

[illegible]

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

```
MM      MM      TTTTTTTTTT  HH      HH      HH      HH      SSSSSSSS  IIIIII  NN      NN      HH      HH
MM      MM      TTTTTTTTTT  HH      HH      HH      HH      SSSSSSSS  IIIIII  NN      NN      HH      HH
MMM     MMM     TT          HH      HH      HH      HH      SS          II      NN      NN      HH      HH
MMM     MMM     TT          HH      HH      HH      HH      SS          II      NN      NN      HH      HH
MM      MM      TT          HH      HH      HH      HH      SS          II      NNNN     NN      HH      HH
MM      MM      TT          HH      HH      HH      HH      SS          II      NNNN     NN      HH      HH
MM      MM      TT          HHHHHHHHHH  HHHHHHHHHH  SSSSSS  II      NN      NN      HHHHHHHHHH
MM      MM      TT          HHHHHHHHHH  HHHHHHHHHH  SSSSSS  II      NN      NN      HHHHHHHHHH
MM      MM      TT          HH      HH      HH      HH      SS          II      NN      NN      HH      HH
MM      MM      TT          HH      HH      HH      HH      SS          II      NN      NN      HH      HH
MM      MM      TT          HH      HH      HH      HH      SS          II      NN      NN      HH      HH
MM      MM      TT          HH      HH      HH      HH      SSSSSSSS  IIIIII  NN      NN      HH      HH
MM      MM      TT          HH      HH      HH      HH      SSSSSSSS  IIIIII  NN      NN      HH      HH
```

```
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
```

(2) 50
(3) 72
(4) 150

HISTORY : Detailed Current Edit History
DECLARATIONS : Declarative Part of Module
MTH\$HSINH - Standard H floating SINH


```
0000 1 .TITLE MTH$HSINH ; H floating Hyperbolic Sine routine
0000 2 ; (HSINH)
0000 3 .IDENT /1-006/ ; File: MTHHSINH.MAR EDIT: RNH1006
0000 4 :
0000 5 :*****
0000 6 :
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :
0000 25 :
0000 26 :*****
0000 27 :
0000 28 :
0000 29 : FACILITY: MATH LIBRARY
0000 30 :++
0000 31 : ABSTRACT:
0000 32 :
0000 33 : MTH$HSINH is a function which returns the H floating hyperbolic sine
0000 34 : of its H floating point argument. The call is standard
0000 35 : call-by-reference.
0000 36 :
0000 37 :--
0000 38 :
0000 39 : VERSION: 1
0000 40 :
0000 41 : HISTORY:
0000 42 : AUTHOR:
0000 43 : John A. Wheeler, 11-Sep-1979: Version 1
0000 44 :
0000 45 : MODIFIED BY:
0000 46 :
0000 47 :
0000 48 :
```

MTH\$HSINH
1-006

M 10
; H floating Hyperbolic Sine routine 16-SEP-1984 01:40:07 VAX/VMS Macro V04-00
HISTORY ; Detailed Current Edit History 6-SEP-1984 11:25:42 [MTHRTL.SRC]MTHHSINH.MAR;1

Page 2
(2)

```
0000 50      .SBTTL HISTORY ; Detailed Current Edit History
0000 51
0000 52
0000 53 : ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000 54 :
0000 55 : Edit History for Version 1 of MTH$HSINH
0000 56 :
0000 57 : 1-001 - Adapted from MTH$GSINH version 1-001. JAW 11-Sep-1979.
0000 58 : 1-002 - Use MTH$HEXP_R6. SBL 3-Oct-1979
0000 59 : 1-003 - Make H_0.25 a literal. SBL 8-Oct-1979
0000 60 : 1-004 - Don't store reserved operand before signal. SBL 7-Feb-1980
0000 61 : 1-005 - Changed lower limit for Chebyshev approximation from 2**-56 to
0000 62 : 2**-57.
0000 63 : - Eliminated second call to EXP for input values between 57*ln2
0000 64 : and 16383*ln2.
0000 65 : - Changed all final floating point divisions by 2 to interger
0000 66 : subtracts of 1 from the exponent field.
0000 67 : - Extended maximum range to 16384*ln2.
0000 68 : - Changed logic for computing EXP(ixi-ln2) to reduce error.
0000 69 : - RNH 10-FEB-81
0000 70 : 1-006 - Changed @hsinh to @hsinh(AP) in error logic RNH 22-Sept-81
```

```
0000 72 .SBTTL DECLARATIONS ; Declarative Part of Module
0000 73
0000 74 :
0000 75 : INCLUDE FILES:
0000 76 :
0000 77 : EXTERNAL SYMBOLS:
0000 78 .DSABL GBL
0000 79 .EXTRN MTH$$SIGNAL
0000 80 .EXTRN MTH$K_FLOOVEMAT
0000 81 .EXTRN MTH$HEXP_R6
0000 82 :
0000 83 : EQUATED SYMBOLS:
0000 84
0000 85
0000 86 :
0000 87 : MACROS: none
0000 88 :
0000 89
0000 90
0000 91 :
0000 92 : PSECT DECLARATIONS:
0000 93
0000 94 .PSECT _MTH$CODE PIC,SHR,LONG,EXE,NOWRT
0000 95 ; program section for math routines
0000 96 :
0000 97 : OWN STORAGE: none
0000 98 :
0000 99 :
0000 100 : CONSTANTS:
0000 101 :
0000 102 :
0000 103 H_57_LOG_2:
6DB93AB1 3C134006 0000 104 .QUAD ^X6DB93AB13C134006
C70997E6 FF9D90B9 0008 105 .QUAD ^XC70997E6FF9D90B9 ; 57*ln2
0010 106 H_16383_LOG_2:
E3E0A45E 62DE400E 0010 107 .QUAD ^XE3E0A45E62DE400E
6B251812 C66964DB 0018 108 .QUAD ^X6B251812C66964DB ; 16383*ln2
0020 109 H_16384_LOG_2:
A39E2FEF 62E4400E 0020 110 .QUAD ^XA39E2FEF62E4400E
07E66730 93C7F357 0028 111 .QUAD ^X07E6673093C7F357 ; 16384*ln2
0030 112 H_LOG_2_HI:
A39E2FEF 62E44000 0030 113 .QUAD ^XA39E2FEF62E44000
40006730 93C7F357 0038 114 .QUAD ^X4000673093C7F357 ; (high 99 bits of ln2)+2**-99
0040 115 H_LOG_2_LO:
FOCB950B C0D0BF9D 0040 116 .QUAD ^XFOCB950BC0D0BF9D
5F359D27 D674CD98 0048 117 .QUAD ^X5F359D27D674CD98 ; ln2 - H_LOG_2_HI
0050 118
0050 119 HSINHTAB:
C377E97C 42343FAD 0050 120 .LONG ^X42343FAD,^XC377E97C ; 6.506912239471978471647518358407046E -26
D9E845CE 6AD9DE9B 0058 121 .LONG ^X6AD9DE9B,^XD9E845CE
5242FAE8 76163FB6 0060 122 .LONG ^X76163FB6,^X5242FAE8 ; 3.867997522833381902976185781667616E -23
BF470195 F929D90B 0068 123 .LONG ^XF929D90B,^XBF470195
5CAAF302 71B83FBF 0070 124 .LONG ^X71B83FBF,^X5CAAF302 ; 1.957294395551957859061134358006348E -20
90698008 F2CF3417 0078 125 .LONG ^XF2CF3417,^X90698008
2521B466 2F493FC8 0080 126 .LONG ^X2F493FC8,^X2521B466 ; 8.220635243500749521323757431341128E -18
56487466 67B8FB06 0088 127 .LONG ^X67B8FB06,^X56487466
0C3D7703 952C3FD0 0090 128 .LONG ^X952C3FD0,^X0C3D7703 ; 2.811457254347797192926287843860811E -15
```


9ED3038D	385C5DA8	0098	129	.LONG	^X385C5DA8,^X9ED3038D	
3B813E73	AE7F3FD8	00A0	130	.LONG	^XAE7F3FD8,^X3B813E73	; 7.647163731819805091328739180656633E -13
A43FC7A1	36713CB8	00A8	131	.LONG	^X36713CB8,^XA43FC7A1	
86D0613A	61243FE0	00B0	132	.LONG	^X61243FE0,^X86D0613A	; 1.605904383682161463834581885720363E -10
6618C597	F1979807	00B8	133	.LONG	^XF1979807,^X6618C597	
544E567F	AE643FE7	00C0	134	.LONG	^XAE643FE7,^X544E567F	; 2.505210838544171877496282899225692E -8
05071710	583338FE	00C8	135	.LONG	^X583338FE,^X05071710	
6C733A55	71DE3FEE	00D0	136	.LONG	^X71DE3FEE,^X6C733A55	; 2.755731922398589065255745053025947E -6
EBFCDB75	AC2438FA	00D8	137	.LONG	^XAC2438FA,^XEBFCDB75	
1A0101A0	A01A3FF4	00E0	138	.LONG	^XA01A3FF4,^X1A0101A0	; 1.984126984126984126984126972636750E -4
DDDB1717	01A0A01A	00E8	139	.LONG	^X01A0A01A,^XDDDB1717	
11111111	11113FFA	00F0	140	.LONG	^X11113FFA,^X11111111	; 8.33333333333333333333333333333385565E -3
98AB1111	11111111	00F8	141	.LONG	^X11111111,^X98AB1111	
55555555	55553FFE	0100	142	.LONG	^X55553FFE,^X55555555	; 1.666666666666666666666666666666657E -1
552F5555	55555555	0108	143	.LONG	^X55555555,^X552F5555	
00000000	00000000	0110	144	.LONG	^X00000000,^X00000000	; 0.0000000000000000000000000000000000E 0
00000000	00000000	0118	145	.LONG	^X00000000,^X00000000	
		0120	146			
0000000D		0120	147	HSINHLEN	= .- HSINHTAB/16	
		0120	148			

```
0120 150      .SBTTL MTH$HSINH - Standard H floating SINH
0120 151
0120 152
0120 153 :++
0120 154 : FUNCTIONAL DESCRIPTION:
0120 155
0120 156 : HSINH - H floating point function
0120 157
0120 158 : HSINH(X) is computed as:
0120 159
0120 160 :     If |X| < 2**(-57), HSINH(X) = X.
0120 161 :     If 2**(-57) <= |X| < 0.25, HSINH(X) = Chebyshev Series.
0120 162 :     If 0.25 <= |X| < 57*ln2, HSINH(X) = (HEXP(X) - HEXP(-X))/2.
0120 163 :     If 57*ln2 <= |X| < 16383*ln2, HSINH(X) = sign(X)*HEXP(|X|)/2.
0120 164 :     If 16383*ln2 <= |X| < 16384*ln2, then HSINH(X) = sign(X)*HEXP(|X|-ln2).
0120 165 :     If 16384*ln2 <= |X|, then overflow.
0120 166
0120 167 : CALLING SEQUENCE:
0120 168
0120 169 :     hsinh.wh.v = MTH$HSINH(x.rh.r)
0120 170
0120 171 :     -or-
0120 172
0120 173 :     CALL MTH$HSINH(hsinh.wh.r, x.rh.r)
0120 174
0120 175 :     Because an H-floating result cannot be expressed in 64 bits, it is
0120 176 :     returned as the first argument, with the input parameter displaced
0120 177 :     to the second argument, in accordance with the Procedure Calling
0120 178 :     Standard.
0120 179
0120 180 : INPUT PARAMETERS:
0120 181
0120 182 :     LONG = 4                                ; define longword multiplier
00000004 0120 183 :     x = 2 * LONG                            ; Contents of x is the argument
00000008 0120 184
0120 185 : IMPLICIT INPUTS:      none
0120 186
0120 187 : OUTPUT PARAMETERS:
0120 188
0120 189 :     hsinh = 1 * LONG
00000004 0120 190
0120 191 : IMPLICIT OUTPUTS:    none
0120 192
0120 193 : COMPLETION CODES:    none
0120 194
0120 195 : SIDE EFFECTS:
0120 196
0120 197 : Signal: MTH$ FLOOVEMAT if 16384*ln2 < |X| with reserved operand in R0/R3
0120 198 : (copied to the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
0120 199 : Associated message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved
0120 200 : operand -0.0 unless a user supplied (or any) error handler changes CHF$MCH_R0/R1
0120 201
0120 202 : NOTE: This procedure disables floating point underflow, enables integer
0120 203 : overflow.
0120 204
0120 205 : ---
0120 206
```



```
0120 207
0120 208
47FC 0120 209 .ENTRY MTH$HSINH, ^M<IV, R2, R3, R4, R5, R6, R7, R8, R9, R10>
0122 210 ; standard call-by-reference entry
0122 211 ; disable DV (and FU), enable IV
0122 212 MTH$FLAG_JACKET ; flag that this is a jacket procedure in
0122
6D 00000000'GF 9E 0122 MOVAB G^MTH$$JACKET_HND, (FP)
0129 ; set handler address to jacket
0129 ; handler
0129
0129 213 ; case of an error in routine
6D 00000000'GF 9E 0129 214 MOVAB G^MTH$$JACKET_HND, (FP) ; Set handler address to jacket handler
0130 215 ; If an error, convert signal to user PC
0130 216 ; and resignal
0130 217 MOVH @x(AP), R0 ; R0/R3 = !X! = @value(AP)
0135 218 MOV0 R0, R7
0139 219 BICW2 #^X8000, R0 ; R0/R3 = !X!
013E 220 CMPW R0, #^X3FFF ; compare !X! with 0.25
0143 221 BGEQ GEQ_TO_0.25 ; branch if !X! >= 0.25
0145 222
0145 223 ;
0145 224 ; !X! < 0.25
0145 225 ;
0145 226
0145 227 CMPW R0, #^X3FC8 ; compare !X! with 2**-57
014A 228 BGEQ GEQ_TO_2M57 ; branch if !X! >= 2**-57
014C 229
014C 230 ;
014C 231 ; !X! < 2**-57
014C 232 ;
014C 233
014C 234 MOV0 R7, @hsinh(AP) ; Result = argument
0151 235 RET ; return
0152 236
0152 237 ;
0152 238 ; 2**-57 <= !X! < 0.25
0152 239 ;
0152 240
0152 241 GEQ_TO_2M57:
0152 242 MULH2 R0, R0 ; Get ARG**2 for POLYG
0156 243 POLYH R0, #HSINHLLEN-1, HSINHTAB ;
015D 244 ; R0/R3 = SUM(Ci*X**i), with
015D 245 ; last coefficient zero
015D 246 MULH2 R7, R0 ; MULH2 by ARG, and then
0161 247 ADDH3 R7, R0, @hsinh(AP) ; add in ARG with overhang and return
0167 248 RET ; with result in first argument
0168 249
0168 250 ;
0168 251 ; 0.25 <= !X!
0168 252 ;
0168 253
0168 254 GEQ_TO_0.25:
0168 255 CMPH R0, H 16383 LOG 2 ; compare !X! with 16383*ln2
016E 256 BGTR GTR_TRAN_16383_LOG_2 ; branch if !X! > 16383*ln2
0170 257
0170 258 ;
```

```
0170 259 : 0.25 =< !X! =< 16383*Ln2
0170 260 :
0170 261 :
FE8A CF 50 71FD 0170 262 CMPH R0, H_57_LOG_2 ; Compare !X! to 57*Ln2. If greater
25 14 0176 263 BGTR ONLY_ONE_TERM ; only one call to HEXP is necessary.
0178 264 :
0178 265 : 0.25 =< !X! < 57*Ln2
0178 266 :
0178 267 :
00000000'EF 16 0178 268 MNEGH R7, R0 ; R0/R3 = -X
7E 50 7DFD 017C 269 JSB MTH$HEXP_R6 ; R0/R3 = HEXP(-X)
50 57 7DFD 0182 270 MOV0 R0, -(SP) ; push HEXP(-X) on stack
00000000'EF 16 018A 271 MOV0 R7, R0 ; R0/R3 = X
50 8E 62FD 0190 272 JSB MTH$HEXP_R6 ; R0/R3 = HEXP(X)
50 01 A2 0194 273 SUBH2 (SP)+, R0 ; R0/R3 = HEXP(X) - HEXP(-X)
04 BC 50 7DFD 0197 274 SUBW2 #^X0001, R0 ; R0/R3 = (HEXP(X)-HEXP(-X))/2
04 019C 275 MOV0 R0, @hsinh(AP) ; Return SINH(x) in first argument
019D 276 RET
019D 277 :
019D 278 : 57*Ln2 =< !X! < 16383*Ln2
019D 279 :
019D 280 :
00000000'EF 16 019D 281 ONLY_ONE_TERM:
57 53FD 01A3 282 JSB MTH$HEXP_R6 ; R0/R3 = HEXP(!X!)
04 14 01A6 283 TSTG R7 ; Check sign of X
50 50 52FD 01A8 284 BGTR POSITIVE ; If negative change sign of
01AC 285 MNEGG R0, R0 ; HEXP(!X!)
01AC 286 POSITIVE:
50 01 A2 01AC 287 SUBW2 #^X0001, R0 ; R0/R3 = sign(X)*HEXP(!X!)/2
04 BC 50 7DFD 01AF 288 MOV0 R0, @hsinh(AP) ; Return SINH(x) in first argument
04 01B4 289 RET
01B5 290 :
01B5 291 : 16383*Ln2 =< !X!
01B5 292 :
01B5 293 :
FE65 CF 50 71FD 01B5 294 GTR_THAN_16383_LOG_2:
23 18 01B8 295 CMPH R0, H_16384_LOG_2 ; Compare !X! to 16384*Ln2
01BD 296 BGEQ ERROR ; if 16384*Ln2 =< !X!, overflow occurs
01BD 297 :
01BD 298 :
01BD 299 :
01BD 300 : 16383*Ln2 < !X! < 16384*Ln2
01BD 301 :
01BD 302 :
50 FE6E CF 62FD 01BD 303 SUBH2 H_LOG_2_HI, R0 ; R0/R3=!X!-H_LOG_2_HI
00000000'EF 16 01C3 304 JSB MTH$HEXP_R6 ; R0/R3 = HEXP(!X!-H_LOG_2_HI)
57 73FD 01C9 305 TSTH R7 ; test the sign of X
04 18 01CC 306 BGEQ 10$ ; branch if X >= 0
50 50 72FD 01CE 307 MNEGH R0, R0 ; R0/R3=sign(X)*HEXP(!X!-H_LOG_2_HI)
54 50 FE69 CF 65FD 01D2 308 10$: MULH3 H_LOG_2_LO, R0, R4 ; R4/R3=HEXP(!X!-H_LOG_2_HI)*H_LOG_2_LO
04 BC 50 54 63FD 01D9 309 SUBH3 R4, R0, @hsinh(AP) ; Return HEXP(!X!-Ln2) in first
01DF 310 RET ; argument
01E0 311 :
01E0 312 :
01E0 313 :
01E0 314 :
01E0 315 : 16384*Ln2 =< !X!, error
```


MTH\$HSINH
1-006

F 11
; H floating Hyperbolic Sine routine
MTH\$HSINH - Standard H floating SINH

16-SEP-1984 01:40:07 VAX/VMS Macro V04-00
6-SEP-1984 11:25:42 [MTHRTL.SRC]MTHHSINH.MAR;1

Page 8
(4)

```

                                01E0 316 ;
                                01E0 317 ;
                                01E0 318 ERROR: MOVZBL #MTH$K_FLOOVEMAT, -(SP) ; condition value
50 7E 01 00'8F 9A 01E0 319 ASHQ #15, #T, R0 ; R0/R3 = result = reserved operand -0.0
                                01E4 320 ; so error handler can modify the result
                                01E8 321
                                01E8 322 CLRQ R2 ;
00000000'EF 52 7C 01E8 323 CALLS #1, MTH$$SIGNAL ; signal error and use real user's PC
04 BC 01 01 FB 01EA 324 MOVO R0, @hsinh(AP) ; Restore result
                                01F1 325 RET ; return
                                01F6 326
                                01F7 327
                                01F7 328
                                01F7 328 .END
```


MTH\$HSINH
Symbol table

G 11
; H floating Hyperbolic Sine routine

16-SEP-1984 01:40:07 VAX/VMS Macro V04-00
6-SEP-1984 11:25:42 [MTHRTL.SRC]MTH\$HSINH.MAR;1

Page 9
(4)

```
ERROR          000001E0 R    01
GEQ_TO_0.25    00000168 R    01
GEQ_TO_2M57    00000152 R    01
GTR_THAN_16383_LOG_2 000001B5 R    01
HSINH          = 00000004
HSINHLEN      = 00000000
HSINHTAB      00000050 R    01
H_16383_LOG_2  00000010 R    01
H_16384_LOG_2  00000020 R    01
H_57_LOG_2     00000000 R    01
H_LOG_2_HI     00000030 R    01
H_LOG_2_LO     00000040 R    01
LONG          = 00000004
MTH$$JACKET_HND ***** X    01
MTH$$SIGNAL    ***** X    00
MTH$HEXP_R6    ***** X    00
MTH$HSINH      00000120 RG   01
MTH$K_FLOOVEMAT ***** X    00
ONLY_ONE_TERM  00000190 R    01
POSITIVE       000001AC 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	000001F7 (503.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG				

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:01.48
Command processing	114	00:00:00.68	00:00:04.33
Pass 1	86	00:00:01.03	00:00:03.54
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	71	00:00:00.78	00:00:02.85
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.01	00:00:00.23
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	309	00:00:02.62	00:00:12.47

The working set limit was 900 pages.
5105 bytes (10 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 1 local symbols.
388 source lines were read in Pass 1, producing 11 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\$:MTHHSINH/OBJ=OBJ\$:MTHHSINH MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

0262 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY