


```

MM      MM      TTTTTTTTTT  HH      HH      CCCCCCCC  DDDDDDDD  SSSSSSSS  QQQQQQ  RRRRRRRR  TTTTTTTTTT
MM      MM      TTTTTTTTTT  HH      HH      CCCCCCCC  DDDDDDDD  SSSSSSSS  QQQQQQ  RRRRRRRR  TTTTTTTTTT
MMMM    MMMM    TT          HH      HH      CC          DD          DD  SS          QQ          QQ  RR          RR  TT
MMMM    MMMM    TT          HH      HH      CC          DD          DD  SS          QQ          QQ  RR          RR  TT
MM      MM      TT          HH      HH      CC          DD          DD  SS          QQ          QQ  RR          RR  TT
MM      MM      TT          HH      HH      CC          DD          DD  SS          QQ          QQ  RR          RR  TT
MM      MM      TT          HHHHHHHHHH  CC          DD          DD  SSSSSS      QQ          QQ  RRRRRRRR  TT
MM      MM      TT          HHHHHHHHHH  CC          DD          DD  SSSSSS      QQ          QQ  RRRRRRRR  TT
MM      MM      TT          HH          HH  CC          DD          DD          SS          QQ      QQ  QQ  RR      RR  TT
MM      MM      TT          HH          HH  CC          DD          DD          SS          QQ      QQ  QQ  RR      RR  TT
MM      MM      TT          HH          HH  CC          DD          DD          SS          QQ      QQ  QQ  RR      RR  TT
MM      MM      TT          HH          HH  CC          DD          DD          SS          QQ      QQ  QQ  RR      RR  TT
MM      MM      TT          HH          HH  CC          DD          DD          SS          QQ      QQ  QQ  RR      RR  TT
MM      MM      TT          HH          HH  CCCCCCCC  DDDDDDDD  SSSSSSSS  QQQQ  QQ  RR      RR  TT
MM      MM      TT          HH          HH  CCCCCCCC  DDDDDDDD  SSSSSSSS  QQQQ  QQ  RR      RR  TT

```

```

...
...
...
...

```

```

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
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```

(2)	49	HISTORY	; Detailed Current Edit History
(3)	58	DECLARATIONS	
(4)	85	MTH\$CDSQRT -	compute D COMPLEX*16 square root

```

0000 1      .TITLE  MTH$CDSQRT
0000 2      .IDENT  /1-003/      ; File: MTHCDSQRT.MAR Edit: SBL1003
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :*  ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :*  TRANSFERRED.
0000 17 :*
0000 18 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :*  CORPORATION.
0000 21 :*
0000 22 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28 :
0000 29 : FACILITY: MATH LIBRARY
0000 30 :++
0000 31 : ABSTRACT:
0000 32 :   This module contains routine MTH$CDSQRT - compute
0000 33 :   D COMPLEX*16 square root.
0000 34 :
0000 35 :--
0000 36 :
0000 37 : VERSION: 1
0000 38 :
0000 39 : HISTORY:
0000 40 :
0000 41 : AUTHOR:
0000 42 :   Steven B. Lionel, 24-July-1979
0000 43 :
0000 44 : MODIFIED BY:
0000 45 :
0000 46 :
0000 47 :

```

MT
Sy
AR
MT
MT
MT
MT
MT
PS
--
M
Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
23
Th
20
1
Ma
--
\$
O
Th
MA

0000 49 .SBTTL HISTORY ; Detailed Current Edit History
0000 50
0000 51
0000 52 ; Edit History
0000 53 :
0000 54 : 1-001 - Adapted from MTH\$CSQRT version 1-003. SBL 24-July-1979
0000 55 : 1-002 - Change shared external reference to G^ RNH 25-Sep-81
0000 56 : 1-003 - Use general mode addressing. SBL 30-Nov-1981

DECLARATIONS

```
0000 58          .SBTTL  DECLARATIONS
0000 59
0000 60 :
0000 61 : INCLUDE FILES:
0000 62 :
0000 63 :
0000 64 :
0000 65 : EXTERNAL SYMBOLS:
0000 66          .DSABL  GBL
0000 67          .EXTRN  MTH$CDABS
0000 68          .EXTRN  MTH$DSQRT_R5
0000 69
0000 70 :
0000 71 : MACROS:
0000 72 :     NONE
0000 73 :
0000 74 :
0000 75 : PSECT DECLARATIONS:
0000 76          .PSECT  _MTH$CODE          PIC, SHR, LONG, EXE, NOWRT
0000 77
0000 78 :
0000 79 : EQUATED SYMBOLS:
0000 80 :
0000 81 :
0000 82 : OWN STORAGE:
0000 83 :     NONE
```

MTH\$CDSQRT - compute D COMPLEX*16 square root 6-SEP-1984 11:20:58 [MTHRTL.SRC]MTHCDSQRT.MAR;1

```

0000 85      .SBTTL MTH$CDSQRT - compute D COMPLEX*16 square root
0000 86
0000 87      :++
0000 88      : FUNCTIONAL DESCRIPTION:
0000 89
0000 90      :   The square root of a complex number (r, i) is computed
0000 91      :   as follows:
0000 92
0000 93      :   ROOT = SQRT((ABS(r) + CABS((r, i))) / 2)
0000 94      :   Q = i / (2*ROOT)
0000 95
0000 96
0000 97      :   r      i      CSQRT((r, i))
0000 98      :   -      -      -----
0000 99
0000 100     :   >=0    any    (ROOT, Q)
0000 101     :   <0     >=0  (Q, ROOT)
0000 102     :   <0     <0   (-Q, -ROOT)
0000 103
0000 104     : CALLING SEQUENCE:
0000 105
0000 106     :   CALL MTH$CDSQRT (result.wdc.r, arg.rdc.r)
0000 107
0000 108     : INPUT PARAMETERS:
0000 109
00000008 0000 110     :   arg      = 8      ; The D COMPLEX*16 argument, passed
0000 111     :                       ; by reference.
0000 112
0000 113     : IMPLICIT INPUTS:
0000 114     :   NONE
0000 115
0000 116     : OUTPUT PARAMETERS:
00000004 0000 117     :
0000 118     :   result = 4      ; The D COMPLEX*16 result, passed by
0000 119     :                       ; reference.
0000 120
0000 121     : IMPLICIT OUTPUTS:
0000 122     :   NONE
0000 123
0000 124     : COMPLETION CODES:
0000 125     :   NONE
0000 126
0000 127     : SIDE EFFECTS:
0000 128
0000 129     :   $$$_ROPRAND    If either part of argument is reserved operand.
0000 130
0000 131     :--
0000 132
0000 133
007C 0000 134     : .ENTRY MTH$CDSQRT,      ^M<R2, R3, R4, R5, R6>
0002 135     : MTH$FLAG_JACKET      ; flag as math routine
0002
6D 00000000'GF 9E 0002     : MOVAB G^MTH$$JACKET_HND, (FP)
0009     :                       ; set handler address to jacket
0009     :                       ; handler
0009
0009 136

```

```

52 08 BC 70 0009 137      MOVD    @arg(AP), R2      ; R2-R3 = r
52 8000 8F AA 000D 138      BICW    #^X8000, R2     ; R2-R3 = ABS(r)
08 AC DD 0012 139      PUSHL   arg(AP)        ; Put address of arg on stack
00000000'GF 01 FB 0015 140     CALLS   #1, G^MTH$CDABS ; R0-R1 = CABS((r, i))
50 52 60 001C 141      ADDD2   R2, R0         ; R0-R1 = ABS(r) + CABS((r, i))
50 00 64 001F 142      MULD2   #0.5, R0       ; R0-R1 = (ABS(r) + CABS((r, i))) / 2
00000000'GF 16 0022 143     JSB     G^MTH$DSQRT_R5 ; R0-R1 = ROOT = SQRT(above)
52 08 AC D0 0028 144      MOVL    arg(AP), R2     ; R2 -> (r, i)
50 73 002C 145      TSTD    R0            ; is ROOT zero?
04 12 002E 146      BNEQ    1$           ; no, go ahead
55 7C 0030 147      CLRD    R5            ; make zero quotient
08 11 0032 148      BRB     2$           ; skip divide
55 08 A2 50 67 0034 149 1$:  DIVD3   R0, 8(R2), R5 ; R5 = i / ROOT
55 00 64 0039 150      MULD2   #0.5, R5      ; R5 = Q = i / (2 * ROOT)
82 73 003C 151 2$:  TSTD    (R2)+         ; if r positive,
15 18 003E 152      BGEQ    RETRN        ; then return (ROOT, Q)
53 50 7D 0040 153      MOVQ    R0, R3        ; else switch ROOT and Q
62 73 0043 154      TSTD    (R2)         ; if i positive
08 18 0045 155      BGEQ    RETRN1       ; then return (Q, ROOT)
50 55 72 0047 156      MNEGD   R5, R0        ; else negate ROOT and Q
55 53 72 004A 157      MNEGD   R3, R5        ; and return (-Q, -ROOT)
06 11 004D 158      BRB     RETRN
004F 159
004F 160 RETRN1:
50 55 7D 004F 161      MOVQ    R5, R0        ; continue to swap ROOT and Q
55 53 7D 0052 162      MOVQ    R3, R5        ; and return (Q, ROOT)
0055 163 RETRN:
52 04 AC D0 0055 164      MOVL    result(AP), R2 ; result address
82 50 7D 0059 165      MOVQ    R0, (R2)+     ; real part
62 55 7D 005C 166      MOVQ    R5, (R2)     ; imaginary part
04 005F 167      RET
0060 168
0060 169
0060 170      .END

```


MTHCDSQRT
Symbol table

H 2

16-SEP-1984 01:07:44
6-SEP-1984 11:20:58

VAX/VMS Macro V04-00
[MTHRTL.SRC]MTHCDSQRT.MAR;1

Page 6
(4)

MTH
1-C

```

ARG = 00000008
MTH$$JACKET_HND ***** X 01
MTH$CDABS ***** X 00
MTHCDSQRT 00000000 RG 01
MTH$DSQRT_RS ***** X 00
RESULT = 00000004
RETRN 00000055 R 01
RETRN1 0000004F R 01

```

```

+-----+
! 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	00000060 (96.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

```

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

```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.39
Command processing	133	00:00:00.71	00:00:04.29
Pass 1	85	00:00:00.67	00:00:04.35
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	45	00:00:00.57	00:00:02.84
Symbol table output	2	00:00:00.02	00:00:00.03
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	299	00:00:02.08	00:00:11.92

The working set limit was 900 pages.
2885 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 8 non-local and 2 local symbols.
230 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\$:MTHCDSQRT/OBJ=OBJ\$:MTHCDSQRT MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MS

0258 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

A large grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window displays a different program or utility, with titles such as MTHCUTDG LIS, MTHDACS LIS, MTHCGABS LIS, MTHCDSINC LIS, MTHCLOG LIS, MTHDASIN LIS, MTHCGLOG LIS, MTHCONVER LIS, MTHDSORT LIS, MTHCGEXP LIS, MTHCGSORT LIS, MTHCINCO LIS, MTHCONJG LIS, MTHCDSORT LIS, MTHCGSINC LIS, and MTHCOSH LIS. Each window contains a mix of text, lists, and data tables.

MTHCUTDG LIS	MTHDACS LIS	MTHCGABS LIS	MTHCDSINC LIS	MTHCLOG LIS	MTHDASIN LIS	MTHCGLOG LIS	MTHCONVER LIS	MTHDSORT LIS	MTHCGEXP LIS
MTHCDSINC LIS	MTHCGABS LIS	MTHCLOG LIS	MTHDASIN LIS	MTHCGLOG LIS	MTHCONVER LIS	MTHDSORT LIS	MTHCGEXP LIS	MTHCGSORT LIS	MTHCINCO LIS
MTHCGEXP LIS	MTHCGSORT LIS	MTHCINCO LIS	MTHCONJG LIS	MTHCDSORT LIS	MTHCGSINC LIS	MTHCOSH LIS			