```
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**MTHDCOSH


```
MM       MM  TTTTTTTTTT  HH      HH  DDDDDDDD      CCCCCCCC    OOOOOO      SSSSSSSS  HH      HH
MM       MM  TTTTTTTTTT  HH      HH  DDDDDDDD      CCCCCCCC    OOOOOO      SSSSSSSS  HH      HH
MMMM   MMMM      TT      HH      HH  DD      DD  CC            OO      OO  SS        HH      HH
MMMM   MMMM      TT      HH      HH  DD      DD  CC            OO      OO  SS        HH      HH
MM  MM  MM      TT      HH      HH  DD      DD  CC            OO      OO  SS        HH      HH
MM  MM  MM      TT      HH      HH  DD      DD  CC            OO      OO  SS        HH      HH
MM      MM      TT      HHHHHHHHHH  DD      DD  CC            OO      OO    SSSSSS  HHHHHHHHHH
MM      MM      TT      HHHHHHHHHH  DD      DD  CC            OO      OO    SSSSSS  HHHHHHHHHH
MM      MM      TT      HH      HH  DD      DD  CC            OO      OO        SS  HH      HH
MM      MM      TT      HH      HH  DD      DD  CC            OO      OO        SS  HH      HH
MM      MM      TT      HH      HH  DD      DD  CC            OO      OO        SS  HH      HH    ....
MM      MM      TT      HH      HH  DD      DD  CC            OO      OO        SS  HH      HH    ....
MM      MM      TT      HH      HH  DDDDDDDD      CCCCCCCC    OOOOOO    SSSSSSSS  HH      HH    ....
MM      MM      TT      HH      HH  DDDDDDDD      CCCCCCCC    OOOOOO    SSSSSSSS  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
```

Table of contents

```
0000     1          .TITLE  MTH$DCOSH       ; Double Floating Hyperbolic Cosine routine
0000     2                                  ; (DCOSH)
0000     3          .IDENT /1-008/          ; File: MTHDCOSH.MAR  Edit: RNH1008
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$DCOSH is a function  which  returns the double floating hyperbolic cosine
0000    34 ; of its double 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$DCOSH
0000     56 ;
0000     57 ; 0-2    MTH$$ERROR changed to MTH$$SIGNAL.
0000     58 ;        MTH$_... changed to MTH__....
0000     59 ;        Changed error handling mechanism. Put error result in R0:R1 before
0000     60 ;        calling MTH$$SIGNAL in order to allow user modify error result.
0000     61 ;
0000     62 ;0-3     Seven term Taylor series, in powers of argument, replaced
0000     63 ;        by six term Chebyshev series, in powers of ARG**2,
0000     64 ;        with overhang, to improve accuracy. 18-May-1978; Mary Payne
0000     65 ; 0-4    Fix LOG(2) constant.  TNH 16-June-78
0000     66 ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000     67 ; 1-002 - Change MTH__FLOOVEMAT to MTH$K_FLOOVEMAT.  JBS 07-DEC-78
0000     68 ; 1-003 - Removed $SSRMDEF - not needed.  JBS 16-DEC-78
0000     69 ; 1-004 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000     70 ; 1-005 - Declare externals.  SBL 17-May-1979
0000     71 ; 1-006 - Use MTH$DEXP_R6.  SBL 27-Sept-1979
0000     72 ; 1-007 - Changed lower limit for Chebyshev approximation from 2**-27 to
0000     73 ;          2**-28.
0000     74 ;        - Eliminated second call to EXP for input values between .25 and
0000     75 ;          28.5*ln2 by computing COSH(x) = (Z + 1/Z)/2, with Z = EXP(|x|).
0000     76 ;        - Eliminated second call to EXP for input values between 28.5*ln2
0000     77 ;          and 127*ln2.
0000     78 ;        - Changed all final floating point divisions by 2 to interger
0000     79 ;          subrtracts of 1 from the exponent field.
0000     80 ;        - Changed entry mask to excluded R7 - no longer needed.
0000     81 ;        - Extended maximum range from 87.69 to 128*ln2=88.72.
0000     82 ;        - Chaneged logic for computing EXP(|x|-ln2) to reduce error.
0000     83 ;        - RNH 10-FEB-81
0000     84 ; 1-003 - Changed W^ to G^ on call to MTH$$SIGNAL RNH 09-Sept-1981
```

```
                     0000     86          .SBTTL  DECLARATIONS    ; Declarative Part of Module
                     0000     87
                     0000     88 ;
                     0000     89 ; INCLUDE FILES:
                     0000     90 ;
                     0000     91
                     0000     92 ;
                     0000     93 ; EXTERNAL SYMBOLS:
                     0000     94 ;
                     0000     95          .DSABL  GBL
                     0000     96          .EXTRN  MTH$DEXP_R6
                     0000     97          .EXTRN  MTH$K_FLOOVEMAT
                     0000     98          .EXTRN  MTH$$SIGNAL
                     0000     99
                     0000    100 ;
                     0000    101 ; EQUATED SYMBOLS:
                     0000    102
00004080             0000    103          SD_1.0  = ^F1.0                 ; 1.0
00000004             0000    104          value   = 4                     ; value.rd.r
                     0000    105
                     0000    106 ;
                     0000    107 ; MACROS:       none
                     0000    108 ;
                     0000    109 ;
                     0000    110 ; PSECT DECLARATIONS:
                     0000    111
00000000             0000    112          .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                     0000    113                                          ; program section for math routines
                     0000    114 ;
                     0000    115 ; OWN STORAGE:  none
                     0000    116 ;
                     0000    117 ;
                     0000    118 ; CONSTANTS:
                     0000    119 ;
                     0000    120
                     0000    121 D_127_LOG_2:
2BDAC7E2 0F3343B0    0000    122          .QUAD   ^X2BDAC7E20F3343B0      ; 127*ln2
                     0008    123 D_128_LOG_2:
CF79F7D1 721743B1    0008    124          .QUAD   ^XCF79F7D1721743B1      ; 128*ln2
                     0010    125 D_2_POWER_28.5:
DE6433F9 04F34EB5    0010    126          .QUAD   ^XDE6433F904F34EB5      ; 2**28.5
                     0018    127 D_LOG_2_HI:
CF80F7D1 72174031    0018    128          .QUAD   ^XCF80F7D172174031      ; (high 49 bits of ln2) + 2**-49
                     0020    129 D_LOG_2_LO:
FF81898C 86C3A5CA    0020    130          .QUAD   ^XFF81898C86C3A5CA      ; ln2 - D_LOG_2_HI
                     0028    131
                     0028    132 DCOSHTAB:
          2853 3594  0028    133          .WORD   ^O032624,^O024123
          DBD2 3EEC  002C    134          .WORD   ^O037354,^O155722       ; DECIMAL: 0.2759648863787355D-06
          0CF1 38D0  0030    135          .WORD   ^O034320,^O006361
          CA4F AB28  0034    136          .WORD   ^O125450,^O145117       ; DECIMAL: 0.2480155975461668D-04
          0B60 3BB6  0038    137          .WORD   ^O035666,^O005540
          FE08 B801  003C    138          .WORD   ^O134001,^O177010       ; DECIMAL: 0.1388888889781712D-02
          AAAA 3E2A  0040    139          .WORD   ^O037052,^O125252
          6FC4 AAAA  0044    140          .WORD   ^O125252,^O067704       ; DECIMAL: 0.4166666666665359D-01
          0000 4000  0048    141          .WORD   ^O040000,0
          0005 0000  004C    142          .WORD   0,^O0000005             ; DECIMAL: 0.5000000000000000D0
```

```
0000 4080  0050   143          .WORD   ^0040200,0
0000 0000  0054   144          .WORD   0,0                     ; DECIMAL: 0.1000000000000000D+01
 00000006  0058   145 DCOSHLEN        = .- DCOSHTAB/8
           0058   146
```

```
                     0058   148          .SBTTL  MTH$DCOSH  - Standard Double Precision Floating DCOSH
                     0058   149
                     0058   150
                     0058   151 ;++
                     0058   152 ; FUNCTIONAL DESCRIPTION:
                     0058   153 ;
                     0058   154 ; DCOSH  - double precision floating point function
                     0058   155 ;
                     0058   156 ; DCOSH(X) is computed as:
                     0058   157 ;
                     0058   158 ;       If |X| < 2**-28, DCOSH(X) = 1.
                     0058   159 ;       IF 2**-28 =< |X| < 0.25, DCOSH(X) = Chebyshev series
                     0058   160 ;       If 0.25 =< |X| =< 28.5*ln2, set Z = DEXP(|X|) and compute
                     0058   161 ;           DCOSH(X) = (Z + 1/Z)/2.
                     0058   162 ;       If 28.5*ln2 < |X| =< 127*ln2  COSH(X) = EXP(|X|)/2
                     0058   163 ;       If 127*ln2 < |X| =< 128*ln2 then DCOSH(X) = DEXP(|X|-LOG(2)).
                     0058   164 ;       If 128*ln2 < |X| then overflow.
                     0058   165 ;
                     0058   166 ; CALLING SEQUENCE:
                     0058   167 ;
                     0058   168 ;       DCOSH.wd.v = MTH$DCOSH(x.rd.r)
                     0058   169 ;
                     0058   170 ; INPUT PARAMETERS:
                     0058   171
           00000004  0058   172          LONG = 4                        ; define longword multiplier
           00000004  0058   173          x = 1 * LONG                    ; Contents of x is the argument
                     0058   174
                     0058   175 ; IMPLICIT INPUTS:      none
                     0058   176 ;
                     0058   177 ; OUTPUT PARAMETERS:
                     0058   178 ;
                     0058   179 ;       VALUE:  double precision floating hyperbolic cosine of the argument
                     0058   180 ;
                     0058   181 ; IMPLICIT OUTPUTS:     none
                     0058   182 ;
                     0058   183 ; COMPLETION CODES:     none
                     0058   184 ;
                     0058   185 ; SIDE EFFECTS:
                     0058   186 ;
                     0058   187 ; Signal: MTH$_FLOOVEMAT if 128*ln2 < |X| with reserved operand in R0/R1
                     0058   188 ; (copied to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
                     0058   189 ; Associated message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved
                     0058   190 ; operand -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1
                     0058   191 ;
                     0058   192 ; NOTE: This procedure disables floating point underflow, enables integer
                     0058   193 ; overflow.
                     0058   194 ;
                     0058   195 ;---
                     0058   196
                     0058   197
                 407C  0058   198          .ENTRY  MTH$DCOSH, ^M<IV, R2, R3, R4, R5, R6 >
                     005A   199                                          ; standard call-by-reference entry
                     005A   200                                          ; disable DV (and FU), enable IV
                     005A   201          MTH$FLAG_JACKET                 ; flag that this is a jacket procedure in
                     005A                                                ;
  6D  00000000'GF     9E   005A          MOVAB   G^MTH$$JACKET_HND, (FP)
                     0061                                                ; set handler address to jacket
```

```
                                0061                                    ; handler
                                0061
                                0061    202                             ; case of an error in routine
                                0061    203                             ; If an error, convert signal to user PC
                                0061    204                             ; and resignal
  50    04 BC    70    0061    205             MOVD    @value(AP), R0          ; R0/R1 = |X| = @value(AP)
50    8000 8F    AA    0065    206             BICW2   #^X8000, R0             ; R0/R1 = |X|
3F80 8F    50    B1    006A    207             CMPW    R0, #^X3F80             ; compare |X| with 0.25
           14    18    006F    208             BGEQ    GEQ_TO_0.25             ; branch if |X| >= 0.25
                               0071    209
                               0071    210 ;
                               0071    211 ; |X  < 0.25
                               0071    212 ;
                               0071    213
3280 8F    50    B1    0071    214             CMPW    R0, #^X3280             ; compare |X| with 2**-28
           04    18    0076    215             BGEQ    GEQ_TO_2M28             ; branch if |X| >= 2**-28
                               0078    216
                               0078    217 ;
                               0078    218 ; |X| < 2**-28
                               0078    219 ;
                               0078    220
     50    08    70    0078    221             MOVD    S^#SD_1.0, R0           ; R0/R1 = 1.0
           04    0078    222             RET                             ; return with result = 1.0
                               007C    223
                               007C    224 ;
                               007C    225 ; 2**-28 =< |X| < 0.25
                               007C    226 ;
                               007C    227
                               007C    228 GEQ_TO_2M28:
     50    50    64    007C    229             MULD    R0,R0                   ;Get ARG**2 for POLYD.
A4 AF      05    50    75    007F    230             POLYD   R0, #DCOSHLEN-1, DCOSHTAB
                               0084    231                                     ; R0/R1 = SUM(Ci*X**i)
           04    0084    232             RET                             ; return with result in R0
                               0085    233
                               0085    234 ;
                               0085    235 ; 0.25 =< |X|
                               0085    236 ;
                               0085    237
                               0085    238 GEQ_TO_0.25:
  FF76 CF    50    71    0085    239             CMPD    R0, D_127_LOG_2         ; compare |X| with 127*ln2
           1A    14    008A    240             BGTR    GTR_THAN_127_LOG_2      ; branch if |X| > 127*ln2
                               008C    241
                               008C    242 ;
                               008C    243 ; 0.25 =< |X| =< 127*ln2
                               008C    244 ;
                               008C    245
 00000000'EF    16    008C    246             JSB     MTH$DEXP_R6             ; R0/R1 = DEXP(X)
FF79 CF    50    71    0092    247             CMPD    R0, D_2_POWER_28.5      ; See if the DEXP(-|X|) is significant
           07    14    0097    248             BGTR    ONE_TERM_ONLY           ; if not skip computation
52    08   50    67    0099    249             DIVD3   R0, S^#SD_1.0, R2       ; R2/R3 = DEXP(-|X|)
     50    52    60    009D    250             ADDD    R2, R0                  ; R0/R1 = DEXP(X) + DEXP(-X)
                               00A0    251 ONE_TERM_ONLY:
50    0080 8F    A2    00A0    252             SUBW    #^X0080, R0             ; R0/R1 = (DEXP(X) + DEXP(-X))/2
           04    00A5    253             RET                             ; return with result in R0/R1
                               00A6    254
                               00A6    255 ;
                               00A6    256 ; 127*ln2 < |X|
```

```
                                00A6    257 ;
                                00A6    258
                                00A6    259 GTR_THAN_127_LOG_2:
        FF5D CF    50    71     00A6    260          CMPD    R0, D_128_LOG_2         ; Check for possible overflow
                   15    14     00AB    261          BGTR    ERROR                   ;
        50   FF67 CF    62     00AD    262          SUBD    D_LOG_2_HI, R0          ;
         00000000'EF    16     00B2    263          JSB     MTH$DEXP_R6             ;
    52  50   FF64 CF    65     00B8    264          MULD3   D_LOG_2_LO, R0, R2      ;
             50   52    62     00BE    265          SUBD    R2, R0
                        04     00C1    266          RET
                                00C2    267 ;
                                00C2    268 ; 128*ln2 < |X|
                                00C2    269 ;
                                00C2    270
          7E    00'8F    9A     00C2    271 ERROR:  MOVZBL  #MTH$K_FLOOVEMAT, -(SP) ; condition value
        50    01    0F    79     00C6    272          ASHQ    #15, #1, R0             ; R0 = result = reserved operand -0.0
                                00CA    273                                          ; goes to signal mechanism vector
                                00CA    274                                          ; (CHF$L_MCH_R0/R1) so error handler
                                00CA    275                                          ; can modify the result.
 00000000'GF    01    FB     00CA    276          CALLS   #1, G^MTH$$SIGNAL       ; signal error and use real user's PC
                                00D1    277                                          ; independent of CALL vs JSB
                        04     00D1    278          RET                             ; return - R0/R1 restored from CHF$L_MCH_R0/
                                00D2    279
                                00D2    280
                                00D2    281
                                00D2    282          .END
```

```
DCOSHLEN                = 00000006
DCOSHTAB                  00000028 R      01
D_127_LOG_2               00000000 R      01
D_128_LOG_2               00000008 R      01
D_2_POWER_28.5            00000010 R      01
D_LOG_2_HI                00000018 R      01
D_LOG_2_LO                00000020 R      01
ERROR                     000000C2 R      01
GEQ_TO_0.25               00000085 R      01
GEQ_TO_2M28               0000007C R      01
GTR_THAN_127_LOG_2        000000A6 R      01
LONG                    = 00000004
MTH$$JACKET_HND           ********    X   01
MTH$$SIGNAL               ********    X   00
MTH$DCOSH                 00000058 RG     01
MTH$DEXP_R6               ********    X   00
MTH$K_FLOOVEMAT           ********    X   00
ONE_TERM_ONLY             000000A0 R      01
SD_1.0                  = 00004080
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	000000D2 (210.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.10	00:00:00.72
Command processing	119	00:00:00.66	00:00:04.63
Pass 1	92	00:00:00.90	00:00:04.19
Symbol table sort	0	00:00:00.01	00:00:00.00
Pass 2	65	00:00:00.70	00:00:02.47
Symbol table output	3	00:00:00.03	00:00:00.08
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	316	00:00:02.42	00:00:12.12

The working set limit was 750 pages.
4120 bytes (9 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 0 local symbols.
342 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$:MTHDCOSH/OBJ=OBJ$:MTHDCOSH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```

MTHDCOSH LIS

MTHDMIN LIS

MTHDLOG LIS

MTHDSINCO LIS

MTHDATANH LIS

MTHDNINT LIS

MTHDSQRT LIS

MTHDCONJG LIS

MTHDINT LIS

MTHDMAX1 LIS

MTHDSIGN LIS

MTHDIM LIS

MTHDMOD LIS

MTHDSINH LIS

MTHDEXP LIS

MTHDFLOOR LIS

MTHDPROD LIS