

[illegible]

3

Sy

MT

MT

MT

MT

MT

MT
MTMT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

MT
MT

MT
MTMT
MT

M1

MI
MIMI
MIM1
M1

10

1

1

10



MM	MM	TTTTTTTTTT	HH	HH	AAAAAA	CCCCCCCC	000000	SSSSSSSS
MM	MM	TTTTTTTTTT	HH	HH	AAAAAA	CCCCCCCC	000000	SSSSSSSS
MMM	MMM	TT	HH	HH	AA	CC	00	SS
MMM	MMM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HHHHHHHHHH	AA	AA	CC	00	SSSSSS
MM	MM	TT	HHHHHHHHHH	AA	AA	CC	00	SSSSSS
MM	MM	TT	HH	HH	AAAAAAAAAA	CC	00	SS
MM	MM	TT	HH	HH	AAAAAAAAAA	CC	00	SS
MM	MM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HH	HH	AA	CC	00	SS
MM	MM	TT	HH	HH	AA	CCCCCCCC	000000	SSSSSSSS
MM	MM	TT	HH	HH	AA	CCCCCCCC	000000	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
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

[illegible]

(2) 54
(3) 77
(4) 119
(5) 182
(6) 273
(7) 336

HISTORY : Detailed Current Edit History
DECLARATIONS : Declarative Part of Module
MTH\$ACOS - Standard Single Precision Floating ACOS
MTH\$ACOS_R4 - Special ACOS routine
MTH\$ACOSD - Standard Single Precision Floating ACOSD
MTH\$ACOSD_R4 - Special ACOSD routine

```
0000 1 .TITLE MTH$ACOS ; Floating Point Arc-cosine routine
0000 2 ; (ACOS,ACOSD)
0000 3 .IDENT /1-014/ ; File: MTHACOS.MAR Edit: JCW1014
0000 4 :
0000 5 :*****
0000 6 :
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :
0000 25 :
0000 26 :*****
0000 27 :
0000 28 :
0000 29 : FACILITY: MATH LIBRARY
0000 30 : ++
0000 31 : ABSTRACT:
0000 32 :
0000 33 : MTH$ACOS is a function which returns the floating point arc-cosine in
0000 34 : radians of its single precision floating point argument. The call is
0000 35 : standard call-by-reference.
0000 36 :
0000 37 : MTH$ACOSD is a function which returns the floating point arc-cosine in
0000 38 : degrees of its single precision floating point argument. The call is
0000 39 : 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$ACOS
0000 60
0000 61 01-2 Changed MOVF #0, R0 into CLRL R0.
0000 62 0-3 MTH$$ERROR changed to MTH$$SIGNAL.
0000 63 MTH$... changed to MTH_....
0000 64 Changed error handling mechanism. Put error result in R0 (or R0:R1)
0000 65 before calling MTH$$SIGNAL in order to allow user modify error result.
0000 66
0000 67 1-006 - Update copyright notice. JBS 16-NOV-78
0000 68 1-007 - Change MTH_INVARG to MTH$K_INVARGMAT. JBS 07-DEC-78
0000 69 1-008 - Add " " to the PSECT directive. JBS 21-DEC-78
0000 70 1-009 - Declare externals. SBL 17-May-1979
0000 71 1-010 - Use MTH$SQRT_R3. SBL 27-Sept-1979
0000 72 1-011 - Change JSB entry to MTH$ACOS R4. RBG 28-Sept-1979
0000 73 1-012 - Added degree entry points. RNH 22-MAR-1981
0000 74 1-013 - Modified computation of 1 - x^2 for |x|>=1/2 RNH 2-Sept-81
0000 75 1-014 - Modified computation of 1 - x^2 for 1/2<=|x|<=0.625 JCW 26-APR-83
```

```
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$SQRT_R3
0000 88      .EXTRN  MTH$ATAN_R4
0000 89      .EXTRN  MTH$ATAN2_R4
0000 90      .EXTRN  MTH$K_INVARGMAT
0000 91      .EXTRN  MTH$SIGNAL
0000 92
0000 93      ; EQUATED SYMBOLS:
0000 94
00004080 0000 95      SF_1.0 = ^F1.0      ; 1.0
0FDB40C9 0000 96      LF_PI_OV_2 = ^0007733a16 + ^0040311
0000 97      ; PI/2
0FDB4149 0000 98      LF_PI = ^0007733a16 + ^0040511
0000 99      ; PI
000043B4 0000 100     LF_90 = ^X000043B4      ; 90
00004434 0000 101     LF_180 = ^X00004434      ; 180
00000004 0000 102     value = 4      ; value.rf.r
0000 103
0000 104      ;
0000 105      ; MACROS:      none
0000 106      ;
0000 107      ; PSECT DECLARATIONS:
0000 108
00000000 0000 109     .PSECT  _MTH$CODE      PIC,SHR,LONG,EXE,NOWRT
0000 110      ; program section for math routines
0000 111      ;
0000 112      ; OWN STORAGE:  none
0000 113      ;
0000 114      ;
0000 115      ; CONSTANTS:
0000 116      ;
0000 117
```



```
0000 119      .SBTTL MTH$ACOS - Standard Single Precision Floating ACOS
0000 120
0000 121
0000 122 :++
0000 123 : FUNCTIONAL DESCRIPTION:
0000 124 :
0000 125 : ACOS - single precision floating point function
0000 126 :
0000 127 : ACOS(X) is computed as:
0000 128 :
0000 129 :     If X = 0, then ACOS(X) = PI/2.
0000 130 :     If X = 1, then ACOS(X) = 0.
0000 131 :     If X = -1, then ACOS(X) = PI.
0000 132 :     If 0 < X < 1/2, then ACOS(X) = ATAN(SQRT(1-X**2)/X).
0000 133 :     If 1/2 < X < 1, then ACOS(X) = ATAN(SQRT((1-X)*(1+X))/X).
0000 134 :     If -1/2 < X < 0, then ACOS(X) = ATAN(SQRT(1-X**2)/X) + PI.
0000 135 :     If -1/2 < X < -1, then ACOS(X) = ATAN(SQRT((1-X)*(1+X))/X) + PI.
0000 136 :     If 1 < |X|, error.
0000 137 :
0000 138 : CALLING SEQUENCE:
0000 139 :
0000 140 :     ACOS.wf.v = MTH$ACOS(x.rf.r)
0000 141 :
0000 142 : INPUT PARAMETERS:
0000 143 :
00000004 0000 144 :     LONG = 4 ; define longword multiplier
00000004 0000 145 :     x = 1 * LONG ; Contents of x is the argument
0000 146 :
0000 147 : IMPLICIT INPUTS: none
0000 148 :
0000 149 : OUTPUT PARAMETERS:
0000 150 :
0000 151 :     VALUE: floating arc-cosine of the argument
0000 152 :
0000 153 : IMPLICIT OUTPUTS: none
0000 154 :
0000 155 : COMPLETION CODES: none
0000 156 :
0000 157 : SIDE EFFECTS:
0000 158 :
0000 159 : Signals: MTH$_INVARG if |X| > 1 with reserved operand in R0 (copied to
0000 160 : the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
0000 161 : Associated message is: "INVALID ARGUMENT". Result is reserved operand -0.0
0000 162 : unless a user supplied (or any) error handler changes CHF$MCH_R0/R1.
0000 163 :
0000 164 : NOTE: This procedure disables floating point underflow, enables integer
0000 165 : overflow.
0000 166 :
0000 167 : ---
0000 168 :
0000 169 :
401C 0000 170 : .ENTRY MTH$ACOS, ^M<IV, R2, R3, R4>
0002 171 : ; standard call-by-reference entry
0002 172 : ; disable DV (and FU), enable IV
0002 173 : MTH$FLAG_JACKET ; flag that this is a jacket procedure in
0002
6D 00000000'GF 9E 0002 MOVAB G^MTH$$JACKET_HND, (FP)
```

```

; Floating Point Arc-cosine routine      G 5      16-SEP-1984 01:02:15  VAX/VMS Macro V04-00      Page 5
MTH$ACOS - Standard Single Precision FL  6-SEP-1984 11:20:11  [MTHRTL.SRC]MTHACOS.MAR;1      (4)

```

			0009			; set handler address to jacket
			0009			; handler
			0009			
			0009	174		; case of an error in routine
			0009	175		; If an error, convert signal to user PC
			0009	176		; and resignal
50	04	BC	50	0009	177	MOVF @value(AP), R0
		01	10	000D	178	BSBB MTH\$ACOS_R4
			04	000F	179	RET
			0010	180		; return with result in R0


```
0010 182 .SBTTL MTH$ACOS_R4 - Special ACOS routine
0010 183
0010 184 ; Special ACOS - used by the standard routine, direct JSB call.
0010 185
0010 186 ; CALLING SEQUENCE:
0010 187 ; save anything needed in R0:R4
0010 188 ; MOVF R0 ; input in R0
0010 189 ; JSB MTH$ACOS_R4
0010 190 ; RSB ; return with result in R0
0010 191 ;
0010 192
0010 193 MTH$ACOS_R4:: ; special ACOS routine
0010 194 MTH$ACOS_R5:: ; Release 1 name
0010 195 ; MOVF R0, R4 ; save X in R4
54 50 50 0010 196 ; BNEQ TEST_FOR_1.0 ; branch if |X| > 0
08 12 0013 197
0015 198 ;
0015 199 ; X = 0
0015 200 ;
0015 201
50 0FDB40C9 8F 50 0015 202 ; MOVF #LF_PI_OV_2, R0 ; R0 = PI/2
05 001C 203 ; RSB ; return PI/2 if |X| = 0
001D 204
001D 205 ;
001D 206 ; 0 < |X|
001D 207 ;
001D 208
001D 209 TEST_FOR_1.0:
50 8000 8F AA 001D 210 ; BICW #X8000, R0 ; R0 = |X|
08 50 51 0022 211 ; CMPF R0, S^#SF 1.0 ; compare |X| with 1.0
30 18 0025 212 ; BGEQ GEQ_TO_1.0 ; branch if |X| >= 1.0
0027 213
0027 214 ;
0027 215 ; 0 < |X| < 1.0
0027 216 ;
0027 217
50 4020 8F B1 0027 218 ; CMPW #X4020, R0 ; Check for loss of significance in
0C 14 002C 219 ; BGTR 1$ ; computing 1 - x^2 and branch
51 08 50 43 002E 220 ; SUBF3 R0, #1, R1 ; R1 = 1 - X
50 08 40 0032 221 ; ADDF #1, R0 ; R0 = 1 + X
50 51 44 0035 222 ; MULF R1, R0 ; R0 = 1 - X^2
07 11 0038 223 ; BRB 2$ ; Join normal flow
50 50 50 44 003A 224 1$: ; MULF2 R0, R0 ; R0 = X**2
08 50 43 003D 225 ; SUBF3 R0, S^#SF 1.0, R0 ; R0 = 1.0 - X**2
00000000 EF 16 0041 226 2$: ; JSB MTH$SQRT_R3 ; R0 = SQRT(1-X**2)
50 54 46 0047 227 ; DIVF R4, R0 ; R0 = SQRT(1-X**2)/X
54 54 DD 004A 228 ; PUSHL R4 ; save sign of X for sign test
00000000 EF 16 004C 229 ; JSB MTH$ATAN_R4 ; R0 = ATAN(SQRT(1-X**2)/X)
54 8E D0 0052 230 ; MOVL (SP)+, R4 ; restore sign of X
04 11 0055 231 ; BRB TEST_SIGN ; branch to TEST_SIGN
0057 232
0057 233 ;
0057 234 ; 1 <= |X|
0057 235 ;
0057 236
0057 237 GEQ_TO_1.0:
0E 14 0057 238 ; BGTR ERROR ; branch to ERROR if |X| > 1.0
```

```
0059 239
0059 240
0059 241 :
0059 242 : |Xi = 1.0
0059 243 :
0059 244
50 D4 0059 245 CLRf R0 ; R0 = 0
005B 246
005B 247
005B 248 :
005B 249 : Test the sign of X in order to decide if add PI to the result
005B 250 :
005B 251
005B 252 TEST_SIGN:
54 53 005B 253 TSTf R4 ; test the sign of X
07 18 005D 254 BGEQ 10$ ; branch if X > 0
50 0FDB4149 8F 40 005F 255 ADDf #LF_PI, R0 ; add PI to R0 if X < 0
05 0066 256 10$: RSB ; return with result in R0
0067 257
0067 258 :
0067 259 : 1 < |Xi|, error
0067 260 :
0067 261
0067 262 ERROR: PUSHf (SP) ; return PC from JSB routine
7E 00 6E DD 0069 263 MOVZBL #MTH$K_INVARGMAT, -(SP) ; condition value
50 01 0F 9A 006D 264 ASHL #15, #T, R0 ; R0 = result = reserved operand -0.0
0071 265 ; goes to signal mechanism vector
0071 266 ; (CHF$L_MCH_R0/R1) so error handler
0071 267 ; can modify the result.
00000000'GF 02 FB 0071 268 CALLS #2, G^MTH$$SIGNAL ; signal error and use real user's PC
0078 269 ; independent of CALL vs JSB
05 0078 270 RSB ; return - R0 restored from CHF$L_MCH_R0/R1
0079 271
```



```
0079 273 .SBTTL MTH$ACOSD - Standard Single Precision Floating ACOSD
0079 274
0079 275
0079 276 ;++
0079 277 : FUNCTIONAL DESCRIPTION:
0079 278 :
0079 279 : ACOSD - single precision floating point function
0079 280 :
0079 281 : ACOSD(X) is computed as:
0079 282 :
0079 283 :     If X = 0, then ACOSD(X) = 90.
0079 284 :     If X = 1, then ACOSD(X) = 0.
0079 285 :     If X = -1, then ACOSD(X) = 180.
0079 286 :     If 0 < X < 1/2, then ACOSD(X) = ATAND(SQRT(1-X**2)/X).
0079 287 :     If 1/2 < X < 1, then ACOSD(X) = ATAND(SQRT((1-X)*(1+X))/X).
0079 288 :     If -1/2 < X < 0, then ACOSD(X) = ATAND(SQRT(1-X**2)/X) + 180.
0079 289 :     If -1 < X < -1/2, then ACOSD(X) = ATAND(SQRT((1-X)*(1+X))/X) + 180.
0079 290 :     If 1 < |X|, error.
0079 291 :
0079 292 : CALLING SEQUENCE:
0079 293 :
0079 294 :     ACOSD.wf.v = MTH$ACOSD(x.rf.r)
0079 295 :
0079 296 : INPUT PARAMETERS:
0079 297 :
00000004 0079 298 :     LONG = 4 ; define longword multiplier
00000004 0079 299 :     x = 1 * LONG ; Contents of x is the argument
0079 300 :
0079 301 : IMPLICIT INPUTS: none
0079 302 :
0079 303 : OUTPUT PARAMETERS:
0079 304 :
0079 305 :     VALUE: floating arc-cosine of the argument
0079 306 :
0079 307 : IMPLICIT OUTPUTS: none
0079 308 :
0079 309 : COMPLETION CODES: none
0079 310 :
0079 311 : SIDE EFFECTS:
0079 312 :
0079 313 : Signals: MTH$_INVARG if |X| > 1 with reserved operand in R0 (co180ed to
0079 314 : the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
0079 315 : Associated message is: "INVALID ARGUMENT". Result is reserved operand -0.0
0079 316 : unless a user supplied (or any) error handler changes CHF$MCH_R0/R1.
0079 317 :
0079 318 : NOTE: This procedure disables floating point underflow, enables integer
0079 319 : overflow.
0079 320 :
0079 321 : ---
0079 322 :
0079 323 :
0079 324 .ENTRY MTH$ACOSD, ^M<IV, R2, R3, R4>
0078 325 : standard call-by-reference entry
0078 326 : disable DV (and FU), enable IV
0078 327 MTH$FLAG_JACKET ; flag that this is a jacket procedure in
0078
6D 00000000'GF 9E 0078 MOVAB G^MTH$$JACKET_HND, (FP)
```

```

; Floating Point Arc-cosine routine      K 5
MTH$ACOSD - Standard Single Precision F  16-SEP-1984 01:02:15 VAX/VMS Macro V04-00      Page 9
                                           6-SEP-1984 11:20:11 [MTHRTL.SRC]MTHACOS.MAR;1      (6)

```

[illegible]


```
0089 336 .SBTTL MTH$ACOSD_R4 - Special ACOSD routine
0089 337
0089 338 ; Special ACOSD - used by the standard routine, direct JSB call.
0089 339
0089 340 CALLING SEQUENCE:
0089 341 ; save anything needed in R0:R4
0089 342 MOVF R0 ; input in R0
0089 343 JSB MTH$ACOSD_R4
0089 344 RSB ; return with result in R0
0089 345
0089 346
0089 347 MTH$ACOSD_R4:: ; special ACOSD routine
54 50 50 0089 348 MOVF R0, R4 ; save X in R4
08 12 008C 349 BNEQ D_TEST_FOR_1.0 ; branch if |X| > 0
008E 350
008E 351 ;
008E 352 ; X = 0
008E 353 ;
008E 354
50 000043B4 8F 50 008E 355 MOVF #LF_90, R0 ; R0 = 90
05 05 0095 356 RSB ; return 90 if |X| = 0
0096 357
0096 358 ;
0096 359 ; 0 < |X|
0096 360 ;
0096 361
0096 362 D_TEST_FOR_1.0:
50 8000 8F AA 0096 363 BICW #X8000, R0 ; R0 = |X|
08 50 51 009B 364 CMPF R0, S^#SF_1.0 ; compare |X| with 1.0
30 18 009E 365 BGEQ D_GEQ_TO_1.0 ; branch if |X| >= 1.0
00A0 366
00A0 367 ;
00A0 368 ; 0 < |X| < 1.0
00A0 369 ;
00A0 370
50 4020 8F B1 00A0 371 CMPW #X4020, R0 ; Check for possible loss of
0C 14 00A5 372 BGTR 1$ ; significance with 0.625
51 08 50 43 00A7 373 SUBF3 R0, #1, R1 ; R1 = 1 - X
50 08 40 00AB 374 ADDF #1, R0 ; R0 = 1 + X
50 51 44 00AE 375 MULF R1, R0 ; R0 = 1 - X^2
07 11 00B1 376 BRB 2$ ; Join main flow
50 50 50 44 00B3 377 1$: MULF2 R0, R0 ; R0 = X**2
08 50 43 00B6 378 SUBF3 R0, S^#SF_1.0, R0 ; R0 = 1.0 - X**2
00000000'EF 16 00BA 379 2$: JSB MTH$SQRT_R3 ; R0 = SQRT(1-X**2)
50 54 46 00C0 380 DIVF R4, R0 ; R0 = SQRT(1-X**2)/X
54 DD 00C3 381 PUSHL R4 ; save sign of X for sign test
00000000'EF 16 00C5 382 JSB MTH$ATAND_R4 ; R0 = ATAND(SQRT(1-X**2)/X)
54 8E D0 00CB 383 MOVL (SP)+, R4 ; restore sign of X
07 11 00CE 384 BRB D_TEST_SIGN ; branch to TEST_SIGN
00D0 385
00D0 386 ;
00D0 387 ; 1 <= |X|
00D0 388 ;
00D0 389
00D0 390 D_GEQ_TO_1.0:
03 13 00D0 391 BEQL 10$ ; branch to ERROR if |X| > 1.0
FF92 31 00D2 392 BRW ERROR
```

```

00D5 393
00D5 394 ;
00D5 395 ; |X| = 1.0
00D5 396 ;
00D5 397
50 D4 00D5 398 10$: CLRf R0 ; R0 = 0
00D7 399
00D7 400
00D7 401 ;
00D7 402 ; Test the sign of X in order to decide if add 180 to the result
00D7 403 ;
00D7 404
00D7 405 D_TEST_SIGN:
54 53 00D7 406 TSTf R4 ; test the sign of X
07 18 00D9 407 BGEQ 10$ ; branch if X > 0
50 00004434 8F 40 00DB 408 ADDf #LF_180, R0 ; add 180 to R0 if X < 0
05 00E2 409 10$: RSB ; return with result in R0
00E3 410
00E3 411 .END

```


MTH\$ACOS
Symbol table

; Floating Point Arc-cosine routine N 5

16-SEP-1984 01:02:15
6-SEP-1984 11:20:11

VAX/VMS Macro V04-00
[MTHRTL.SRC]MTHACOS.MAR;1

Page 12
(7)

```
D_GEQ_TO_1.0      000000D0 R    01
D_TEST_FOR_1.0    00000096 R    01
D_TEST_SIGN       000000D7 R    01
ERROR             00000067 R    01
GEQ_TO_1.0        00000057 R    01
LF_T80            = 00004434
LF_90             = 000043B4
LF_PI             = 0FDB4149
LF_PI_OV_2        = 0FDB40C9
LONG              = 00000004
MTH$$JACKET_HND   ***** X    01
MTH$$SIGNAL       ***** X    00
MTH$ACOS          00000000 RG    01
MTH$ACOSD         00000079 RG    01
MTH$ACOSD_R4      00000089 RG    01
MTH$ACOS_R4       00000010 RG    01
MTH$ACOS_R5       00000010 RG    01
MTH$ATAN_R4       ***** X    00
MTH$ATAN_R4       ***** X    00
MTH$K_INVARGMAT   ***** X    00
MTH$SORT_R3       ***** X    00
SF_1.0            = 00004080
TEST_FOR_1.0      0000001D R    01
TEST_SIGN         0000005B R    01
VALUE             = 00000004
```

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

PSECT name	Allocation	PSECT No.	Attributes												
ABS	00000000 (0.)	00 (0.)	NOPI	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE		
MTH\$CODE	000000E3 (227.)	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	00:00:01.11
Command processing	115	00:00:00.59	00:00:03.37
Pass 1	94	00:00:01.10	00:00:03.69
Symbol table sort	0	00:00:00.00	00:00:00.24
Pass 2	82	00:00:00.97	00:00:02.86
Symbol table output	3	00:00:00.03	00:00:00.03
Psect synopsis output	3	00:00:00.03	00:00:00.17
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	329	00:00:02.80	00:00:11.48

The working set limit was 900 pages.
6148 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.
471 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\$:MTHACOS/OBJ=OBJ\$:MTHACOS MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC\$:

0257

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY