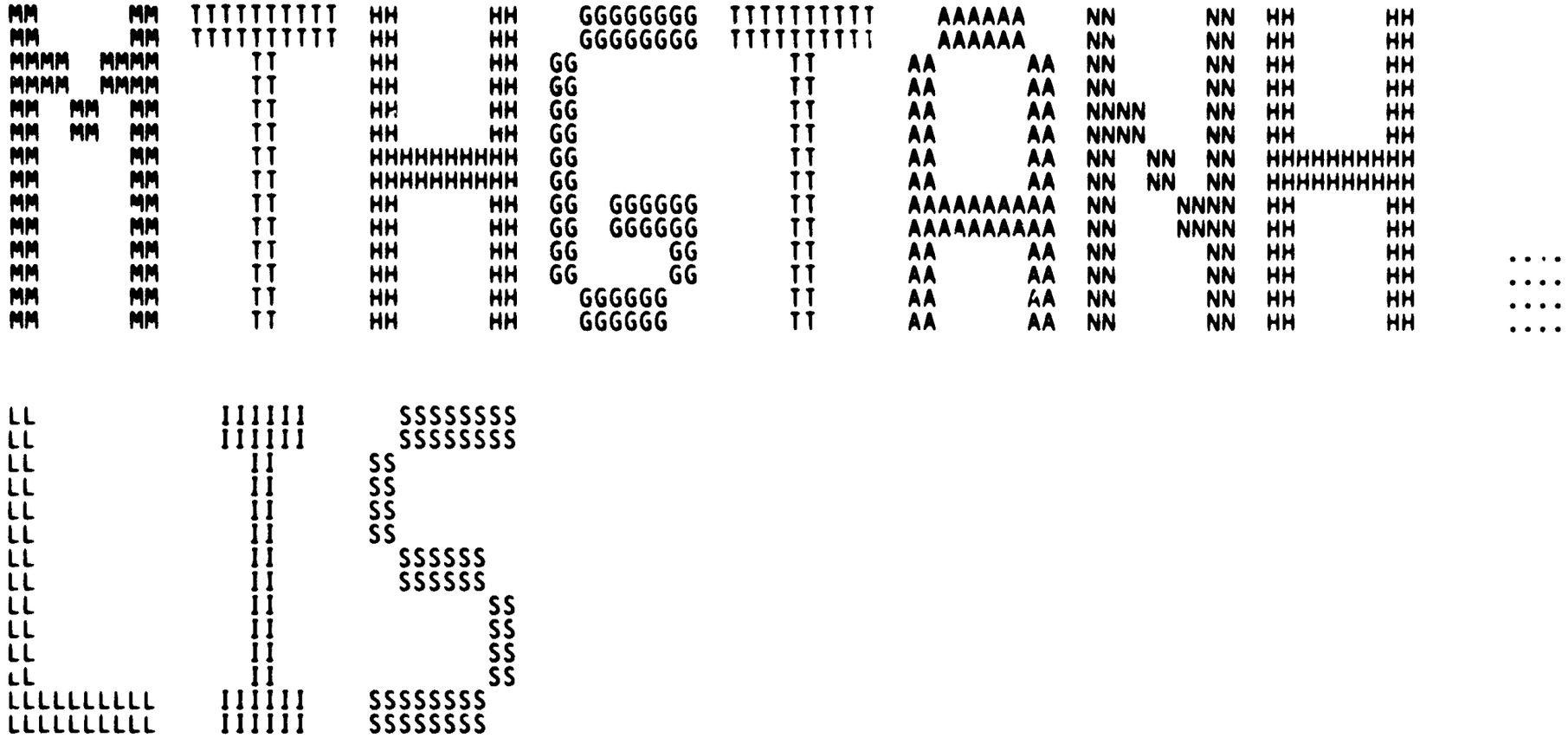
```
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMMMMM    MMMMMM        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 GGGGGGGG TTTTTTTTTT AAAAAA NN NN HH HH
MM MM TTTTTTTTTT HH HH GGGGGGGG TTTTTTTTTT AAAAAA NN NN HH HH
MMMM MMMM TT HH HH GG TT AA AA NN NN HH HH
MMMM MMMM TT HH HH GG TT AA AA NN NN HH HH
MM MM MM TT HH HH GG TT AA AA NNNN NN HH HH
MM MM MM TT HH HH GG TT AA AA NNNN NN HH HH
MM MM TT HHHHHHHHHH GG TT AA AA NN NN NN HHHHHHHHHH
MM MM TT HHHHHHHHHH GG TT AA AA NN NN NN HHHHHHHHHH
MM MM TT HH HH GG GGGGG TT AAAAAAAAAA NN NNNN HH HH
MM MM TT HH HH GG GGGGG TT AAAAAAAAAA NN NNNN HH HH
MM MM TT HH HH GG GG TT AA AA NN NN HH HH
MM MM TT HH HH GG GG TT AA AA NN NN HH HH
MM MM TT HH HH GGGGGG TT AA AA NN NN HH HH
MM MM TT HH HH GGGGGG TT AA AA NN NN HH HH
LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSS
LLLLLLLLLL IIIIII SSSSSSS

```
0000     1          .TITLE  MTH$GTANH       ; G Floating Hyperbolic Tangent routine
0000     2                                  ; (DTANH)
0000     3          .IDENT /1-005/          ; File: MTHGTANH.MAR  Edit: JCW1005
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$GTANH is a function  which  returns the G floating hyperbolic tangent
0000    34 ; of its G 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 ;       Steven B. Lionel, 26-Jan-79: Version 1
0000    44 ;
0000    45 ; MODIFIED BY:
0000    46 ;
0000    47 ;
0000    48 ;
```

```
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$GTANH
0000     56 ;
0000     57 ; 1-001 - Adapted from MTH$DTANH version 1-006.  SBL 26-Jan-79
0000     58 ; 1-002 - Use MTH$GEXP_R6.  SBL 27-Sept-1979
0000     59 ; 1-003 - Change constant 16.0 to 20.0 to correct inaccuracy.  The
0000     60 ;          value of X above which 1.0 is the best machine approximation
0000     61 ;          to GTANH(X) is about 19.06.  The next higher number that can
0000     62 ;          be represented as a short literal is 20.0.  JAW 19-Sep-80
0000     63 ; 1-004 - Eliminated symbolic short literals. RNH 15-Oct-81
0000     64 ; 1-005 - Changed the constant 2^-14 to 2^-27 to correct inaccuracy. For
0000     65 ;          values of !X! between 2^-14 and 2^-27 GTANH was only accurate
0000     66 ;          to 8 decimal places since the assumption that GTANH(x)=x in that
0000     67 ;          range of values is false. GTANH(X)=X for !X!<=2^-27 and
0000     68 ;          GTANH(X)=GSINH(X)/GCOSH(X) for 2^-27<!X!<=.25 . All appropriate
0000     69 ;          references to 2^-14 have been changed to 2^-27.  JCW 10-Jan-83
0000     70
```

```
                                0000     72          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                                0000     73
                                0000     74 ;
                                0000     75 ; INCLUDE FILES:
                                0000     76 ;
                                0000     77 ; EXTERNAL SYMBOLS:     MTH$JACKET_HDLR
                                0000     78 ;
                                0000     79          .DSABL  GBL                     ; Force .EXTRN on all symbols
                                0000     80          .EXTRN  MTH$GCOSH               ; GCOSH
                                0000     81          .EXTRN  MTH$GSINH               ; GSINH
                                0000     82          .EXTRN  MTH$GEXP_R6             ; EXP
                                0000     83 ; EQUATED SYMBOLS:
                                0000     84
                     00000004   0000     85          value   = 4                     ; value.rg.r
                                0000     86
                                0000     87 ;
                                0000     88 ; MACROS:       none
                                0000     89 ;
                                0000     90 ; PSECT DECLARATIONS:
                                0000     91
                     00000000   0000     92          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                                0000     93                                  ; program section for math routines
                                0000     94 ;
                                0000     95 ; OWN STORAGE:  none
                                0000     96 ;
                                0000     97 ;
                                0000     98 ; CONSTANTS:
                                0000     99 ;
                                0000    100
                                0000    101 G_0.25:
0000 0000 0000 3FF0             0000    102          .WORD   ^X3FF0, 0, 0, 0         ; 0.25
                                0008    103 G_2_POWER_M27:
0000 0000 0000 3E60             0008    104          .WORD   ^X3E60, 0, 0, 0         ; 2**-27
                                0010    105
```

```
                     0010   107          .SBTTL  MTH$GTANH  - G Double Precision Floating DTANH
                     0010   108
                     0010   109
                     0010   110 ;++
                     0010   111 ; FUNCTIONAL DESCRIPTION:
                     0010   112 ;
                     0010   113 ; GTANH  - G floating point function
                     0010   114 ;
                     0010   115 ; GTANH(X) is computed as:
                     0010   116 ;
                     0010   117 ;       If |X| =< 2**-27, then GTANH(X) = X.
                     0010   118 ;       If 2**-27 < |X| =< 0.25, then GTANH(X) = GSINH(X)/GCOSH(X).
                     0010   119 ;       If 0.25 < |X  < 20.0, then GTANH(X) = (GEXP(2*X) - 1) / (GEXP(2*X) + 1)
                     0010   120 ;       If 20.0 =< |X|, then GTANH(X) = sign(X) * 1
                     0010   121 ;
                     0010   122 ; CALLING SEQUENCE:
                     0010   123 ;
                     0010   124 ;       GTANH.wg.v = MTH$GTANH(x.rg.r)
                     0010   125 ;
                     0010   126 ; INPUT PARAMETERS:
                     0010   127
00000004             0010   128          LONG = 4                        ; define longword multiplier
00000004             0010   129          x = 1 * LONG                    ; Contents of x is the argument
                     0010   130
                     0010   131 ; IMPLICIT INPUTS:      none
                     0010   132 ;
                     0010   133 ; OUTPUT PARAMETERS:
                     0010   134 ;
                     0010   135 ;       VALUE:  G floating hyperbolic tangent of the argument
                     0010   136 ;
                     0010   137 ; IMPLICIT OUTPUTS:     none
                     0010   138 ;
                     0010   139 ; COMPLETION CODES:     none
                     0010   140 ;
                     0010   141 ; SIDE EFFECTS: none
                     0010   142 ;
                     0010   143 ; NOTE: This procedure disables floating point underflow, enables integer
                     0010   144 ; overflow.
                     0010   145 ;
                     0010   146 ;---
                     0010   147
                     0010   148
                 407C 0010   149          .ENTRY  MTH$GTANH, ^M<IV, R2, R3, R4, R5, R6>
                     0012   150                                          ;standard call-by-reference entry
                     0012   151                                          ; disable DV (and FU), enable IV
                     0012   152          MTH$FLAG_JACKET                 ; flag that this is a jacket procedure in
                     0012
6D   00000000'GF  9E 0012                MOVAB   G^MTH$$JACKET_HND, (FP)
                     0019                                                ; set handler address to jacket
                     0019                                                ; handler
                     0019
                     0019   153                                          ; case of an error in routine
                     0019   154                                          ; If an error, convert signal to user PC
                     0019   155                                          ; and resignal
  50   04 BC 50FD  0019   156          MOVG    @value(AP), R0              ; R0/R1 = |X| = @value(AP)
50   8000 8F  AA  001E   157          BICW    #^X8000, R0                 ; R0/R1 = |X|
  E0 AF  50 51FD  0023   158          CMPG    R0, G_2_POWER_M27           ; compare |X| with 2**-27
```

```
                     47   15  0028   159          BLEQ    OUT_X                   ; branch if .X' =< 2**-27
                                 002A   160
                                 002A   161 ;
                                 002A   162 ; 2**-27 <  X
                                 002A    63 ;
                                 002A   164
                     2A  50 51FD 002A   165          CMPG    R0, #20                 ; compare !X  with 20.0
                         32   18 002E   166          BGEQ    GEQ_TO_20.0             ; branch if !X! >= 20.0
                                 0030   167
                                 0030   168 ;
                                 0030   169 ; 2**-27 < !X' < 20.0
                                 0030   170 ;
                                 0030   171
                  CB AF  50 51FD 0030   172          CMPG    R0, G_0.25              ; compare !X! with 0.25
                         19   15 0035   173          BLEQ    LEQ_TO_0.25             ; branch if .X  =< 0.25
                                 0037   174
                                 0037   175 ;
                                 0037   176 ; 0.25 <  X! < 20.0
                                 0037   177 ;
                                 0037   178
           50  04 BC   04 BC 41FD 0037   179          ADDG3   @value(AP), @value(AP), R0
                                 003E   180                                          ; R0/R1 = 2*X
                0000'CF   16     003E   181          JSB     W^MTH$GEXP_R6           ; R0/R1 = GEXP(2*X)
                52  50   08 41FD 0042   182          ADDG3   #1, R0, R2              ; R2/R3 = GEXP(2*X) + 1
                    50   08 42FD 0047   183          SUBG2   #1, R0                  ; R0/R1 = GEXP(2*X) - 1
                    50   52 46FD 004B   184          DIVG2   R2, R0                  ; R0/R1 = (GEXP(2*X) - 1) / (GEXP(2*X) + 1)
                              04 004F   185          RET                             ; return with result in R0/R1
                                 0050   186
                                 0050   187 ;
                                 0050   188 ; 2**-1R6 < !X! =< 0.25
                                 0050   189 ;
                                 0050   190
                                 0050   191 LEQ_TO_0.25:
           0000'CF   6C   FA     0050   192          CALLG   (AP), W^MTH$GCOSH               ; R0/R1 = GCOSH(X)
                52   50   7D     0055   193          MOVQ    R0, R2                  ; R2/R3 = GCOSH(X)
           0000'CF   6C   FA     0058   194          CALLG   (AP), W^MTH$GSINH               ; R0/R1 = GSINH(X)
                50   52 46FD     005D   195          DIVG2   R2, R0                  ; R0/R1 = GSINH(X) / GCOSH(X)
                              04 0061   196          RET                             ; return with result in R0
                                 0062   197
                                 0062   198 ;
                                 0062   199 ; !X! >= 20.0
                                 0062   200 ;
                                 0062   201
                                 0062   202 GEQ_TO_20.0:
                     50   08 50FD 0062   203          MOVG    #1, R0                  ; R0/R1 = 1.0
                       04 BC 53FD 0066   204          TSTG    @value(AP)              ; test the sign of X
                          04   18 006A   205          BGEQ    10$                     ; branch if X >= 0
                     50   50 52FD 006C   206          MNEGG   R0, R0                  ; R0/R1 = -1
                              04 0070   207 10$:       RET                             ; return with result in R0
                                 0071   208
                                 0071   209 ;
                                 0071   210 ; !X! =< 2**-27
                                 0071   211 ;
                                 0071   212
                  50   04 BC   7D 0071   213 OUT_X:   MOVQ    @value(AP), R0          ; R0/R1 = GTANH(X) = X
                              04 0075   214          RET                             ; return with result in R0/R1
                                 0076   215
```

```
0076   216
0076   217
0076   218          .END
```

```
GEQ_TO_20.0         00000062 R      01
G_0.25              00000000 R      01
G_2_POWER_M27       00000008 R      01
LEQ_TO_0.25         00000050 R      01
LONG            =   00000004
MTH$$JACKET_HND     ********    X   01
MTH$GCOSH           ********    X   00
MTH$GEXP_R6         ********    X   00
MTH$GSINH           ********    X   00
MTH$GTANH           00000010 RG     01
OUT_X               00000071 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	00000076 (118.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.09	00:00:00.74
Command processing	112	00:00:00.65	00:00:03.02
Pass 1	83	00:00:00.70	00:00:04.34
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	52	00:00:00.58	00:00:01.95
Symbol table output	2	00:00:00.02	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.18
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	283	00:00:02.06	00:00:10.31

The working set limit was 900 pages.
3309 bytes (7 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 13 non-local and 1 local symbols.
278 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.

There were no errors, warnings or information messages.

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