
0000    36 ;
0000    37 ; MTH$HASIND is a function which returns the H floating point arcsine
0000    38 ; in degrees of its H floating point argument. The call is standard call-
0000    39 ; by-reference.
0000    40 ;
0000    41 ;--
0000    42 ;
0000    43 ; VERSION: 1
0000    44 ;
0000    45 ; HISTORY:
0000    46 ; AUTHOR:
0000    47 ;       John A. Wheeler, 20-Oct-1979: Version 1
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 ; Edit History for Version 1 of MTH$HASIN
0000     58 ;
0000     59 ; 1-001 - Adapted from MTH$GASIN version 1-002.  JAW 20-Oct-1979
0000     60 ; 1-002 - Added degree entry points.  RNH 29-MAR-1981
0000     61 ; 1-003 - Change shared external references to G^ RNH 25-Sep-81
0000     62 ; 1-004 - Eliminated symbolic short literals.   RNH 15-Oct-81
```

```
                                                .SBTTL  DECLARATIONS    ; Declarative Part of Module
                                        ;
                                        ; INCLUDE FILES:          OTSPARAMS.MAR
                                        ;
                                        ; EXTERNAL SYMBOLS:
                                        
                                                .DSABL  GBL                     ; Force error for undefineds
                                                .EXTRN  MTH$HSQRT_R8            ; Square root routine
                                                .EXTRN  MTH$HATAN_R8            ; Arctangent routine
                                                .EXTRN  MTH$HATAND_R8           ; Arctangent routine
                                                .EXTRN  MTH$$SIGNAL             ; Math signal routine
                                                .EXTRN  MTH$K_INVARGMAT         ; Error code
                                        
                                        ;
                                        ; EQUATED SYMBOLS:
                                        
                                        
                                        ;
                                        ; MACROS:         none
                                        ;
                                        ; PSECT DECLARATIONS:
                                        
                                00000000        .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                                                                        ; Program section for math routines
                                        ;
                                        ; OWN STORAGE:  none
                                        ;
                                        ;
                                        ; CONSTANTS:
                                        ;
                                        
                                        H_PI_OVER_2:
42D1B544 921F4001               0000            .LONG   ^X921F4001, ^X42D1B544  ;  1.57079632679489661923132169163975144209
01B8C517 898C8469               0008            .LONG   ^X898C8469, ^X01B8C517
                                0010    H_90:
00000000 68004007               0010            .LONG   ^X68004007, ^X0         ;  90
00000000 00000000               0018            .LONG   ^X0,        ^X0
                                0020
```

```
                        0020   104          .SBTTL  MTH$HASIN  - Standard H Floating Arcsine
                        0020   105
                        0020   106
                        0020   107 ;++
                        0020   108 ; FUNCTIONAL DESCRIPTION:
                        0020   109 ;
                        0020   110 ; HASIN  - H floating point arcsine function
                        0020   111 ;
                        0020   112 ; HASIN(X) is computed as:
                        0020   113 ;
                        0020   114 ;       If X = 0, then HASIN(X) = 0.
                        0020   115 ;       If X = 1, then HASIN(X) = PI/2.
                        0020   116 ;       If X = -1, then HASIN(X) = -PI/2.
                        0020   117 ;       If 0 < |X| < 1, then HASIN(X) = ATAN(X/SQRT(1-X**2)).
                        0020   118 ;       If 1 < |X|, error.
                        0020   119 ;
                        0020   120 ; CALLING SEQUENCE:
                        0020   121 ;
                        0020   122 ;       hasin.wh.v = MTH$HASIN(x.rh.r)
                        0020   123 ;
                        0020   124 ;               -or-
                        0020   125 ;
                        0020   126 ;       CALL MTH$HASIN(hasin.wh.r, x.rh.r)
                        0020   127 ;
                        0020   128 ;       Because an H-floating result cannot be expressed in 64 bits, it is
                        0020   129 ;       returned as the first argument, with the input parameter displaced
                        0020   130 ;       to the second argument, in accordance with the Procedure Calling
                        0020   131 ;       Standard.
                        0020   132 ;
                        0020   133 ; INPUT PARAMETERS:
                        0020   134 ;
00000004                0020   135          LONG = 4                        ; Define longword multiplier
00000008                0020   136          x = 2 * LONG                    ; Contents of x is the argument
                        0020   137 ;
                        0020   138 ; IMPLICIT INPUTS:      none
                        0020   139 ;
                        0020   140 ; OUTPUT PARAMETERS:
                        0020   141 ;
00000004                0020   142          hasin = 1 * LONG                ; hasin is the result
                        0020   143 ;
                        0020   144 ;       VALUE:  H floating arcsine of the argument
                        0020   145 ;
                        0020   146 ; IMPLICIT OUTPUTS:     none
                        0020   147 ;
                        0020   148 ; COMPLETION CODES:     none
                        0020   149 ;
                        0020   150 ; SIDE EFFECTS:
                        0020   151 ;
                        0020   152 ; Signals: MTH$_INVARGMAT if |X| > 1 with reserved operand in R0/R3
                        0020   153 ; (copied to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                        0020   154 ; Associated message is: "INVALID ARGUMENT". Result is reserved operand
                        0020   155 ; -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1.
                        0020   156 ;
                        0020   157 ; NOTE: This procedure disables floating point underflow, enables integer
                        0020   158 ; overflow.
                        0020   159 ;
                        0020   160 ;---
```

```
                        0020   161
                        0020   162
                 41FC   0020   163          .ENTRY  MTH$HASIN, ^M<IV, R2, R3, R4, R5, R6, R7, R8>
                        0022   164                                              ; Standard call-by-reference entry
                        0022   165                                              ; Disable DV (and FU), enable IV
                        0022   166          MTH$FLAG_JACKET                     ; Flag that this is a jacket procedure in
                        0022                                                     ;
    6D  00000000'GF      9E 0022            MOVAB   G^MTH$$JACKET_HND, (FP)
                        0029                                                     ; set handler address to jacket
                        0029                                                     ; handler
                        0029
                        0029   167                                              ; case of an error in routine
                        0029   168                                              ; If an error, convert signal to user PC
                        0029   169                                              ; and resignal
     50    08 BC 70FD   0029   170          MOVH    @x(AP), R0                  ; R0/R3 = X = @x(AP)
           06   10      002E   171          @SBB    MTH$HASIN_R8                ; Call special HASIN routine
     04 BC   50 7DFD    0030   172          MOVQ    R0, @hasin(AP)              ; Store result in first argument
                 04    0035   173          RET                                 ; Return to caller
                        0036   174
```

```
                        0036   176          .SBTTL  MTH$HASIN_R8 - Special HASIN routine
                        0036   177
                        0036   178 ; Special HASIN - used by the standard routine and direct JSB call.
                        0036   179 ;
                        0036   180 ; CALLING SEQUENCE:
                        0036   181 ;       save anything needed in R0:R8
                        0036   182 ;       MOVH    ...,R0                  ; Input in R0/R3
                        0036   183 ;       JSB     MTH$HASIN_R8
                        0036   184 ;       return with result in R0/R3
                        0036   185 ;
                        0036   186
                        0036   187 MTH$HASIN_R8::                           ; Special HASIN routine
                50   B5 0036   188          TSTW    R0                      ; Compare X with 0
                39   13 0038   189          BEQL    RETURN                  ; If X = 0, return HASIN(0) = 0
                        003A   190
                        003A   191 ;
                        003A   192 ; 0 < |X|
                        003A   193 ;
                        003A   194
          7E    50 70FD 003A   195          MOVH    R0,-(SP)                ; stack = R0/R3 = X
    50    8000  8F   AA 003E   196          BICW    #^X8000,R0              ; R0/R3 = |X|
          08    50 71FD 0043   197          CMPH    R0,#1                   ; Compare  |X| with 1.0
                1A   18 0047   198          BGEQ    GEQ_TO_1.0              ; Branch if |X| >= 1.0
                        0049   199
                        0049   200 ;
                        0049   201 ; 0 < |X| < 1.0
                        0049   202 ;
                        0049   203
          50    50 64FD 0049   204          MULH2   R0,R0                   ; R0/R3 = X**2
    50    08    50 63FD 004D   205          SUBH3   R0,#1,R0                ; R0/R3 = 1.0 - X**2
 00000000'EF      16   0052   206          JSB     MTH$HSQRT_R8            ; R0/R3 = HSQRT(1-X**2)
    50  8E    50 67FD 0058   207          DIVH3   R0,(SP)+,R0             ; R0/R3 = X/HSQRT(1-X**2)
                        005D   208                                          ; Also clear stack
 00000000'GF      17   005D   209          JMP     G^MTH$HATAN_R8          ; R0/R3 = HATAN(X/HSQRT(1-X**2))
                        0063   210
                        0063   211 ;
                        0063   212 ; 1 =< |X|
                        0063   213 ;
                        0063   214
                        0063   215 GEQ_TO_1.0:
                0F   14 0063   216          BGTR    ERROR                   ; Branch to ERROR if |X| > 1.0
                        0065   217
                        0065   218 ;
                        0065   219 ; |X| = 1.0
                        0065   220 ;
                        0065   221
    50    97    AF 7DFD 0065   222          MOVO    H_PI_OVER_2,R0          ; R0/R3 = PI/2
                8E 73FD 006A   223          TSTH    (SP)+                   ; Test the sign of X and clear stack
                04   18 006D   224          BGEQ    RETURN                  ; Branch if X > 0
          50    50 72FD 006F   225          MNEGH   R0,R0                   ; R0/R3 = -PI/2
                     05 0073   226 RETURN:  RSB
                        0074   227
                        0074   228 ;
                        0074   229 ; 1 < |X|, error
                        0074   230 ;
                        0074   231
                8E 73FD 0074   232 ERROR:  TSTH    (SP)+                   ; Clear stack
```

```
                    6E   DD  0077   233          PUSHL   (SP)                          ; Return PC from JSB routine
                7E  00'8F   9A  0079   234          MOVZBL  #MTH$K_INVARGMAT, -(SP)       ; Condition value
              50    01    0F  79  007D   235          ASHQ    #15, #T, R0                   ; R0 = result = reserved operand -0.0
                                0081   236                                                ; Goes to signal mechanism vector
                                0081   237                                                ; (CHF$L_MCH_R0/R1) so error handler
                                0081   238                                                ; Can modify the result.
                    52   7C  0081   239          CLRQ    R2                            ;
                                                                                          ; ...
        00000000'GF     02  FB  0083   240          CALLS   #2, G^MTH$$SIGNAL             ; Signal error and use real user's PC
                                008A   241                                                ; Independent of CALL vs JSB
                                05  008A   242          RSB                           ; Return - R0 restored from CHF$L_MCH_R0/R1
                                008B   243
```

```
                     008B   245          .SBTTL  MTH$HASIND  - Standard H Floating Arcsine
                     008B   246
                     008B   247
                     008B   248 ;++
                     008B   249 ; FUNCTIONAL DESCRIPTION:
                     008B   250 ;
                     008B   251 ; HASIND  - H floating point arcsine function
                     008B   252 ;
                     008B   253 ; HASIND(X) is computed as:
                     008B   254 ;
                     008B   255 ;       If X = 0, then HASIND(X) = 0.
                     008B   256 ;       If X = 1, then HASIND(X) = 90.
                     008B   257 ;       If X = -1, then HASIND(X) = -90.
                     008B   258 ;       If 0 < |X| < 1, then HASIND(X) = ATAND'X/SQRT(1-X**2)).
                     008B   259 ;       If 1 < |X|, error.
                     008B   260 ;
                     008B   261 ; CALLING SEQUENCE:
                     008B   262 ;
                     008B   263 ;       hasino.wh.v = MTH$HASIND(x.rh.r)
                     008B   264 ;
                     008B   265 ;               -or-
                     008B   266 ;
                     008B   267 ;       CALL MTH$HASIND(hasind.wh.r, x.rh.r)
                     008B   268 ;
                     008B   269 ;       Because an H-floating result cannot be expressed in 64 bits, it is
                     008B   270 ;       returned as the first argument, with the input parameter displaced
                     008B   271 ;       to the second argument, in accordance with the Procedure Calling
                     008B   272 ;       Standard.
                     008B   273 ;
                     008B   274 ; INPUT PARAMETERS:
                     008B   275 ;
00000004             008B   276          LONG = 4                        ; Define longword multiplier
00000008             008B   277          x = 2 * LONG                    ; Contents of x is the argument
                     008B   278 ;
                     008B   279 ; IMPLICIT INPUTS:      none
                     008B   280 ;
                     008B   281 ; OUTPUT PARAMETERS:
                     008B   282 ;
00000004             008B   283          hasind = 1 * LONG               ; hasind is the result
                     008B   284 ;
                     008B   285 ;       VALUE:  H floating arcsine of the argument
                     008B   286 ;
                     008B   287 ; IMPLICIT OUTPUTS:     none
                     008B   288 ;
                     008B   289 ; COMPLETION CODES:     none
                     008B   290 ;
                     008B   291 ; SIDE EFFECTS:
                     008B   292 ;
                     008B   293 ; Signals: MTH$_INVARGMAT if |X| > 1 with reserved operand in R0/R3
                     008B   294 ; (copied to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                     008B   295 ; Associated message is: "INVALID ARGUMENT". Result is reserved operand
                     008B   296 ; -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1.
                     008B   297 ;
                     008B   298 ; NOTE: This procedure disables floating point underflow, enables integer
                     008B   299 ; overflow.
                     008B   300 ;
                     008B   301 ;---
```

```
                                008B   302
                                008B   303
                         41FC   008B   304          .ENTRY  MTH$HASIND, ^M<IV, R2, R3, R4, R5, R6, R7, R8>
                                008D   305                                                 ; Standard call-by-reference entry
                                008D   306                                                 ; Disable DV (and FU), enable IV
                                008D   307          MTH$FLAG_JACKET                        ; Flag that this is a jacket procedure in
                                008D                                                       ;
    6D    00000000'GF    9E    008D                MOVAB   G^MTH$$JACKET_HND, (FP)
                                0094                                                       ; set handler address to jacket
                                0094                                                       ; handler
                                0094
                                0094   308                                                 ; case of an error in routine
                                0094   309                                                 ; If an error, convert signal to user PC
                                0094   310                                                 ; and resignal
     50    08 BC 70FD           0094   311          MOVH    @x(AP), R0                     ; R0/R3 = X = @x(AP)
              06   10           0099   312          BSBB    MTH$HASIND_R8                  ; Call special HASIND routine
     04 BC    50 7DFD           009B   313          MOVQ    R0, @hasind(AP)                ; Store result in first argument
                04              00A0   314          RET                                    ; Return to caller
                                00A1   315
```

```
                        00A1   317          .SBTTL  MTH$HASIND_R8 - Special HASIND routine
                        00A1   318
                        00A1   319 ; Special HASIND - used by the standard routine and direct JSB call.
                        00A1   320 ;
                        00A1   321 ; CALLING SEQUENCE:
                        00A1   322 ;       save anything needed in R0:R8
                        00A1   323 ;       MOVH    ...,R0                  ; Input in R0/R3
                        00A1   324 ;       JSB     MTH$HASIND_R8
                        00A1   325 ;       return with result in R0/R3
                        00A1   326 ;
                        00A1   327
                        00A1   328 MTH$HASIND_R8::                          ; Special HASIND routine
                 50  B5 00A1   329          TSTW    R0                      ; Compare X with 0
                 3A  13 00A3   330          BEQL    D_RETURN                ; If X = 0, return HASIND(0) = 0
                        00A5   331
                        00A5   332 ;
                        00A5   333 ; 0 < |X|
                        00A5   334 ;
                        00A5   335
              7E  50 70FD 00A5   336          MOVH    R0,-(SP)                ; stack = R0/R3 = X
        50    8000 8F  AA 00A9   337          BICW    #^X8000,R0              ; R0/R3 = |X|
              08  50 71FD 00AE   338          CMPH    R0,#1                   ; Compare |X| with 1.0
                  1A  18 00B2   339          BGEQ    D_GEQ_TO_1.0            ; Branch if |X| >= 1.0
                        00B4   340
                        00B4   341 ;
                        00B4   342 ; 0 < |X| < 1.0
                        00B4   343 ;
                        00B4   344
              50  50 64FD 00B4   345          MULH2   R0,R0                   ; R0/R3 = X**2
        50    08  50 63FD 00B8   346          SUBH3   R0,#1,R0                ; R0/R3 = 1.0 - X**2
         00000000'EF    16 00BD   347          JSB     MTH$HSQRT_R8            ; R0/R3 = HSQRT(1-X**2)
        50    8E  50 67FD 00C3   348          DIVH3   R0,(SP)+,R0             ; R0/R3 = X/HSQRT(1-X**2)
                        00C8   349                                          ; Also clear stack
         00000000'GF    17 00C8   350          JMP     G^MTH$HATAND_R8         ; R0/R3 = HATAND(X/HSQRT(1-X**2))
                        00CE   351
                        00CE   352 ;
                        00CE   353 ; 1 =< |X|
                        00CE   354 ;
                        00CE   355
                        00CE   356 D_GEQ_TO_1.0:
                  A4    14 00CE   357          BGTR    ERROR                   ; Branch to ERROR if |X| > 1.0
                        00D0   358
                        00D0   359 ;
                        00D0   360 ; |X| = 1.0
                        00D0   361 ;
                        00D0   362
        50    FF3B CF 7DFD 00D0   363          MOVO    H_90,R0                 ; R0/R3 = 90
                     8E 73FD 00D6   364          TSTH    (SP)+                   ; Test the sign of X and clear stack
                     04  18 00D9   365          BGEQ    D_RETURN                ; Branch if X > 0
              50  50 72FD 00DB   366          MNEGH   R0,R0                   ; R0/R3 = -90
                        05 00DF   367 D_RETURN:       RSB
                        00E0   368
                        00E0   369          .END
```

```
D_GEQ_TO_1.0        000000CE R      01
D_RETURN            000000DF R      01
ERROR               00000074 R      01
GEQ_TO_1.0          00000063 R      01
HASIN             = 00000004
HASIND            = 00000004
H_90                00000010 R      01
H_PI_OVER_2         00000000 R      01
LONG              = 00000004
MTH$$JACKET_HND     ********   X    01
MTH$$SIGNAL         ********   X    00
MTH$HASIN           00000020 RG     01
MTH$HASIND          0000008B RG     01
MTH$HASIND_R8       000000A1 RG     01
MTH$HASIN_R8        00000036 RG     01
MTH$HATAND_R8       ********   X    00
MTH$HATAN_R8        ********   X    00
MTH$HSQRT_R8        ********   X    00
MTH$K_INVARGMAT     ********   X    00
RETURN              00000073 R      01
X                 = 00000008
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_MTH$CODE	000000E0 (224.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.07	00:00:00.64
Command processing	122	00:00:00.56	00:00:04.29
Pass 1	90	00:00:00.98	00:00:04.08
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	75	00:00:00.81	00:00:03.45
Symbol table output	3	00:00:00.02	00:00:00.03
Psect synopsis output	3	00:00:00.01	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	325	00:00:02.48	00:00:12.62

The working set limit was 750 pages.
5419 bytes (11 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 21 non-local and 0 local symbols.
429 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.

```
                                        +----------------------------+
                                        ! 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$:MTHHASIN/OBJ=OBJ$:MTHHASIN MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```