```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                        2 ************************************************************************
                                                        3 *
                                                        4 *Testcase IEEE CONVERT FROM LOGICAL 64
                                                        5 *  Test case capability includes ieee exceptions trappable and otherwise.
                                                        6 *  Test result, FPC flags, and DXC saved for all tests.  (Convert From
                                                        7 *  Logical 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-009-cvtfrlog64.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 three conversion instructions
                                                      66 *   CONVERT FROM LOGICAL (64 to short BFP, RRF-e)
                                                      67 *   CONVERT FROM LOGICAL (64 to long BFP, RRF-e)
                                                      68 *   CONVERT FROM LOGICAL (64 to extended BFP, RRF-e)
                                                      69 *
                                                      70 * Limited test data is compiled into this program.  The test script
                                                      71 * that runs this program can provide alternative test data through
                                                      72 * Hercules R commands.
                                                      73 *
                                                      74 * Test Case Order
                                                      75 * 1) Uint-64 to Short BFP
                                                      76 * 2) Uint-64 to Short BFP with all rounding modes
                                                      77 * 3) Uint-64 to Long BFP
                                                      78 * 4) Uint-64 to Long BFP with all rounding modes
                                                      79 * 5) Uint-64 to Extended BFP
                                                      80 *
                                                      81 * Provided test data is:
                                                      82 *         1, 2, 4,
                                                      83 *         9 007 199 254 740 991(0x001FFFFFFFFFFFF)
                                                      84 *        18 014 398 509 481 983(0x003FFFFFFFFFFFF)
                                                      85 *    18 446 744 073 709 551 615 (0xFFFFFFFFFFFFFFFF)
                                                      86 *
                                                      87 *    The fourth value oveflows a short BFP but fits in a long BFP.
                                                      88 *    The fifth  value oveflows both short BFP and long BFP.  The
                                                      89 *    last value also overflows both, but fits in an extended BFP.
                                                      90 *
                                                      91 * Also tests the following floating point support instructions
                                                      92 *    LOAD  (Short)
                                                      93 *    LOAD  (Long)
                                                      94 *    LOAD FPC
                                                      95 *    SET BFP ROUNDING MODE 2-BIT
                                                      96 *    SET BFP ROUNDING MODE 3-BIT
                                                      97 *    STORE (Short)
                                                      98 *    STORE (Long)
                                                      99 *    STORE FPC
                                                     100 *
                                                     101 ********************************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                          103 *
                                          104 *  Note: for compatibility with the z/CMS test rig, do not change
                                          105 *  or use R11, R14, or R15.  Everything else is fair game.
                                          106 *



                          00000000  000055B3   108 BFPCVTFL START 0
                          00000000  00000001   109 STRTLABL EQU   *
                          00000000  00000001   110 R0       EQU   0                 Work register for cc extraction
                          00000001  00000001   111 R1       EQU   1
                          00000002  00000001   112 R2       EQU   2                 Holds count of test input values
                          00000003  00000001   113 R3       EQU   3                 Points to next test input value(s)
                          00000004  00000001   114 R4       EQU   4                 Available
                          00000005  00000001   115 R5       EQU   5                 Available
                          00000006  00000001   116 R6       EQU   6                 Available
                          00000007  00000001   117 R7       EQU   7                 Pointer to next result value(s)
                          00000008  00000001   118 R8       EQU   8                 Pointer to next FPCR result
                          00000009  00000001   119 R9       EQU   9                 Rounding tests top of outer loop
                          0000000A  00000001   120 R10      EQU   10                Pointer to test address list
                          0000000B  00000001   121 R11      EQU   11                **Reserved for z/CMS test rig
                          0000000C  00000001   122 R12      EQU   12                Holds number of test cases in set
                          0000000D  00000001   123 R13      EQU   13                Mainline return address
                          0000000E  00000001   124 R14      EQU   14                **Return address for z/CMS test rig
                          0000000F  00000001   125 R15      EQU   15                **Base register on z/CMS or Hyperion
                                          126 *
                                          127 * Floating Point Register equates to keep the cross reference clean
                                          128 *
                          00000000  00000001   129 FPR0     EQU   0
                          00000001  00000001   130 FPR1     EQU   1
                          00000002  00000001   131 FPR2     EQU   2
                          00000003  00000001   132 FPR3     EQU   3
                          00000004  00000001   133 FPR4     EQU   4
                          00000005  00000001   134 FPR5     EQU   5
                          00000006  00000001   135 FPR6     EQU   6
                          00000007  00000001   136 FPR7     EQU   7
                          00000008  00000001   137 FPR8     EQU   8
                          00000009  00000001   138 FPR9     EQU   9
                          0000000A  00000001   139 FPR10    EQU   10
                          0000000B  00000001   140 FPR11    EQU   11
                          0000000C  00000001   141 FPR12    EQU   12
                          0000000D  00000001   142 FPR13    EQU   13
                          0000000E  00000001   143 FPR14    EQU   14
                          0000000F  00000001   144 FPR15    EQU   15
                                          145 *
00000000                  00000000            146          USING *,R15
00000000                  00005200            147          USING HELPERS,R12
                                          148 *
                                          149 * Above works on real iron (R15=0 after sysclear)
                                          150 * and in z/CMS (R15 points to start of load module)
                                          151 *



                                          153 ************************************************************************
                                          154 *
                                          155 * Low core definitions, Restart PSW, and Program Check Routine.
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT
                                                     156 *
                                                     157 ********************************************************************************


00000000                        00000000  0000008E   159          ORG   STRTLABL+X'8E'      Program check interrution code
0000008E  0000                                       160 PCINTCD  DS    H
                                                     161 *
                                00000150  00000001   162 PCOLDPSW EQU   STRTLABL+X'150'     z/Arch Program check old PSW
                                                     163 *
00000090                        00000090  000001A0   164          ORG   STRTLABL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                          165          DC    X'0000000180000000',AD(START)
                                                     166 *
000001B0                        000001B0  000001D0   167          ORG   STRTLABL+X'1D0'     z/Arch Program check NEW PSW
000001D0  00000000 00000000                          168          DC    X'0000000000000000',AD(PROGCHK)
                                                     169 *
                                                     170 * Program check routine.  If Data Exception, continue execution at
                                                     171 * the instruction following the program check.  Otherwise, hard wait.
                                                     172 * No need to collect data.  All interesting DXC stuff is captured
                                                     173 * in the FPCR.
                                                     174 *
000001E0                        000001E0  00000200   175          ORG   STRTLABL+X'200'
00000200                                             176 PROGCHK  DS    0H              Program check occured...
00000200  9507 F08F                     0000008F   177          CLI   PCINTCD+1,X'07'  Data Exception?
00000204  A774 0004                     0000020C   178          JNE   PCNOTDTA       ..no, hardwait (not sure if R15 is ok)
00000208  B2B2 F150                     00000150   179          LPSWE PCOLDPSW       ..yes, resume program execution

0000020C  900F F23C                     0000023C   181 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                     0000027C   182          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                     00005200   183          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                     0000023C   184          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                       186          LTR   R14,R14        Return address provided?
0000021E  077E                                       187          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                     00000228   188          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                          189 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2D8                     000002D8   190 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                          191 SAVEREGS DC    16F'0'           Registers save area
0000027C  00005200                                   192 AHELPERS DC    A(HELPERS)       Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    194 ************************************************************************
                                                    195 *
                                                    196 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                    197 *
                                                    198 ************************************************************************


00000280  B600 F2E8                       000002E8  200 START    STCTL R0,R0,CTLR0    Store CR0 to enable AFP
00000284  9604 F2E9                       000002E9  201          OI    CTLR0+1,X'04'  Turn on AFP bit
00000288  B700 F2E8                       000002E8  202          LCTL  R0,R0,CTLR0    Reload updated CR0
                                                    203 *
0000028C  41A0 F2F4                       000002F4  204          LA    R10,SHORTS     Point to uint-64 test inputs
00000290  4DD0 F344                       00000344  205          BAS   R13,CELGBR     Convert values from fixed to short BFP
00000294  41A0 F324                       00000324  206          LA    R10,RMSHORTS   Point to inputs for rounding mode tests
00000298  4DD0 F388                       00000388  207          BAS   R13,CELGBRA    Convert using all rounding mode options
                                                    208 *
0000029C  41A0 F304                       00000304  209          LA    R10,LONGS      Point to uint-64 test inputs
000002A0  4DD0 F45C                       0000045C  210          BAS   R13,CDLGBR     Convert values from fixed to long
000002A4  41A0 F334                       00000334  211          LA    R10,RMLONGS    Point to inputs for rounding mode tests
000002A8  4DD0 F4A0                       000004A0  212          BAS   R13,CDLGBRA    Convert using all rounding mode options
                                                    213 *
000002AC  41A0 F314                       00000314  214          LA    R10,EXTDS      Point to uint-64 test inputs
000002B0  4DD0 F574                       00000574  215          BAS   R13,CXLGBR     Convert values from fixed to extended
                                                    216 *
                                                    217 * uint-64 always fits in extended BFP exactly.  No rounding nor
                                                    218 * loss of precision, so no need for exhaustive rounding tests
                                                    219 *
                                                    220 ************************************************************************
                                                    221 *                    Verify test results...
                                                    222 ************************************************************************
                                                    223 *
000002B4  58C0 F27C                       0000027C  224          L     R12,AHELPERS     Get address of helper subroutines
000002B8  4DD0 C0A0                       000052A0  225          BAS   R13,VERISUB      Go verify results
000002BC  12EE                                      226          LTR   R14,R14          Was return address provided?
000002BE  077E                                      227          BNZR  R14              Yes, return to z/CMS test rig.
000002C0  B2B2 F2C8                       000002C8  228          LPSWE GOODPSW          Load SUCCESS PSW
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000002C8                                          230          DS    0D             Ensure correct alignment for PSW
000002C8  00020000 00000000                       231 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002D8  00020000 00000000                       232 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                  233 *
000002E8  00000000                                234 CTLR0    DS    F
000002EC  00000000                                235 FPCREGNT DC    X'00000000'  FPCR, trap all IEEE exceptions, zero flags
000002F0  F8000000                                236 FPCREGTR DC    X'F8000000'  FPCR, trap no IEEE exceptions, zero flags
                                                  237 *
                                                  238 * Input values parameter list, four fullwords:
                                                  239 *      1) Count,
                                                  240 *      2) Address of inputs,
                                                  241 *      3) Address to place results, and
                                                  242 *      4) Address to place DXC/Flags/cc values.
                                                  243 *
000002F4                                          244 SHORTS   DS    0F
000002F4  00000006                                245          DC    A(INTCOUNT/8)
000002F8  000005C0                                246          DC    A(INTIN)
000002FC  00001000                                247          DC    A(SBFPOUT)
00000300  00001100                                248          DC    A(SBFPFLGS)
                                                  249 *
00000304                                          250 LONGS    DS    0F          uint-64 inputs for long BFP testing
00000304  00000006                                251          DC    A(INTCOUNT/8)
00000308  000005C0                                252          DC    A(INTIN)
0000030C  00002000                                253          DC    A(LBFPOUT)
00000310  00002100                                254          DC    A(LBFPFLGS)
                                                  255 *
00000314                                          256 EXTDS    DS    0F          uint-64 inputs for Extended BFP testing
00000314  00000006                                257          DC    A(INTCOUNT/8)
00000318  000005C0                                258          DC    A(INTIN)
0000031C  00003000                                259          DC    A(XBFPOUT)
00000320  00003200                                260          DC    A(XBFPFLGS)
                                                  261 *
00000324  00000003                                262 RMSHORTS DC    A(SINTRMCT/8)
00000328  000005F0                                263          DC    A(SINTRMIN)
0000032C  00001200                                264          DC    A(SBFPRMO)   Space for rounding mode tests
00000330  00001500                                265          DC    A(SBFPRMOF)  Space for rounding mode test flags
                                                  266 *
00000334  00000003                                267 RMLONGS  DC    A(LINTRMCT/8)
00000338  00000608                                268          DC    A(LINTRMIN)  Last two uint-64 are only concerns
0000033C  00002200                                269          DC    A(LBFPRMO)   Space for rounding mode tests
00000340  00002700                                270          DC    A(LBFPRMOF)  Space for rounding mode test flags
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    272 ************************************************************************
                                                    273 *
                                                    274 * Convert integers to short BFP format.  A pair of results is generated
                                                    275 * for each input: one with all exceptions non-trappable, and the second
                                                    276 * with all exceptions trappable.   The FPCR is stored for each result.
                                                    277 *
                                                    278 ************************************************************************


00000344  9823 A000                 00000000   280 CELGBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000348  1222                                 281          LTR   R2,R2         Any test cases?
0000034A  078D                                 282          BZR   R13           ..No, return to caller
0000034C  9878 A008                 00000008   283          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000350  0DC0                                 284          BASR  R12,0         Set top of loop
                                               285 *
00000352  E310 3000 0004            00000000   286          LG    R1,0(,R3)     Get integer test value
00000358  B29D F2EC                 000002EC   287          LFPC  FPCREGNT      Set exceptions non-trappable
0000035C  B3A0 0081                            288          CELGBR FPR8,0,R1,0  Cvt uint in GPR1 to float in FPR8
00000360  7080 7000                 00000000   289          STE   FPR8,0(,R7)   Store short BFP result
00000364  B29C 8000                 00000000   290          STFPC 0(R8)         Store resulting FPC flags and DXC
                                               291 *
00000368  B29D F2F0                 000002F0   292          LFPC  FPCREGTR      Set exceptions trappable
0000036C  B3A0 0081                            293          CELGBR FPR8,0,R1,0  Cvt uint in GPR1 to float in FPR8
00000370  7080 7004                 00000004   294          STE   FPR8,4(,R7)   Store short BFP result
00000374  B29C 8004                 00000004   295          STFPC 4(R8)         Store resulting FPC flags and DXC
00000378  4130 3008                 00000008   296          LA    R3,8(,R3)     Point to next input values
0000037C  4170 7008                 00000008   297          LA    R7,8(,R7)     Point to next short BFP converted values
00000380  4180 8008                 00000008   298          LA    R8,8(,R8)     Point to next FPCR/CC result area
00000384  062C                                 299          BCTR  R2,R12        Convert next input value.
00000386  07FD                                 300          BR    R13           All converted; return.
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                     302 ************************************************************************
                                                     303 *
                                                     304 * Convert uint-64 to short BFP format using every rounding mode.
                                                     305 * Ten test results are generated for each input.  A 48-byte test result
                                                     306 * section is used to keep results sets aligned on a quad-double word.
                                                     307 *
                                                     308 * The first four tests use rounding modes specified in the FPCR with
                                                     309 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for the
                                                     310 * first two FPCR-controlled tests and SRNMB (3-bit) is used for the
                                                     311 * last two To get full coverage of that instruction pair.
                                                     312 *
                                                     313 * The next six results use instruction-specified rounding modes.
                                                     314 *
                                                     315 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                     316 * prior tests used the default rounding mode.  RNTE is tested
                                                     317 * explicitly as a rounding mode in this section.
                                                     318 *
                                                     319 ************************************************************************


00000388  9823 A000                 00000000    321 CELGBRA  LM    R2,R3,0(R10)  Get count and address of test input values
0000038C  1222                                   322          LTR   R2,R2         Any test cases?
0000038E  078D                                   323          BZR   R13           ..No, return to caller
00000390  9878 A008                 00000008    324          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000394  0DC0                                   325          BASR  R12,0         Set top of loop
                                                 326 *
00000396  E310 3000 0004            00000000    327          LG    R1,0(,R3)     Get uint-64 test value
                                                 328 *
                                                 329 * Test cases using rounding mode specified in the FPCR
                                                 330 *
0000039C  B29D F2EC                 000002EC    331          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003A0  B2B8 0001                 00000001    332          SRNMB 1             SET FPC to RZ, towards zero.
000003A4  B3A0 0481                             333          CELGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003A8  6080 7000                 00000000    334          STD   FPR8,0*4(,R7) Store short BFP result
000003AC  B29C 8000                 00000000    335          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                 336 *
000003B0  B29D F2EC                 000002EC    337          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003B4  B2B8 0002                 00000002    338          SRNMB 2             SET FPC to RP, to +infinity
000003B8  B3A0 0481                             339          CELGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003BC  6080 7004                 00000004    340          STD   FPR8,1*4(,R7) Store short BFP result
000003C0  B29C 8004                 00000004    341          STFPC 1*4(R8)       Store resulting FPC flags and DXC
                                                 342 *
000003C4  B29D F2EC                 000002EC    343          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003C8  B2B8 0003                 00000003    344          SRNMB 3             SET FPC to RM, to -infinity
000003CC  B3A0 0481                             345          CELGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003D0  6080 7008                 00000008    346          STD   FPR8,2*4(,R7) Store short BFP result
000003D4  B29C 8008                 00000008    347          STFPC 2*4(R8)       Store resulting FPC flags and DXC
                                                 348 *
000003D8  B29D F2EC                 000002EC    349          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003DC  B2B8 0007                 00000007    350          SRNMB 7             RFS, Prepare for Shorter Precision
000003E0  B3A0 0481                             351          CELGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003E4  6080 700C                 0000000C    352          STD   FPR8,3*4(,R7) Store short BFP result
000003E8  B29C 800C                 0000000C    353          STFPC 3*4(R8)       Store resulting FPC flags and DXC
                                                 354 *
000003EC  B29D F2EC                 000002EC    355          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003F0  B3A0 1081                             356          CELGBR FPR8,1,R1,B'0000'  RNTA, to nearest, ties away
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000003F4  7080 7010                    00000010   357          STE   FPR8,4*4(,R7) Store short BFP result
000003F8  B29C 8010                    00000010   358          STFPC 4*4(R8)       Store resulting FPC flags and DXC
                                                  359 *
000003FC  B29D F2EC                    000002EC   360          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000400  B3A0 3081                                361          CELGBR FPR8,3,R1,B'0000'  RFS, prepare for shorter precision
00000404  7080 7014                    00000014   362          STE   FPR8,5*4(,R7) Store short BFP result
00000408  B29C 8014                    00000014   363          STFPC 5*4(R8)       Store resulting FPC flags and DXC
                                                  364 *
0000040C  B29D F2EC                    000002EC   365          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000410  B3A0 4081                                366          CELGBR FPR8,4,R1,B'0000'  RNTE, to nearest, ties to even
00000414  7080 7018                    00000018   367          STE   FPR8,6*4(,R7) Store short BFP result
00000418  B29C 8018                    00000018   368          STFPC 6*4(R8)       Store resulting FPC flags and DXC
                                                  369 *
0000041C  B29D F2EC                    000002EC   370          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000420  B3A0 5081                                371          CELGBR FPR8,5,R1,B'0000'  RZ, toward zero
00000424  7080 701C                    0000001C   372          STE   FPR8,7*4(,R7) Store short BFP result
00000428  B29C 801C                    0000001C   373          STFPC 7*4(R8)       Store resulting FPC flags and DXC
                                                  374 *
0000042C  B29D F2EC                    000002EC   375          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000430  B3A0 6081                                376          CELGBR FPR8,6,R1,B'0000'  RP, to +inf
00000434  7080 7020                    00000020   377          STE   FPR8,8*4(,R7) Store short BFP result
00000438  B29C 8020                    00000020   378          STFPC 8*4(R8)       Store resulting FPC flags and DXC
                                                  379 *
0000043C  B29D F2EC                    000002EC   380          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000440  B3A0 7081                                381          CELGBR FPR8,7,R1,B'0000'  RM, to -inf
00000444  7080 7024                    00000024   382          STE   FPR8,9*4(,R7) Store short BFP result
00000448  B29C 8024                    00000024   383          STFPC 9*4(R8)       Store resulting FPC flags and DXC
                                                  384 *
0000044C  4130 3008                    00000008   385          LA    R3,8(,R3)     Point to next input values
00000450  4170 7030                    00000030   386          LA    R7,12*4(,R7)  Point to next short BFP converted values
00000454  4180 8030                    00000030   387          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
00000458  062C                                     388          BCTR  R2,R12        Convert next input value.
0000045A  07FD                                     389          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    391 ************************************************************************
                                                    392 *
                                                    393 * Convert integers to long BFP format.  A pair of results is generated
                                                    394 * for each input: one with all exceptions non-trappable, and the second
                                                    395 * with all exceptions trappable.   The FPCR is stored for each result.
                                                    396 * Conversion of a 64-bit integer to long is always exact; no exceptions
                                                    397 * are expected
                                                    398 *
                                                    399 ************************************************************************


0000045C  9823 A000                 00000000    401 CDLGBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000460  9878 A008                 00000008    402          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000464  1222                                  403          LTR   R2,R2         Any test cases?
00000466  078D                                  404          BZR   R13           ..No, return to caller
00000468  0DC0                                  405          BASR  R12,0         Set top of loop
                                                406 *
0000046A  E310 3000 0004            00000000    407          LG    R1,0(,R3)     Get integer test value
00000470  B29D F2EC                 000002EC    408          LFPC  FPCREGNT      Set exceptions non-trappable
00000474  B3A1 0081                             409          CDLGBR FPR8,0,R1,0  Cvt uint in GPR1 to float in FPR8
00000478  6080 7000                 00000000    410          STD   FPR8,0(,R7)   Store long BFP result
0000047C  B29C 8000                 00000000    411          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                412 *
00000480  B29D F2F0                 000002F0    413          LFPC  FPCREGTR      Set exceptions trappable
00000484  B3A1 0081                             414          CDLGBR FPR8,0,R1,0  Cvt uint in GPR1 to float in FPR8
00000488  6080 7008                 00000008    415          STD   FPR8,8(,R7)   Store long BFP result
0000048C  B29C 8004                 00000004    416          STFPC 4(R8)         Store resulting FPC flags and DXC
00000490  4130 3008                 00000008    417          LA    R3,8(,R3)     Point to next input value
00000494  4170 7010                 00000010    418          LA    R7,16(,R7)    Point to next long BFP result pair
00000498  4180 8008                 00000008    419          LA    R8,8(,R8)     Point to next FPCR/CC contents pair
0000049C  062C                                  420          BCTR  R2,R12        Convert next input value.
0000049E  07FD                                  421          BR    R13           All converted; return.
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                     423 ********************************************************************************
                                                     424 *
                                                     425 * Convert uint-64 to short BFP format using every rounding mode.
                                                     426 * Ten test results are generated for each input.  A 48-byte test result
                                                     427 * section is used to keep results sets aligned on a quad-double word.
                                                     428 *
                                                     429 * The first four tests use rounding modes specified in the FPCR with
                                                     430 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for the
                                                     431 * first two FPCR-controlled tests and SRNMB (3-bit) is used for the
                                                     432 * last two To get full coverage of that instruction pair.
                                                     433 *
                                                     434 * The next six results use instruction-specified rounding modes.
                                                     435 *
                                                     436 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                     437 * prior tests used the default rounding mode.  RNTE is tested
                                                     438 * explicitly as a rounding mode in this section.
                                                     439 *
                                                     440 ********************************************************************************


000004A0  9823 A000                 00000000    442 CDLGBRA  LM    R2,R3,0(R10)  Get count and address of test input values
000004A4  9878 A008                 00000008    443          LM    R7,R8,8(R10)  Get address of result area and flag area.
000004A8  1222                                  444          LTR   R2,R2         Any test cases?
000004AA  078D                                  445          BZR   R13           ..No, return to caller
000004AC  0DC0                                  446          BASR  R12,0         Set top of loop
                                                447 *
000004AE  E310 3000 0004            00000000    448          LG    R1,0(,R3)     Get uint-64 test value
                                                449 *
                                                450 * Test cases using rounding mode specified in the FPCR
                                                451 *
000004B4  B29D F2EC                 000002EC    452          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004B8  B2B8 0001                 00000001    453          SRNMB 1             SET FPC to RZ, towards zero.
000004BC  B3A1 0481                             454          CDLGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004C0  6080 7000                 00000000    455          STD   FPR8,0*8(,R7) Store short BFP result
000004C4  B29C 8000                 00000000    456          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                457 *
000004C8  B29D F2EC                 000002EC    458          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004CC  B2B8 0002                 00000002    459          SRNMB 2             SET FPC to RP, to +infinity
000004D0  B3A1 0481                             460          CDLGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004D4  6080 7008                 00000008    461          STD   FPR8,1*8(,R7) Store short BFP result
000004D8  B29C 8004                 00000004    462          STFPC 1*4(R8)       Store resulting FPC flags and DXC
                                                463 *
000004DC  B29D F2EC                 000002EC    464          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004E0  B2B8 0003                 00000003    465          SRNMB 3             SET FPC to RM, to -infinity
000004E4  B3A1 0481                             466          CDLGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004E8  6080 7010                 00000010    467          STD   FPR8,2*8(,R7) Store short BFP result
000004EC  B29C 8008                 00000008    468          STFPC 2*4(R8)       Store resulting FPC flags and DXC
                                                469 *
000004F0  B29D F2EC                 000002EC    470          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004F4  B2B8 0007                 00000007    471          SRNMB 7             RFS, Prepare for Shorter Precision
000004F8  B3A1 0481                             472          CDLGBR FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004FC  6080 7018                 00000018    473          STD   FPR8,3*8(,R7) Store short BFP result
00000500  B29C 800C                 0000000C    474          STFPC 3*4(R8)       Store resulting FPC flags and DXC
                                                475 *
00000504  B29D F2EC                 000002EC    476          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000508  B3A1 1081                             477          CDLGBR FPR8,1,R1,B'0000'  RNTA, to nearest, ties away
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

0000050C  6080 7020                    00000020   478          STD   FPR8,4*8(,R7) Store short BFP result
00000510  B29C 8010                    00000010   479          STFPC 4*4(R8)       Store resulting FPC flags and DXC
                                                  480 *
00000514  B29D F2EC                    000002EC   481          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000518  B3A1 3081                               482          CDLGBR FPR8,3,R1,B'0000'  RFS, prepare for shorter precision
0000051C  6080 7028                    00000028   483          STD   FPR8,5*8(,R7) Store short BFP result
00000520  B29C 8014                    00000014   484          STFPC 5*4(R8)       Store resulting FPC flags and DXC
                                                  485 *
00000524  B29D F2EC                    000002EC   486          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000528  B3A1 4081                               487          CDLGBR FPR8,4,R1,B'0000'  RNTE, to nearest, ties to even
0000052C  6080 7030                    00000030   488          STD   FPR8,6*8(,R7)  Store short BFP result
00000530  B29C 8018                    00000018   489          STFPC 6*4(R8)       Store resulting FPC flags and DXC
                                                  490 *
00000534  B29D F2EC                    000002EC   491          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000538  B3A1 5081                               492          CDLGBR FPR8,5,R1,B'0000'  RZ, toward zero
0000053C  6080 7038                    00000038   493          STD   FPR8,7*8(,R7) Store short BFP result
00000540  B29C 801C                    0000001C   494          STFPC 7*4(R8)       Store resulting FPC flags and DXC
                                                  495 *
00000544  B29D F2EC                    000002EC   496          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000548  B3A1 6081                               497          CDLGBR FPR8,6,R1,B'0000'  RP, to +inf
0000054C  6080 7040                    00000040   498          STD   FPR8,8*8(,R7) Store short BFP result
00000550  B29C 8020                    00000020   499          STFPC 8*4(R8)       Store resulting FPC flags and DXC
                                                  500 *
00000554  B29D F2EC                    000002EC   501          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000558  B3A1 7081                               502          CDLGBR FPR8,7,R1,B'0000'  RM, to -inf
0000055C  6080 7048                    00000048   503          STD   FPR8,9*8(,R7) Store short BFP result
00000560  B29C 8024                    00000024   504          STFPC 9*4(R8)       Store resulting FPC flags and DXC
                                                  505 *
00000564  4130 3008                    00000008   506          LA    R3,8(,R3)     Point to next input values
00000568  4170 7050                    00000050   507          LA    R7,10*8(,R7)  Point to next long BFP converted values
0000056C  4180 8030                    00000030   508          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
00000570  062C                                    509          BCTR  R2,R12        Convert next input value.
00000572  07FD                                    510          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    512 ************************************************************************
                                                    513 *
                                                    514 * Convert integers to extended BFP format.  A pair of results is
                                                    515 * generated for each input: one with all exceptions non-trappable,
                                                    516 * and the second with all exceptions trappable.   The FPCR is
                                                    517 * stored for each result.  Conversion of a 64-bit integer to
                                                    518 * extended is always exact; no exceptions are expected
                                                    519 *
                                                    520 ************************************************************************


00000574  9823 A000                 00000000    522 CXLGBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000578  9878 A008                 00000008    523          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000057C  1222                                  524          LTR   R2,R2         Any test cases?
0000057E  078D                                  525          BZR   R13           ..No, return to caller
00000580  0DC0                                  526          BASR  R12,0         Set top of loop
                                                527 *
00000582  E310 3000 0004            00000000    528          LG    R1,0(,R3)     Get integer test value
00000588  B29D F2EC                 000002EC    529          LFPC  FPCREGNT      Set exceptions non-trappable
0000058C  B3A2 0081                             530          CXLGBR FPR8,0,R1,0  Cvt uint in GPR1 to float in FPR8-FPR10
00000590  6080 7000                 00000000    531          STD   FPR8,0(,R7)   Store extended BFP result part 1
00000594  60A0 7008                 00000008    532          STD   FPR10,8(,R7)  Store extended BFP result part 2
00000598  B29C 8000                 00000000    533          STFPC 0(R8)         Store resulting FPC flags and DXC
                                                534 *
0000059C  B29D F2F0                 000002F0    535          LFPC  FPCREGTR      Set exceptions trappable
000005A0  B3A2 0081                             536          CXLGBR FPR8,0,R1,0  Cvt uint in GPR1 to float in FPR8-FPR10
000005A4  6080 7010                 00000010    537          STD   FPR8,16(,R7)  Store extended BFP result part 1
000005A8  60A0 7018                 00000018    538          STD   FPR10,24(,R7) Store extended BFP result part 2
000005AC  B29C 8004                 00000004    539          STFPC 4(R8)         Store resulting FPC flags and DXC
000005B0  4130 3008                 00000008    540          LA    R3,8(,R3)     Point to next input value
000005B4  4170 7020                 00000020    541          LA    R7,32(,R7)    Point to next extended BFP result pair
000005B8  4180 8008                 00000008    542          LA    R8,8(,R8)     Point to next FPCR/CC result pair
000005BC  062C                                  543          BCTR  R2,R12        Convert next input value.
000005BE  07FD                                  544          BR    R13           All converted; return.
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT
                                                     546 ************************************************************************
                                                     547 *
                                                     548 * long integer inputs for Convert From Fixed testing.  The same set of
                                                     549 * inputs are used for short, long, and extended formats.  The last two
                                                     550 * values are used for rounding mode tests for short and long only;
                                                     551 * conversion of uint-64 to extended is always exact.
                                                     552 *
                                                     553 ************************************************************************


                                                     555 *
                                                     556 * int-64 inputs for basic tests
                                                     557 *
000005C0                                             558 INTIN    DS    0D
000005C0  00000000 00000001                         559          DC    FD'U1'
000005C8  00000000 00000002                         560          DC    FD'U2'
000005D0  00000000 00000004                         561          DC    FD'U4'
000005D8  FFFFFF00 00000000                         562          DC    XL8'FFFFFF0000000000'  Exact long and short BFP
000005E0  FFFFFFFF FFFFF800                         563          DC    XL8'FFFFFFFFFFFFF800'  Exact long BFP, inexact short bfp
000005E8  FFFFFFFF FFFFFFFF                         564          DC    XL8'FFFFFFFFFFFFFFFF'  Inexact short & long BFP,
                                                     565 *                                     ..exact extended BFP
                            00000030  00000001       566 INTCOUNT EQU   *-INTIN      Count of integers in list
                                                     567 *
                                                     568 * uint-64 inputs for exhaustive short BFP rounding mode tests
                                                     569 *
000005F0  FFFFFFC0 00000000                         570 SINTRMIN DC    XL8'FFFFFFC000000000'  Rounds nearest up
000005F8  FFFFFF80 00000000                         571          DC    XL8'FFFFFF8000000000'  Tie
00000600  FFFFFF40 00000000                         572          DC    XL8'FFFFFF4000000000'  Rounds nearest down
00000608                                            573          DS    0F           required by asma for following EQU to work.
                            00000018  00000001       574 SINTRMCT EQU   *-SINTRMIN   Count of integers for rounding mode tests
                                                     575 *
                                                     576 * uint-64 inputs for exhaustive long BFP rounding mode tests
                                                     577 *
00000608  FFFFFFFF FFFFFE00                         578 LINTRMIN DC    XL8'FFFFFFFFFFFFFE00'  Rounds nearest up
00000610  FFFFFFFF FFFFFC00                         579          DC    XL8'FFFFFFFFFFFFFC00'  Tie
00000618  FFFFFFFF FFFFFA00                         580          DC    XL8'FFFFFFFFFFFFFA00'  Rounds nearest down
                            00000018  00000001       581 LINTRMCT EQU   *-LINTRMIN   Count of integers for rounding mode tests
```

```
 LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                    583 ********************************************************************************
                                                    584 *                    ACTUAL results saved here
                                                    585 ********************************************************************************
                                                    586 *
                                                    587 *                 Locations for ACTUAL results
                                                    588 *
                                                    589 *
                          00001000  00000001        590 SBFPOUT  EQU   STRTLABL+X'1000'    Short BFP values from uint-64
                                                    591 *                                 ..6 pairs used, room for 16 pairs
                          00001100  00000001        592 SBFPFLGS EQU   STRTLABL+X'1100'    FPCR flags and DXC from short BFP
                                                    593 *                                 ..6 pairs used, room for 32 pairs
                          00001200  00000001        594 SBFPRMO  EQU   STRTLABL+X'1200'    Short BFP rounding mode results
                                                    595 *                                 ..2 sets used, room for 16 sets
                          00001500  00000001        596 SBFPRMOF EQU   STRTLABL+X'1500'    Short BFP rounding mode FPCR
                                                    597 *                                 ..2 sets used, room for 16+ sets
                                                    598 *                                 ..16 sets ends x'2A00'
                                                    599 *
                          00002000  00000001        600 LBFPOUT  EQU   STRTLABL+X'2000'    Long BFP values from uint-64
                                                    601 *                                 ..6 pairs used, room for 16 pairs
                          00002100  00000001        602 LBFPFLGS EQU   STRTLABL+X'2100'    FPCR flags and DXC from long BFP
                                                    603 *                                 ..6 pairs used, room for 32 pairs
                          00002200  00000001        604 LBFPRMO  EQU   STRTLABL+X'2200'    Long BFP rounding mode results
                                                    605 *                                 ..2 sets used, room for 16 sets
                          00002700  00000001        606 LBFPRMOF EQU   STRTLABL+X'2700'    Long BFP rounding mode FPCR
                                                    607 *                                 ..2 sets used, room for 16+ sets
                                                    608 *                                 ..16 sets ends x'2A00'
                                                    609 *
                          00003000  00000001        610 XBFPOUT  EQU   STRTLABL+X'3000'    Extended BFP values from uint-64
                                                    611 *                                 ..6 pairs used, room for 16 pairs
                          00003200  00000001        612 XBFPFLGS EQU   STRTLABL+X'3200'    Extended BFP rounding mode FPCR
                                                    613 *                                 ..6 pairs used, room for 32 pairs
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    615 ********************************************************************************
                                                    616 *                      EXPECTED results
                                                    617 ********************************************************************************
                                                    618 *
00000620                      00000620  00004000    619          ORG   STRTLABL+X'4000'   (past end of actual results)
                                                    620 *
                              00004000  00000001    621 SBFPOUT_GOOD EQU *
00004000  C3C5D3C7 C2D94099                         622  DC CL48'CELGBR result pairs 1-2'
00004030  3F800000 3F800000                         623  DC XL16'3F8000003F800004000000040000000'
00004040  C3C5D3C7 C2D94099                         624  DC CL48'CELGBR result pairs 3-4'
00004070  40800000 40800000                         625  DC XL16'40800000408000005F7FFFFF5F7FFFFF'
00004080  C3C5D3C7 C2D94099                         626  DC CL48'CELGBR result pairs 5-6'
000040B0  5F800000 5F800000                         627  DC XL16'5F8000005F8000005F8000005F800000'
                              00000003  00000001    628 SBFPOUT_NUM EQU (*-SBFPOUT_GOOD)/64
                                                    629 *
                                                    630 *
                              000040C0  00000001    631 SBFPFLGS_GOOD EQU *
000040C0  C3C5D3C7 C2D940C6                         632  DC CL48'CELGBR FPC pairs 1-2'
000040F0  00000000 F8000000                         633  DC XL16'00000000F8000000000000F8000000'
00004100  C3C5D3C7 C2D940C6                         634  DC CL48'CELGBR FPC pairs 3-4'
00004130  00000000 F8000000                         635  DC XL16'00000000F8000000000000F8000000'
00004140  C3C5D3C7 C2D940C6                         636  DC CL48'CELGBR FPC pairs 5-6'
00004170  00080000 F8000C00                         637  DC XL16'00080000F8000C0000080000F8000C00'
                              00000003  00000001    638 SBFPFLGS_NUM EQU (*-SBFPFLGS_GOOD)/64
                                                    639 *
                                                    640 *
                              00004180  00000001    641 SBFPRMO_GOOD EQU *
00004180  C3C5D3C7 C2D9404E                         642  DC CL48'CELGBR + result FPC modes 1-3, 7'
000041B0  5F7FFFFF 5F800000                         643  DC XL16'5F7FFFFF5F8000005F7FFFFF5F7FFFFF'
000041C0  C3C5D3C7 C2D9404E                         644  DC CL48'CELGBR + result M3 modes 1, 3-5'
000041F0  5F800000 5F7FFFFF                         645  DC XL16'5F8000005F7FFFFF5F8000005F7FFFFF'
00004200  C3C5D3C7 C2D9404E                         646  DC CL48'CELGBR + result M3 modes 6, 7'
00004230  5F800000 5F7FFFFF                         647  DC XL16'5F8000005F7FFFFF0000000000000000'
00004240  C3C5D3C7 C2D94060                         648  DC CL48'CELGBR - result FPC modes 1-3, 7'
00004270  5F7FFFFF 5F800000                         649  DC XL16'5F7FFFFF5F8000005F7FFFFF5F7FFFFF'
00004280  C3C5D3C7 C2D94060                         650  DC CL48'CELGBR - result M3 modes 1, 3-5'
000042B0  5F800000 5F7FFFFF                         651  DC XL16'5F8000005F7FFFFF5F8000005F7FFFFF'
000042C0  C3C5D3C7 C2D94060                         652  DC CL48'CELGBR - result M3 modes 6, 7'
000042F0  5F800000 5F7FFFFF                         653  DC XL16'5F8000005F7FFFFF0000000000000000'
00004300  C3C5D3C7 C2D94060                         654  DC CL48'CELGBR - result FPC modes 1-3, 7'
00004330  5F7FFFFF 5F800000                         655  DC XL16'5F7FFFFF5F8000005F7FFFFF5F7FFFFF'
00004340  C3C5D3C7 C2D94060                         656  DC CL48'CELGBR - result M3 modes 1, 3-5'
00004370  5F7FFFFF 5F7FFFFF                         657  DC XL16'5F7FFFFF5F7FFFFF5F7FFFFF5F7FFFFF'
00004380  C3C5D3C7 C2D94060                         658  DC CL48'CELGBR - result M3 modes 6, 7'
000043B0  5F800000 5F7FFFFF                         659  DC XL16'5F8000005F7FFFFF0000000000000000'
                              00000009  00000001    660 SBFPRMO_NUM EQU (*-SBFPRMO_GOOD)/64
                                                    661 *
                                                    662 *
                              000043C0  00000001    663 SBFPRMOF_GOOD EQU *
000043C0  C3C5D3C7 C2D9404E                         664  DC CL48'CELGBR + FPC modes 1-3, 7 FPCR'
000043F0  00000001 00000002                         665  DC XL16'00000001000000020000000300000007'
00004400  C3C5D3C7 C2D9404E                         666  DC CL48'CELGBR + M3 modes 1, 3-5 FPCR'
00004430  00080000 00080000                         667  DC XL16'00080000000800000008000000080000'
00004440  C3C5D3C7 C2D9404E                         668  DC CL48'CELGBR + M3 modes 6, 7 FPCR'
00004470  00080000 00080000                         669  DC XL16'00080000000800000000000000000000'
00004480  C3C5D3C7 C2D94060                         670  DC CL48'CELGBR - FPC modes 1-3, 7 FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000044B0  00000001 00000002                          671  DC XL16'00000001000000020000000300000007'
000044C0  C3C5D3C7 C2D94060                          672  DC CL48'CELGBR - M3 modes 1, 3-5 FPCR'
000044F0  00080000 00080000                          673  DC XL16'00080000000800000008000000080000'
00004500  C3C5D3C7 C2D94060                          674  DC CL48'CELGBR - M3 modes 6, 7 FPCR'
00004530  00080000 00080000                          675  DC XL16'00080000000800000000000000000000'
00004540  C3C5D3C7 C2D94060                          676  DC CL48'CELGBR - FPC modes 1-3, 7 FPCR'
00004570  00000001 00000002                          677  DC XL16'00000001000000020000000300000007'
00004580  C3C5D3C7 C2D94060                          678  DC CL48'CELGBR - M3 modes 1, 3-5 FPCR'
000045B0  00080000 00080000                          679  DC XL16'00080000000800000008000000080000'
000045C0  C3C5D3C7 C2D94060                          680  DC CL48'CELGBR - M3 modes 6, 7 FPCR'
000045F0  00080000 00080000                          681  DC XL16'00080000000800000000000000000000'
                            00000009  00000001     682 SBFPRMOF_NUM EQU (*-SBFPRMOF_GOOD)/64
                                                     683 *
                                                     684 *
                            00004600  00000001     685 LBFPOUT_GOOD EQU *
00004600  C3C4D3C7 C2D94099                          686  DC CL48'CDLGBR result pair 1'
00004630  3FF00000 00000000                          687  DC XL16'3FF00000000000003FF0000000000000'
00004640  C3C4D3C7 C2D94099                          688  DC CL48'CDLGBR result pair 2'
00004670  40000000 00000000                          689  DC XL16'40000000000000004000000000000000'
00004680  C3C4D3C7 C2D94099                          690  DC CL48'CDLGBR result pair 3'
000046B0  40100000 00000000                          691  DC XL16'40100000000000004010000000000000'
000046C0  C3C4D3C7 C2D94099                          692  DC CL48'CDLGBR result pair 4'
000046F0  43EFFFFF E0000000                          693  DC XL16'43EFFFFFE000000043EFFFFFE0000000'
00004700  C3C4D3C7 C2D94099                          694  DC CL48'CDLGBR result pair 5'
00004730  43EFFFFF FFFFFFFF                          695  DC XL16'43EFFFFFFFFFFFFF43EFFFFFFFFFFFFF'
00004740  C3C4D3C7 C2D94099                          696  DC CL48'CDLGBR result pair 6'
00004770  43F00000 00000000                          697  DC XL16'43F00000000000043F0000000000000'
                            00000006  00000001     698 LBFPOUT_NUM EQU (*-LBFPOUT_GOOD)/64
                                                     699 *
                                                     700 *
                            00004780  00000001     701 LBFPFLGS_GOOD EQU *
00004780  C3C4D3C7 C2D940C6                          702  DC CL48'CDLGBR FPC pairs 1-2'
000047B0  00000000 F8000000                          703  DC XL16'00000000F800000000000000F8000000'
000047C0  C3C4D3C7 C2D940C6                          704  DC CL48'CDLGBR FPC pairs 3-4'
000047F0  00000000 F8000000                          705  DC XL16'00000000F800000000000000F8000000'
00004800  C3C4D3C7 C2D940C6                          706  DC CL48'CDLGBR FPC pairs 5-6'
00004830  00000000 F8000000                          707  DC XL16'00000000F800000000080000F8000C00'
                            00000003  00000001     708 LBFPFLGS_NUM EQU (*-LBFPFLGS_GOOD)/64
                                                     709 *
                                                     710 *
                            00004840  00000001     711 LBFPRMO_GOOD EQU *
00004840  C3C4D3C7 C2D9404E                          712  DC CL48'CDLGBR + FPC modes 1, 2'
00004870  43EFFFFF FFFFFFFF                          713  DC XL16'43EFFFFFFFFFFFF43F0000000000000'
00004880  C3C4D3C7 C2D9404E                          714  DC CL48'CDLGBR + FPC modes 3, 7'
000048B0  43EFFFFF FFFFFFFF                          715  DC XL16'43EFFFFFFFFFFFF43EFFFFFFFFFFFF'
000048C0  C3C4D3C7 C2D9404E                          716  DC CL48'CDLGBR + M3 modes 1, 3'
000048F0  43F00000 00000000                          717  DC XL16'43F00000000000043EFFFFFFFFFFFF'
00004900  C3C4D3C7 C2D9404E                          718  DC CL48'CDLGBR + M3 modes 4, 5'
00004930  43F00000 00000000                          719  DC XL16'43F00000000000043EFFFFFFFFFFFF'
00004940  C3C4D3C7 C2D9404E                          720  DC CL48'CDLGBR + M3 modes 6, 7'
00004970  43F00000 00000000                          721  DC XL16'43F00000000000043EFFFFFFFFFFFF'
00004980  C3C4D3C7 C2D94060                          722  DC CL48'CDLGBR - FPC modes 1, 2'
000049B0  43EFFFFF FFFFFFFF                          723  DC XL16'43EFFFFFFFFFFFF43F0000000000000'
000049C0  C3C4D3C7 C2D94060                          724  DC CL48'CDLGBR - FPC modes 3, 7'
000049F0  43EFFFFF FFFFFFFF                          725  DC XL16'43EFFFFFFFFFFFF43EFFFFFFFFFFFF'
00004A00  C3C4D3C7 C2D94060                          726  DC CL48'CDLGBR - M3 modes 1, 3'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00004A30  43F00000 00000000                          727  DC XL16'43F000000000000043EFFFFFFFFFFFFF'
00004A40  C3C4D3C7 C2D94060                          728  DC CL48'CDLGBR - M3 modes 4, 5'
00004A70  43F00000 00000000                          729  DC XL16'43F000000000000043EFFFFFFFFFFFFF'
00004A80  C3C4D3C7 C2D94060                          730  DC CL48'CDLGBR - M3 modes 6, 7'
00004AB0  43F00000 00000000                          731  DC XL16'43F000000000000043EFFFFFFFFFFFFF'
00004AC0  C3C4D3C7 C2D94060                          732  DC CL48'CDLGBR - FPC modes 1, 2'
00004AF0  43EFFFFF FFFFFFFF                          733  DC XL16'43EFFFFFFFFFFFF43F0000000000000'
00004B00  C3C4D3C7 C2D94060                          734  DC CL48'CDLGBR - FPC modes 3, 7'
00004B30  43EFFFFF FFFFFFFF                          735  DC XL16'43EFFFFFFFFFFFF43EFFFFFFFFFFFF'
00004B40  C3C4D3C7 C2D94060                          736  DC CL48'CDLGBR - M3 modes 1, 3'
00004B70  43EFFFFF FFFFFFFF                          737  DC XL16'43EFFFFFFFFFFFF43EFFFFFFFFFFFF'
00004B80  C3C4D3C7 C2D94060                          738  DC CL48'CDLGBR - M3 modes 4, 5'
00004BB0  43EFFFFF FFFFFFFF                          739  DC XL16'43EFFFFFFFFFFFF43EFFFFFFFFFFFF'
00004BC0  C3C4D3C7 C2D94060                          740  DC CL48'CDLGBR - M3 modes 6, 7'
00004BF0  43F00000 00000000                          741  DC XL16'43F000000000000043EFFFFFFFFFFFFF'
                            0000000F  00000001       742 LBFPRMO_NUM EQU (*-LBFPRMO_GOOD)/64
                                                     743 *
                                                     744 *
                            00004C00  00000001       745 LBFPRMOF_GOOD EQU *
00004C00  C3C4D3C7 C2D9404E                          746  DC CL48'CDLGBR + FPC modes 1-3, 7 FPCR'
00004C30  00000001 00000002                          747  DC XL16'00000001000000020000000300000007'
00004C40  C3C4D3C7 C2D9404E                          748  DC CL48'CDLGBR + M3 modes 1, 3-5 FPCR'
00004C70  00080000 00080000                          749  DC XL16'00080000000800000008000000080000'
00004C80  C3C4D3C7 C2D9404E                          750  DC CL48'CDLGBR + M3 modes 6, 7 FPCR'
00004CB0  00080000 00080000                          751  DC XL16'00080000000800000000000000000000'
00004CC0  C3C4D3C7 C2D94060                          752  DC CL48'CDLGBR - FPC modes 1-3, 7 FPCR'
00004CF0  00000001 00000002                          753  DC XL16'00000001000000020000000300000007'
00004D00  C3C4D3C7 C2D94060                          754  DC CL48'CDLGBR - M3 modes 1, 3-5 FPCR'
00004D30  00080000 00080000                          755  DC XL16'00080000000800000008000000080000'
00004D40  C3C4D3C7 C2D94060                          756  DC CL48'CDLGBR - M3 modes 6, 7 FPCR'
00004D70  00080000 00080000                          757  DC XL16'00080000000800000000000000000000'
00004D80  C3C4D3C7 C2D94060                          758  DC CL48'CDLGBR - FPC modes 1-3, 7 FPCR'
00004DB0  00000001 00000002                          759  DC XL16'00000001000000020000000300000007'
00004DC0  C3C4D3C7 C2D94060                          760  DC CL48'CDLGBR - M3 modes 1, 3-5 FPCR'
00004DF0  00080000 00080000                          761  DC XL16'00080000000800000008000000080000'
00004E00  C3C4D3C7 C2D94060                          762  DC CL48'CDLGBR - M3 modes 6, 7 FPCR'
00004E30  00080000 00080000                          763  DC XL16'00080000000800000000000000000000'
                            00000009  00000001       764 LBFPRMOF_NUM EQU (*-LBFPRMOF_GOOD)/64
                                                     765 *
                                                     766 *
                            00004E40  00000001       767 XBFPOUT_GOOD EQU *
00004E40  C3E7D3C7 C2D94099                          768  DC CL48'CXLGBR result 1a'
00004E70  3FFF0000 00000000                          769  DC XL16'3FFF0000000000000000000000000000'
00004E80  C3E7D3C7 C2D94099                          770  DC CL48'CXLGBR result 1b'
00004EB0  3FFF0000 00000000                          771  DC XL16'3FFF0000000000000000000000000000'
00004EC0  C3E7D3C7 C2D94099                          772  DC CL48'CXLGBR result 2a'
00004EF0  40000000 00000000                          773  DC XL16'40000000000000000000000000000000'
00004F00  C3E7D3C7 C2D94099                          774  DC CL48'CXLGBR result 2b'
00004F30  40000000 00000000                          775  DC XL16'40000000000000000000000000000000'
00004F40  C3E7D3C7 C2D94099                          776  DC CL48'CXLGBR result 3a'
00004F70  40010000 00000000                          777  DC XL16'40010000000000000000000000000000'
00004F80  C3E7D3C7 C2D94099                          778  DC CL48'CXLGBR result 3b'
00004FB0  40010000 00000000                          779  DC XL16'40010000000000000000000000000000'
00004FC0  C3E7D3C7 C2D94099                          780  DC CL48'CXLGBR result 4a'
00004FF0  403EFFFF FE000000                          781  DC XL16'403EFFFFFE000000000000000000000'
00005000  C3E7D3C7 C2D94099                          782  DC CL48'CXLGBR result 4b'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00005030  403EFFFF FE000000                          783  DC XL16'403EFFFFFE00000000000000000000'
00005040  C3E7D3C7 C2D94099                          784  DC CL48'CXLGBR result 5a'
00005070  403EFFFF FFFFFFFF                          785  DC XL16'403EFFFFFFFFFFFF000000000000000'
00005080  C3E7D3C7 C2D94099                          786  DC CL48'CXLGBR result 5b'
000050B0  403EFFFF FFFFFFFF                          787  DC XL16'403EFFFFFFFFFFFF000000000000000'
000050C0  C3E7D3C7 C2D94099                          788  DC CL48'CXLGBR result 6a'
000050F0  403EFFFF FFFFFFFF                          789  DC XL16'403EFFFFFFFFFFFFFE000000000000'
00005100  C3E7D3C7 C2D94099                          790  DC CL48'CXLGBR result 6b'
00005130  403EFFFF FFFFFFFF                          791  DC XL16'403EFFFFFFFFFFFFFE000000000000'
                            0000000C  00000001    792 XBFPOUT_NUM EQU (*-XBFPOUT_GOOD)/64
                                                  793 *
                                                  794 *
                            00005140  00000001    795 XBFPFLGS_GOOD EQU *
00005140  C3E7D3C7 C2D940C6                          796  DC CL48'CXLGBR FPC pairs 1-2'
00005170  00000000 F8000000                          797  DC XL16'00000000F800000000000000F8000000'
00005180  C3E7D3C7 C2D940C6                          798  DC CL48'CXLGBR FPC pairs 3-4'
000051B0  00000000 F8000000                          799  DC XL16'00000000F800000000000000F8000000'
000051C0  C3E7D3C7 C2D940C6                          800  DC CL48'CXLGBR FPC pairs 5-6'
000051F0  00000000 F8000000                          801  DC XL16'00000000F800000000000000F8000000'
                            00000003  00000001    802 XBFPFLGS_NUM EQU (*-XBFPFLGS_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00005200                                          804 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                  806 ********************************************************************
                                                  807 *                 REPORT UNEXPECTED PROGRAM CHECK
                                                  808 ********************************************************************

00005200                                          810 PGMCK    DS    0H
00005200  F342 C072 F08E      00005272  0000008E  811          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
00005206  926B C076                     00005276  812          MVI   PGMCOMMA,C','
0000520A  DC03 C072 C178      00005272  00005378  813          TR    PROGCODE,HEXTRTAB

00005210  F384 C07C F150      0000527C  00000150  815          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
00005216  9240 C084                     00005284  816          MVI   PGMPSW+(0*9)+8,C' '
0000521A  DC07 C07C C178      0000527C  00005378  817          TR    PGMPSW+(0*9)(8),HEXTRTAB

00005220  F384 C085 F154      00005285  00000154  819          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
00005226  9240 C08D                     0000528D  820          MVI   PGMPSW+(1*9)+8,C' '
0000522A  DC07 C085 C178      00005285  00005378  821          TR    PGMPSW+(1*9)(8),HEXTRTAB

00005230  F384 C08E F158      0000528E  00000158  823          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
00005236  9240 C096                     00005296  824          MVI   PGMPSW+(2*9)+8,C' '
0000523A  DC07 C08E C178      0000528E  00005378  825          TR    PGMPSW+(2*9)(8),HEXTRTAB

00005240  F384 C097 F15C      00005297  0000015C  827          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
00005246  9240 C09F                     0000529F  828          MVI   PGMPSW+(3*9)+8,C' '
0000524A  DC07 C097 C178      00005297  00005378  829          TR    PGMPSW+(3*9)(8),HEXTRTAB

00005250  4100 0042                     00000042  831          LA    R0,L'PROGMSG     R0 <== length of message
00005254  4110 C05E                     0000525E  832          LA    R1,PROGMSG       R1 --> the message text itself
00005258  4520 C27A                     0000547A  833          BAL   R2,MSG           Go display this message
                                                  834
0000525C  07FD                                    835          BR    R13              Return to caller

0000525E                                          837 PROGMSG  DS   0CL66
0000525E  D7D9D6C7 D9C1D440                       838          DC    CL20'PROGRAM CHECK! CODE '
00005272  88888888                                839 PROGCODE DC    CL4'hhhh'
00005276  6B                                      840 PGMCOMMA DC    CL1','
00005277  40D7E2E6 40                             841          DC    CL5' PSW '
0000527C  88888888 88888888                       842 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                      844 ************************************************************************
                                                      845 *                      VERIFICATION ROUTINE
                                                      846 ************************************************************************

000052A0                                              848 VERISUB  DS    0H
                                                      849 *
                                                      850 **       Loop through the VERIFY TABLE...
                                                      851 *

000052A0  4110 C32C                 0000552C          853          LA    R1,VERIFTAB      R1 --> Verify table
000052A4  4120 000A                 0000000A          854          LA    R2,VERIFLEN      R2 <== Number of entries
000052A8  0D30                                        855          BASR  R3,0             Set top of loop

000052AA  9846 1000                 00000000          857          LM    R4,R6,0(R1)      Load verify table values
000052AE  4D70 C0C2                 000052C2          858          BAS   R7,VERIFY        Verify results
000052B2  4110 100C                 0000000C          859          LA    R1,12(,R1)       Next verify table entry
000052B6  0623                                        860          BCTR  R2,R3            Loop through verify table

000052B8  9500 C278                 00005478          862          CLI   FAILFLAG,X'00'   Did all tests verify okay?
000052BC  078D                                        863          BER   R13              Yes, return to caller
000052BE  47F0 F238                 00000238          864          B     FAIL             No, load FAILURE disabled wait PSW



                                                      866 *
                                                      867 **       Loop through the ACTUAL / EXPECTED results...
                                                      868 *

000052C2  0D80                                        870 VERIFY   BASR  R8,0             Set top of loop

000052C4  D50F 4000 5030      00000000  00000030      872          CLC   0(16,R4),48(R5)  Actual results == Expected results?
000052CA  4770 C0DA                     000052DA      873          BNE   VERIFAIL         No, show failure
000052CE  4140 4010                     00000010      874 VERINEXT LA    R4,16(,R4)       Next actual result
000052D2  4150 5040                     00000040      875          LA    R5,64(,R5)       Next expected result
000052D6  0668                                        876          BCTR  R6,R8            Loop through results

000052D8  07F7                                        878          BR    R7               Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                   880 ********************************************************************
                                                   881 *                     Report the failure...
                                                   882 ********************************************************************

000052DA  9005 C250                      00005450   884 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
000052DE  92FF C278                      00005478   885          MVI   FAILFLAG,X'FF'   Remember verification failure
                                                   886 *
                                                   887 **       First, show them the description...
                                                   888 *
000052E2  D22F C1E0 5000      000053E0   00000000   889          MVC   FAILDESC,0(R5)   Save results/test description
000052E8  4100 0044                      00000044   890          LA    R0,L'FAILMSG1    R0 <== length of message
000052EC  4110 C1CC                      000053CC   891          LA    R1,FAILMSG1      R1 --> the message text itself
000052F0  4520 C27A                      0000547A   892          BAL   R2,MSG           Go display this message
                                                   893 *
                                                   894 **       Save address of actual and expected results
                                                   895 *
000052F4  5040 C24C                      0000544C   896          ST    R4,AACTUAL       Save A(actual results)
000052F8  4150 5030                      00000030   897          LA    R5,48(,R5)       R5 ==> expected results
000052FC  5050 C248                      00005448   898          ST    R5,AEXPECT       Save A(expected results)
                                                   899 *
                                                   900 **       Format and show them the EXPECTED ("Want") results...
                                                   901 *
00005300  D205 C210 C3A8      00005410   000055A8   902          MVC   WANTGOT,=CL6'Want: '
00005306  F384 C216 C248      00005416   00005448   903          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
0000530C  9240 C21E                      0000541E   904          MVI   BLANKEQ,C' '
00005310  DC07 C216 C178      00005416   00005378   905          TR    FAILADR,HEXTRTAB

00005316  F384 C221 5000      00005421   00000000   907          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
0000531C  9240 C229                      00005429   908          MVI   FAILVALS+(0*9)+8,C' '
00005320  DC07 C221 C178      00005421   00005378   909          TR    FAILVALS+(0*9)(8),HEXTRTAB

00005326  F384 C22A 5004      0000542A   00000004   911          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
0000532C  9240 C232                      00005432   912          MVI   FAILVALS+(1*9)+8,C' '
00005330  DC07 C22A C178      0000542A   00005378   913          TR    FAILVALS+(1*9)(8),HEXTRTAB

00005336  F384 C233 5008      00005433   00000008   915          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
0000533C  9240 C23B                      0000543B   916          MVI   FAILVALS+(2*9)+8,C' '
00005340  DC07 C233 C178      00005433   00005378   917          TR    FAILVALS+(2*9)(8),HEXTRTAB

00005346  F384 C23C 500C      0000543C   0000000C   919          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
0000534C  9240 C244                      00005444   920          MVI   FAILVALS+(3*9)+8,C' '
00005350  DC07 C23C C178      0000543C   00005378   921          TR    FAILVALS+(3*9)(8),HEXTRTAB

00005356  4100 0035                      00000035   923          LA    R0,L'FAILMSG2    R0 <== length of message
0000535A  4110 C210                      00005410   924          LA    R1,FAILMSG2      R1 --> the message text itself
0000535E  4520 C27A                      0000547A   925          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    927 *
                                                    928 **       Format and show them the ACTUAL ("Got") results...
                                                    929 *
00005362  D205 C210 C3AE      00005410  000055AE    930          MVC   WANTGOT,=CL6'Got:  '
00005368  F384 C216 C24C      00005416  0000544C    931          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
0000536E  9240 C21E                     0000541E    932          MVI   BLANKEQ,C' '
00005372  DC07 C216 C178      00005416  00005378    933          TR    FAILADR,HEXTRTAB

00005378  F384 C221 4000      00005421  00000000    935          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
0000537E  9240 C229                     00005429    936          MVI   FAILVALS+(0*9)+8,C' '
00005382  DC07 C221 C178      00005421  00005378    937          TR    FAILVALS+(0*9)(8),HEXTRTAB

00005388  F384 C22A 4004      0000542A  00000004    939          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
0000538E  9240 C232                     00005432    940          MVI   FAILVALS+(1*9)+8,C' '
00005392  DC07 C22A C178      0000542A  00005378    941          TR    FAILVALS+(1*9)(8),HEXTRTAB

00005398  F384 C233 4008      00005433  00000008    943          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
0000539E  9240 C23B                     0000543B    944          MVI   FAILVALS+(2*9)+8,C' '
000053A2  DC07 C233 C178      00005433  00005378    945          TR    FAILVALS+(2*9)(8),HEXTRTAB

000053A8  F384 C23C 400C      0000543C  0000000C    947          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
000053AE  9240 C244                     00005444    948          MVI   FAILVALS+(3*9)+8,C' '
000053B2  DC07 C23C C178      0000543C  00005378    949          TR    FAILVALS+(3*9)(8),HEXTRTAB

000053B8  4100 0035                     00000035    951          LA    R0,L'FAILMSG2    R0 <== length of message
000053BC  4110 C210                     00005410    952          LA    R1,FAILMSG2      R1 --> the message text itself
000053C0  4520 C27A                     0000547A    953          BAL   R2,MSG           Go display this message

000053C4  9805 C250                     00005450    955          LM    R0,R5,SAVER0R5   Restore registers
000053C8  47F0 C0CE                     000052CE    956          B     VERINEXT         Continue with verification...



000053CC                                            958 FAILMSG1 DS    0CL68
000053CC  C3D6D4D7 C1D9C9E2                         959          DC    CL20'COMPARISON FAILURE! '
000053E0  4D8485A2 83998997                         960 FAILDESC DC    CL48'(description)'


00005410                                            962 FAILMSG2 DS    0CL53
00005410  40404040 4040                             963 WANTGOT  DC    CL6' '           'Want: ' -or- 'Got:  '
00005416  C1C1C1C1 C1C1C1C1                         964 FAILADR  DC    CL8'AAAAAAAA'
0000541E  407E40                                    965 BLANKEQ  DC    CL3' = '
00005421  88888888 88888888                         966 FAILVALS DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '

00005448  00000000                                  968 AEXPECT  DC    F'0'             ==> Expected ("Want") results
0000544C  00000000                                  969 AACTUAL  DC    F'0'             ==> Actual ("Got") results
00005450  00000000 00000000                         970 SAVER0R5 DC    6F'0'            Registers R0 - R5 save area
00005468  F0F1F2F3 F4F5F6F7                         971 CHARHEX  DC    CL16'0123456789ABCDEF'
                            00005378  00000010      972 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
00005478  00                                        973 FAILFLAG DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                            975 ********************************************************************************
                                            976 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                            977 ********************************************************************************

0000547A  4900 C3A4                 000055A4   979 MSG      CH    R0,=H'0'                Do we even HAVE a message?
0000547E  07D2                              980          BNHR  R2                      No, ignore

00005480  9002 C2B0                 000054B0   982          STM   R0,R2,MSGSAVE           Save registers

00005484  4900 C3A6                 000055A6   984          CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00005488  47D0 C290                 00005490   985          BNH   MSGOK                   Yes, continue
0000548C  4100 005F                 0000005F   986          LA    R0,L'MSGMSG             No, set to maximum

00005490  1820                              988 MSGOK    LR    R2,R0                   Copy length to work register
00005492  0620                              989          BCTR  R2,0                    Minus-1 for execute
00005494  4420 C2BC                 000054BC   990          EX    R2,MSGMVC               Copy message to O/P buffer

00005498  4120 200A                 0000000A   992          LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
0000549C  4110 C2C2                 000054C2   993          LA    R1,MSGCMD               Point to true command

000054A0  83120008                          995          DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
000054A4  4780 C2AA                 000054AA   996          BZ    MSGRET                  Return if successful
000054A8  0000                              997          DC    H'0'                    CRASH for debugging purposes

000054AA  9802 C2B0                 000054B0   999 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
000054AE  07F2                             1000          BR    R2                      Return to caller
```

```
000054B0  00000000 00000000                1002 MSGSAVE  DC    3F'0'                   Registers save area
000054BC  D200 C2CB 1000      000054CB  00000000  1003 MSGMVC   MVC   MSGMSG(0),0(R1)       Executed instruction

000054C2  D4E2C7D5 D6C8405C                1005 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
000054CB  40404040 40404040                1006 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1008 ********************************************************************************
                                                  1009 *                            VERIFY TABLE
                                                  1010 ********************************************************************************
                                                  1011 *
                                                  1012 *        A(actual results), A(expected results), A(#of results)
                                                  1013 *
                                                  1014 ********************************************************************************

0000552C                                          1016 VERIFTAB DC    0F'0'
0000552C  00001000                                1017          DC    A(SBFPOUT)
00005530  00004000                                1018          DC    A(SBFPOUT_GOOD)
00005534  00000003                                1019          DC    A(SBFPOUT_NUM)
                                                  1020 *
00005538  00001100                                1021          DC    A(SBFPFLGS)
0000553C  000040C0                                1022          DC    A(SBFPFLGS_GOOD)
00005540  00000003                                1023          DC    A(SBFPFLGS_NUM)
                                                  1024 *
00005544  00001200                                1025          DC    A(SBFPRMO)
00005548  00004180                                1026          DC    A(SBFPRMO_GOOD)
0000554C  00000009                                1027          DC    A(SBFPRMO_NUM)
                                                  1028 *
00005550  00001500                                1029          DC    A(SBFPRMOF)
00005554  000043C0                                1030          DC    A(SBFPRMOF_GOOD)
00005558  00000009                                1031          DC    A(SBFPRMOF_NUM)
                                                  1032 *
0000555C  00002000                                1033          DC    A(LBFPOUT)
00005560  00004600                                1034          DC    A(LBFPOUT_GOOD)
00005564  00000006                                1035          DC    A(LBFPOUT_NUM)
                                                  1036 *
00005568  00002100                                1037          DC    A(LBFPFLGS)
0000556C  00004780                                1038          DC    A(LBFPFLGS_GOOD)
00005570  00000003                                1039          DC    A(LBFPFLGS_NUM)
                                                  1040 *
00005574  00002200                                1041          DC    A(LBFPRMO)
00005578  00004840                                1042          DC    A(LBFPRMO_GOOD)
0000557C  0000000F                                1043          DC    A(LBFPRMO_NUM)
                                                  1044 *
00005580  00002700                                1045          DC    A(LBFPRMOF)
00005584  00004C00                                1046          DC    A(LBFPRMOF_GOOD)
00005588  00000009                                1047          DC    A(LBFPRMOF_NUM)
                                                  1048 *
0000558C  00003000                                1049          DC    A(XBFPOUT)
00005590  00004E40                                1050          DC    A(XBFPOUT_GOOD)
00005594  0000000C                                1051          DC    A(XBFPOUT_NUM)
                                                  1052 *
00005598  00003200                                1053          DC    A(XBFPFLGS)
0000559C  00005140                                1054          DC    A(XBFPFLGS_GOOD)
000055A0  00000003                                1055          DC    A(XBFPFLGS_NUM)
                                                  1056 *
                          0000000A  00000001      1057 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000055A4                                            1059          END
000055A4  0000                                      1060                =H'0'
000055A6  005F                                      1061                =AL2(L'MSGMSG)
000055A8  E68195A3 7A40                             1062                =CL6'Want: '
000055AE  C796A37A 4040                             1063                =CL6'Got:  '
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AACTUAL	F	00544C	4	969	896 931
AEXPECT	F	005448	4	968	898 903
AHELPERS	A	00027C	4	192	182 224
BFPCVTFL	J	000000	21940	108	
BLANKEQ	C	00541E	3	965	904 932
CDLGBR	I	00045C	4	401	210
CDLGBRA	I	0004A0	4	442	212
CELGBR	I	000344	4	280	205
CELGBRA	I	000388	4	321	207
CHARHEX	C	005468	16	971	972
CTLR0	F	0002E8	4	234	200 201 202
CXLGBR	I	000574	4	522	215
EXTDS	F	000314	4	256	214
FAIL	I	000238	4	190	864
FAILADR	C	005416	8	964	903 905 931 933
FAILDESC	C	0053E0	48	960	889
FAILFLAG	X	005478	1	973	862 885
FAILMSG1	C	0053CC	68	958	890 891
FAILMSG2	C	005410	53	962	923 924 951 952
FAILPSW	X	0002D8	8	232	190
FAILVALS	C	005421	36	966	907 908 909 911 912 913 915 916 917 919 920 921 935 936 937 939 940 941 943 944 945 947 948 949
FPCREGNT	X	0002EC	4	235	287 331 337 343 349 355 360 365 370 375 380 408 452 458 464 470 476 481 486 491 496 501 529
FPCREGTR	X	0002F0	4	236	292 413 535
FPR0	U	000000	1	129	
FPR1	U	000001	1	130	
FPR10	U	00000A	1	139	532 538
FPR11	U	00000B	1	140	
FPR12	U	00000C	1	141	
FPR13	U	00000D	1	142	
FPR14	U	00000E	1	143	
FPR15	U	00000F	1	144	
FPR2	U	000002	1	131	
FPR3	U	000003	1	132	
FPR4	U	000004	1	133	
FPR5	U	000005	1	134	
FPR6	U	000006	1	135	
FPR7	U	000007	1	136	
FPR8	U	000008	1	137	288 289 293 294 333 334 339 340 345 346 351 352 356 357 361 362 366 367 371 372 376 377 381 382 409 410 414 415 454 455 460 461 466 467 472 473 477 478 482 483 487 488 492 493 497 498 502 503 530 531 536 537
FPR9	U	000009	1	138	
GOODPSW	X	0002C8	8	231	228
HELPERS	H	005200	2	804	147 192
HEXTRTAB	U	005378	16	972	813 817 821 825 829 905 909 913 917 921 933 937 941 945 949
IMAGE	1	000000	21940	0	
INTCOUNT	U	000030	1	566	245 251 257
INTIN	D	0005C0	8	558	566 246 252 258
LBFPFLGS	U	002100	1	602	254 1037
LBFPFLGS_GOOD	U	004780	1	701	708 1038
LBFPFLGS_NUM	U	000003	1	708	1039
LBFPOUT	U	002000	1	600	253 1033
LBFPOUT_GOOD	U	004600	1	685	698 1034

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
LBFPOUT_NUM	U	000006	1	698	1035
LBFPRMO	U	002200	1	604	269 1041
LBFPRMOF	U	002700	1	606	270 1045
LBFPRMOF_GOOD	U	004C00	1	745	764 1046
LBFPRMOF_NUM	U	000009	1	764	1047
LBFPRMO_GOOD	U	004840	1	711	742 1042
LBFPRMO_NUM	U	00000F	1	742	1043
LINTRMCT	U	000018	1	581	267
LINTRMIN	X	000608	8	578	581 268
LONGS	F	000304	4	250	209
MSG	I	00547A	4	979	833 892 925 953
MSGCMD	C	0054C2	9	1005	992 993
MSGMSG	C	0054CB	95	1006	986 1003 984
MSGMVC	I	0054BC	6	1003	990
MSGOK	I	005490	2	988	985
MSGRET	I	0054AA	4	999	996
MSGSAVE	F	0054B0	4	1002	982 999
PCINTCD	H	00008E	2	160	177 811
PCNOTDTA	I	00020C	4	181	178
PCOLDPSW	U	000150	1	162	179 815 819 823 827
PGMCK	H	005200	2	810	183
PGMCOMMA	C	005276	1	840	812
PGMPSW	C	00527C	36	842	815 816 817 819 820 821 823 824 825 827 828 829
PROGCHK	H	000200	2	176	168
PROGCODE	C	005272	4	839	811 813
PROGMSG	C	00525E	66	837	831 832
PROGPSW	D	000228	8	189	188
R0	U	000000	1	110	181 184 200 202 831 884 890 923 951 955 979 982 984 986 988 999
R1	U	000001	1	111	286 288 293 327 333 339 345 351 356 361 366 371 376 381 407 409 414 448 454 460 466 472 477 482 487 492 497 502 528 530 536 832 853 857 859 891 924 952 993 1003
R10	U	00000A	1	120	204 206 209 211 214 280 283 321 324 401 402 442 443 522 523
R11	U	00000B	1	121	
R12	U	00000C	1	122	147 182 224 284 299 325 388 405 420 446 509 526 543
R13	U	00000D	1	123	183 205 207 210 212 215 225 282 300 323 389 404 421 445 510 525 544 835 863
R14	U	00000E	1	124	186 187 226 227
R15	U	00000F	1	125	146 181 184
R2	U	000002	1	112	280 281 299 321 322 388 401 403 420 442 444 509 522 524 543 833 854 860 892 925 953 980 982 988 989 990 992 999 1000
R3	U	000003	1	113	280 286 296 321 327 385 401 407 417 442 448 506 522 528 540 855 860
R4	U	000004	1	114	857 872 874 896 935 939 943 947
R5	U	000005	1	115	872 875 884 889 897 898 907 911 915 919 955
R6	U	000006	1	116	857 876
R7	U	000007	1	117	283 289 294 297 324 334 340 346 352 357 362 367 372 377 382 386 402 410 415 418 443 455 461 467 473 478 483 488 493 498 503 507 523 531 532 537 538 541 858 878
R8	U	000008	1	118	283 290 295 298 324 335 341 347 353 358 363 368 373 378 383 387 402 411 416 419 443 456 462 468 474 479 484 489 494 499 504 508 523 533 539 542 870 876
R9	U	000009	1	119	
RMLONGS	A	000334	4	267	211

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

RMSHORTS             A    000324       4   262   206
SAVER0R5             F    005450       4   970   884   955
SAVEREGS             F    00023C       4   191   181   184
SBFPFLGS             U    001100       1   592   248  1021
SBFPFLGS_GOOD        U    0040C0       1   631   638  1022
SBFPFLGS_NUM         U    000003       1   638  1023
SBFPOUT              U    001000       1   590   247  1017
SBFPOUT_GOOD         U    004000       1   621   628  1018
SBFPOUT_NUM          U    000003       1   628  1019
SBFPRMO              U    001200       1   594   264  1025
SBFPRMOF             U    001500       1   596   265  1029
SBFPRMOF_GOOD        U    0043C0       1   663   682  1030
SBFPRMOF_NUM         U    000009       1   682  1031
SBFPRMO_GOOD         U    004180       1   641   660  1026
SBFPRMO_NUM          U    000009       1   660  1027
SHORTS               F    0002F4       4   244   204
SINTRMCT             U    000018       1   574   262
SINTRMIN             X    0005F0       8   570   574   263
START                I    000280       4   200   165
STRTLABL             U    000000       1   109   159   162   164   167   175   590   592   594   596   600   602   604   606   610
                                               612   619
VERIFAIL             I    0052DA       4   884   873
VERIFLEN             U    00000A       1  1057   854
VERIFTAB             F    00552C       4  1016  1057   853
VERIFY               I    0052C2       2   870   858
VERINEXT             I    0052CE       4   874   956
VERISUB              H    0052A0       2   848   225
WANTGOT              C    005410       6   963   902   930
XBFPFLGS             U    003200       1   612   260  1053
XBFPFLGS_GOOD        U    005140       1   795   802  1054
XBFPFLGS_NUM         U    000003       1   802  1055
XBFPOUT              U    003000       1   610   259  1049
XBFPOUT_GOOD         U    004E40       1   767   792  1050
XBFPOUT_NUM          U    00000C       1   792  1051
=AL2(L'MSGMSG)       R    0055A6       2  1061   984
=CL6'Got:  '         C    0055AE       6  1063   930
=CL6'Want: '         C    0055A8       6  1062   902
=H'0'                H    0055A4       2  1060   979
```

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 0

Image       IMAGE      21940  0000-55B3  0000-55B3
  Region               21940  0000-55B3  0000-55B3
    CSECT   BFPCVTFL   21940  0000-55B3  0000-55B3
```

```
   STMT                                                    FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-009-cvtfrlog64\bfp-009-cvtfrlog64.asm

** NO ERRORS FOUND **
```