```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     2 ************************************************************************
                                                     3 *
                                                     4 *Testcase IEEE CONVERT FROM FIXED 32
                                                     5 *  Test case capability includes IEEE exceptions trappable and
                                                     6 *  otherwise.  Test result, FPC flags, and DXC saved for all tests.
                                                     7 *  Convert From Fixed does not set the condition code.
                                                     8 *
                                                     9 *
                                                    10 *                        ********************
                                                    11 *                        **   IMPORTANT!   **
                                                    12 *                        ********************
                                                    13 *
                                                    14 *        This test uses the Hercules Diagnose X'008' interface
                                                    15 *        to display messages and thus your .tst runtest script
                                                    16 *        MUST contain a "DIAG8CMD ENABLE" statement within it!
                                                    17 *
                                                    18 *
                                                    19 ************************************************************************


                                                    21 ************************************************************************
                                                    22 *
                                                    23 *                          bfp-010-cvtfrfix.asm
                                                    24 *
                                                    25 *        This assembly-language source file is part of the
                                                    26 *        Hercules Binary Floating Point Validation Package
                                                    27 *                          by Stephen R. Orso
                                                    28 *
                                                    29 * Copyright 2016 by Stephen R Orso.
                                                    30 * Runtest *Compare dependency removed by Fish on 2022-08-16
                                                    31 * PADCSECT macro/usage removed by Fish on 2022-08-16
                                                    32 *
                                                    33 * Redistribution and use in source and binary forms, with or without
                                                    34 * modification, are permitted provided that the following conditions
                                                    35 * are met:
                                                    36 *
                                                    37 * 1. Redistributions of source code must retain the above copyright
                                                    38 *    notice, this list of conditions and the following disclaimer.
                                                    39 *
                                                    40 * 2. Redistributions in binary form must reproduce the above copyright
                                                    41 *    notice, this list of conditions and the following disclaimer in
                                                    42 *    the documentation and/or other materials provided  with the
                                                    43 *    distribution.
                                                    44 *
                                                    45 * 3. The name of the author may not be used to endorse or promote
                                                    46 *    products derived from this software without specific prior written
                                                    47 *    permission.
                                                    48 *
                                                    49 * DISCLAMER: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER "AS IS"
                                                    50 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
                                                    51 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
                                                    52 * PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT
                                                    53 * HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
                                                    54 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
                                                    55 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
                                                    56 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
```

```
 LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                      57 * OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
                                                      58 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
                                                      59 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
                                                      60 *
                                                      61 ********************************************************************************



                                                      63 ********************************************************************************
                                                      64 *
                                                      65 * Tests the following six conversion instructions
                                                      66 *   CONVERT FROM FIXED (32 to short BFP, RRE)
                                                      67 *   CONVERT FROM FIXED (32 to long BFP, RRE)
                                                      68 *   CONVERT FROM FIXED (32 to extended BFP, RRE)
                                                      69 *   CONVERT FROM FIXED (32 to short BFP, RRF-e)
                                                      70 *   CONVERT FROM FIXED (32 to long BFP, RRF-e)
                                                      71 *   CONVERT FROM FIXED (32 to extended BFP, RRF-e)
                                                      72 *
                                                      73 * Test data is compiled into this program.  The test script that runs
                                                      74 * this program can provide alternative test data through Hercules R
                                                      75 * commands.
                                                      76 *
                                                      77 * Test Case Order
                                                      78 * 1) Int-32 to Short BFP
                                                      79 * 2) Int-32 to Short BFP with all rounding modes
                                                      80 * 3) Int-32 to Long BFP
                                                      81 * 4) Int-32 to Extended BFP
                                                      82 *
                                                      83 * Provided test data is 1, 2, 4, -2, 2 147 483 647, -2 147 483 647.
                                                      84 *   The last two values will trigger inexact exceptions when converted
                                                      85 *   to short BFP.  The last two values are also used to test rounding
                                                      86 *   mode and inexact supression in the CEFBRA instruction.
                                                      87 *
                                                      88 * Also tests the following floating point support instructions
                                                      89 *   LOAD  (Short)
                                                      90 *   LOAD  (Long)
                                                      91 *   LOAD FPC
                                                      92 *   SET BFP ROUNDING MODE 2-BIT
                                                      93 *   SET BFP ROUNDING MODE 3-BIT
                                                      94 *   STORE (Short)
                                                      95 *   STORE (Long)
                                                      96 *   STORE FPC
                                                      97 *
                                                      98 ********************************************************************************
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                      100 *
                                                      101 *  Note: for compatibility with the z/CMS test rig, do not change
                                                      102 *  or use R11, R14, or R15.  Everything else is fair game.
                                                      103 *



                              00000000  0000515B    105 BFPCVTFF START 0
                              00000000  00000001    106 STRTLABL EQU   *
                              00000000  00000001    107 R0       EQU   0                  Work register for cc extraction
                              00000001  00000001    108 R1       EQU   1
                              00000002  00000001    109 R2       EQU   2                  Holds count of test input values
                              00000003  00000001    110 R3       EQU   3                  Points to next test input value(s)
                              00000004  00000001    111 R4       EQU   4                  Available
                              00000005  00000001    112 R5       EQU   5                  Available
                              00000006  00000001    113 R6       EQU   6                  Available
                              00000007  00000001    114 R7       EQU   7                  Pointer to next result value(s)
                              00000008  00000001    115 R8       EQU   8                  Pointer to next FPCR result
                              00000009  00000001    116 R9       EQU   9                  Rounding tests top of outer loop
                              0000000A  00000001    117 R10      EQU   10                 Pointer to test address list
                              0000000B  00000001    118 R11      EQU   11                 **Reserved for z/CMS test rig
                              0000000C  00000001    119 R12      EQU   12                 Holds number of test cases in set
                              0000000D  00000001    120 R13      EQU   13                 Mainline return address
                              0000000E  00000001    121 R14      EQU   14                 **Return address for z/CMS test rig
                              0000000F  00000001    122 R15      EQU   15                 **Base register on z/CMS or Hyperion
                                                    123 *
                                                    124 * Floating Point Register equates to keep the cross reference clean
                                                    125 *
                              00000000  00000001    126 FPR0     EQU   0
                              00000001  00000001    127 FPR1     EQU   1
                              00000002  00000001    128 FPR2     EQU   2
                              00000003  00000001    129 FPR3     EQU   3
                              00000004  00000001    130 FPR4     EQU   4
                              00000005  00000001    131 FPR5     EQU   5
                              00000006  00000001    132 FPR6     EQU   6
                              00000007  00000001    133 FPR7     EQU   7
                              00000008  00000001    134 FPR8     EQU   8
                              00000009  00000001    135 FPR9     EQU   9
                              0000000A  00000001    136 FPR10    EQU   10
                              0000000B  00000001    137 FPR11    EQU   11
                              0000000C  00000001    138 FPR12    EQU   12
                              0000000D  00000001    139 FPR13    EQU   13
                              0000000E  00000001    140 FPR14    EQU   14
                              0000000F  00000001    141 FPR15    EQU   15
                                                    142 *
00000000                      00000000              143          USING *,R15
00000000                      00004DC0              144          USING HELPERS,R12
                                                    145 *
                                                    146 * Above works on real iron (R15=0 after sysclear)
                                                    147 * and in z/CMS (R15 points to start of load module)
                                                    148 *



                                                    150 ************************************************************************
                                                    151 *
                                                    152 * Low core definitions, Restart PSW, and Program Check Routine.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    153 *
                                                    154 ********************************************************************************


00000000                      00000000  0000008E    156          ORG   STRTLABL+X'8E'      Program check interrution code
0000008E  0000                                    157 PCINTCD  DS    H
                                                    158 *
                              00000150  00000001    159 PCOLDPSW EQU   STRTLABL+X'150'     z/Arch Program check old PSW
                                                    160 *
00000090                      00000090  000001A0    161          ORG   STRTLABL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                         162          DC    X'0000000180000000',AD(START)
                                                    163 *
000001B0                      000001B0  000001D0    164          ORG   STRTLABL+X'1D0'     z/Arch Program check NEW PSW
000001D0  00000000 00000000                         165          DC    X'0000000000000000',AD(PROGCHK)
                                                    166 *
                                                    167 * Program check routine.  If Data Exception, continue execution at
                                                    168 * the instruction following the program check.  Otherwise, hard wait.
                                                    169 * No need to collect data.  All interesting DXC stuff is captured
                                                    170 * in the FPCR.
                                                    171 *
000001E0                      000001E0  00000200    172          ORG   STRTLABL+X'200'
00000200                                            173 PROGCHK  DS    0H              Program check occured...
00000200  9507 F08F                     0000008F    174          CLI   PCINTCD+1,X'07'  Data Exception?
00000204  A774 0004                     0000020C    175          JNE   PCNOTDTA       ..no, hardwait (not sure if R15 is ok)
00000208  B2B2 F150                     00000150    176          LPSWE PCOLDPSW       ..yes, resume program execution

0000020C  900F F23C                     0000023C    178 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                     0000027C    179          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                     00004DC0    180          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                     0000023C    181          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                      183          LTR   R14,R14        Return address provided?
0000021E  077E                                      184          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                     00000228    185          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                         186 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2D0                     000002D0    187 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                         188 SAVEREGS DC    16F'0'          Registers save area
0000027C  00004DC0                                  189 AHELPERS DC    A(HELPERS)      Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    191 ************************************************************************
                                                    192 *
                                                    193 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                    194 *
                                                    195 ************************************************************************


00000280  B600 F2E0                        000002E0   197 START    STCTL R0,R0,CTLR0    Store CR0 to enable AFP
00000284  9604 F2E1                        000002E1   198          OI    CTLR0+1,X'04'  Turn on AFP bit
00000288  B700 F2E0                        000002E0   199          LCTL  R0,R0,CTLR0    Reload updated CR0
                                                    200 *
0000028C  41A0 F2EC                        000002EC   201          LA    R10,SHORTS     Point to integer test inputs
00000290  4DD0 F32C                        0000032C   202          BAS   R13,CEFBR      Convert values from fixed to short BFP
                                                    203 *
00000294  41A0 F31C                        0000031C   204          LA    R10,RMSHORTS   Point to inputs for rounding mode tests
00000298  4DD0 F36E                        0000036E   205          BAS   R13,CEFBRA     Convert using all rounding mode options
                                                    206 *
0000029C  41A0 F2FC                        000002FC   207          LA    R10,LONGS      Point to integer test inputs
000002A0  4DD0 F440                        00000440   208          BAS   R13,CDFBR      Convert values from fixed to long BFP
                                                    209 *
000002A4  41A0 F30C                        0000030C   210          LA    R10,EXTDS      Point to integer test inputs
000002A8  4DD0 F482                        00000482   211          BAS   R13,CXFBR      Convert values from fixed to extended
                                                    212 *
                                                    213 ************************************************************************
                                                    214 *                    Verify test results...
                                                    215 ************************************************************************
                                                    216 *
000002AC  58C0 F27C                        0000027C   217          L     R12,AHELPERS   Get address of helper subroutines
000002B0  4DD0 C0A0                        00004E60   218          BAS   R13,VERISUB    Go verify results
000002B4  12EE                                     219          LTR   R14,R14        Was return address provided?
000002B6  077E                                     220          BNZR  R14            Yes, return to z/CMS test rig.
000002B8  B2B2 F2C0                        000002C0   221          LPSWE GOODPSW        Load SUCCESS PSW
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000002C0                                           223          DS    0D            Ensure correct alignment for PSW
000002C0  00020000 00000000                        224 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002D0  00020000 00000000                        225 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                   226 *
000002E0  00000000                                 227 CTLR0    DS    F
000002E4  00000000                                 228 FPCREGNT DC    X'00000000'    FPC Reg IEEE exceptions Not Trappable
000002E8  F8000000                                 229 FPCREGTR DC    X'F8000000'    FPC Reg IEEE exceptions TRappable
                                                   230 *
                                                   231 * Input values parameter list, four fullwords:
                                                   232 *      1) Count,
                                                   233 *      2) Address of inputs,
                                                   234 *      3) Address to place results, and
                                                   235 *      4) Address to place DXC/Flags/cc values.
                                                   236 *
000002EC                                           237 SHORTS   DS    0F
000002EC  00000007                                 238          DC    A(INTCOUNT)
000002F0  000004CC                                 239          DC    A(INTIN)
000002F4  00001000                                 240          DC    A(SBFPOUT)
000002F8  00001100                                 241          DC    A(SBFPFLGS)
                                                   242 *
000002FC                                           243 LONGS    DS    0F           int-32 inputs for long BFP testing
000002FC  00000007                                 244          DC    A(INTCOUNT)
00000300  000004CC                                 245          DC    A(INTIN)
00000304  00002000                                 246          DC    A(LBFPOUT)
00000308  00002100                                 247          DC    A(LBFPFLGS)
                                                   248 *
0000030C                                           249 EXTDS    DS    0F           int-32 inputs for Extended BFP testing
0000030C  00000007                                 250          DC    A(INTCOUNT)
00000310  000004CC                                 251          DC    A(INTIN)
00000314  00003000                                 252          DC    A(XBFPOUT)
00000318  00003200                                 253          DC    A(XBFPFLGS)
                                                   254 *
0000031C  00000003                                 255 RMSHORTS DC    A(INTRMCT)
00000320  000004E8                                 256          DC    A(INTINRM)   Last two int-32 are only concerns
00000324  00001200                                 257          DC    A(SBFPRMO)   Space for rounding mode tests
00000328  00001500                                 258          DC    A(SBFPRMOF)  Space for rounding mode test flags
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    260 ********************************************************************************
                                                    261 *
                                                    262 * Convert int-32 to short BFP format.  A pair of results is generated
                                                    263 * for each input: one with all exceptions non-trappable, and the second
                                                    264 * with all exceptions trappable.   The FPCR is stored for each result.
                                                    265 *
                                                    266 ********************************************************************************


0000032C  9823 A000                 00000000    268 CEFBR    LM    R2,R3,0(R10)  Get count and address of test input values
00000330  9878 A008                 00000008    269          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000334  1222                                  270          LTR   R2,R2         Any test cases?
00000336  078D                                  271          BZR   R13           ..No, return to caller
00000338  0DC0                                  272          BASR  R12,0         Set top of loop
                                                    273 *
0000033A  5810 3000                 00000000    274          L     R1,0(,R3)     Get integer test value
0000033E  B29D F2E4                 000002E4    275          LFPC  FPCREGNT      Set exceptions non-trappable
00000342  B394 0081                             276          CEFBR FPR8,R1       Cvt Int in GPR1 to float in FPR8
00000346  7080 7000                 00000000    277          STE   FPR8,0(,R7)   Store short BFP result
0000034A  B29C 8000                 00000000    278          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                    279 *
0000034E  B29D F2E8                 000002E8    280          LFPC  FPCREGTR      Set exceptions trappable
00000352  B394 0081                             281          CEFBR FPR8,R1       Cvt Int in GPR1 to float in FPR8
00000356  7080 7004                 00000004    282          STE   FPR8,4(,R7)   Store short BFP result
0000035A  B29C 8004                 00000004    283          STFPC 4(R8)         Store resulting FPC flags and DXC
0000035E  4130 3004                 00000004    284          LA    R3,4(,R3)     Foint to next input values
00000362  4170 7008                 00000008    285          LA    R7,8(,R7)     Point to next short BFP converted values
00000366  4180 8008                 00000008    286          LA    R8,8(,R8)     Point to next FPCR/CC result area
0000036A  062C                                  287          BCTR  R2,R12        Convert next input value.
0000036C  07FD                                  288          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    290 ************************************************************************
                                                    291 *
                                                    292 * Convert int-32 to short BFP format using each possible rounding mode.
                                                    293 * Ten test results are generated for each input.  A 48-byte test result
                                                    294 * section is used to keep results sets aligned on a quad-double word.
                                                    295 *
                                                    296 * The first four tests use rounding modes specified in the FPC with the
                                                    297 * IEEE Inexact exception supressed.  SRNM (2-bit) is used  for the
                                                    298 * first two FPCR-controlled tests and SRNMB (3-bit) is used for the
                                                    299 * last two to get full coverage of that instruction pair.
                                                    300 *
                                                    301 * The next six results use instruction-specified rounding modes.
                                                    302 *
                                                    303 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    304 * prior tests used the default rounding mode.  RNTE is tested
                                                    305 * explicitly as a rounding mode in this section.
                                                    306 *
                                                    307 ************************************************************************


0000036E  9823 A000                 00000000    309 CEFBRA   LM    R2,R3,0(R10)  Get count and address of test input values
00000372  9878 A008                 00000008    310          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000376  1222                                  311          LTR   R2,R2         Any test cases?
00000378  078D                                  312          BZR   R13           ..No, return to caller
0000037A  0DC0                                  313          BASR  R12,0         Set top of loop
                                                314 *
0000037C  5810 3000                 00000000    315          L     R1,0(,R3)     Get integer test value
                                                316 *
                                                317 * Test cases using rounding mode specified in the FPCR
                                                318 *
00000380  B29D F2E4                 000002E4    319          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000384  B299 0001                 00000001    320          SRNM  1             SET FPCR to RZ, towards zero
00000388  B394 0481                             321          CEFBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
0000038C  7080 7000                 00000000    322          STE   FPR8,0*4(,R7) Store short BFP result
00000390  B29C 8000                 00000000    323          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                324 *
00000394  B29D F2E4                 000002E4    325          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000398  B299 0002                 00000002    326          SRNM  2             SET FPCR to RP, to +infinity
0000039C  B394 0481                             327          CEFBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003A0  7080 7004                 00000004    328          STE   FPR8,1*4(,R7) Store short BFP result
000003A4  B29C 8004                 00000004    329          STFPC 1*4(R8)       Store resulting FPC flags and DXC
                                                330 *
000003A8  B29D F2E4                 000002E4    331          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003AC  B2B8 0003                 00000003    332          SRNMB 3             SET FPCR to RM, to -infinity
000003B0  B394 0481                             333          CEFBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003B4  7080 7008                 00000008    334          STE   FPR8,2*4(,R7) Store short BFP result
000003B8  B29C 8008                 00000008    335          STFPC 2*4(R8)       Store resulting FPC flags and DXC
                                                336 *
000003BC  B29D F2E4                 000002E4    337          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003C0  B2B8 0007                 00000007    338          SRNMB 7             RFS, Prepare For Shorter Precision
000003C4  B394 0481                             339          CEFBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003C8  7080 700C                 0000000C    340          STE   FPR8,3*4(,R7) Store short BFP result
000003CC  B29C 800C                 0000000C    341          STFPC 3*4(R8)       Store resulting FPC flags and DXC
                                                342 *
000003D0  B29D F2E4                 000002E4    343          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003D4  B394 1081                             344          CEFBRA FPR8,1,R1,B'0000'  RNTA, to nearest, ties away
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000003D8  7080 7010                    00000010   345          STE   FPR8,4*4(,R7) Store short BFP result
000003DC  B29C 8010                    00000010   346          STFPC 4*4(R8)       Store resulting FPC flags and DXC
                                                  347 *
000003E0  B29D F2E4                    000002E4   348          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003E4  B394 3081                               349          CEFBRA FPR8,3,R1,B'0000'  RPS, prepare for shorter precision
000003E8  7080 7014                    00000014   350          STE   FPR8,5*4(,R7) Store short BFP result
000003EC  B29C 8014                    00000014   351          STFPC 5*4(R8)       Store resulting FPC flags and DXC
                                                  352 *
000003F0  B29D F2E4                    000002E4   353          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003F4  B394 4081                               354          CEFBRA FPR8,4,R1,B'0000'  RNTE to nearest, ties to even
000003F8  7080 7018                    00000018   355          STE   FPR8,6*4(,R7) Store short BFP result
000003FC  B29C 8018                    00000018   356          STFPC 6*4(R8)       Store resulting FPC flags and DXC
                                                  357 *
00000400  B29D F2E4                    000002E4   358          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000404  B394 5081                               359          CEFBRA FPR8,5,R1,B'0000'  RZ, toward zero
00000408  7080 701C                    0000001C   360          STE   FPR8,7*4(,R7) Store short BFP result
0000040C  B29C 801C                    0000001C   361          STFPC 7*4(R8)       Store resulting FPC flags and DXC
                                                  362 *
00000410  B29D F2E4                    000002E4   363          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000414  B394 6081                               364          CEFBRA FPR8,6,R1,B'0000'  RP, to +inf
00000418  7080 7020                    00000020   365          STE   FPR8,8*4(,R7) Store short BFP result
0000041C  B29C 8020                    00000020   366          STFPC 8*4(R8)       Store resulting FPC flags and DXC
                                                  367 *
00000420  B29D F2E4                    000002E4   368          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000424  B394 7081                               369          CEFBRA FPR8,7,R1,B'0000'  RM, to -inf
00000428  7080 7024                    00000024   370          STE   FPR8,9*4(,R7) Store short BFP result
0000042C  B29C 8024                    00000024   371          STFPC 9*4(R8)       Store resulting FPC flags and DXC
                                                  372 *
00000430  4130 3004                    00000004   373          LA    R3,4(,R3)     Point to next input values
00000434  4170 7030                    00000030   374          LA    R7,12*4(,R7)  Point to next short BFP converted values
00000438  4180 8030                    00000030   375          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
0000043C  062C                                    376          BCTR  R2,R12        Convert next input value.
0000043E  07FD                                    377          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    379 ********************************************************************************
                                                    380 *
                                                    381 * Convert int-32 to long BFP format.  A pair of results is generated
                                                    382 * for each input: one with all exceptions non-trappable, and the second
                                                    383 * with all exceptions trappable.   The FPCR is stored for each result.
                                                    384 * Conversion of a 32-bit integer to long is always exact; no exceptions
                                                    385 * are expected
                                                    386 *
                                                    387 ********************************************************************************

00000440  9823 A000                 00000000    389 CDFBR    LM    R2,R3,0(R10)  Get count and address of test input values
00000444  9878 A008                 00000008    390          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000448  1222                                  391          LTR   R2,R2         Any test cases?
0000044A  078D                                  392          BZR   R13           ..No, return to caller
0000044C  0DC0                                  393          BASR  R12,0         Set top of loop
                                                    394 *
0000044E  5810 3000                 00000000    395          L     R1,0(,R3)     Get integer test value
00000452  B29D F2E4                 000002E4    396          LFPC  FPCREGNT      Set exceptions non-trappable
00000456  B395 0081                             397          CDFBR FPR8,R1       Cvt Int in GPR1 to float in FPR8
0000045A  6080 7000                 00000000    398          STD   FPR8,0(,R7)   Store long BFP result
0000045E  B29C 8000                 00000000    399          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                    400 *
00000462  B29D F2E8                 000002E8    401          LFPC  FPCREGTR      Set exceptions trappable
00000466  B395 0081                             402          CDFBR FPR8,R1       Cvt Int in GPR1 to float in FPR8
0000046A  6080 7008                 00000008    403          STD   FPR8,8(,R7)   Store long BFP result
0000046E  B29C 8004                 00000004    404          STFPC 4(R8)         Store resulting FPC flags and DXC
                                                    405 *
00000472  4130 3004                 00000004    406          LA    R3,4(,R3)     Point to next input values
00000476  4170 7010                 00000010    407          LA    R7,16(,R7)    Point to next long BFP converted value
0000047A  4180 8008                 00000008    408          LA    R8,8(,R8)     Point to next FPCR/CC result area
0000047E  062C                                  409          BCTR  R2,R12        Convert next input value.
00000480  07FD                                  410          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    412 ********************************************************************************
                                                    413 *
                                                    414 * Convert int-32 to extended BFP format.  A pair of results is
                                                    415 * generated for each input: one with all exceptions non-trappable,
                                                    416 * and the second with all exceptions trappable.   The FPCR is
                                                    417 * stored for each result.  Conversion of a 32-bit integer to
                                                    418 * extended is always exact; no exceptions are expected
                                                    419 *
                                                    420 ********************************************************************************


00000482  9823 A000                 00000000   422 CXFBR    LM    R2,R3,0(R10)  Get count and address of test input values
00000486  9878 A008                 00000008   423          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000048A  1222                                 424          LTR   R2,R2         Any test cases?
0000048C  078D                                 425          BZR   R13           ..No, return to caller
0000048E  0DC0                                 426          BASR  R12,0         Set top of loop
                                               427 *
00000490  5810 3000                 00000000   428          L     R1,0(,R3)     Get integer test value
00000494  B29D F2E4                 000002E4   429          LFPC  FPCREGNT      Set exceptions non-trappable
00000498  B396 0081                            430          CXFBR FPR8,R1       Cvt Int in GPR1 to float in FPR8-FPR10
0000049C  6080 7000                 00000000   431          STD   FPR8,0(,R7)   Store extended BFP result part 1
000004A0  60A0 7008                 00000008   432          STD   FPR10,8(,R7)  Store extended BFP result part 2
000004A4  B29C 8000                 00000000   433          STFPC 0(R8)         Store resulting FPC flags and DXC
                                               434 *
000004A8  B29D F2E8                 000002E8   435          LFPC  FPCREGTR      Set exceptions trappable
000004AC  B396 0081                            436          CXFBR FPR8,R1       Cvt Int in GPR1 to float in FPR8-FPR10
000004B0  6080 7010                 00000010   437          STD   FPR8,16(,R7)  Store extended BFP result part 1
000004B4  60A0 7018                 00000018   438          STD   FPR10,24(,R7) Store extended BFP result part 2
000004B8  B29C 8004                 00000004   439          STFPC 4(R8)         Store resulting FPC flags and DXC
                                               440 *
000004BC  4130 3004                 00000004   441          LA    R3,4(,R3)     Point to next input values
000004C0  4170 7020                 00000020   442          LA    R7,32(,R7)    Point to next extended BFP converted value
000004C4  4180 8008                 00000008   443          LA    R8,8(,R8)     Point to next FPCR/CC result area
000004C8  062C                                 444          BCTR  R2,R12        Convert next input value.
000004CA  07FD                                 445          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    447 ********************************************************************************
                                                    448 *
                                                    449 * Short integer inputs for Convert From Fixed testing.  The same set of
                                                    450 * inputs are used for short, long, and extended formats.  The last two
                                                    451 * values are used for rounding mode tests for short only; conversion of
                                                    452 * int-32 to long or extended are always exact.
                                                    453 *
                                                    454 ********************************************************************************


000004CC                                            456 INTIN    DS    0F
000004CC  00000001                                  457          DC    F'1'
000004D0  00000002                                  458          DC    F'2'
000004D4  00000004                                  459          DC    F'4'
000004D8  FFFFFFFE                                  460          DC    F'-2'
000004DC  7FFFFFFF                                  461          DC    F'2147483647'  should compile to X'7FFFFFFF' - inexact
000004E0  80000001                                  462          DC    F'-2147483647' should compile to X'80000001' - inexact
000004E4  7FFFFF80                                  463          DC    XL4'7FFFFF80'  Fits in short BFP
                          00000007  00000001        464 INTCOUNT EQU   (*-INTIN)/4    Count of integers in list
                                                    465 *
                                                    466 * Short BFP Exhaustive rounding mode tests.  int-32 always fits in
                                                    467 * long BFP or extended BFP with no loss of precision, so no basis for
                                                    468 * exhaustive rounding tests for long or extended
                                                    469 *
000004E8                                            470 INTINRM  DS    0F
000004E8  7FFFFFE0                                  471          DC    XL4'7FFFFFE0'  Inexact, normally rounds up
000004EC  7FFFFFC0                                  472          DC    XL4'7FFFFFC0'  Inexact, Tie
000004F0  7FFFFFA0                                  473          DC    XL4'7FFFFFA0'  Inexact, normally rounds down
                          00000003  00000001        474 INTRMCT  EQU   (*-INTINRM)/4  Count of rounding mode test inputs
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                      476 ********************************************************************************
                                                      477 *                    ACTUAL results saved here
                                                      478 ********************************************************************************
                                                      479 *
                                                      480 *                 Locations for ACTUAL results
                                                      481 *
                                                      482 *
                              00001000  00000001      483 SBFPOUT  EQU   STRTLABL+X'1000'    Short BFP results from Int-32 inputs
                                                      484 *                                 ..6 pairs used, room for 32 pairs
                              00001100  00000001      485 SBFPFLGS EQU   STRTLABL+X'1100'    FPCR flags and DXC from short BFP
                                                      486 *                                 ..6 pairs used, room for 32 pairs
                              00001200  00000001      487 SBFPRMO  EQU   STRTLABL+X'1200'    Short BFP rounding mode results
                                                      488 *                                 ..2 sets used, room for 16
                              00001500  00000001      489 SBFPRMOF EQU   STRTLABL+X'1500'    Short BFP rndg mode FPCR contents
                                                      490 *                                 ..2 sets used, room for +16
                                                      491 *                                 ..next set at X'1800'
                                                      492 *
                              00002000  00000001      493 LBFPOUT  EQU   STRTLABL+X'2000'    Long BFP results from Int-32 inputs
                                                      494 *                                 ..6 pairs used, room for 16 pairs
                              00002100  00000001      495 LBFPFLGS EQU   STRTLABL+X'2100'    Long BFP FPCR contents
                                                      496 *                                 ..6 pairs used, room for 32+ pairs
                                                      497 *                                 ..next pair at X'2200'
                                                      498 *
                              00003000  00000001      499 XBFPOUT  EQU   STRTLABL+X'3000'    Extended BFP results from Int-32
                                                      500 *                                 ..6 pairs used, room for 16 pairs
                              00003200  00000001      501 XBFPFLGS EQU   STRTLABL+X'3200'    Extended BFP FPCR contents
                                                      502 *                                 ..6 pairs used, room for 16 pairs
                                                      503 *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    505 ********************************************************************************
                                                    506 *                       EXPECTED results
                                                    507 ********************************************************************************
                                                    508 *
000004F4                      000004F4  00004000    509          ORG   STRTLABL+X'4000'   (past end of actual results)
                                                    510 *
                              00004000  00000001    511 SBFPOUT_GOOD EQU *
00004000  C3C5C6C2 D9409985                         512  DC CL48'CEFBR result pairs 1-2'
00004030  3F800000 3F800000                         513  DC XL16'3F8000003F800004000000040000000'
00004040  C3C5C6C2 D9409985                         514  DC CL48'CEFBR result pairs 3-4'
00004070  40800000 40800000                         515  DC XL16'4080000040800000C0000000C0000000'
00004080  C3C5C6C2 D9409985                         516  DC CL48'CEFBR result pairs 5-6'
000040B0  4F000000 4F000000                         517  DC XL16'4F0000004F000000CF000000CF000000'
000040C0  C3C5C6C2 D9409985                         518  DC CL48'CEFBR result pair 7'
000040F0  4EFFFFFF 4EFFFFFF                         519  DC XL16'4EFFFFFF4EFFFFFF0000000000000000'
                              00000004  00000001    520 SBFPOUT_NUM EQU (*-SBFPOUT_GOOD)/64
                                                    521 *
                                                    522 *
                              00004100  00000001    523 SBFPFLGS_GOOD EQU *
00004100  C3C5C6C2 D940C6D7                         524  DC CL48'CEFBR FPC pairs 1-2'
00004130  00000000 F8000000                         525  DC XL16'00000000F8000000000000000F8000000'
00004140  C3C5C6C2 D940C6D7                         526  DC CL48'CEFBR FPC pairs 3-4'
00004170  00000000 F8000000                         527  DC XL16'00000000F8000000000000000F8000000'
00004180  C3C5C6C2 D940C6D7                         528  DC CL48'CEFBR FPC pairs 5-6'
000041B0  00080000 F8000C00                         529  DC XL16'00080000F8000C0000080000F8000C00'
000041C0  C3C5C6C2 D940C6D7                         530  DC CL48'CEFBR FPC pair 7'
000041F0  00000000 F8000000                         531  DC XL16'00000000F8000000000000000000000'
                              00000004  00000001    532 SBFPFLGS_NUM EQU (*-SBFPFLGS_GOOD)/64
                                                    533 *
                                                    534 *
                              00004200  00000001    535 SBFPRMO_GOOD EQU *
00004200  C3C5C6C2 D9C140D9                         536  DC CL48'CEFBRA RU FPC modes 1-3, 7'
00004230  4EFFFFFF 4F000000                         537  DC XL16'4EFFFFFF4F0000004EFFFFF4EFFFFFF'
00004240  C3C5C6C2 D9C140D9                         538  DC CL48'CEFBRA RU M3 modes 1, 3-5'
00004270  4F000000 4EFFFFFF                         539  DC XL16'4F0000004EFFFFF4F0000004EFFFFFF'
00004280  C3C5C6C2 D9C140D9                         540  DC CL48'CEFBRA RU M3 modes 6, 7'
000042B0  4F000000 4EFFFFFF                         541  DC XL16'4F0000004EFFFFF0000000000000000'
000042C0  C3C5C6C2 D9C140E3                         542  DC CL48'CEFBRA Tie FPC modes 1-3, 7'
000042F0  4EFFFFFF 4F000000                         543  DC XL16'4EFFFFFF4F0000004EFFFFF4EFFFFFF'
00004300  C3C5C6C2 D9C140E3                         544  DC CL48'CEFBRA Tie M3 modes 1, 3-5'
00004330  4F000000 4EFFFFFF                         545  DC XL16'4F0000004EFFFFF4F0000004EFFFFFF'
00004340  C3C5C6C2 D9C140E3                         546  DC CL48'CEFBRA Tie M3 modes 6, 7'
00004370  4F000000 4EFFFFFF                         547  DC XL16'4F0000004EFFFFF0000000000000000'
00004380  C3C5C6C2 D9C140D9                         548  DC CL48'CEFBRA RD FPC modes 1-3, 7'
000043B0  4EFFFFFF 4F000000                         549  DC XL16'4EFFFFFF4F0000004EFFFFF4EFFFFFF'
000043C0  C3C5C6C2 D9C140D9                         550  DC CL48'CEFBRA RD M3 modes 1, 3-5'
000043F0  4EFFFFFF 4EFFFFFF                         551  DC XL16'4EFFFFFF4EFFFFFF4EFFFFF4EFFFFFF'
00004400  C3C5C6C2 D9C140D9                         552  DC CL48'CEFBRA RD M3 modes 6, 7'
00004430  4F000000 4EFFFFFF                         553  DC XL16'4F0000004EFFFFF0000000000000000'
                              00000009  00000001    554 SBFPRMO_NUM EQU (*-SBFPRMO_GOOD)/64
                                                    555 *
                                                    556 *
                              00004440  00000001    557 SBFPRMOF_GOOD EQU *
00004440  C3C5C6C2 D9C140D9                         558  DC CL48'CEFBRA RU FPC modes 1-3, 7 FCPR'
00004470  00000001 00000002                         559  DC XL16'00000001000000020000000300000007'
00004480  C3C5C6C2 D9C140D9                         560  DC CL48'CEFBRA RU M3 modes 1, 3-5 FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000044B0  00080000 00080000                          561  DC XL16'00080000000800000008000000080000'
000044C0  C3C5C6C2 D9C140D9                          562  DC CL48'CEFBRA RU M3 modes 6, 7 FPCR'
000044F0  00080000 00080000                          563  DC XL16'00080000000800000000000000000000'
00004500  C3C5C6C2 D9C140E3                          564  DC CL48'CEFBRA Tie FPC modes 1-3, 7 FPCR'
00004530  00000001 00000002                          565  DC XL16'00000001000000020000000300000007'
00004540  C3C5C6C2 D9C140E3                          566  DC CL48'CEFBRA Tie M3 modes 1, 3-5 FPCR'
00004570  00080000 00080000                          567  DC XL16'00080000000800000008000000080000'
00004580  C3C5C6C2 D9C140E3                          568  DC CL48'CEFBRA Tie M3 modes 6, 7 FPCR'
000045B0  00080000 00080000                          569  DC XL16'00080000000800000000000000000000'
000045C0  C3C5C6C2 D9C140D9                          570  DC CL48'CEFBRA RD FPC modes 1-3, 7 FPCR'
000045F0  00000001 00000002                          571  DC XL16'00000001000000020000000300000007'
00004600  C3C5C6C2 D9C140D9                          572  DC CL48'CEFBRA RD M3 modes 1, 3-5 FPCR'
00004630  00080000 00080000                          573  DC XL16'00080000000800000008000000080000'
00004640  C3C5C6C2 D9C140D9                          574  DC CL48'CEFBRA RD M3 modes 6, 7 FPCR'
00004670  00080000 00080000                          575  DC XL16'00080000000800000000000000000000'
                             00000009  00000001    576 SBFPRMOF_NUM EQU (*-SBFPRMOF_GOOD)/64
                                                   577 *
                                                   578 *
                             00004680  00000001    579 LBFPOUT_GOOD EQU *
00004680  C3C4C6C2 D9409985                          580  DC CL48'CDFBR result pair 1'
000046B0  3FF00000 00000000                          581  DC XL16'3FF00000000000003FF0000000000000'
000046C0  C3C4C6C2 D9409985                          582  DC CL48'CDFBR result pair 2'
000046F0  40000000 00000000                          583  DC XL16'40000000000000004000000000000000'
00004700  C3C4C6C2 D9409985                          584  DC CL48'CDFBR result pair 3'
00004730  40100000 00000000                          585  DC XL16'40100000000000004010000000000000'
00004740  C3C4C6C2 D9409985                          586  DC CL48'CDFBR result pair 4'
00004770  C0000000 00000000                          587  DC XL16'C000000000000000C000000000000000'
00004780  C3C4C6C2 D9409985                          588  DC CL48'CDFBR result pair 5'
000047B0  41DFFFFF FFC00000                          589  DC XL16'41DFFFFFFFC0000041DFFFFFFFC00000'
000047C0  C3C4C6C2 D9409985                          590  DC CL48'CDFBR result pair 6'
000047F0  C1DFFFFF FFC00000                          591  DC XL16'C1DFFFFFFFC00000C1DFFFFFFFC00000'
00004800  C3C4C6C2 D9409985                          592  DC CL48'CDFBR result pair 7'
00004830  41DFFFFF E0000000                          593  DC XL16'41DFFFFFE000000041DFFFFFE0000000'
                             00000007  00000001    594 LBFPOUT_NUM EQU (*-LBFPOUT_GOOD)/64
                                                   595 *
                                                   596 *
                             00004840  00000001    597 LBFPFLGS_GOOD EQU *
00004840  C3C4C6C2 D940C6D7                          598  DC CL48'CDFBR FPC pairs 1-2'
00004870  00000000 F8000000                          599  DC XL16'00000000F800000000000000F8000000'
00004880  C3C4C6C2 D940C6D7                          600  DC CL48'CDFBR FPC pairs 3-4'
000048B0  00000000 F8000000                          601  DC XL16'00000000F800000000000000F8000000'
000048C0  C3C4C6C2 D940C6D7                          602  DC CL48'CDFBR FPC pairs 5-6'
000048F0  00000000 F8000000                          603  DC XL16'00000000F800000000000000F8000000'
00004900  C3C4C6C2 D940C6D7                          604  DC CL48'CDFBR FPC pair 7'
00004930  00000000 F8000000                          605  DC XL16'00000000F800000000000000000000'
                             00000004  00000001    606 LBFPFLGS_NUM EQU (*-LBFPFLGS_GOOD)/64
                                                   607 *
                                                   608 *
                             00004940  00000001    609 XBFPOUT_GOOD EQU *
00004940  C3E7C6C2 D9409985                          610  DC CL48'CXFBR result 1a'
00004970  3FFF0000 00000000                          611  DC XL16'3FFF0000000000000000000000000000'
00004980  C3E7C6C2 D9409985                          612  DC CL48'CXFBR result 1b'
000049B0  3FFF0000 00000000                          613  DC XL16'3FFF0000000000000000000000000000'
000049C0  C3E7C6C2 D9409985                          614  DC CL48'CXFBR result 2a'
000049F0  40000000 00000000                          615  DC XL16'40000000000000000000000000000000'
00004A00  C3E7C6C2 D9409985                          616  DC CL48'CXFBR result 2b'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00004A30  40000000 00000000                          617  DC XL16'40000000000000000000000000000000'
00004A40  C3E7C6C2 D9409985                          618  DC CL48'CXFBR result 3a'
00004A70  40010000 00000000                          619  DC XL16'40010000000000000000000000000000'
00004A80  C3E7C6C2 D9409985                          620  DC CL48'CXFBR result 3b'
00004AB0  40010000 00000000                          621  DC XL16'40010000000000000000000000000000'
00004AC0  C3E7C6C2 D9409985                          622  DC CL48'CXFBR result 4a'
00004AF0  C0000000 00000000                          623  DC XL16'C0000000000000000000000000000000'
00004B00  C3E7C6C2 D9409985                          624  DC CL48'CXFBR result 4b'
00004B30  C0000000 00000000                          625  DC XL16'C0000000000000000000000000000000'
00004B40  C3E7C6C2 D9409985                          626  DC CL48'CXFBR result 5a'
00004B70  401DFFFF FFFC0000                          627  DC XL16'401DFFFFFFFC0000000000000000000'
00004B80  C3E7C6C2 D9409985                          628  DC CL48'CXFBR result 5b'
00004BB0  401DFFFF FFFC0000                          629  DC XL16'401DFFFFFFFC0000000000000000000'
00004BC0  C3E7C6C2 D9409985                          630  DC CL48'CXFBR result 6a'
00004BF0  C01DFFFF FFFC0000                          631  DC XL16'C01DFFFFFFFC0000000000000000000'
00004C00  C3E7C6C2 D9409985                          632  DC CL48'CXFBR result 6b'
00004C30  C01DFFFF FFFC0000                          633  DC XL16'C01DFFFFFFFC0000000000000000000'
00004C40  C3E7C6C2 D9409985                          634  DC CL48'CXFBR result 7a'
00004C70  401DFFFF FE000000                          635  DC XL16'401DFFFFFE0000000000000000000'
00004C80  C3E7C6C2 D9409985                          636  DC CL48'CXFBR result 7b'
00004CB0  401DFFFF FE000000                          637  DC XL16'401DFFFFFE0000000000000000000'
                            0000000E  00000001    638 XBFPOUT_NUM EQU (*-XBFPOUT_GOOD)/64
                                                  639 *
                                                  640 *
                            00004CC0  00000001    641 XBFPFLGS_GOOD EQU *
00004CC0  C3E7C6C2 D940C6D7                          642  DC CL48'CXFBR FPC pairs 1-2'
00004CF0  00000000 F8000000                          643  DC XL16'00000000F800000000000000F8000000'
00004D00  C3E7C6C2 D940C6D7                          644  DC CL48'CXFBR FPC pairs 3-4'
00004D30  00000000 F8000000                          645  DC XL16'00000000F800000000000000F8000000'
00004D40  C3E7C6C2 D940C6D7                          646  DC CL48'CXFBR FPC pairs 5-6'
00004D70  00000000 F8000000                          647  DC XL16'00000000F800000000000000F8000000'
00004D80  C3E7C6C2 D940C6D7                          648  DC CL48'CXFBR FPC pair 7'
00004DB0  00000000 F8000000                          649  DC XL16'00000000F800000000000000000000'
                          00000004  00000001    650 XBFPFLGS_NUM EQU (*-XBFPFLGS_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00004DC0                                        652 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                654 ********************************************************************
                                                655 *                 REPORT UNEXPECTED PROGRAM CHECK
                                                656 ********************************************************************

00004DC0                                        658 PGMCK    DS    0H
00004DC0  F342 C072 F08E      00004E32  0000008E  659          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
00004DC6  926B C076                     00004E36  660          MVI   PGMCOMMA,C','
00004DCA  DC03 C072 C178      00004E32  00004F38  661          TR    PROGCODE,HEXTRTAB

00004DD0  F384 C07C F150      00004E3C  00000150  663          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
00004DD6  9240 C084                     00004E44  664          MVI   PGMPSW+(0*9)+8,C' '
00004DDA  DC07 C07C C178      00004E3C  00004F38  665          TR    PGMPSW+(0*9)(8),HEXTRTAB

00004DE0  F384 C085 F154      00004E45  00000154  667          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
00004DE6  9240 C08D                     00004E4D  668          MVI   PGMPSW+(1*9)+8,C' '
00004DEA  DC07 C085 C178      00004E45  00004F38  669          TR    PGMPSW+(1*9)(8),HEXTRTAB

00004DF0  F384 C08E F158      00004E4E  00000158  671          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
00004DF6  9240 C096                     00004E56  672          MVI   PGMPSW+(2*9)+8,C' '
00004DFA  DC07 C08E C178      00004E4E  00004F38  673          TR    PGMPSW+(2*9)(8),HEXTRTAB

00004E00  F384 C097 F15C      00004E57  0000015C  675          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
00004E06  9240 C09F                     00004E5F  676          MVI   PGMPSW+(3*9)+8,C' '
00004E0A  DC07 C097 C178      00004E57  00004F38  677          TR    PGMPSW+(3*9)(8),HEXTRTAB

00004E10  4100 0042                     00000042  679          LA    R0,L'PROGMSG     R0 <== length of message
00004E14  4110 C05E                     00004E1E  680          LA    R1,PROGMSG       R1 --> the message text itself
00004E18  4520 C27A                     0000503A  681          BAL   R2,MSG           Go display this message
                                                  682
00004E1C  07FD                                    683          BR    R13              Return to caller

00004E1E                                        685 PROGMSG  DS   0CL66
00004E1E  D7D9D6C7 D9C1D440                     686          DC    CL20'PROGRAM CHECK! CODE '
00004E32  88888888                              687 PROGCODE DC    CL4'hhhh'
00004E36  6B                                    688 PGMCOMMA DC    CL1','
00004E37  40D7E2E6 40                           689          DC    CL5' PSW '
00004E3C  88888888 88888888                     690 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                      692 ************************************************************************
                                                      693 *                      VERIFICATION ROUTINE
                                                      694 ************************************************************************

00004E60                                              696 VERISUB  DS    0H
                                                      697 *
                                                      698 **       Loop through the VERIFY TABLE...
                                                      699 *

00004E60  4110 C32C                 000050EC          701          LA    R1,VERIFTAB      R1 --> Verify table
00004E64  4120 0008                 00000008          702          LA    R2,VERIFLEN      R2 <== Number of entries
00004E68  0D30                                        703          BASR  R3,0             Set top of loop

00004E6A  9846 1000                 00000000          705          LM    R4,R6,0(R1)      Load verify table values
00004E6E  4D70 C0C2                 00004E82          706          BAS   R7,VERIFY         Verify results
00004E72  4110 100C                 0000000C          707          LA    R1,12(,R1)       Next verify table entry
00004E76  0623                                        708          BCTR  R2,R3            Loop through verify table

00004E78  9500 C278                 00005038          710          CLI   FAILFLAG,X'00'   Did all tests verify okay?
00004E7C  078D                                        711          BER   R13              Yes, return to caller
00004E7E  47F0 F238                 00000238          712          B     FAIL             No, load FAILURE disabled wait PSW


                                                      714 *
                                                      715 **       Loop through the ACTUAL / EXPECTED results...
                                                      716 *

00004E82  0D80                                        718 VERIFY   BASR  R8,0            Set top of loop

00004E84  D50F 4000 5030      00000000  00000030      720          CLC   0(16,R4),48(R5)  Actual results == Expected results?
00004E8A  4770 C0DA                     00004E9A      721          BNE   VERIFAIL         No, show failure
00004E8E  4140 4010                     00000010      722 VERINEXT LA    R4,16(,R4)       Next actual result
00004E92  4150 5040                     00000040      723          LA    R5,64(,R5)       Next expected result
00004E96  0668                                        724          BCTR  R6,R8            Loop through results

00004E98  07F7                                        726          BR    R7               Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                   728 ***************************************************************************
                                                   729 *                       Report the failure...
                                                   730 ***************************************************************************

00004E9A  9005 C250                     00005010   732 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
00004E9E  92FF C278                     00005038   733          MVI   FAILFLAG,X'FF'   Remember verification failure
                                                   734 *
                                                   735 **       First, show them the description...
                                                   736 *
00004EA2  D22F C1E0 5000      00004FA0  00000000   737          MVC   FAILDESC,0(R5)   Save results/test description
00004EA8  4100 0044                     00000044   738          LA    R0,L'FAILMSG1    R0 <== length of message
00004EAC  4110 C1CC                     00004F8C   739          LA    R1,FAILMSG1      R1 --> the message text itself
00004EB0  4520 C27A                     0000503A   740          BAL   R2,MSG           Go display this message
                                                   741 *
                                                   742 **       Save address of actual and expected results
                                                   743 *
00004EB4  5040 C24C                     0000500C   744          ST    R4,AACTUAL       Save A(actual results)
00004EB8  4150 5030                     00000030   745          LA    R5,48(,R5)       R5 ==> expected results
00004EBC  5050 C248                     00005008   746          ST    R5,AEXPECT       Save A(expected results)
                                                   747 *
                                                   748 **       Format and show them the EXPECTED ("Want") results...
                                                   749 *
00004EC0  D205 C210 C390      00004FD0  00005150   750          MVC   WANTGOT,=CL6'Want: '
00004EC6  F384 C216 C248      00004FD6  00005008   751          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
00004ECC  9240 C21E                     00004FDE   752          MVI   BLANKEQ,C' '
00004ED0  DC07 C216 C178      00004FD6  00004F38   753          TR    FAILADR,HEXTRTAB

00004ED6  F384 C221 5000      00004FE1  00000000   755          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
00004EDC  9240 C229                     00004FE9   756          MVI   FAILVALS+(0*9)+8,C' '
00004EE0  DC07 C221 C178      00004FE1  00004F38   757          TR    FAILVALS+(0*9)(8),HEXTRTAB

00004EE6  F384 C22A 5004      00004FEA  00000004   759          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
00004EEC  9240 C232                     00004FF2   760          MVI   FAILVALS+(1*9)+8,C' '
00004EF0  DC07 C22A C178      00004FEA  00004F38   761          TR    FAILVALS+(1*9)(8),HEXTRTAB

00004EF6  F384 C233 5008      00004FF3  00000008   763          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
00004EFC  9240 C23B                     00004FFB   764          MVI   FAILVALS+(2*9)+8,C' '
00004F00  DC07 C233 C178      00004FF3  00004F38   765          TR    FAILVALS+(2*9)(8),HEXTRTAB

00004F06  F384 C23C 500C      00004FFC  0000000C   767          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
00004F0C  9240 C244                     00005004   768          MVI   FAILVALS+(3*9)+8,C' '
00004F10  DC07 C23C C178      00004FFC  00004F38   769          TR    FAILVALS+(3*9)(8),HEXTRTAB

00004F16  4100 0035                     00000035   771          LA    R0,L'FAILMSG2    R0 <== length of message
00004F1A  4110 C210                     00004FD0   772          LA    R1,FAILMSG2      R1 --> the message text itself
00004F1E  4520 C27A                     0000503A   773          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    775 *
                                                    776 **       Format and show them the ACTUAL ("Got") results...
                                                    777 *
00004F22  D205 C210 C396      00004FD0  00005156  778          MVC   WANTGOT,=CL6'Got:  '
00004F28  F384 C216 C24C      00004FD6  0000500C  779          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
00004F2E  9240 C21E                     00004FDE  780          MVI   BLANKEQ,C' '
00004F32  DC07 C216 C178      00004FD6  00004F38  781          TR    FAILADR,HEXTRTAB

00004F38  F384 C221 4000      00004FE1  00000000  783          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
00004F3E  9240 C229                     00004FE9  784          MVI   FAILVALS+(0*9)+8,C' '
00004F42  DC07 C221 C178      00004FE1  00004F38  785          TR    FAILVALS+(0*9)(8),HEXTRTAB

00004F48  F384 C22A 4004      00004FEA  00000004  787          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
00004F4E  9240 C232                     00004FF2  788          MVI   FAILVALS+(1*9)+8,C' '
00004F52  DC07 C22A C178      00004FEA  00004F38  789          TR    FAILVALS+(1*9)(8),HEXTRTAB

00004F58  F384 C233 4008      00004FF3  00000008  791          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
00004F5E  9240 C23B                     00004FFB  792          MVI   FAILVALS+(2*9)+8,C' '
00004F62  DC07 C233 C178      00004FF3  00004F38  793          TR    FAILVALS+(2*9)(8),HEXTRTAB

00004F68  F384 C23C 400C      00004FFC  0000000C  795          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
00004F6E  9240 C244                     00005004  796          MVI   FAILVALS+(3*9)+8,C' '
00004F72  DC07 C23C C178      00004FFC  00004F38  797          TR    FAILVALS+(3*9)(8),HEXTRTAB

00004F78  4100 0035                     00000035  799          LA    R0,L'FAILMSG2    R0 <== length of message
00004F7C  4110 C210                     00004FD0  800          LA    R1,FAILMSG2      R1 --> the message text itself
00004F80  4520 C27A                     0000503A  801          BAL   R2,MSG           Go display this message

00004F84  9805 C250                     00005010  803          LM    R0,R5,SAVER0R5   Restore registers
00004F88  47F0 C0CE                     00004E8E  804          B     VERINEXT         Continue with verification...



00004F8C                                          806 FAILMSG1 DS    0CL68
00004F8C  C3D6D4D7 C1D9C9E2                       807          DC    CL20'COMPARISON FAILURE! '
00004FA0  4D8485A2 83998997                       808 FAILDESC DC    CL48'(description)'


00004FD0                                          810 FAILMSG2 DS    0CL53
00004FD0  40404040 4040                           811 WANTGOT  DC    CL6' '           'Want: ' -or- 'Got:  '
00004FD6  C1C1C1C1 C1C1C1C1                       812 FAILADR  DC    CL8'AAAAAAAA'
00004FDE  407E40                                  813 BLANKEQ  DC    CL3' = '
00004FE1  88888888 88888888                       814 FAILVALS DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '


00005008  00000000                                816 AEXPECT  DC    F'0'             ==> Expected ("Want") results
0000500C  00000000                                817 AACTUAL  DC    F'0'             ==> Actual ("Got") results
00005010  00000000 00000000                       818 SAVER0R5 DC    6F'0'            Registers R0 - R5 save area
00005028  F0F1F2F3 F4F5F6F7                       819 CHARHEX  DC    CL16'0123456789ABCDEF'
                             00004F38  00000010   820 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
00005038  00                                      821 FAILFLAG DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          823 ********************************************************************************
                                          824 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                          825 ********************************************************************************

0000503A  4900 C38C                 0000514C   827 MSG      CH    R0,=H'0'                Do we even HAVE a message?
0000503E  07D2                                 828          BNHR  R2                      No, ignore

00005040  9002 C2B0                 00005070   830          STM   R0,R2,MSGSAVE           Save registers

00005044  4900 C38E                 0000514E   832          CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00005048  47D0 C290                 00005050   833          BNH   MSGOK                   Yes, continue
0000504C  4100 005F                 0000005F   834          LA    R0,L'MSGMSG             No, set to maximum

00005050  1820                                 836 MSGOK    LR    R2,R0                   Copy length to work register
00005052  0620                                 837          BCTR  R2,0                    Minus-1 for execute
00005054  4420 C2BC                 0000507C   838          EX    R2,MSGMVC               Copy message to O/P buffer

00005058  4120 200A                 0000000A   840          LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
0000505C  4110 C2C2                 00005082   841          LA    R1,MSGCMD               Point to true command

00005060  83120008                             843          DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
00005064  4780 C2AA                 0000506A   844          BZ    MSGRET                  Return if successful
00005068  0000                                 845          DC    H'0'                    CRASH for debugging purposes

0000506A  9802 C2B0                 00005070   847 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
0000506E  07F2                                 848          BR    R2                      Return to caller
```

```
00005070  00000000 00000000                    850 MSGSAVE  DC    3F'0'                   Registers save area
0000507C  D200 C2CB 1000      0000508B  00000000   851 MSGMVC   MVC   MSGMSG(0),0(R1)       Executed instruction

00005082  D4E2C7D5 D6C8405C                    853 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
0000508B  40404040 40404040                    854 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     856 ********************************************************************************
                                                     857 *                            VERIFY TABLE
                                                     858 ********************************************************************************
                                                     859 *
                                                     860 *        A(actual results), A(expected results), A(#of results)
                                                     861 *
                                                     862 ********************************************************************************
000050EC                                             864 VERIFTAB DC    0F'0'
000050EC  00001000                                   865          DC    A(SBFPOUT)
000050F0  00004000                                   866          DC    A(SBFPOUT_GOOD)
000050F4  00000004                                   867          DC    A(SBFPOUT_NUM)
                                                     868 *
000050F8  00001100                                   869          DC    A(SBFPFLGS)
000050FC  00004100                                   870          DC    A(SBFPFLGS_GOOD)
00005100  00000004                                   871          DC    A(SBFPFLGS_NUM)
                                                     872 *
00005104  00001200                                   873          DC    A(SBFPRMO)
00005108  00004200                                   874          DC    A(SBFPRMO_GOOD)
0000510C  00000009                                   875          DC    A(SBFPRMO_NUM)
                                                     876 *
00005110  00001500                                   877          DC    A(SBFPRMOF)
00005114  00004440                                   878          DC    A(SBFPRMOF_GOOD)
00005118  00000009                                   879          DC    A(SBFPRMOF_NUM)
                                                     880 *
0000511C  00002000                                   881          DC    A(LBFPOUT)
00005120  00004680                                   882          DC    A(LBFPOUT_GOOD)
00005124  00000007                                   883          DC    A(LBFPOUT_NUM)
                                                     884 *
00005128  00002100                                   885          DC    A(LBFPFLGS)
0000512C  00004840                                   886          DC    A(LBFPFLGS_GOOD)
00005130  00000004                                   887          DC    A(LBFPFLGS_NUM)
                                                     888 *
00005134  00003000                                   889          DC    A(XBFPOUT)
00005138  00004940                                   890          DC    A(XBFPOUT_GOOD)
0000513C  0000000E                                   891          DC    A(XBFPOUT_NUM)
                                                     892 *
00005140  00003200                                   893          DC    A(XBFPFLGS)
00005144  00004CC0                                   894          DC    A(XBFPFLGS_GOOD)
00005148  00000004                                   895          DC    A(XBFPFLGS_NUM)
                                                     896 *
                           00000008  00000001       897 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

0000514C                                              899          END
0000514C  0000                                        900                =H'0'
0000514E  005F                                        901                =AL2(L'MSGMSG)
00005150  E68195A3 7A40                               902                =CL6'Want: '
00005156  C796A37A 4040                               903                =CL6'Got:  '
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AACTUAL	F	00500C	4	817	744 779
AEXPECT	F	005008	4	816	746 751
AHELPERS	A	00027C	4	189	179 217
BFPCVTFF	J	000000	20828	105	
BLANKEQ	C	004FDE	3	813	752 780
CDFBR	I	000440	4	389	208
CEFBR	I	00032C	4	268	202
CEFBRA	I	00036E	4	309	205
CHARHEX	C	005028	16	819	820
CTLR0	F	0002E0	4	227	197 198 199
CXFBR	I	000482	4	422	211
EXTDS	F	00030C	4	249	210
FAIL	I	000238	4	187	712
FAILADR	C	004FD6	8	812	751 753 779 781
FAILDESC	C	004FA0	48	808	737
FAILFLAG	X	005038	1	821	710 733
FAILMSG1	C	004F8C	68	806	738 739
FAILMSG2	C	004FD0	53	810	771 772 799 800
FAILPSW	X	0002D0	8	225	187
FAILVALS	C	004FE1	36	814	755 756 757 759 760 761 763 764 765 767 768 769 783 784 785 787 788 789 791 792 793 795 796 797
FPCREGNT	X	0002E4	4	228	275 319 325 331 337 343 348 353 358 363 368 396 429
FPCREGTR	X	0002E8	4	229	280 401 435
FPR0	U	000000	1	126	
FPR1	U	000001	1	127	
FPR10	U	00000A	1	136	432 438
FPR11	U	00000B	1	137	
FPR12	U	00000C	1	138	
FPR13	U	00000D	1	139	
FPR14	U	00000E	1	140	
FPR15	U	00000F	1	141	
FPR2	U	000002	1	128	
FPR3	U	000003	1	129	
FPR4	U	000004	1	130	
FPR5	U	000005	1	131	
FPR6	U	000006	1	132	
FPR7	U	000007	1	133	
FPR8	U	000008	1	134	276 277 281 282 321 322 327 328 333 334 339 340 344 345 349 350 354 355 359 360 364 365 369 370 397 398 402 403 430 431 436 437
FPR9	U	000009	1	135	
GOODPSW	X	0002C0	8	224	221
HELPERS	H	004DC0	2	652	144 189
HEXTRTAB	U	004F38	16	820	661 665 669 673 677 753 757 761 765 769 781 785 789 793 797
IMAGE	1	000000	20828	0	
INTCOUNT	U	000007	1	464	238 244 250
INTIN	F	0004CC	4	456	464 239 245 251
INTINRM	F	0004E8	4	470	474 256
INTRMCT	U	000003	1	474	255
LBFPFLGS	U	002100	1	495	247 885
LBFPFLGS_GOOD	U	004840	1	597	606 886
LBFPFLGS_NUM	U	000004	1	606	887
LBFPOUT	U	002000	1	493	246 881
LBFPOUT_GOOD	U	004680	1	579	594 882
LBFPOUT_NUM	U	000007	1	594	883
LONGS	F	0002FC	4	243	207
MSG	I	00503A	4	827	681 740 773 801

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
MSGCMD	C	005082	9	853	840 841
MSGMSG	C	00508B	95	854	834 851 832
MSGMVC	I	00507C	6	851	838
MSGOK	I	005050	2	836	833
MSGRET	I	00506A	4	847	844
MSGSAVE	F	005070	4	850	830 847
PCINTCD	H	00008E	2	157	174 659
PCNOTDTA	I	00020C	4	178	175
PCOLDPSW	U	000150	1	159	176 663 667 671 675
PGMCK	H	004DC0	2	658	180
PGMCOMMA	C	004E36	1	688	660
PGMPSW	C	004E3C	36	690	663 664 665 667 668 669 671 672 673 675 676 677
PROGCHK	H	000200	2	173	165
PROGCODE	C	004E32	4	687	659 661
PROGMSG	C	004E1E	66	685	679 680
PROGPSW	D	000228	8	186	185
R0	U	000000	1	107	178 181 197 199 679 732 738 771 799 803 827 830 832 834 836 847
R1	U	000001	1	108	274 276 281 315 321 327 333 339 344 349 354 359 364 369 395 397 402 428 430 436 680 701 705 707 739 772 800 841 851
R10	U	00000A	1	117	201 204 207 210 268 269 309 310 389 390 422 423
R11	U	00000B	1	118	
R12	U	00000C	1	119	144 179 217 272 287 313 376 393 409 426 444
R13	U	00000D	1	120	180 202 205 208 211 218 271 288 312 377 392 410 425 445 683 711
R14	U	00000E	1	121	183 184 219 220
R15	U	00000F	1	122	143 178 181
R2	U	000002	1	109	268 270 287 309 311 376 389 391 409 422 424 444 681 702 708 740 773 801 828 830 836 837 838 840 847 848
R3	U	000003	1	110	268 274 284 309 315 373 389 395 406 422 428 441 703 708
R4	U	000004	1	111	705 720 722 744 783 787 791 795
R5	U	000005	1	112	720 723 732 737 745 746 755 759 763 767 803
R6	U	000006	1	113	705 724
R7	U	000007	1	114	269 277 282 285 310 322 328 334 340 345 350 355 360 365 370 374 390 398 403 407 423 431 432 437 438 442 706 726
R8	U	000008	1	115	269 278 283 286 310 323 329 335 341 346 351 356 361 366 371 375 390 399 404 408 423 433 439 443 718 724
R9	U	000009	1	116	
RMSHORTS	A	00031C	4	255	204
SAVER0R5	F	005010	4	818	732 803
SAVEREGS	F	00023C	4	188	178 181
SBFPFLGS	U	001100	1	485	241 869
SBFPFLGS_GOOD	U	004100	1	523	532 870
SBFPFLGS_NUM	U	000004	1	532	871
SBFPOUT	U	001000	1	483	240 865
SBFPOUT_GOOD	U	004000	1	511	520 866
SBFPOUT_NUM	U	000004	1	520	867
SBFPRMO	U	001200	1	487	257 873
SBFPRMOF	U	001500	1	489	258 877
SBFPRMOF_GOOD	U	004440	1	557	576 878
SBFPRMOF_NUM	U	000009	1	576	879
SBFPRMO_GOOD	U	004200	1	535	554 874
SBFPRMO_NUM	U	000009	1	554	875
SHORTS	F	0002EC	4	237	201
START	I	000280	4	197	162
STRTLABL	U	000000	1	106	156 159 161 164 172 483 485 487 489 493 495 499 501 509
VERIFAIL	I	004E9A	4	732	721
VERIFLEN	U	000008	1	897	702

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
VERIFTAB	F	0050EC	4	864	897 701
VERIFY	I	004E82	2	718	706
VERINEXT	I	004E8E	4	722	804
VERISUB	H	004E60	2	696	218
WANTGOT	C	004FD0	6	811	750 778
XBFPFLGS	U	003200	1	501	253 893
XBFPFLGS_GOOD	U	004CC0	1	641	650 894
XBFPFLGS_NUM	U	000004	1	650	895
XBFPOUT	U	003000	1	499	252 889
XBFPOUT_GOOD	U	004940	1	609	638 890
XBFPOUT_NUM	U	00000E	1	638	891
=AL2(L'MSGMSG)	R	00514E	2	901	832
=CL6'Got: '	C	005156	6	903	778
=CL6'Want: '	C	005150	6	902	750
=H'0'	H	00514C	2	900	827

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 0

Image       IMAGE      20828  0000-515B  0000-515B
  Region               20828  0000-515B  0000-515B
    CSECT   BFPCVTFF   20828  0000-515B  0000-515B
```

```
   STMT                                                FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-010-cvtfrfix\bfp-010-cvtfrfix.asm

** NO ERRORS FOUND **
```