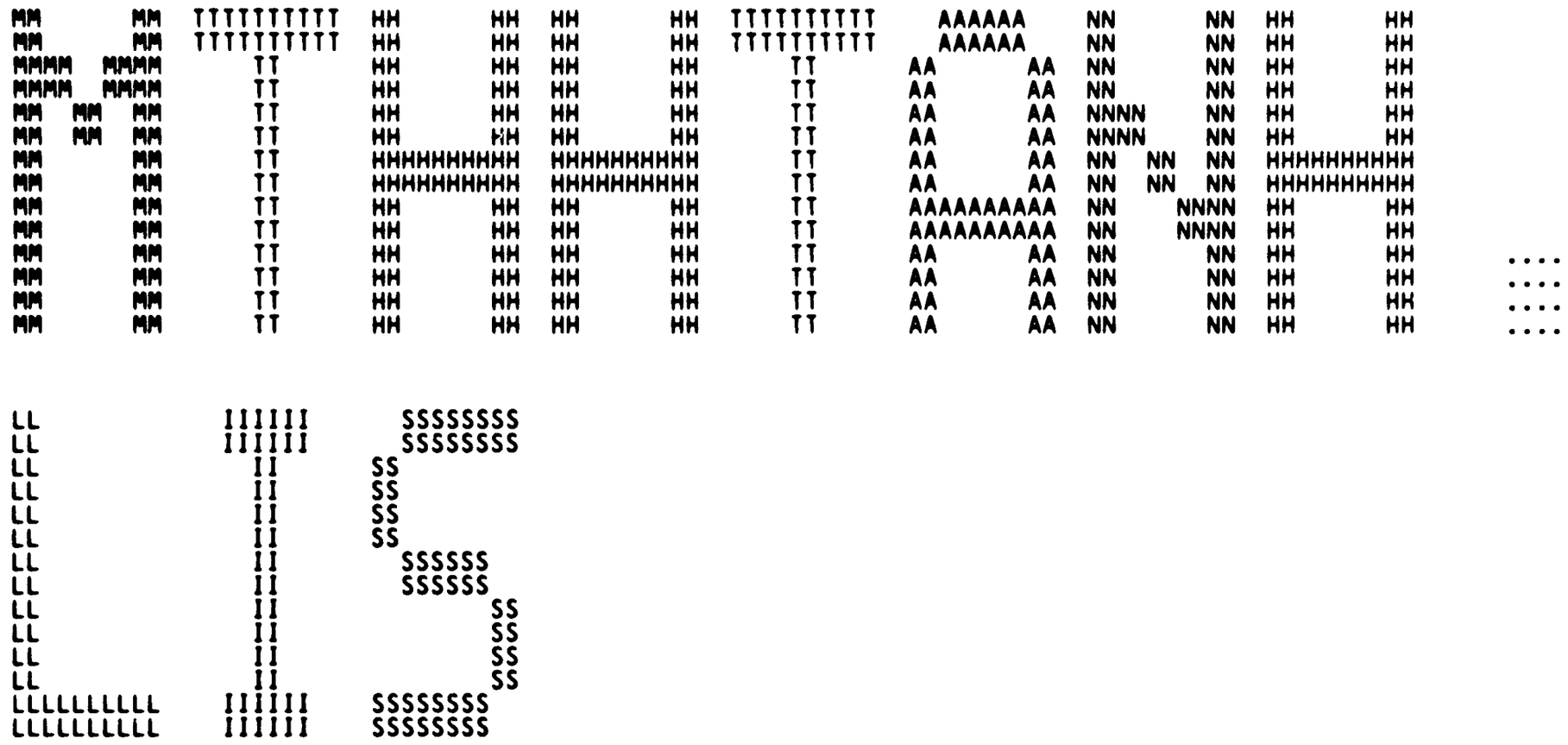
```
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMMMMM    MMMMMM        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**MTHHTANH



```
0000     1          .TITLE  MTH$HTANH       ; H Floating Hyperbolic Tangent routine
0000     2                                  ; (HTANH)
0000     3          .IDENT /1-006/          ; File: MTHHTANH.MAR  Edit:  JCW1006
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$HTANH is a function  which  returns the H floating hyperbolic tangent
0000    34 ; of its H floating argument. The call is standard call-by-reference.
0000    35 ;
0000    36 ;--
0000    37 ;
0000    38 ; VERSION: 1
0000    39 ;
0000    40 ; HISTORY:
0000    41 ; AUTHOR:
0000    42 ;       John A. Wheeler, 20-Oct-1979: Version 1
0000    43 ;
0000    44 ; MODIFIED BY:
0000    45 ;
0000    46 ;
0000    47 ;
```

```
                        .SBTTL  HISTORY ; Detailed Current Edit History

; Edit History for Version 1 of MTH$HTANH
;
; 1-001 - Adapted from MTH$GTANH version 1-002.  JAW 20-Oct-1979
; 1-002 - Change constant 16.0 to 40.0 to correct inaccuracy.  The
;         value of X above which 1.0 is the best machine approximation
;         to HTANH(X) is about 39.86.  The next higher number that can
;         be represented as a short literal is 40.0.  JAW 19-Sep-80
; 1-003 - Eliminated symbolic short literals.  RNH 15-Oct-81
; 1-004 - Changed #40 to S^#40 to circumvent assembler problem RNH 23-Oct-81
; 1-005 - Use general mode addressing.  SBL 30-Nov-1981
; 1-006 - Changed the constant 2^-59 to 2^-57 to correct inaccuracy. For
;         values of !X! between 2^-59 and 2^-57 no noticeable loss of
;         significance was noticed by the assumption that HTANH(x)=x starting
;         at !X!<=2^-59 instead of 2^-57, but a loss of performance was felt
;         do to unnecessary computation of HSINH(X)/HCOSH(X). All appropriate
;         references to 2^-59 have been changed to 2^-57.  JCW 10-Jan-1983
```

```
                                0000     69          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                                0000     70
                                0000     71 ;
                                0000     72 ; INCLUDE FILES:
                                0000     73 ;
                                0000     74 ; EXTERNAL SYMBOLS:     MTH$JACKET_HDLR
                                0000     75 ;
                                0000     76          .DSABL  GBL                     ; Force .EXTRN on all symbols
                                0000     77          .EXTRN  MTH$HCOSH               ; HCOSH
                                0000     78          .EXTRN  MTH$HSINH               ; HSINH
                                0000     79          .EXTRN  MTH$HEXP_R6             ; EXP
                                0000     80 ;
                                0000     81 ; EQUATED SYMBOLS:
                                0000     82 ;
                                0000     83
                       00000004 0000     84          y = 4                           ; Offset for first argument
                                0000     85
                                0000     86 ;
                                0000     87 ; MACROS:       none
                                0000     88 ;
                                0000     89 ; PSECT DECLARATIONS:
                                0000     90
                       00000000 0000     91          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                                0000     92                                  ; Program section for math routines
                                0000     93 ;
                                0000     94 ; OWN STORAGE:  none
                                0000     95 ;
                                0000     96 ;
                                0000     97 ; CONSTANTS:
                                0000     98 ;
                                0000     99
                                0000    100 H_0.25:
00000000 00003FFF               0000    101          .LONG   ^X00003FFF, ^X00000000  ; 0.25
00000000 00000000               0008    102          .LONG   ^X00000000, ^X00000000
                                0010    103 H_2_POWER_M57:
00000000 00003FC8               0010    104          .LONG   ^X00003FC8, ^X00000000  ; 2**-57
00000000 00000000               0018    105          .LONG   ^X00000000, ^X00000000
                                0020    106
```

```
                     0020   108          .SBTTL  MTH$HTANH  - H Floating Point TANH
                     0020   109
                     0020   110
                     0020   111 ;++
                     0020   112 ; FUNCTIONAL DESCRIPTION:
                     0020   113 ;
                     0020   114 ; HTANH  - H floating point hyperbolic tangent function
                     0020   115 ;
                     0020   116 ; HTANH(X) is computed as:
                     0020   117 ;
                     0020   118 ;       If |X| <= 2**-57, then HTANH(X) = X.
                     0020   119 ;       If 2**-57 < |X| <= 0.25, then HTANH(X) = HSINH(X)/HCOSH(X).
                     0020   120 ;       If 0.25 < |X| < 40.0, then HTANH(X) = (HEXP(2*X) - 1) / (HEXP(2*X) + 1)
                     0020   121 ;       If 40.0 <= |X|, then HTANH(X) = sign(X) * 1
                     0020   122 ;
                     0020   123 ; CALLING SEQUENCE:
                     0020   124 ;
                     0020   125 ;       htanh.wh.v = MTH$HTANH(x.rh.r)
                     0020   126 ;
                     0020   127 ;               -or-
                     0020   128 ;
                     0020   129 ;       CALL MTH$HTANH(htanh.wh.r, x.rh.r)
                     0020   130 ;
                     0020   131 ;       Because an H-floating result cannot be expressed in 64 bits, it is
                     0020   132 ;       returned as the first argument, with the input parameter displaced
                     0020   133 ;       to the second argument, in accordance with the Procedure Calling
                     0020   134 ;       Standard.
                     0020   135 ;
                     0020   136 ; INPUT PARAMETERS:
                     0020   137
00000004             0020   138          LONG = 4                        ; Define longword multiplier
00000008             0020   139          x = 2 * LONG                    ; Contents of x is the argument
                     0020   140
                     0020   141 ; IMPLICIT INPUTS:      none
                     0020   142 ;
                     0020   143 ; OUTPUT PARAMETERS:
                     0020   144 ;
                     0020   145 ;       VALUE:  H floating hyperbolic tangent of the argument
                     0020   146 ;
00000004             0020   147          htanh = 1 * LONG                ; htanh is the result
                     0020   148 ;
                     0020   149 ; IMPLICIT OUTPUTS:     none
                     0020   150 ;
                     0020   151 ; COMPLETION CODES:     none
                     0020   152 ;
                     0020   153 ; SIDE EFFECTS: none
                     0020   154 ;
                     0020   155 ; NOTE: This procedure disables floating point underflow, enables integer
                     0020   156 ; overflow.
                     0020   157 ;
                     0020   158 ;--
                     0020   159
                     0020   160
                40FC 0020   161          .ENTRY  MTH$HTANH, ^M<IV, R2, R3, R4, R5, R6, R7>
                     0022   162                                          ; Standard call-by-reference entry
                     0022   163                                          ; Disable DV (and FU), enable IV
                     0022   164          MTH$FLAG_JACKET                 ; Flag that this is a jacket procedure in
```

```
                                0022
                                0022
            6D  00000000'GF  9E  0022                MOVAB   G^MTH$$JACKET_HND, (FP)
                                0029                                            ; set handler address to jacket
                                0029                                            ; handler
                                0029
                                0029   165                                      ; case of an error in routine
                                0029   166                                      ; If an error, convert signal to user PC
                                0029   167                                      ; and resignal
                  50  08 BC 70FD  0029   168                MOVH    @x(AP), R0       ; R0/R3 = X = @x(AP)
                50  8000 8F  AA  002E   169                BICW    #^X8000, R0      ; R0/R3 = |X|
                  D8 AF  50 71FD  0033   170                CMPH    R0, H_2_POWER_M57 ; Compare |X| with 2**-57
                         5B  15  0038   171                BLEQ    OUT_X            ; Branch if |X| <= 2**-57
                                003A   172
                                003A   173 ;
                                003A   174 ; 2**-57 < |X|
                                003A   175 ;
                                003A   176
                     32  50 71FD  003A   177                CMPH    R0, S^#40        ; Compare |X| with 40.0
                         41  18  003E   178                BGEQ    GEQ_TO_40.0      ; Branch if |X| >= 40.0
                                0040   179
                                0040   180 ;
                                0040   181 ; 2**-57 < |X| < 40.0
                                0040   182 ;
                                0040   183
                  BB AF  50 71FD  0040   184                CMPH    R0, H_0.25       ; Compare |X| with 0.25
                         1C  15  0045   185                BLEQ    LEQ_TO_0.25      ; Branch if |X| <= 0.25
                                0047   186
                                0047   187 ;
                                0047   188 ; 0.25 < |X| < 40.0
                                0047   189 ;
                                0047   190
        50  08 BC  08 BC 61FD  0047   191                ADDH3   @x(AP), @x(AP), R0       ; R0/R3 = 2*X
              00000000'GF  16  004E   192                JSB     G^MTH$HEXP_R6            ; R0/R3 = HEXP(2*X)
            54  50  08 61FD  0054   193                ADDH3   #1, R0, R4               ; R4/R7 = HEXP(2*X) + 1
                50  08 62FD  0059   194                SUBH2   #1, R0                   ; R0/R3 = HEXP(2*X) - 1
                50  54 66FD  005D   195                DIVH2   R4, R0                   ; R0/R3 = (HEXP(2*X) - 1) / (HEXP(2*X) + 1)
                    2C  11  0061   196                BRB     RETURN                   ; Store result and return
                                0063   197
                                0063   198 ;
                                0063   199 ; 2**-57 < |X| <= 0.25
                                0063   200 ;
                                0063   201
                                0063   202 LEQ_TO_0.25:
      00000000'GF   6C  FA  0063   203                CALLG   (AP), G^MTH$HCOSH        ; @y(AP) = HCOSH(X)
            54  04 BC 7DFD  006A   204                MOVQ    @y(AP), R4               ; R4/R7 = HCOSH(X)
      00000000'GF   6C  FA  006F   205                CALLG   (AP), G^MTH$HSINH        ; @y(AP) = HSINH(X)
            50  04 BC 7DFD  0076   206                MOVQ    @y(AP), R0               ; R0/R3 = HSINH(X)
                50  54 66FD  007B   207                DIVH2   R4, R0                   ; R0/R3 = HSINH(X) / HCOSH(X)
                    0E  11  007F   208                BRB     RETURN                   ; Store result and return
                                0081   209
                                0081   210 ;
                                0081   211 ; |X| >= 40.0
                                0081   212 ;
                                0081   213
                                0081   214 GEQ_TO_40.0:
                50  08 70FD  0081   215                MOVH    #1, R0           ; R0/R3 = 1.0
                  08 BC 73FD  0085   216                TSTH    @x(AP)           ; Test the sign of X
```

```
                     04   18  0089   217          BGEQ    RETURN                  ; Branch if X >= 0
                  50 50 72FD  008B   218          MNEGH   R0, R0                  ; R0/R3 = -1
               04 BC 50 7DFD  008F   219 RETURN:  MOVQ    R0, @htanh(AP)          ; Store result in first argument
                        04    0094   220          RET                             ; Return to caller
                              0095   221
                              0095   222 ;
                              0095   223 ; |X| <= 2**-57
                              0095   224 ;
                              0095   225
            04 BC    08 BC 7DFD  0095   226 OUT_X:  MOVQ    @x(AP), @htanh(AP)      ; Store result in first argument
                        04    009B   227          RET                             ; Return to caller
                              009C   228
                              009C   229          .END
```

```
GEQ_TO_40.0         00000081 R      01
HTANH             = 00000004
H_0.25              00000000 R      01
H_2_POWER_M57       00000010 R      01
LEQ_TO_0.25         00000063 R      01
LONG              = 00000004
MTH$$JACKET_HND     ********   X    01
MTH$HCOSH           ********   X    00
MTH$HEXP_R6         ********   X    00
MTH$HSINH           ********   X    00
MTH$HTANH           00000020 RG     01
OUT_X               00000095 R      01
RETURN              0000008F R      01
X                 = 00000008
Y                 = 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	0000009C (156.)	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	00:00:01.03
Command processing	117	00:00:00.64	00:00:04.22
Pass 1	88	00:00:00.70	00:00:05.40
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	57	00:00:00.63	00:00:02.68
Symbol table output	2	00:00:00.03	00:00:00.04
Psect synopsis output	2	00:00:00.02	00:00:00.06
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	298	00:00:02.10	00:00:13.43

The working set limit was 750 pages.
3485 bytes (7 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 15 non-local and 0 local symbols.
289 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
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

MTH$HTANH ; H Floating Hyperbolic Tangent routine 16-SEP-1984 01:41:24 VAX/VMS Macro V04-00 Page 8
VAX-11 Macro Run Statistics 6-SEP-1984 11:25:54 [MTHRTL.SRC]MTHHTANH.MAR;1 (4)

There were no errors, warnings or information messages.

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MTHIISIGN
LIS

MTHHFLOOR
LIS

MTHHSIGN
LIS

MTHHMIN
LIS

MTHHLOG
LIS

MTHHTAN
LIS

MTHIIONNT
LIS

MTHIIHNNT
LIS

MTHHSQRT
LIS

MTHIMAX0
LIS

MTHHNINT
LIS

MTHHSINH
LIS

MTHHTANH
LIS

MTHHINT
LIS

MTHHMAX1
LIS

MTHHSINCO
LIS

MTHHMOD
LIS

MTHIIGNNT
LIS