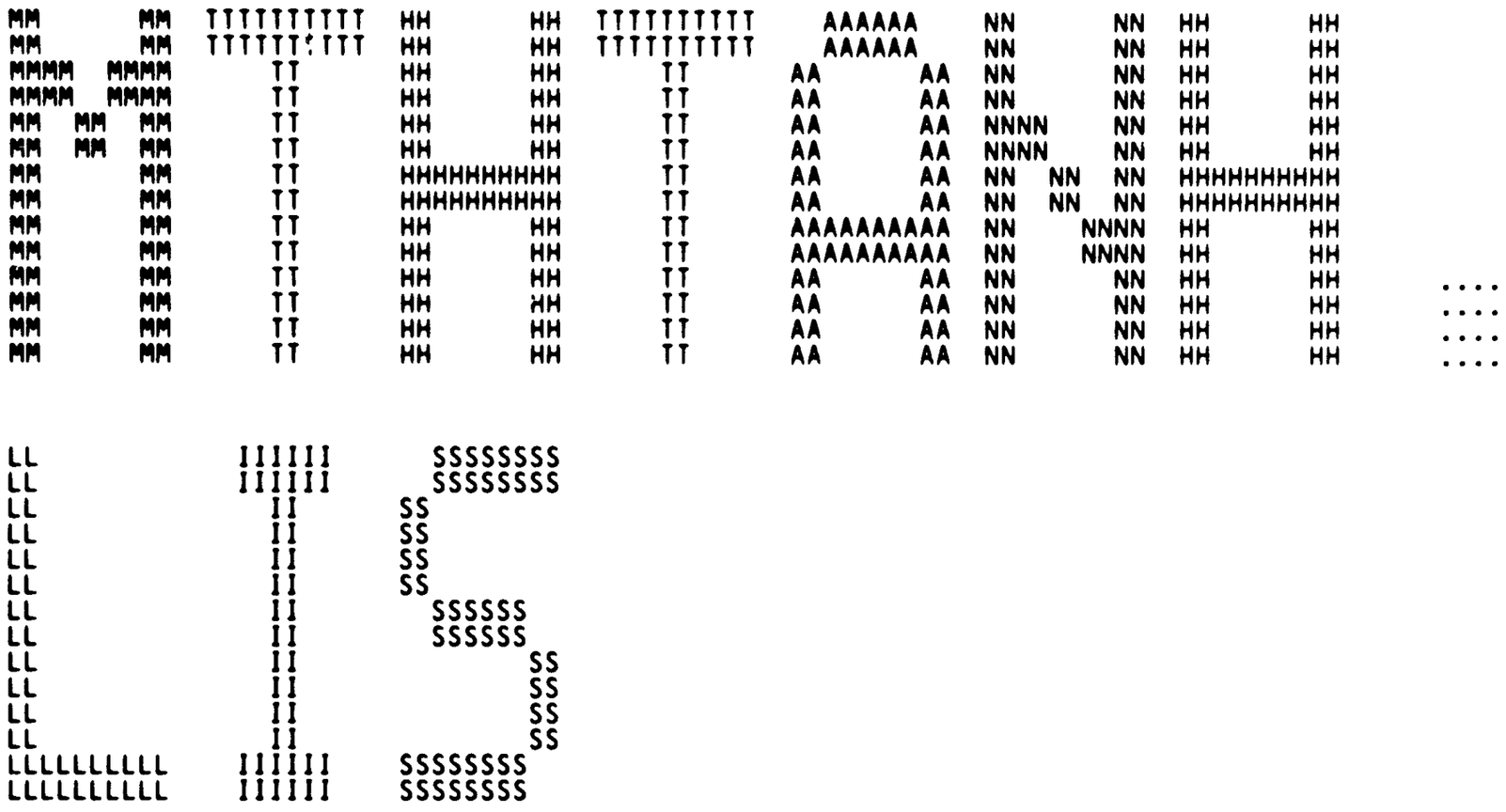
```
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMM          MMM  TTTTTTTTTTTTTTT   HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT   LLL
MMMMMM    MMMMMM        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 TTTTTTTTTT AAAAAA NN NN HH HH
MM MM TTTTTTTTTT HH HH TTTTTTTTTT AAAAAA NN NN HH HH
MMMM MMMM TT HH HH TT AA AA NN NN HH HH
MMMM MMMM TT HH HH TT AA AA NN NN HH HH
MM MM MM TT HH HH TT AA AA NNNN NN HH HH
MM MM MM TT HH HH TT AA AA NNNN NN HH HH
MM MM TT HHHHHHHHHH TT AA AA NN NN NN HHHHHHHHHH
MM MM TT HHHHHHHHHH TT AA AA NN NN NN HHHHHHHHHH
MM MM TT HH HH TT AAAAAAAAAA NN NNNN HH HH
MM MM TT HH HH TT AAAAAAAAAA NN NNNN HH HH
MM MM TT HH HH TT AA AA NN NN HH HH
MM MM TT HH HH TT AA AA NN NN HH HH
MM MM TT HH HH TT AA AA NN NN HH HH
LL II SSSSSSS
LL II 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$TANH        ; Floating Point Hyperbolic Tangent routine
0000     2                                  ; (TANH)
0000     3          .IDENT /1-009/          ; File: MTHTANH.MAR  Edit: JCW1009
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$TANH is a function  which  returns the floating point hyperbolic tangent
0000    34 ; of its single precision floating point argument. The call is standard
0000    35 ; call-by-reference.
0000    36 ;
0000    37 ;--
0000    38 ;
0000    39 ; VERSION: 01
0000    40 ;
0000    41 ; HISTORY:
0000    42 ; AUTHOR:
0000    43 ;       Peter Yuo, 29-Jun-77: Version 01
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 01 of MTH$TANH
0000     56 ;
0000     57 ; 0-4    - Call MTH$COSH, MTH$SINH directly instead of using POLY.  TNH 13-June-78
0000     58 ; 0-5    - Declare externals, word offsets.  TNH 13-June-78
0000     59 ; 1-006 - Put version number in standard format, with three digits of
0000     60 ;           edit number, and update copyright notice.  JBS 16-NOV-78
0000     61 ; 1-007 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000     62 ; 1-008 - Change constant 16.0 to 10.0 for improved efficiency.  The
0000     63 ;           value of X above which 1.0 is the best machine approximation
0000     64 ;           to TANH(X) is about 9.01.  The next higher number that can be
0000     65 ;           represented as a short literal is 10.0.  JAW 19-Sep-80
0000     66 ; 1-009 - Changed the constant 2^-14 to 2^-12 to correct inaccuracy. For
0000     67 ;           values of !X! between 2^-14 and 2^-12 no noticeable loss of
0000     68 ;           significance was noticed by the assumption that TANH(x)=x starting
0000     69 ;           at !X!<=2^-14 instead of 2^-12, but a loss of performance was felt
0000     70 ;           do to an unnecessary computation of SINH(X)/COSH(X). All appropriate
0000     71 ;           references to 2^-14 have been changed to 2^-12.  JCW 10-Jan-83
0000     72
```

```
                     0000     74          .SBTTL  DECLARATIONS    : Declarative Part of Module
                     0000     75
                     0000     76 ;
                     0000     77 ; INCLUDE FILES:         OTSPARAMS.MAR
                     0000     78 ;
                     0000     79 ; EXTERNAL SYMBOLS:      MTH$JACKET_HDLR
                     0000     80 ;
                     0000     81          .DSABL  GBL                         ; Force all symbols to need .EXTERN
                     0000     82          .EXTRN  MTH$EXP_R4                  ; EXP
                     0000     83          .EXTRN  MTH$COSH                    ; COSH
                     0000     84          .EXTRN  MTH$SINH                    ; SINH
                     0000     85 ; EQUATED SYMBOLS:
                     0000     86
00003F80             0000     87          LF_0.25 = 0@16 + ^X3F80             ; 0.25
00003A80             0000     88          LF_2_POWER_M12  = 0@16 + ^X3A80     ; 2**-12
00004080             0000     89          SF_1.0  = ^F1.0                     ; 1.0
00004220             0000     90          SF_10.0 = ^F10.0                            ; 10.0
00000004             0000     91          value   = 4                         ; value.rf.r
                     0000     92
                     0000     93 ;
                     0000     94 ; MACROS:        none
                     0000     95 ;
                     0000     96 ; PSECT DECLARATIONS:
                     0000     97
                 00000000     98          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                     0000     99                                              ; program section for math routines
                     0000    100 ;
                     0000    101 ; OWN STORAGE:  none
                     0000    102 ;
                     0000    103 ;
                     0000    104 ; CONSTANTS:
                     0000    105 ;
                     0000    106 ;       NONE
```

```
                        0000   108          .SBTTL  MTH$TANH  - Standard Single Precision Floating TANH
                        0000   109
                        0000   110
                        0000   111 ;++
                        0000   112 ; FUNCTIONAL DESCRIPTION:
                        0000   113 ;
                        0000   114 ; TANH  - single precision floating point function
                        0000   115 ;
                        0000   116 ; TANH(X) is computed as:
                        0000   117 ;
                        0000   118 ;       If |X| =< 2**-12, then TANH(X) = X.
                        0000   119 ;       If 2**-12 < |X| =< 0.25, then TANH(X) = SINH(X)/COSH(X).
                        0000   120 ;       If 0.25 < |X| < 10.0, then TANH(X) = (EXP(2*X) - 1) / (EXP(2*X) + 1)
                        0000   121 ;       If 10.0 =< |X|, then TANH(X) = sign(X) * 1
                        0000   122 ;
                        0000   123 ; CALLING SEQUENCE:
                        0000   124 ;
                        0000   125 ;       TANH.wf.v = MTH$TANH(x.rf.r)
                        0000   126 ;
                        0000   127 ; INPUT PARAMETERS:
                        0000   128
00000004                0000   129          LONG = 4                        ; define longword multiplier
00000004                0000   130          x = 1 * LONG                    ; Contents of x is the argument
                        0000   131
                        0000   132 ; IMPLICIT INPUTS:      none
                        0000   133 ;
                        0000   134 ; OUTPUT PARAMETERS:
                        0000   135 ;
                        0000   136 ;       VALUE:  floating hyperbolic tangent of the argument
                        0000   137 ;
                        0000   138 ; IMPLICIT OUTPUTS:     none
                        0000   139 ;
                        0000   140 ; COMPLETION CODES:     none
                        0000   141 ;
                        0000   142 ; SIDE EFFECTS: none
                        0000   143 ;
                        0000   144 ; NOTE: This procedure disables floating point underflow, enables integer
                        0000   145 ; overflow.
                        0000   146 ;
                        0000   147 ;--
                        0000   148
                        0000   149
                  401C  0000   150          .ENTRY  MTH$TANH, ^M<IV, R2, R3, R4>
                        0002   151                                          ; standard call-by-reference entry
                        0002   152                                          ; disable DV (and FU), enable IV
                        0002   153          MTH$FLAG_JACKET                ; flag that this is a jacket procedure in
                        0002
6D    00000000'GF   9E  0002               MOVAB   G^MTH$$JACKET_HND, (FP)
                        0009                                                ; set handler address to jacket
                        0009                                                ; handler
                        0009
                        0009   154                                          ; case of an error in routine
                        0009   155                                          ; If an error, convert signal to user PC
                        0009   156                                          ; and resignal
    50    04 BC   50    0009   157          MOVF    @value(AP), R0         ; R0 = |X| = @value(AP)
 50    8000 8F   AA     000D   158          BICW    #^X8000, R0            ; R0 = |X|
00003A80 8F    50   51  0012   159          CMPF    R0, #LF_2_POWER_M12    ; compare |X| with 2**-12
```

```
                     46   15  0019   160          BLEQ    OUT_X                   ; branch if !X. =< 2**-12
                                 001B   161
                                 001B   162 ;
                                 001B   163 ; 2**-12 < !X'
                                 001B   164 ;
                                 001B   165
                          22     50   51  001B   166          CMPF    R0, S^#SF_10.0          ; compare 'X! with 10.0
                                 35   18  001E   167          BGEQ    GEQ_TO_10.0             ; branch if 'X. >= 10.0
                                 0020   168
                                 0020   169 ;
                                 0020   170 ; 2**-12 < !X! < 10.0
                                 0020   171 ;
                                 0020   172
                00003F80 8F      50   51  0020   173          CMPF    R0, #LF_0.25            ; compare !X! with 0.25
                                 17   15  0027   174          BLEQ    LEQ_TO_0.25             ; branch if !X! =< 0.25
                                 0029   175
                                 0029   176 ;
                                 0029   177 ; 0.25 < !X! < 10.0
                                 0029   178 ;
                                 0029   179
           50    04 BC    04 BC   41   0029   180          ADDF3   @value(AP), @value(AP), R0
                                 002F   181                                           ; R0 = 2*X
                 00000000'GF     16   002F   182          JSB     G^MTH$EXP_R4            ; R0 = EXP(2*X)
                 51    50    08   41   0035   183          ADDF3   S^#SF_1.0, R0, R1       ; R1 = EXP(2*X) + 1
                       50    08   42   0039   184          SUBF    S^#SF_1.0, R0           ; R0 = EXP(2*X) - 1
                       50    51   46   003C   185          DIVF    R1, R0                  ; R0 = (EXP(2*X) - 1) / (EXP(2*X) + 1)
                                 04   003F   186          RET                             ; return with result in R0
                                 0040   187
                                 0040   188 ;
                                 0040   189 ; 2**-12 < !X! =< 0.25
                                 0040   190 ;
                                 0040   191
                                 0040   192 LEQ_TO_0.25:
00000000'GF      6C       FA     0040   193          CALLG   (AP), G^MTH$COSH                ; R0 = COSH(X)
          52     50       50     0047   194          MOVF    R0, R2                  ; R2 = COSH(X)
00000000'GF      6C       FA     004A   195          CALLG   (AP), G^MTH$SINH                ; R0 = SINH(X)
          50     52       46     0051   196          DIVF    R2, R0                  ; R0 = SINH(X) / COSH(X)
                                 04   0054   197          RET                             ; return with result in R0
                                 0055   198
                                 0055   199 ;
                                 0055   200 ; !X! >= 10.0
                                 0055   201 ;
                                 0055   202
                                 0055   203 GEQ_TO_10.0:
                50     08        50   0055   204          MOVF    S^#SF_1.0, R0           ; R0 = 1.0
                  04 BC           53   0058   205          TSTF    @value(AP)              ; test the sign of X
                       03        18   005B   206          BGEQ    10$                     ; branch if X >= 0
                50     50        52   005D   207          MNEGF   R0, R0                  ; R0 = -1
                                 04   0060   208 10$:      RET                             ; return with result in R0
                                 0061   209
                                 0061   210 ;
                                 0061   211 ; !X! =< 2**-12
                                 0061   212 ;
                                 0061   213
           50    04 BC           50   0061   214 OUT_X:   MOVF    @value(AP), R0          ; R0 = TANH(X) = X
                                 04   0065   215          RET                             ; return with result in R0
                                 0066   216
```

```
0066   217
0066   218
0066   219          .END
```

```
GEQ_TO_10.0         00000055 R      01
LEQ_TO_0.25         00000040 R      01
LF_0.25         =   00003F80
LF_2_POWER_M12  =   00003A80
LONG            =   00000004
MTH$$JACKET_HND     ********    X   01
MTH$COSH            ********    X   00
MTH$EXP_R4          ********    X   00
MTH$SINA            ********    X   00
MTH$TANH            00000000 RG     01
OUT_X               00000061 R      01
SF_1.0          =   00004080
SF_10.0         =   00004220
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	00000066 (102.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.10	00:00:01.48
Command processing	119	00:00:00.69	00:00:03.92
Pass 1	86	00:00:00.73	00:00:04.03
Symbol table sort	0	00:00:00.01	00:00:00.00
Pass 2	53	00:00:00.55	00:00:03.08
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	294	00:00:02.12	00:00:12.65

The working set limit was 900 pages.
3329 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 1 local symbols.
279 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$:MTHTANH/OBJ=OBJ$:MTHTANH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:
```

OTSMULCD
LIS

OTSPOWCGC
LIS

OTSDIVC
LIS

OTSPOWDD
LIS

OTSPOWCC
LIS

OTSPOWGG
LIS

OTSPOWGJ
LIS

OTSPOWCDJ
LIS

MTHTAN
LIS

MTHVECTOR
LIS

OTSDIVCG
LIS

OTSPOWCJ
LIS

OTSPOWDLJ
LIS

MTHTANH
LIS

OTSMULCG
LIS

OTSPOWCGJ
LIS

OTSPOWDJ
LIS

OTSDIVCD
LIS

OTSPOWCDC
LIS