


```

000000  TTTTTTTTTT  SSSSSSSS  CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR  HH      HH  PPPPPPPP
000000  TTTTTTTTTT  SSSSSSSS  CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR  HH      HH  PPPPPPPP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
00      00    TT      SS      CC      VV      VV  TT      RR      RR  HH      HH  PP      PP
000000  TT      SSSSSSSS  CCCCCCCC  VV      VV  TT      RR      RR  HH      HH  PP      PP
000000  TT      SSSSSSSS  CCCCCCCC  VV      VV  TT      RR      RR  HH      HH  PP      PP

```

```

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

```

```

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

```

(2) 49
(3) 63
(4) 101

HISTORY ; Detailed Current Edit History
DECLARATIONS
OTSSCVTRHP_R9

```
0000 1 .TITLE OTSSCVTRHP_R9 Convert Rounded H Floating to Packed
0000 2 .IDENT /1-006/ ; File: OTSCVTRHP.MAR Edit: MDL1006
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: LANGUAGE INDEPENDENT SUPPORT
0000 29 :++
0000 30 : ABSTRACT:
0000 31 : This module contains the routine that converts H floating
0000 32 : numbers to packed with rounding.
0000 33 :
0000 34 :
0000 35 : --
0000 36 :
0000 37 : VERSION: 1
0000 38 :
0000 39 : HISTORY:
0000 40 :
0000 41 : AUTHOR:
0000 42 : Pamela Levesque, 21-Jan-1982
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 OTSCVTRHP
0000 53 :
0000 54 : 1-001 - Original. PLL 21-Jan-1982
0000 55 : 1-002 - Call h floating conversion routine. PLL 2-Mar-1982
0000 56 : 1-003 - Expect a 2 digit exponent from the conversion routine. PLL 8-Mar-1982
0000 57 : 1-004 - determine the number of digits in the exponent on the fly.
0000 58 : MDL 9-Apr-1984
0000 59 : 1-005 - Make sure result will not overflow destination after calculating
0000 60 : longword exponent. DG 9-Apr-1984
0000 61 : 1-006 - complete edit 1-004. MDL 12-Apr-1984
```

```
0000 63      .SBTTL  DECLARATIONS
0000 64
0000 65 :
0000 66 : INCLUDE FILES:
0000 67 :
0000 68      $DSCDEF
0000 69
0000 70 :
0000 71 : EXTERNAL SYMBOLS:
0000 72
0000 73      .DSABL  GBL           ; Prevent undeclared symbols from being
0000 74                                     ; automatically global
0000 75
0000 76      .EXTRN  OTSSCNVOUT_H ; E-format conversion
0000 77      .EXTRN  LIB$STOP    ; Error halt
0000 78      .EXTRN  OTSS_FATINTERR ; Fatal internal error code
0000 79 :
0000 80
0000 81 :
0000 82 : MACROS:
0000 83 :     NONE
0000 84 :
0000 85
0000 86 :
0000 87 : PSECT DECLARATIONS:
0000 88      .PSECT  _OTSSCODE    PIC, SHR, LONG, EXE, NOWRT,-
0000 89                                     USR, CON, REL, LCL, RD
0000 90
0000 91 :
0000 92 : EQUATED SYMBOLS:
0000 93 :     NONE
0000 94 :
0000 95
0000 96 :
0000 97 : OWN STORAGE:
0000 98 :     NONE
0000 99 :
```

```

0000 101      .SBTTL  OTSSCVTRHP_R9
0000 102
0000 103      :++
0000 104      : FUNCTIONAL DESCRIPTION:
0000 105      :
0000 106      :     Converts an h floating number to packed with rounding.
0000 107      :
0000 108      : CALLING SEQUENCE:
0000 109      :
0000 110      :     JSB OTSSCVTRHP_R9 (scale.rl.v, src.rh.r, dstlen.rl.v, dst.wp.r)
0000 111      :
0000 112      :     Arguments are passed in R6, R7, R8 and R9.
0000 113      :
0000 114      : INPUT PARAMETERS:
0000 115      :
0000 116      :     SCALE.rl.v           The power of ten by which the internal
0000 117      :                               representation of the source must be
0000 118      :                               multiplied to scale the same as the
0000 119      :                               internal representation of the dest.
0000 120      :     SRC.rh.r             The number to be converted
0000 121      :     DSTLEN.rl.v         The number of digits in the destination
0000 122      :
0000 123      : IMPLICIT INPUTS:
0000 124      :
0000 125      :     All of the trap bits in the PSL are assumed off.
0000 126      :
0000 127      : OUTPUT PARAMETERS:
0000 128      :
0000 129      :     DST.wp.r             The place to store the converted number
0000 130      :
0000 131      : IMPLICIT OUTPUTS:
0000 132      :
0000 133      :     NONE
0000 134      :
0000 135      : FUNCTION VALUE:
0000 136      :
0000 137      :     1 = SUCCESS, 0 = FAILURE
0000 138      :
0000 139      : SIDE EFFECTS:
0000 140      :
0000 141      :     Destroys registers R0 through R9.
0000 142      :
0000 143      :--
0000 144
0000 145
0000 146 OTSSCVTRHP R9::
0000 147     SUBL2  #56,SP           ; Allocate temp space
6E   38   00   5E   38   C2 0003 148     MOVCS  #0,(SP),#^A'',#56,(SP) ; Clear temp space
6E   67   70FD 0009 149     MOVH   (R7),(SP)           ; Get input number
0000 150
0000 151 : Make a descriptor for the temporary string.
0000 152 :
0000 153     PUSHAB 16(SP)           ; Address = space reserved
7E   10   AE   9F 0010 154     MOVB  #DSC$K_CLASS_S,-(SP) ; Class = static
7E   0E   90 0013 155     MOVB  #DSC$K_DTYPE_T,-(SP) ; Data type = ASCII string
7E   26   B0 0016 156     MOVW  #38,-(SP)           ; Length in bytes
0019 157 :

```

```

0019 158 : Call OTSSCNVOUT.
0019 159 :
      1F DD 0019 160 : PUSHL #31 : Digits in fraction
04 AE 9F 001B 161 : PUSHAB 4(SP) : Output string descriptor
10 AE 9F 001E 162 : PUSHAB 16(SP) : Number to convert
00000000'GF 03 FB 0021 163 : CALLS #3,G^OTSSCNVOUT_H : Call conversion routine
63 50 E9 0028 164 : BLBC R0,20$ : Should never fail
      002B 165 :
      002B 166 : Determine the number of digits in the exponent.
      002B 167 :
      002B 168 :
1C AE 22 2B 3A 002B 169 : LOCC #^A'+',#34,28(SP) : find beginning of exponent (sign byte)
      07 12 0030 170 : BNEQ 5$ : look for sign byte (+)
1C AE 22 2D 3A 0032 171 : LOCC #^A'-',#34,28(SP) : branch if we found it
      55 13 0037 172 : BEQL 20$ : look for sign byte (-)
      0039 173 : : branch to error if we didn't find it
      08 AE 51 D0 0039 174 5$: MOVL R1,8(SP) : save location of sign byte
      003D 175 : : (contained in R1 after above LOCC)
      003D 176 :
08 BE 05 00 3A 003D 177 : LOCC #^A'',#5,@B^8(SP) : look for NUL (end of string)
OC AE 51 08 AE C3 0042 178 : SUBL3 8(SP),R1,12(SP) : determine number of digits by subtracting
      0048 179 : : location of beginning of exponent from
      0048 180 : : location of end of exponent, and save it
      0C AE D7 0048 181 : DECL 12(SP) : adjust for sign byte
      004B 182 : : at this point, 12(SP) = length of exponent
      004B 183 :
      004B 184 : Convert the exponent and correct for scale factor.
      004B 185 :
6E OC AE 08 BE OC AE 09 004B 186 : CVTSP 12(SP),@B^8(SP),12(SP),(SP) : Make packed exponent
      50 6E OC AE 36 0053 187 : CVTPL 12(SP),(SP),R0 : Make longword exponent
      0058 188 :
      0058 189 : Here must check that the result will not overflow
      0058 190 : : the destination. The destination length (total number of
      0058 191 : : digits) minus the scale factor will give the number of digits
      0058 192 : : to the left of the decimal point. If that number is greater than
      0058 193 : : the longword exponent, then the result WILL overflow the dest.
      0058 194 :
      54 58 56 C3 0058 195 : SUBL3 R6,R8,R4 : Determine num of digits on left
      54 50 D1 005C 196 : CMPL R0,R4 : Compare with longword exponent
      23 14 005F 197 : BGTR 15$ : Branch if overflowed
      0061 198 :
      56 E2 A640 9E 0061 199 : MOVAB -30(R6)[R0],R6 : Correct for fraction size and scale
      0066 200 :
      0066 201 : Convert the fraction to packed.
      0066 202 :
      0066 203 :
69 6E 1A AE 18 AE 90 0066 203 : MOVAB 24(SP),26(SP) : Move sign over ""
      58 05 1E 1A AE 1E 09 006B 204 : CVTSP #30,26(SP),#30,(SP) : Make packed fraction
      06E 1E 56 F8 0071 205 : ASHP R6,#30,(SP),#5,R8,(R9) : Scale to destination
      0078 206 : : (also clears R0)
      02 1D 0078 207 : BVS 10$ : Branch if overflowed
      50 D6 007A 208 : INCL R0 : Indicate success, R0 = 1
      5E 00000040 8F C0 007C 209 10$: ADDL2 #64,SP : Delete stack temps
      0083 210 :
      0084 211 :
      5E 00000040 50 D4 0084 212 15$: CLRL R0 : Indicate failure, R0 = 0
      8F C0 0086 213 : ADDL2 #64,SP : Delete stack temps
      008D 214 : RSB

```


00000000'8F
00000000'GF 01

DD
FB

008E 215 :
008E 216 :
008E 217 :
008E 218 :
0094 219 :
009B 220 :
009B 221 :

: Come here on failure of OTSS\$CNVOUT. This should never happen.
20\$: PUSHL #OTSS\$FATINTERR ; OTS fatal error message
CALLS #1,G^CIB\$STOP ; Signal and don't return
.END

OTSSCVTRHP R9
Symbol table

Convert Rounded H Floating to Packed^{C 6}

16-SEP-1984 00:28:00
6-SEP-1984 11:13:30

VAX/VMS Macro V04-00
[LIBRTL.SRC]OTSCVTRHP.MAR;1

Page 7
(4)

DSCSK_CLASS_S	=	00000001		
DSCSK_DTYPE_T	=	0000000E		
LIB\$STOP		*****	X	00
OTSSCNVOUT_H		*****	X	00
OTSSCVTRHP_R9		00000000	RG	02
OTSS_FATINTERR		*****	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					
_OTSSCODE	00000098 (155.)	02 (2.)	PIC USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG					

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.03	00:00:01.01
Command processing	129	00:00:00.31	00:00:01.82
Pass 1	132	00:00:01.24	00:00:05.98
Symbol table sort	0	00:00:00.11	00:00:00.71
Pass 2	57	00:00:00.36	00:00:01.97
Symbol table output	2	00:00:00.01	00:00:00.20
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	357	00:00:02.08	00:00:11.71

The working set limit was 1050 pages.
9083 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 4 local symbols.
221 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\$:OTSCVTRHP/OBJ=OBJ\$:OTSCVTRHP MSRC\$:OTSCVTRHP/UPDATE=(ENH\$:OTSCVTRHP)

