



```

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

```

```

LL      IIIIIII  SSSSSSSS
LL      IIIIIII  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  IIIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIIII  SSSSSSSS

```

(2)	49
(3)	66
(4)	104

HISTORY	; Detailed Current Edit History
DECLARATIONS	
OTSSCVTHP_R9	

```

0000 1      .TITLE  OTSSCVTHP_R9      Convert H Floating to Packed
0000 2      .IDENT  /1-006/          ; File: OTSCVTMP.MAR Edit: DG1006
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.
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, 18-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 OTSCVTMP
0000 53 :
0000 54 : 1-001 - Original. PLL 18-Jan-1982
0000 55 : 1-002 - Call the 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 how many digits there are in the exponent on the fly.
0000 58 : Correct allocation of space for temporary descriptor so that
0000 59 : it doesn't point into the middle of the H-floating number.
0000 60 : MDL 13-Sep-1982
0000 61 : 1-005 - correct minor scaling bug in previous edit. MDL 7-Oct-1982
0000 62 : 1-006 - Make sure result will not overflow destination after calculating
0000 63 : longword exponent. DG 3-Apr-1984
0000 64 :
```

```
0000 66      .SBTTL  DECLARATIONS
0000 67
0000 68 :
0000 69 : INCLUDE FILES:
0000 70 :
0000 71      $DSCDEF
0000 72
0000 73 :
0000 74 : EXTERNAL SYMBOLS:
0000 75      .DSABL  GBL          ; Prevent undeclared symbols from
0000 76                                ; being automatically global
0000 77
0000 78      .EXTRN  OTSS$CNVOUT_H ; E-format conversion
0000 79      .EXTRN  LIB$STOP      ; error halt
0000 80      .EXTRN  OTSS$_FATINTERR ; Fatal internal error code
0000 81
0000 82 :
0000 83 :
0000 84 :
0000 85 : MACROS:
0000 86 :     NONE
0000 87 :
0000 88 :
0000 89 :
0000 90 : PSECT DECLARATIONS:
0000 91      .PSECT  _OTSS$CODE    PIC, SHR, LONG, EXE, NOWRT,-
0000 92                                USR, CON, REL, LCL, RD
0000 93
0000 94 :
0000 95 : EQUATED SYMBOLS:
0000 96 :     NONE
0000 97 :
0000 98 :
0000 99 :
0000 100 : OWN STORAGE:
0000 101 :     NONE
0000 102 :
```

```

0000 104      .SBTTL OTSS$CVTHP_R9
0000 105
0000 106 :++
0000 107 : FUNCTIONAL DESCRIPTION:
0000 108 :
0000 109 :     Converts an h floating number to packed.
0000 110 :
0000 111 : CALLING SEQUENCE:
0000 112 :
0000 113 :     JSB OTSS$CVTHP_R9 (scale.rl.v, src.rh.r, dstlen.rl.v, dst.wp.r)
0000 114 :
0000 115 :     Arguments are passed in R6, R7, R8 and R9.
0000 116 :
0000 117 : INPUT PARAMETERS:
0000 118 :
0000 119 :     SCALE.rl.v           The power of ten by which the internal
0000 120 :                          representation of the source must be
0000 121 :                          multiplied to scale the same as the
0000 122 :                          internal representation of the dest.
0000 123 :     SRC.rh.r             The number to be converted
0000 124 :     DSTLEN.rl.v         The number of digits in the destination
0000 125 :
0000 126 : IMPLICIT INPUTS:
0000 127 :
0000 128 :     All of the trap bits in the PSL are assumed off.
0000 129 :
0000 130 : OUTPUT PARAMETERS:
0000 131 :
0000 132 :     DST.wp.r             The place to store the converted number
0000 133 :
0000 134 : IMPLICIT OUTPUTS:
0000 135 :
0000 136 :     NONE
0000 137 :
0000 138 : FUNCTION VALUE:
0000 139 :
0000 140 :     1 = SUCCESS, 0 = FAILURE
0000 141 :
0000 142 : SIDE EFFECTS:
0000 143 :
0000 144 :     Destroys registers R0 through R9.
0000 145 :
0000 146 : --
0000 147 :
0000 148 :
0000 149 : OTSS$CVTHP_R9::
0000 150 :     SOBL2 #56,SP           ; Allocate temp space
0000 151 :     MOVCS #0,(SP),#^A'',#56,(SP) ; Clear temp space
0000 152 :     MOVH (R7),(SP)        ; Get input number
0000 153 :
0000 154 :     ; Make a descriptor for the temporary string.
0000 155 :
0000 156 :     PUSHAB 16(SP)         ; Address = space reserved
0000 157 :     MOVB #DSC$K_CLASS_S,-(SP) ; Class = static
0000 158 :     MOVB #DSC$K_DTYPE_T,-(SP) ; Data type = ASCII string
0000 159 :     MOVW #38,-(SP)        ; Length in bytes
0000 160 :

```

```

6E 38 00 5E 38 C2
6E 00 2C 0003 151
6E 67 70FD 0009 152
10 AE 9F 000D 153
7E 01 90 0010 157
7E 0E 90 0013 158
7E 26 80 0016 159
0019 160

```

```

0019 161 ; Call OTSSCNVOUT.
0019 162 ;
04 1F DD 0019 163 ; PUSHL #31 ; Digits in fraction
10 AE 9F 001B 164 ; PUSHAB 4(SP) ; Output string descriptor
00000000'GF 03 FB 001E 165 ; PUSHAB 16(SP) ; Number to convert
63 50 E9 0021 166 ; CALLS #3,G^OTSSCNVOUT_H ; Call conversion routine
002B 167 ; BLBC R0,20$ ; Should never fail
002B 168 ;
002B 169 ; Determine the number of digits in the exponent.
002B 170 ;
002B 171 ;
1C AE 22 2B 3A 002B 172 ; LOCC #^A'+',#34,28(SP) ; find beginning of exponent (sign byte)
07 12 0030 173 ; BNEQ 5$ ; look for sign byte (+)
1C AE 22 2D 3A 0032 174 ; LOCC #^A'-',#34,28(SP) ; branch if we found it
55 13 0037 175 ; BEQL 20$ ; look for sign byte (-)
0039 176 ; ; branch to error if we didn't find it
08 AE 51 D0 0039 177 5$: ; MOVL R1,8(SP) ; save location of sign byte
003D 178 ; ; (contained in R1 after above LOCC)
003D 179 ;
08 BE 05 00 3A 003D 180 ; LOCC #^A'',#5,@B^8(SP) ; look for NUL (end of string)
OC AE 51 08 AE C3 0042 181 ; SUBL3 8(SP),R1,12(SP) ; determine number of digits by subtracting
0048 182 ; ; location of beginning of exponent from
0048 183 ; ; location of end of exponent, and save it
OC AE D7 0048 184 ; DECL 12(SP) ; adjust for sign byte
004B 185 ; ; at this point, 12(SP) = length of exponent
004B 186 ;
004B 187 ; Convert the exponent and correct for scale factor.
004B 188 ;
6E OC AE 08 BE OC AE 09 004B 189 ; CVTSP 12(SP),@B^8(SP),12(SP),(SP) ; Make packed exponent
50 6E OC AE 36 0053 190 ; CVTPL 12(SP),(SP),R0 ; Make longword exponent
0058 191 ;
0058 192 ; Here must check that the result will not overflow
0058 193 ; the destination. The destination length (total number of
0058 194 ; digits) minus the scale factor will give the number of digits
0058 195 ; to the left of the decimal point. If that number is greater than
0058 196 ; the longword exponent, then the result WILL overflow the dest.
0058 197 ;
54 58 56 C3 0058 198 ; SUBL3 R6,R8,R4 ; Determine num of digits on left
54 50 D1 005C 199 ; CMPL R0,R4 ; Compare with longword exponent
23 14 005F 200 ; BGTR 15$ ; Branch if overflowed
56 E2 A640 9E 0061 201 ;
0061 202 ; MOVAB -30(R6)[R0],R6 ; Correct for fraction size and scale
0066 203 ;
0066 204 ; Convert the fraction to packed.
0066 205 ;
69 6E 1E 1A AE 18 AE 90 0066 206 ; MOVB 24(SP),26(SP) ; Move sign over ""
58 00 6E 1A AE 1E 56 F8 006B 207 ; CVTSP #30,26(SP),#30,(SP) ; Make packed fraction
0071 208 ; ASHP R6,#30,(SP),#0,R8,(R9) ; Scale to destination
0078 209 ; ; (also clears R0)
02 1D 0078 210 ; BVS 10$ ; Branch if overflowed
50 D6 007A 211 ; INCL R0 ; Indicate success, R0 = 1
SE 00000040 8F C0 007C 212 10$: ; ADDL2 #64,SP ; Delete stack temps
0083 213 ; RSB
0084 214 ;
SE 00000040 50 D4 0084 215 15$: ; CLRL R0 ; Indicate failure, R0 = 0
8F C0 0086 216 ; ADDL2 #64,SP ; Delete stack temps
008D 217 ; RSB

```



```
00000000'8F DD 008E 218 ;  
00000000'GF 01 FB 008E 219 ; Come here on failure of OTSSCNVOUT. This should never happen.  
008E 220 ;  
008E 221 20$: PUSHL #OTSS_FATINTERR ; OTS fatal error message  
009B 222 CALLS #1,G^CIB$STOP ; Signal and don't return  
009B 223 ;  
009B 224 .END
```

```
DSC&K_CLASS_S      = 00000001
DSC&K_DTYPE_T      = 0000000E
LIB&STOP           ***** X 00
OTSSCNVOUT H       ***** X 00
OTSSCVTHP 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
\$ABSS	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	29	00:00:00.04	00:00:01.88
Command processing	117	00:00:00.33	00:00:02.39
Pass 1	135	00:00:01.18	00:00:05.76
Symbol table sort	0	00:00:00.10	00:00:00.59
Pass 2	57	00:00:00.42	00:00:02.53
Symbol table output	3	00:00:00.02	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	345	00:00:02.10	00:00:13.19

The working set limit was 1050 pages.  
 9106 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.  
 224 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\$:OTSCVTMP/OBJ=OBJ\$:OTSCVTMP MSRCS:OTSCVTMP/UPDATE=(ENHS:OTSCVTMP)

