

| | | | | | |
|--------------|----------|--------------|-------------|----------------|------------------|
| CCCCCCCCCCCC | 00000000 | 888888888888 | RRRRRRRRRRR | TTTTTTTTTTTTTT | LLL |
| CCCCCCCCCCCC | 00000000 | 888888888888 | RRRRRRRRRRR | TTTTTTTTTTTTTT | LLL |
| CCCCCCCCCCCC | 00000000 | 888888888888 | RRRRRRRRRRR | TTTTTTTTTTTTTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCC | 000 | 888 | RRR | TTT | LLL |
| CCCCCCCCCCCC | 00000000 | 888888888888 | RRR | TTT | LLLLLLLLLLLLLLLL |
| CCCCCCCCCCCC | 00000000 | 888888888888 | RRR | TTT | LLLLLLLLLLLLLLLL |
| CCCCCCCCCCCC | 00000000 | 888888888888 | RRR | TTT | LLLLLLLLLLLLLLLL |

```
CCCCCCCC 000000 BBBB8888 CCCCCCCC VV VV TTTTTT;TTT QQQQQQ DDDDDDDD
CCCCCCCC 000000 000000 BBBB8888 CCCCCCCC VV VV TTTTTTTTTT QQQQQQ DDDDDDDD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BBBB8888  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BBBB8888  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CC          00      00  BB      BB  CC          VV VV VV      TT      QQ      DD      DD
CCCCCCCC 000000 BBBB8888 CCCCCCCC          VV VV      TT      QQ      DD      DD
CCCCCCCC 000000 BBBB8888 CCCCCCCC          VV VV      TT      QQ      DD      DD
```

```
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
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
```

| | |
|-----|-----|
| (2) | 49 |
| (3) | 60 |
| (4) | 102 |

| | |
|--------------|---------------------------------|
| HISTORY | : Detailed Current Edit History |
| DECLARATIONS | |
| COBSCVTQD_R8 | |

```
0000 1 .TITLE COBSCVTQD_R8 COBOL Convert Quad to Double
0000 2 .IDENT /1-004/ ; File: COBSCVTQD.MAR
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 FACILITY: COBOL SUPPORT
0000 29 ++
0000 30 ABSTRACT:
0000 31 This module contains the routine that converts quadword numbers
0000 32 to double floating.
0000 33
0000 34
0000 35 --
0000 36
0000 37 VERSION: 1
0000 38
0000 39 HISTORY:
0000 40
0000 41 AUTHOR:
0000 42 Marty Jack, 14-Mar-1979
0000 43
0000 44 MODIFIED BY:
0000 45
0000 46
0000 47
```

```
0000 49 .SBTTL HISTORY ; Detailed Current Edit History
0000 50
0000 51
0000 52 ; Edit History for Version 1 of COBSCVTQD
0000 53 :
0000 54 : 1-001 - Original. MLJ 14-Mar-1979
0000 55 : 1-002 - Make external references explicit. RKR 17-JULY-1979
0000 56 : 1-003 - Change all references to FOR$CNV_IN_DEFG to OTSSCVT_T_D
0000 57 : RKR 27-SEPT-79
0000 58 : 1-004 - Cosmetic changes. RKR 18-OCT-79
```

```

0000 60      .SBTTL  DECLARATIONS
0000 61
0000 62 :
0000 63 : INCLUDE FILES:
0000 64 :
0000 65      $DSCDEF
0000 66
0000 67 :
0000 68 : EXTERNAL SYMBOLS:
0000 69
0000 70      .DSABL  GBL           ; Prevent undeclared symbols from being
0000 71                                ; automatically global
0000 72
0000 73      .EXTRN  OTSSCVT_T_D   ; D, E, F, G Conversion Routine
0000 74
0000 75 :
0000 76
0000 77 :
0000 78 : MACROS:
0000 79 :     NONE
0000 80 :
0000 81 :
0000 82 :
0000 83 : PSECT DECLARATIONS:
0000 84 :     .PSECT  _COB$CODE       PIC, SHR, LONG, EXE, NOWRT
0000 85
0000 86 :
0000 87 : EQUATED SYMBOLS:
0000 88 :     NONE
0000 89 :
0000 90 :
0000 91 :
0000 92 : OWN STORAGE:
0000 93 :
0000 94 :+
0000 95 : The following constant has the value 2**32. It is used for scaling
0000 96 : the high 32 bits and for compensating for unsigned arithmetic.
0000 97 :-
6C 29 67 49 29 04 0000 98 BIAS:   .PACKED 4294967296   ; 2**32
0000 99 BIAS_DIGITS=10
0006
0006 100 :

```

```

0006 102      .SBTTL COB$CVTQD_R8
0006 103
0006 104      :++
0006 105      : FUNCTIONAL DESCRIPTION:
0006 106      :
0006 107      :     Converts 64-bit (quadword) numbers to double floating.
0006 108      :
0006 109      : CALLING SEQUENCE:
0006 110      :
0006 111      :     JSB COB$CVTQD_R8 (scale.rl.v, src.rq.r, dst.wd.r)
0006 112      :
0006 113      :     Arguments are passed in R6, R7, and R8.
0006 114      :
0006 115      : INPUT PARAMETERS:
0006 116      :
0006 117      :     SCALE.rl.v           The power of ten by which the internal
0006 118      :                               representation of the source must be
0006 119      :                               multiplied to scale the same as the
0006 120      :                               internal representation of the dest.
0006 121      :     SRC.rq.r           The number to be converted
0006 122      :
0006 123      : IMPLICIT INPUTS:
0006 124      :
0006 125      :     ALL of the trap bits in the PSL are assumed off.
0006 126      :
0006 127      : OUTPUT PARAMETERS:
0006 128      :
0006 129      :     DST.wd.r           The place to store the converted number
0006 130      :
0006 131      : IMPLICIT OUTPUTS:
0006 132      :
0006 133      :     NONE
0006 134      :
0006 135      : FUNCTION VALUE:
0006 136      :
0006 137      :     1 = SUCCESS, 0 = FAILURE
0006 138      :
0006 139      : SIDE EFFECTS:
0006 140      :
0006 141      :     Destroys registers R0 through R8.
0006 142      :
0006 143      :--
0006 144
0006 145
0006 146 COB$CVTQD_R8::
0006 147     SOBL2 #40,SP           ; Space for temp string and result
0009 148 :
0009 149 : Convert the quadword input to packed.
0009 150 :
0009 151     CMPV #31,#1,(R7),4(R7) ; Is number in longword range?
000F 152     BNEQ 11$             ; Br if not to slower code
0011 153     CVTLP (R7),#19,8(SP) ; Convert low order longword
0016 154     BRB 13$              ; To common code
0018 155 11$: CVTLP 4(R7),#10,(SP) ; Convert high order longword
13  6E 0A DF AF 0A 25 001D 156     MULP #BIAS_DIGITS,BIAS,#10,(SP),#19,8(SP)
0024
0026 157                               ; Multiply by 2**32

```

```

        6E  0A  67  F9  0026  158      CVTLP  (R7),#10,(SP)      ; Convert low order longword
        06  18  002A  159      BGEQ   12$           ; Br if nonnegative
        6E  0A  00  AF  0A  20  002C  160      ADDP4  #BIAS_DIGITS,BIAS,#10,(SP)
        0032  161
        08  AE  13  6E  0A  20  0032  162  12$:  ADDP4  #10,(SP),#19,8(SP)  ; Correct for signed conversion
        0038  163           ; Sum low and high order parts
        0038  164           ;
        0038  165           ; Convert the packed intermediate to leading separate.
        14  AE  13  08  AE  13  08  0038  166  13$:  CVTPS  #19,8(SP),#19,20(SP) ; Make a separate sign string
        003F  167           ;
        003F  168           ; Make a descriptor for the leading separate string.
        003F  169           ;
        003F  170           ;
        7E  01  90  0041  171      PUSHL  R3           ; Address = temp string
        7E  0E  90  0044  172      MOVB   #DSC$K_CLASS_S,-(SP) ; Class = static
        7E  14  80  0047  173      MOVB   #DSC$K_DTYPE_T,-(SP) ; Data type = ASCII text
        004A  174           ; Length = 20 bytes
        004A  175           ;
        004A  176           ; Now call the conversion routine.
        7E  56  CE  004A  177      MNEGL  R6,-(SP)     ; Scale factor
        00  DD  004D  178      PUSHL  #0           ; Digits in fraction
        10  AE  9F  004F  179      PUSHAB 16(SP)      ; Address of result area
        0C  AE  9F  0052  180      PUSHAB 12(SP)      ; Address of descriptor
        00000000'GF 04  FB  0055  181      CALLS  #4,G^OTSSCVT_T_D ; Call the routine
        08  50  E9  005C  182      BLBC   R0,15$      ; Failure, must be overflow
        68  08  AE  70  005F  183      MOVD   8(SP),(R8)  ; Store result
        50  01  D0  0063  184      MOVL   #1,R0       ; Indicate success
        5E  30  C0  0066  185  14$:  ADDL2  #48,SP       ; Delete stack temps
        05  0069  186      RSB           ; Return
        006A  187           ;
        006A  188           ; Come here on overflow to store the reserved operand.
        006A  189           ;
        68  01  0F  79  006A  190  15$:  ASHQ   #15,#1,(R8) ; Store reserved operand
        50  D4  006E  191      CLRL   R0           ; Indicate failure
        F4  11  0070  192      BRB   14$         ; Delete stack temps and return
        0072  193           ;
        0072  194           ;
        .END

```


COB\$CVTQD R8
Symbol table

COBOL Convert Quad to Double

H 13

15-SEP-1984 23:39:56
6-SEP-1984 10:43:26

VAX/VMS Macro V04-00
[COBRTL.SRC]COB\$CVTQD.MAR;1

Page 6
(4)

BIAS
BIAS DIGITS
COB\$CVTQD R8
DSC\$K_CLASS_S
DSC\$K_DTYPE_T
OT\$SCVT_T_D

00000000 R 02
= 0000000A
00000006 RG 02
= 00000001
= 0000000E
***** X 00

! Psect synopsis !

| PSECT name | Allocation | PSECT No. | Attributes |
|------------|------------------|-----------|---|
| . ABS . | 00000000 (0.) | 00 (0.) | NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE |
| \$AB\$\$ | 00000000 (0.) | 01 (1.) | NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE |
| _COB\$CODE | 00000072 (114.) | 02 (2.) | PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG |

! Performance indicators !

| Phase | Page faults | CPU Time | Elapsed Time |
|------------------------|-------------|-------------|--------------|
| Initialization | 29 | 00:00:00.07 | 00:00:01.13 |
| Command processing | 123 | 00:00:00.30 | 00:00:02.27 |
| Pass 1 | 140 | 00:00:01.23 | 00:00:05.34 |
| Symbol table sort | 0 | 00:00:00.09 | 00:00:00.93 |
| Pass 2 | 48 | 00:00:00.36 | 00:00:02.29 |
| Symbol table output | 3 | 00:00:00.01 | 00:00:00.02 |
| Psect synopsis output | 2 | 00:00:00.01 | 00:00:00.01 |
| Cross-reference output | 0 | 00:00:00.00 | 00:00:00.00 |
| Assembler run totals | 347 | 00:00:02.09 | 00:00:12.00 |

The working set limit was 900 pages.
8720 bytes (18 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 135 non-local and 5 local symbols.
194 source lines were read in Pass 1, producing 10 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

! Macro library statistics !

| Macro library name | Macros defined |
|-------------------------------------|----------------|
| _\$255\$DUA28:[SYSLIB]STARLET.MLB;2 | 4 |

190 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:COB\$CVTQD/OBJ=OBJ\$:COB\$CVTQD MSRC\$:COB\$CVTQD/UPDATE=(ENH\$:COB\$CVTQD)

