```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                      2 ************************************************************************
                                                      3 *
                                                      4 *Testcase IEEE LOAD LENGTHENED
                                                      5 *  Test case capability includes IEEE exceptions trappable and
                                                      6 *  otherwise. Test results, FPCR flags, and any DXC are saved for all
                                                      7 *  tests.  Load Lengthened 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-017-loadl.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 *   LOAD LENGTHENED (short BFP, RRE)
                                                        67 *   LOAD LENGTHENED (long BFP, RRE)
                                                        68 *   LOAD LENGTHENED (extended BFP, RRE)
                                                        69 *   LOAD LENGTHENED (short BFP, RXE)
                                                        70 *   LOAD LENGTHENED (long BFP, RXE)
                                                        71 *   LOAD LENGTHENED (extended BFP, RXE)
                                                        72 *
                                                        73 * Test data is compiled into this program.  The test script that runs
                                                        74 * this program can provide alternative test data through Hercules R
                                                        75 * commands.
                                                        76 *
                                                        77 * Test Case Order
                                                        78 * 1) Short to log BFP non-finite tests, non-trap and trap
                                                        79 * 2) Short to log BFP finite tests, non-trappable
                                                        80 * 3) Long to extended BFP non-finite tests, non-trap and trap
                                                        81 * 4) Long to extended BFP finite tests, non-trappable
                                                        82 * 5) Short to extended BFP non-finite tests, non-trap and trap
                                                        83 * 6) Short to extended BFP finite tests, non-trappable
                                                        84 *
                                                        85 * Two input test data sets are provided, one each for short and long
                                                        86 *   precision BFP inputs.  The same short BFP inputs are used for
                                                        87 *   short to long testing and short to extended testing.
                                                        88 *
                                                        89 * Also tests the following floating point support instructions
                                                        90 *   LOAD  (Short)
                                                        91 *   LOAD  (Long)
                                                        92 *   LFPC  (Load Floating Point Control Register)
                                                        93 *   STORE (Short)
                                                        94 *   STORE (Long)
                                                        95 *   STFPC (Store Floating Point Control Register)
                                                        96 *
                                                        97 ********************************************************************************
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                          99 *
                                         100 *  Note: for compatibility with the z/CMS test rig, do not change
                                         101 *  or use R11, R14, or R15.  Everything else is fair game.
                                         102 *
                            00000000  00006E0B   103 BFPLDLEN START 0                   Load Lengthened Testing
                            00000000  00000001   104 STRTLABL EQU   *
                            00000000  00000001   105 R0       EQU   0                   Work register for cc extraction
                            00000001  00000001   106 R1       EQU   1                   Available
                            00000002  00000001   107 R2       EQU   2                   Holds count of test input values
                            00000003  00000001   108 R3       EQU   3                   Points to next test input value(s)
                            00000004  00000001   109 R4       EQU   4                   Available
                            00000005  00000001   110 R5       EQU   5                   Available
                            00000006  00000001   111 R6       EQU   6                   Available
                            00000007  00000001   112 R7       EQU   7                   Pointer to next result value(s)
                            00000008  00000001   113 R8       EQU   8                   Pointer to next FPCR result
                            00000009  00000001   114 R9       EQU   9                   Available
                            0000000A  00000001   115 R10      EQU   10                  Pointer to test address list
                            0000000B  00000001   116 R11      EQU   11                  **Reserved for z/CMS test rig
                            0000000C  00000001   117 R12      EQU   12                  Holds number of test cases in set
                            0000000D  00000001   118 R13      EQU   13                  Mainline return address
                            0000000E  00000001   119 R14      EQU   14                  **Return address for z/CMS test rig
                            0000000F  00000001   120 R15      EQU   15                  **Base register on z/CMS or Hyperion
                                         121 *
                                         122 * Floating Point Register equates to keep the cross reference clean
                                         123 *
                            00000000  00000001   124 FPR0     EQU   0
                            00000001  00000001   125 FPR1     EQU   1
                            00000002  00000001   126 FPR2     EQU   2
                            00000003  00000001   127 FPR3     EQU   3
                            00000004  00000001   128 FPR4     EQU   4
                            00000005  00000001   129 FPR5     EQU   5
                            00000006  00000001   130 FPR6     EQU   6
                            00000007  00000001   131 FPR7     EQU   7
                            00000008  00000001   132 FPR8     EQU   8
                            00000009  00000001   133 FPR9     EQU   9
                            0000000A  00000001   134 FPR10    EQU   10
                            0000000B  00000001   135 FPR11    EQU   11
                            0000000C  00000001   136 FPR12    EQU   12
                            0000000D  00000001   137 FPR13    EQU   13
                            0000000E  00000001   138 FPR14    EQU   14
                            0000000F  00000001   139 FPR15    EQU   15
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00000000                      00000000              141          USING *,R15
00000000                      00006A40              142          USING HELPERS,R12
                                                    143 *
                                                    144 * Above works on real iron (R15=0 after sysclear)
                                                    145 * and in z/CMS (R15 points to start of load module)
                                                    146 *


                                                    148 ********************************************************************************
                                                    149 *
                                                    150 * Low core definitions, Restart PSW, and Program Check Routine.
                                                    151 *
                                                    152 ********************************************************************************


00000000                      00000000  0000008E    154          ORG   STRTLABL+X'8E'      Program check interrution code
0000008E  0000                                    155 PCINTCD  DS    H
                                                    156 *
                              00000150  00000001    157 PCOLDPSW EQU   STRTLABL+X'150'     z/Arch Program check old PSW
                                                    158 *
00000090                      00000090  000001A0    159          ORG   STRTLABL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                         160          DC    X'0000000180000000',AD(START)
                                                    161 *
000001B0                      000001B0  000001D0    162          ORG   STRTLABL+X'1D0'     z/Arch Program check NEW PSW
000001D0  00000000 00000000                         163          DC    X'0000000000000000',AD(PROGCHK)
                                                    164 *
                                                    165 * Program check routine.  If Data Exception, continue execution at
                                                    166 * the instruction following the program check.  Otherwise, hard wait.
                                                    167 * No need to collect data.  All interesting DXC stuff is captured
                                                    168 * in the FPCR.
                                                    169 *
000001E0                    000001E0  00000200    170          ORG   STRTLABL+X'200'
00000200                                            171 PROGCHK  DS    0H              Program check occured...
00000200  9507 F08F                   0000008F    172          CLI   PCINTCD+1,X'07'  Data Exception?
00000204  A774 0004                   0000020C    173          JNE   PCNOTDTA        ..no, hardwait (not sure if R15 is ok)
00000208  B2B2 F150                   00000150    174          LPSWE PCOLDPSW        ..yes, resume program execution

0000020C  900F F23C                   0000023C    176 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                   0000027C    177          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                   00006A40    178          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                   0000023C    179          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                      181          LTR   R14,R14        Return address provided?
0000021E  077E                                      182          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                   00000228    183          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                         184 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2E0                   000002E0    185 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                         186 SAVEREGS DC    16F'0'          Registers save area
0000027C  00006A40                                  187 AHELPERS DC    A(HELPERS)      Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    189 ********************************************************************************
                                                    190 *
                                                    191 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                    192 *
                                                    193 ********************************************************************************


00000280                                            195 START    DS    0H
00000280  B600 F2F0                 000002F0        196          STCTL R0,R0,CTLR0   Store CR0 to enable AFP
00000284  9604 F2F1                 000002F1        197          OI    CTLR0+1,X'04' Turn on AFP bit
00000288  B700 F2F0                 000002F0        198          LCTL  R0,R0,CTLR0   Reload updated CR0
                                                    199 *
0000028C  41A0 F2FC                 000002FC        200          LA    R10,SHORTNF   Point to short BFP non-finite test data
00000290  4DD0 F35C                 0000035C        201          BAS   R13,LDEBRNF   Convert short BFP to long BFP
00000294  41A0 F30C                 0000030C        202          LA    R10,SHORTF    Point to short BFP finite test data
00000298  4DD0 F3CA                 000003CA        203          BAS   R13,LDEBRF    Convert short BFP to long BFP
                                                    204 *
0000029C  41A0 F31C                 0000031C        205          LA    R10,LONGNF    Point to long BFP non-finite test data
000002A0  4DD0 F412                 00000412        206          BAS   R13,LXDBRNF   Convert long BFP to integer long BFP
000002A4  41A0 F32C                 0000032C        207          LA    R10,LONGF     Point to long BFP finite test data
000002A8  4DD0 F490                 00000490        208          BAS   R13,LXDBRF    Convert using all rounding mode options
                                                    209 *
000002AC  41A0 F33C                 0000033C        210          LA    R10,XTNDNF    Point to short BFP non-finite test data
000002B0  4DD0 F4DC                 000004DC        211          BAS   R13,LXEBRNF   Convert short BFP to extended BFP
000002B4  41A0 F34C                 0000034C        212          LA    R10,XTNDF     Point to short BFP finite test data
000002B8  4DD0 F55A                 0000055A        213          BAS   R13,LXEBRF    Convert short BFP to extended BFP
                                                    214 *
                                                    215 ********************************************************************************
                                                    216 *                    Verify test results...
                                                    217 ********************************************************************************
                                                    218 *
000002BC  58C0 F27C                 0000027C        219          L     R12,AHELPERS     Get address of helper subroutines
000002C0  4DD0 C0A0                 00006AE0        220          BAS   R13,VERISUB      Go verify results
000002C4  12EE                                      221          LTR   R14,R14          Was return address provided?
000002C6  077E                                      222          BNZR  R14              Yes, return to z/CMS test rig.
000002C8  B2B2 F2D0                 000002D0        223          LPSWE GOODPSW          Load SUCCESS PSW
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000002D0                                          225          DS    0D            Ensure correct alignment for PSW
000002D0  00020000 00000000                       226 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002E0  00020000 00000000                       227 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                  228 *
000002F0  00000000                                229 CTLR0    DS    F
000002F4  00000000                                230 FPCREGNT DC    X'00000000'  FPCR, trap all IEEE exceptions, zero flags
000002F8  F8000000                                231 FPCREGTR DC    X'F8000000'  FPCR, trap no IEEE exceptions, zero flags
                                                  232 *
                                                  233 * Input values parameter list, four fullwords:
                                                  234 *      1) Count,
                                                  235 *      2) Address of inputs,
                                                  236 *      3) Address to place results, and
                                                  237 *      4) Address to place DXC/Flags/cc values.
                                                  238 *
000002FC                                          239 SHORTNF  DS    0F            Short to long BFP non-finite inputs
000002FC  00000008                                240          DC    A(SBFPNFCT)
00000300  000005A8                                241          DC    A(SBFPNFIN)
00000304  00001000                                242          DC    A(SBFPNFOT)
00000308  00001100                                243          DC    A(SBFPNFFL)
                                                  244 *
0000030C                                          245 SHORTF   DS    0F            Short to long BFP finite inputs
0000030C  0000000A                                246          DC    A(SBFPFCT)
00000310  000005C8                                247          DC    A(SBFPFIN)
00000314  00001200                                248          DC    A(SBFPFOT)
00000318  00001300                                249          DC    A(SBFPFOF)
                                                  250 *
0000031C                                          251 LONGNF   DS    0F            Long to extended BFP non-finite inputs
0000031C  00000008                                252          DC    A(LBFPNFCT)
00000320  000005F0                                253          DC    A(LBFPNFIN)
00000324  00002000                                254          DC    A(LBFPNFOT)
00000328  00002200                                255          DC    A(LBFPNFFL)
                                                  256 *
0000032C                                          257 LONGF    DS    0F            Long to extended BFP finite inputs
0000032C  0000000A                                258          DC    A(LBFPFCT)
00000330  00000630                                259          DC    A(LBFPFIN)
00000334  00002300                                260          DC    A(LBFPFOT)
00000338  00002500                                261          DC    A(LBFPFOF)
                                                  262 *
0000033C                                          263 XTNDNF   DS    0F            Short to extended BFP non-finite inputs
0000033C  00000008                                264          DC    A(SBFPNFCT)
00000340  000005A8                                265          DC    A(SBFPNFIN)
00000344  00003000                                266          DC    A(XBFPNFOT)
00000348  00003200                                267          DC    A(XBFPNFFL)
                                                  268 *
0000034C                                          269 XTNDF    DS    0F            Short to extended BFP finite inputs
0000034C  0000000A                                270          DC    A(SBFPFCT)
00000350  000005C8                                271          DC    A(SBFPFIN)
00000354  00003300                                272          DC    A(XBFPFOT)
00000358  00003500                                273          DC    A(XBFPFOF)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    275 ************************************************************************
                                                    276 *
                                                    277 * Load Lengthened Short to Long non-finite tests
                                                    278 *
                                                    279 * Lengthen short BFP inputs to long BFP.  Two pairs of results are
                                                    280 * generated, one pair for RRE and one for RXE.  Each pair consists of
                                                    281 * a result with all exceptions non-trappable and a second with all
                                                    282 * exceptions trappable.   The FPCR is stored for each result.  The
                                                    283 * Condition Code in not changed by Load Lengthened and is not stored.
                                                    284 *
                                                    285 ************************************************************************


0000035C                                            287 LDEBRNF  DS    0H             Convert short BFP inputs to long BFP
0000035C  9823 A000                 00000000       288          LM    R2,R3,0(R10)   Get count and address of test input values
00000360  9878 A008                 00000008       289          LM    R7,R8,8(R10)   Get address of result area and flag area.
00000364  1222                                     290          LTR   R2,R2          Any test cases?
00000366  078D                                     291          BZR   R13            ..No, return to caller
00000368  0DC0                                     292          BASR  R12,0          Set top of loop
                                                    293 *
0000036A  7880 3000                 00000000       294          LE    FPR8,0(,R3)    Get short BFP test value
0000036E  B29D F2F4                 000002F4       295          LFPC  FPCREGNT       Set exceptions non-trappable
00000372  B304 0018                                296          LDEBR FPR1,FPR8      Lengthen short in FPR8 to long in FPR1
00000376  6010 7000                 00000000       297          STD   FPR1,0(,R7)    Store long BFP result
0000037A  B29C 8000                 00000000       298          STFPC 0(R8)          Store resulting FPCR flags and DXC
                                                    299 *
0000037E  B29D F2F8                 000002F8       300          LFPC  FPCREGTR       Set exceptions trappable
00000382  B374 0010                                301          LZER  FPR1           Eliminate any residual results
00000386  B304 0018                                302          LDEBR FPR1,FPR8      Lengthen short in FPR8 to long in FPR1
0000038A  6010 7008                 00000008       303          STD   FPR1,8(,R7)    Store long BFP result
0000038E  B29C 8004                 00000004       304          STFPC 4(R8)          Store resulting FPCR flags and DXC
                                                    305 *
00000392  B29D F2F4                 000002F4       306          LFPC  FPCREGNT       Set exceptions non-trappable
00000396  ED10 3000 0004            00000000       307          LDEB  FPR1,0(,R3)    Lengthen short BFP to long in FPR1
0000039C  6010 7010                 00000010       308          STD   FPR1,16(,R7)   Store long BFP result
000003A0  B29C 8008                 00000008       309          STFPC 8(R8)          Store resulting FPCR flags and DXC
                                                    310 *
000003A4  B29D F2F8                 000002F8       311          LFPC  FPCREGTR       Set exceptions trappable
000003A8  B374 0010                                312          LZER  FPR1           Eliminate any residual results
000003AC  ED10 3000 0004            00000000       313          LDEB  FPR1,0(,R3)    Lengthen short BFP to long in FPR1
000003B2  6010 7018                 00000018       314          STD   FPR1,24(,R7)   Store long BFP result
000003B6  B29C 800C                 0000000C       315          STFPC 12(R8)         Store resulting FPCR flags and DXC
                                                    316 *
000003BA  4130 3004                 00000004       317          LA    R3,4(,R3)      Point to next input value
000003BE  4170 7020                 00000020       318          LA    R7,4*8(,R7)    Point to next long BFP result set
000003C2  4180 8010                 00000010       319          LA    R8,4*4(,R8)    Point to next FPCR result area
000003C6  062C                                     320          BCTR  R2,R12         Convert next input value.
000003C8  07FD                                     321          BR    R13            All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    323 ************************************************************************
                                                    324 *
                                                    325 * Load Lengthened Short to Long finite tests
                                                    326 *
                                                    327 * Convert short BFP to integer BFP using each possible rounding mode.
                                                    328 * Ten test results are generated for each input.  A 48-byte test result
                                                    329 * section is used to keep results sets aligned on a quad-double word.
                                                    330 *
                                                    331 * The first four tests use rounding modes specified in the FPCR with
                                                    332 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    333 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    334 * the last two To get full coverage of that instruction pair.
                                                    335 *
                                                    336 * The next six results use instruction-specified rounding modes.
                                                    337 *
                                                    338 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    339 * prior tests used the default rounding mode.  RNTE is tested
                                                    340 * explicitly as a rounding mode in this section.
                                                    341 *
                                                    342 ************************************************************************

000003CA  9823 A000                 00000000    344 LDEBRF   LM    R2,R3,0(R10)  Get count and address of test input values
000003CE  9878 A008                 00000008    345          LM    R7,R8,8(R10)  Get address of result area and flag area.
000003D2  1222                                  346          LTR   R2,R2         Any test cases?
000003D4  078D                                  347          BZR   R13           ..No, return to caller
000003D6  0DC0                                  348          BASR  R12,0         Set top of loop
                                                    349 *
000003D8  7880 3000                 00000000    350          LE    FPR8,0(,R3)    Get short BFP test value
                                                    351 *
000003DC  B29D F2F4                 000002F4    352          LFPC  FPCREGNT      Set exceptions non-trappable
000003E0  B304 0018                             353          LDEBR FPR1,FPR8     Lengthen short in FPR8 to long in FPR1
000003E4  6010 7000                 00000000    354          STD   FPR1,0(,R7)   Store long BFP result
000003E8  B29C 8000                 00000000    355          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                    356 *
000003EC  B29D F2F4                 000002F4    357          LFPC  FPCREGNT      Set exceptions non-trappable
000003F0  B374 0010                             358          LZER  FPR1          Eliminate any residual results
000003F4  ED10 3000 0004            00000000    359          LDEB  FPR1,0(,R3)   Lengthen short in FPR8 to long in FPR1
000003FA  6010 7008                 00000008    360          STD   FPR1,8(,R7)   Store long BFP result
000003FE  B29C 8004                 00000004    361          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                                    362 *
00000402  4130 3004                 00000004    363          LA    R3,4(,R3)     Point to next input value
00000406  4170 7010                 00000010    364          LA    R7,2*8(,R7)   Point to next long BFP result set
0000040A  4180 8008                 00000008    365          LA    R8,2*4(,R8)   Point to next FPCR result area
0000040E  062C                                  366          BCTR  R2,R12        Convert next input value.
00000410  07FD                                  367          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    369 ************************************************************************
                                                    370 *
                                                    371 * Load Lengthened Long to Extended non-finite tests
                                                    372 *
                                                    373 * Lengthen long BFP inputs to extended BFP.  Two pairs of results are
                                                    374 * generated, one pair for RRE and one for RXE.  Each pair consists of
                                                    375 * a result with all exceptions non-trappable and a second with all
                                                    376 * exceptions trappable.   The FPCR is stored for each result.  The
                                                    377 * Condition Code in not changed by Load Lengthened and is not stored.
                                                    378 *
                                                    379 ************************************************************************


00000412  9823 A000                 00000000   381 LXDBRNF  LM    R2,R3,0(R10)  Get count and address of test input values
00000416  9878 A008                 00000008   382          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000041A  1222                                 383          LTR   R2,R2         Any test cases?
0000041C  078D                                 384          BZR   R13           ..No, return to caller
0000041E  0DC0                                 385          BASR  R12,0         Set top of loop
                                               386 *
00000420  6880 3000                 00000000   387          LD    FPR8,0(,R3)   Get long BFP test value
00000424  B29D F2F4                 000002F4   388          LFPC  FPCREGNT      Set exceptions non-trappable
00000428  B305 0018                            389          LXDBR FPR1,FPR8     Lengthen long in FPR8 to extended in FPR1
0000042C  6010 7000                 00000000   390          STD   FPR1,0(,R7)   Store extended BFP result part 1
00000430  6030 7008                 00000008   391          STD   FPR3,8(,R7)   Store extended BFP result part 2
00000434  B29C 8000                 00000000   392          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                               393 *
00000438  B29D F2F8                 000002F8   394          LFPC  FPCREGTR      Set exceptions trappable
0000043C  B376 0010                            395          LZXR  FPR1          Eliminate any residual results
00000440  B305 0018                            396          LXDBR FPR1,FPR8     Lengthen long in FPR8 to extended in FPR1
00000444  6010 7010                 00000010   397          STD   FPR1,16(,R7)  Store extended BFP result part 1
00000448  6030 7018                 00000018   398          STD   FPR3,24(,R7)  Store extended BFP result part 2
0000044C  B29C 8004                 00000004   399          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                               400 *
00000450  B29D F2F4                 000002F4   401          LFPC  FPCREGNT      Set exceptions non-trappable
00000454  ED10 3000 0005            00000000   402          LXDB  FPR1,0(,R3)   Lengthen long BFP to extended in FPR1-3
0000045A  6010 7020                 00000020   403          STD   FPR1,32(,R7)  Store extended BFP result part 1
0000045E  6030 7028                 00000028   404          STD   FPR3,40(,R7)  Store extended BFP result part 2
00000462  B29C 8008                 00000008   405          STFPC 8(R8)         Store resulting FPCR flags and DXC
                                               406 *
00000466  B29D F2F8                 000002F8   407          LFPC  FPCREGTR      Set exceptions trappable
0000046A  B376 0010                            408          LZXR  FPR1          Eliminate any residual results
0000046E  ED10 3000 0005            00000000   409          LXDB  FPR1,0(,R3)   Lengthen long BFP to extended in FPR1-3
00000474  6010 7030                 00000030   410          STD   FPR1,48(,R7)  Store extended BFP result part 1
00000478  6030 7038                 00000038   411          STD   FPR3,56(,R7)  Store extended BFP result part 2
0000047C  B29C 800C                 0000000C   412          STFPC 12(R8)        Store resulting FPCR flags and DXC
                                               413 *
00000480  4130 3008                 00000008   414          LA    R3,8(,R3)     Point to next input value
00000484  4170 7040                 00000040   415          LA    R7,4*16(,R7)  Point to next extended BFP result set
00000488  4180 8010                 00000010   416          LA    R8,4*4(,R8)   Point to next FPCR result area
0000048C  062C                                 417          BCTR  R2,R12        Convert next input value.
0000048E  07FD                                 418          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                    420 ************************************************************************
                                                    421 *
                                                    422 * Convert long BFP to integers using each possible rounding mode.
                                                    423 * Ten test results are generated for each input.  A 48-byte test result
                                                    424 * section is used to keep results sets aligned on a quad-double word.
                                                    425 *
                                                    426 * The first four tests use rounding modes specified in the FPCR with
                                                    427 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    428 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    429 * the last two To get full coverage of that instruction pair.
                                                    430 *
                                                    431 * The next six results use instruction-specified rounding modes.
                                                    432 *
                                                    433 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    434 * prior tests used the default rounding mode.  RNTE is tested
                                                    435 * explicitly as a rounding mode in this section.
                                                    436 *
                                                    437 ************************************************************************


00000490  9823 A000                 00000000    439 LXDBRF   LM    R2,R3,0(R10)  Get count and address of test input values
00000494  9878 A008                 00000008    440          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000498  1222                                  441          LTR   R2,R2         Any test cases?
0000049A  078D                                  442          BZR   R13           ..No, return to caller
0000049C  0DC0                                  443          BASR  R12,0         Set top of loop
                                                    444 *
0000049E  6880 3000                 00000000    445          LD    FPR8,0(,R3)   Get long BFP test value
000004A2  B29D F2F4                 000002F4    446          LFPC  FPCREGNT      Set exceptions non-trappable
000004A6  B305 0018                             447          LXDBR FPR1,FPR8     Lengthen long in FPR8 to extended in FPR1
000004AA  6010 7000                 00000000    448          STD   FPR1,0(,R7)   Store extended BFP result part 1
000004AE  6030 7008                 00000008    449          STD   FPR3,8(,R7)   Store extended BFP result part 2
000004B2  B29C 8000                 00000000    450          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                    451 *
000004B6  B29D F2F4                 000002F4    452          LFPC  FPCREGNT      Set exceptions non-trappable
000004BA  ED10 3000 0005            00000000    453          LXDB  FPR1,0(,R3)   Lengthen long BFP to extended in FPR1-3
000004C0  6010 7010                 00000010    454          STD   FPR1,16(,R7)  Store extended BFP result part 1
000004C4  6030 7018                 00000018    455          STD   FPR3,24(,R7)  Store extended BFP result part 2
000004C8  B29C 8004                 00000004    456          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                                    457 *
000004CC  4130 3008                 00000008    458          LA    R3,8(,R3)     Point to next input value
000004D0  4170 7020                 00000020    459          LA    R7,2*16(,R7)  Point to next extended BFP result set
000004D4  4180 8008                 00000008    460          LA    R8,2*4(,R8)   Point to next FPCR result area
000004D8  062C                                  461          BCTR  R2,R12        Convert next input value.
000004DA  07FD                                  462          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    464 ************************************************************************
                                                    465 *
                                                    466 * Load Lengthened Short to Extended non-finite tests
                                                    467 *
                                                    468 * Lengthen short BFP inputs to extended BFP.  Two pairs of results are
                                                    469 * generated, one pair for RRE and one for RXE.  Each pair consists of
                                                    470 * a result with all exceptions non-trappable and a second with all
                                                    471 * exceptions trappable.   The FPCR is stored for each result.  The
                                                    472 * Condition Code in not changed by Load Lengthened and is not stored.
                                                    473 *
                                                    474 ************************************************************************


000004DC  9823 A000                 00000000   476 LXEBRNF  LM    R2,R3,0(R10)  Get count and address of test input values
000004E0  9878 A008                 00000008   477          LM    R7,R8,8(R10)  Get address of result area and flag area.
000004E4  1222                                 478          LTR   R2,R2         Any test cases?
000004E6  078D                                 479          BZR   R13           ..No, return to caller
000004E8  0DC0                                 480          BASR  R12,0         Set top of loop
                                               481 *
000004EA  7880 3000                 00000000   482          LE    FPR8,0(,R3)   Get long BFP test value
000004EE  B29D F2F4                 000002F4   483          LFPC  FPCREGNT      Set exceptions non-trappable
000004F2  B306 0018                            484          LXEBR FPR1,FPR8     Lengthen long in FPR8 to extended in FPR1
000004F6  6010 7000                 00000000   485          STD   FPR1,0(,R7)   Store extended BFP result part 1
000004FA  6030 7008                 00000008   486          STD   FPR3,8(,R7)   Store extended BFP result part 2
000004FE  B29C 8000                 00000000   487          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                               488 *
00000502  B29D F2F8                 000002F8   489          LFPC  FPCREGTR      Set exceptions trappable
00000506  B376 0010                            490          LZXR  FPR1          Eliminate any residual results
0000050A  B306 0018                            491          LXEBR FPR1,FPR8     Lengthen long in FPR8 to extended in FPR1
0000050E  6010 7010                 00000010   492          STD   FPR1,16(,R7)  Store extended BFP result part 1
00000512  6030 7018                 00000018   493          STD   FPR3,24(,R7)  Store extended BFP result part 2
00000516  B29C 8004                 00000004   494          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                               495 *
0000051A  B29D F2F4                 000002F4   496          LFPC  FPCREGNT      Set exceptions non-trappable
0000051E  ED10 3000 0006            00000000   497          LXEB  FPR1,0(,R3)   Lengthen long BFP to extended in FPR1-3
00000524  6010 7020                 00000020   498          STD   FPR1,32(,R7)  Store extended BFP result part 1
00000528  6030 7028                 00000028   499          STD   FPR3,40(,R7)  Store extended BFP result part 2
0000052C  B29C 8008                 00000008   500          STFPC 8(R8)         Store resulting FPCR flags and DXC
                                               501 *
00000530  B29D F2F8                 000002F8   502          LFPC  FPCREGTR      Set exceptions trappable
00000534  B376 0010                            503          LZXR  FPR1          Eliminate any residual results
00000538  ED10 3000 0006            00000000   504          LXEB  FPR1,0(,R3)   Lengthen long BFP to extended in FPR1-3
0000053E  6010 7030                 00000030   505          STD   FPR1,48(,R7)  Store extended BFP result part 1
00000542  6030 7038                 00000038   506          STD   FPR3,56(,R7)  Store extended BFP result part 2
00000546  B29C 800C                 0000000C   507          STFPC 12(R8)         Store resulting FPCR flags and DXC
                                               508 *
0000054A  4130 3004                 00000004   509          LA    R3,4(,R3)     Point to next input value
0000054E  4170 7040                 00000040   510          LA    R7,4*16(,R7)  Point to next extended BFP result set
00000552  4180 8010                 00000010   511          LA    R8,4*4(,R8)   Point to next FPCR result area
00000556  062C                                 512          BCTR  R2,R12        Convert next input value.
00000558  07FD                                 513          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    515 ************************************************************************
                                                    516 *
                                                    517 * Convert extended BFP to integers using each possible rounding mode.
                                                    518 * Ten test results are generated for each input.  A 48-byte test result
                                                    519 * section is used to keep results sets aligned on a quad-double word.
                                                    520 *
                                                    521 * The first four tests use rounding modes specified in the FPCR with
                                                    522 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    523 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    524 * the last two To get full coverage of that instruction pair.
                                                    525 *
                                                    526 * The next six results use instruction-specified rounding modes.
                                                    527 *
                                                    528 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    529 * prior tests used the default rounding mode.  RNTE is tested
                                                    530 * explicitly as a rounding mode in this section.
                                                    531 *
                                                    532 ************************************************************************


0000055A  9823 A000                 00000000    534 LXEBRF   LM    R2,R3,0(R10)  Get count and address of test input values
0000055E  9878 A008                 00000008    535          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000562  1222                                  536          LTR   R2,R2         Any test cases?
00000564  078D                                  537          BZR   R13           ..No, return to caller
00000566  0DC0                                  538          BASR  R12,0         Set top of loop
                                                    539 *
00000568  7880 3000                 00000000    540          LE    FPR8,0(,R3)   Get long BFP test value
0000056C  B29D F2F4                 000002F4    541          LFPC  FPCREGNT      Set exceptions non-trappable
00000570  B306 0018                             542          LXEBR FPR1,FPR8     Lengthen long in FPR8 to extended in FPR1
00000574  6010 7000                 00000000    543          STD   FPR1,0(,R7)   Store extended BFP result part 1
00000578  6030 7008                 00000008    544          STD   FPR3,8(,R7)   Store extended BFP result part 2
0000057C  B29C 8000                 00000000    545          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                    546 *
00000580  B29D F2F4                 000002F4    547          LFPC  FPCREGNT      Set exceptions non-trappable
00000584  ED10 3000 0006            00000000    548          LXEB  FPR1,0(,R3)   Lengthen long BFP to extended in FPR1-3
0000058A  6010 7010                 00000010    549          STD   FPR1,16(,R7)  Store extended BFP result part 1
0000058E  6030 7018                 00000018    550          STD   FPR3,24(,R7)  Store extended BFP result part 2
00000592  B29C 8008                 00000008    551          STFPC 8(R8)         Store resulting FPCR flags and DXC
                                                    552 *
00000596  4130 3004                 00000004    553          LA    R3,4(,R3)     Point to next input value
0000059A  4170 7020                 00000020    554          LA    R7,2*16(,R7)  Point to next extended BFP result set
0000059E  4180 8008                 00000008    555          LA    R8,2*4(,R8)   Point to next FPCR result area
000005A2  062C                                  556          BCTR  R2,R12        Convert next input value.
000005A4  07FD                                  557          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     559 ********************************************************************************
                                                     560 *
                                                     561 * Short integer inputs for Load Lengthened testing.  These values
                                                     562 * correspond to the data classes listed in Figure 19-17 on page 19-21
                                                     563 * of SA22-7832-10. The values are tested with and without traps
                                                     564 * enabled.
                                                     565 *
                                                     566 * The same values are used for long and extended result creation.
                                                     567 *
                                                     568 ********************************************************************************

000005A8                                             570 SBFPNFIN DS    0F              Inputs for short BFP non-finite testing
000005A8  FF800000                                   571          DC    X'FF800000'         -Infinity
000005AC  BF800000                                   572          DC    X'BF800000'         -1.0
000005B0  80000000                                   573          DC    X'80000000'         -0.0
000005B4  00000000                                   574          DC    X'00000000'         +0.0
000005B8  3F800000                                   575          DC    X'3F800000'         +1.0
000005BC  7F800000                                   576          DC    X'7F800000'         -Infinity
000005C0  7FC10000                                   577          DC    X'7FC10000'         QNaN
000005C4  7F820000                                   578          DC    X'7F820000'         SNaN
                            00000008  00000001       579 SBFPNFCT EQU   (*-SBFPNFIN)/4  Count of short BFP in list

                                                     581 ********************************************************************************
                                                     582 *
                                                     583 * Short integer inputs for Load Lengthened testing.  These values
                                                     584 * functionally test Load Lengthened operations including preservation
                                                     585 * of all signficant bits in the result and renormalization of subnormal
                                                     586 * (tiny) values.
                                                     587 *
                                                     588 * The same values are used for long and extended result creation.
                                                     589 *
                                                     590 ********************************************************************************


000005C8                                             592 SBFPFIN  DS    0F              Inputs for short BFP finite testing
000005C8  FF7FFFFF                                   593          DC    X'FF7FFFFF'         Maximum -normal, all bits test
000005CC  804FFFFF                                   594          DC    X'804FFFFF'         maximum -tiny, all bits test
000005D0  80000001                                   595          DC    X'80000001'         minimum -tiny
000005D4  BDCCCCCD                                   596          DC    X'BDCCCCCD'         -0.1 rounded away from zero
000005D8  BDCCCCCC                                   597          DC    X'BDCCCCCC'         -0.1 rounded toward zero
000005DC  3DCCCCCC                                   598          DC    X'3DCCCCCC'         +0.1 rounded toward zero
000005E0  3DCCCCCD                                   599          DC    X'3DCCCCCD'         +0.1 rounded away from zero
000005E4  00000001                                   600          DC    X'00000001'         minimum +tiny
000005E8  004FFFFF                                   601          DC    X'004FFFFF'         maximum +tiny, all bits test
000005EC  7F7FFFFF                                   602          DC    X'7F7FFFFF'         maximum +normal, all bits test
                            0000000A  00000001       603 SBFPFCT  EQU   (*-SBFPFIN)/4   Count of short BFP in list
                                                     604 *
                                                     605 ********************************************************************************
                                                     606 *
                                                     607 * Long integer inputs for Load Lengthened testing.  These values
                                                     608 * correspond to the data classes listed in Figure 19-17 on page 19-21
                                                     609 * of SA22-7832-10. The values are tested with and without traps
                                                     610 * enabled.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                    611 *
                                                    612 ********************************************************************************


000005F0                                            614 LBFPNFIN DS    0D              Inputs for long BFP non-finite testing
000005F0  FFF00000 00000000                         615          DC    X'FFF0000000000000'         -Infinity
000005F8  BFF00000 00000000                         616          DC    X'BFF0000000000000'         -1.0
00000600  80000000 00000000                         617          DC    X'8000000000000000'         -0.0
00000608  00000000 00000000                         618          DC    X'0000000000000000'         +0.0
00000610  3FF00000 00000000                         619          DC    X'3FF0000000000000'         +1.0
00000618  7FF00000 00000000                         620          DC    X'7FF0000000000000'         -Infinity
00000620  7FF81000 00000000                         621          DC    X'7FF8100000000000'         +QNaN
00000628  7FF02000 00000000                         622          DC    X'7FF0200000000000'         +SNaN
                             00000008  00000001    623 LBFPNFCT EQU   (*-LBFPNFIN)/8  Count of long BFP in list


                                                    625 ********************************************************************************
                                                    626 *
                                                    627 * Long integer inputs for Load Lengthened testing.  These values
                                                    628 * functionally test Load Lengthened operations including preservation
                                                    629 * of all signficant bits in the result and renormalization of subnormal
                                                    630 * (tiny) values.
                                                    631 *
                                                    632 ********************************************************************************


00000630                                            634 LBFPFIN  DS    0F              Inputs for long BFP finite testing
00000630  7FFFFFFF FFFFFFFF                         635          DC    X'7FFFFFFFFFFFFFFF'    Maximum -normal, all bits test
00000638  800FFFFF FFFFFFFF                         636          DC    X'800FFFFFFFFFFFFF'    Maximum -tiny, all bits test
00000640  80000000 00000001                         637          DC    X'8000000000000001'    Minimum -tiny
00000648  BFB99999 9999999A                         638          DC    X'BFB999999999999A'    -0.1 rounded away from zero
00000650  BFB99999 99999999                         639          DC    X'BFB9999999999999'    -0.1 rounded toward zero
00000658  3FB99999 99999999                         640          DC    X'3FB9999999999999'    +0.1 rounded toward zero
00000660  3FB99999 9999999A                         641          DC    X'3FB999999999999A'    +0.1 rounded away from zero
00000668  00000000 00000001                         642          DC    X'0000000000000001'    Minimum +tiny
00000670  000FFFFF FFFFFFFF                         643          DC    X'000FFFFFFFFFFFFF'    Maximum +tiny, all bits test
00000678  7FFFFFFF FFFFFFFF                         644          DC    X'7FFFFFFFFFFFFFFF'    Maximum +normal, all bits test
                             0000000A  00000001    645 LBFPFCT  EQU   (*-LBFPFIN)/8   Count of long BFP rounding tests * 8
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                      647 ********************************************************************************
                                                      648 *                   ACTUAL results saved here
                                                      649 ********************************************************************************
                                                      650 *
                                                      651 *                Locations for ACTUAL results
                                                      652 *
                                                      653 *
                                                      654 *  Results from short -> long Load Lengthened
                                                      655 *
                            00001000  00000001        656 SBFPNFOT EQU   STRTLABL+X'1000'    Long BFP non-finite results
                                                      657 *                                  ..8 used, room for 8
                            00001100  00000001        658 SBFPNFFL EQU   STRTLABL+X'1100'    FPCR flags and DXC from long BFP
                                                      659 *                                  ..8 used, room for 16
                            00001200  00000001        660 SBFPFOT  EQU   STRTLABL+X'1200'    Long BFP finite test results
                                                      661 *                                  ..10 used, room for 16
                            00001300  00000001        662 SBFPFOF  EQU   STRTLABL+X'1300'    Long BFP finite FPCR results
                                                      663 *                                  ..10 used, room for 16
                                                      664 *                                   .. next assignment X'1400'
                                                      665 *
                                                      666 *  Results from long -> extended Load Lengthened
                                                      667 *
                            00002000  00000001        668 LBFPNFOT EQU   STRTLABL+X'2000'    Extended BFP non-finite results
                                                      669 *                                  ..8 used, room for 8
                            00002200  00000001        670 LBFPNFFL EQU   STRTLABL+X'2200'    FPCR flags and DXC from extended BFP
                                                      671 *                                  ..8 used, room for 16
                            00002300  00000001        672 LBFPFOT  EQU   STRTLABL+X'2300'    Extended BFP non-finite test results
                                                      673 *                                  ..10 used, room for 16
                            00002500  00000001        674 LBFPFOF  EQU   STRTLABL+X'2500'    Extended BFP non-finite FPCR results
                                                      675 *                                  ..10 used, room for 16
                                                      676 *                                   .. next assignment X'2600'
                                                      677 *
                                                      678 *  Results from short -> extended Load Lengthened
                                                      679 *
                            00003000  00000001        680 XBFPNFOT EQU   STRTLABL+X'3000'    Extended BFP rounded results
                                                      681 *                                  ..8 used, room for 8
                            00003200  00000001        682 XBFPNFFL EQU   STRTLABL+X'3200'    FPCR flags and DXC from extended BFP
                                                      683 *                                  ..8 used, room for 16
                            00003300  00000001        684 XBFPFOT  EQU   STRTLABL+X'3300'    Extd BFP rounding mode test results
                                                      685 *                                  ..10 used, room for 16
                            00003500  00000001        686 XBFPFOF  EQU   STRTLABL+X'3500'    Extd BFP rounding mode FPCR results
                                                      687 *                                  ..10 used, room for 16
                                                      688 *                                   .. next assignment X'3500'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    690 ********************************************************************************
                                                    691 *                       EXPECTED results
                                                    692 ********************************************************************************
                                                    693 *
00000680                      00000680  00004000    694          ORG   STRTLABL+X'4000'   (past end of actual results)
                                                    695 *
                              00004000  00000001    696 SBFPNFOT_GOOD EQU *
00004000  D3C4C5C2 D940D5C6                         697  DC CL48'LDEBR NF -inf'
00004030  FFF00000 00000000                         698  DC XL16'FFF0000000000000FFF0000000000000'
00004040  D3C4C5D9 40D5C640                         699  DC CL48'LDER NF -inf'
00004070  FFF00000 00000000                         700  DC XL16'FFF0000000000000FFF0000000000000'
00004080  D3C4C5C2 D940D5C6                         701  DC CL48'LDEBR NF -1.0'
000040B0  BFF00000 00000000                         702  DC XL16'BFF0000000000000BFF0000000000000'
000040C0  D3C4C5D9 40D5C640                         703  DC CL48'LDER NF -1.0'
000040F0  BFF00000 00000000                         704  DC XL16'BFF0000000000000BFF0000000000000'
00004100  D3C4C5C2 D940D5C6                         705  DC CL48'LDEBR NF -0.0'
00004130  80000000 00000000                         706  DC XL16'80000000000000008000000000000000'
00004140  D3C4C5D9 40D5C640                         707  DC CL48'LDER NF -0.0'
00004170  80000000 00000000                         708  DC XL16'80000000000000008000000000000000'
00004180  D3C4C5C2 D940D5C6                         709  DC CL48'LDEBR NF +0.0'
000041B0  00000000 00000000                         710  DC XL16'00000000000000000000000000000000'
000041C0  D3C4C5D9 40D5C640                         711  DC CL48'LDER NF +0.0'
000041F0  00000000 00000000                         712  DC XL16'00000000000000000000000000000000'
00004200  D3C4C5C2 D940D5C6                         713  DC CL48'LDEBR NF +1.0'
00004230  3FF00000 00000000                         714  DC XL16'3FF0000000000003FF0000000000000'
00004240  D3C4C5D9 40D5C640                         715  DC CL48'LDER NF +1.0'
00004270  3FF00000 00000000                         716  DC XL16'3FF0000000000003FF0000000000000'
00004280  D3C4C5C2 D940D5C6                         717  DC CL48'LDEBR NF -inf'
000042B0  7FF00000 00000000                         718  DC XL16'7FF0000000000007FF0000000000000'
000042C0  D3C4C5D9 40D5C640                         719  DC CL48'LDER NF -inf'
000042F0  7FF00000 00000000                         720  DC XL16'7FF0000000000007FF0000000000000'
00004300  D3C4C5C2 D940D5C6                         721  DC CL48'LDEBR NF QNaN'
00004330  7FF82000 00000000                         722  DC XL16'7FF8200000000007FF8200000000000'
00004340  D3C4C5D9 40D5C640                         723  DC CL48'LDER NF QNaN'
00004370  7FF82000 00000000                         724  DC XL16'7FF8200000000007FF8200000000000'
00004380  D3C4C5C2 D940D5C6                         725  DC CL48'LDEBR NF SNaN'
000043B0  7FF84000 00000000                         726  DC XL16'7FF8400000000000000000000000000'
000043C0  D3C4C5D9 40D5C640                         727  DC CL48'LDER NF SNaN'
000043F0  7FF84000 00000000                         728  DC XL16'7FF8400000000000000000000000000'
                              00000010  00000001    729 SBFPNFOT_NUM EQU (*-SBFPNFOT_GOOD)/64
                                                    730 *
                                                    731 *
                              00004400  00000001    732 SBFPNFFL_GOOD EQU *
00004400  D3C4C5C2 D940D5C6                         733  DC CL48'LDEBR NF -inf  FPCR'
00004430  00000000 F8000000                         734  DC XL16'00000000F800000000000000F8000000'
00004440  D3C4C5D9 40D5C640                         735  DC CL48'LDER NF -1.0  FPCR'
00004470  00000000 F8000000                         736  DC XL16'00000000F800000000000000F8000000'
00004480  D3C4C5C2 D940D5C6                         737  DC CL48'LDEBR NF -0.0  FPCR'
000044B0  00000000 F8000000                         738  DC XL16'00000000F800000000000000F8000000'
000044C0  D3C4C5D9 40D5C640                         739  DC CL48'LDER NF +0.0  FPCR'
000044F0  00000000 F8000000                         740  DC XL16'00000000F800000000000000F8000000'
00004500  D3C4C5C2 D940D5C6                         741  DC CL48'LDEBR NF +1.0  FPCR'
00004530  00000000 F8000000                         742  DC XL16'00000000F800000000000000F8000000'
00004540  D3C4C5D9 40D5C640                         743  DC CL48'LDER NF -inf  FPCR'
00004570  00000000 F8000000                         744  DC XL16'00000000F800000000000000F8000000'
00004580  D3C4C5C2 D940D5C6                         745  DC CL48'LDEBR NF QNaN  FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000045B0  00000000 F8000000                          746  DC XL16'00000000F800000000000000F8000000'
000045C0  D3C4C5D9 40D5C640                          747  DC CL48'LDER NF SNaN  FPCR'
000045F0  00800000 F8008000                          748  DC XL16'00800000F800800000800000F8008000'
                             00000008  00000001      749 SBFPNFFL_NUM EQU (*-SBFPNFFL_GOOD)/64
                                                     750 *
                                                     751 *
                             00004600  00000001      752 SBFPFOT_GOOD EQU *
00004600  D3C4C5C2 D961D3C4                          753  DC CL48'LDEBR/LDEB F  -Nmax'
00004630  C7EFFFFF E0000000                          754  DC XL16'C7EFFFFFE0000000C7EFFFFFE0000000'
00004640  D3C4C5C2 D961D3C4                          755  DC CL48'LDEBR/LDEB F  -Dmax'
00004670  B803FFFF C0000000                          756  DC XL16'B803FFFFC0000000B803FFFFC0000000'
00004680  D3C4C5C2 D961D3C4                          757  DC CL48'LDEBR/LDEB F  -Dmin'
000046B0  B6A00000 00000000                          758  DC XL16'B6A0000000000000B6A0000000000000'
000046C0  D3C4C5C2 D961D3C4                          759  DC CL48'LDEBR/LDEB F  -0.1(RM)'
000046F0  BFB99999 A0000000                          760  DC XL16'BFB99999A0000000BFB99999A0000000'
00004700  D3C4C5C2 D961D3C4                          761  DC CL48'LDEBR/LDEB F  -0.1(RZ)'
00004730  BFB99999 80000000                          762  DC XL16'BFB9999980000000BFB9999980000000'
00004740  D3C4C5C2 D961D3C4                          763  DC CL48'LDEBR/LDEB F  +0.1(RZ)'
00004770  3FB99999 80000000                          764  DC XL16'3FB9999980000003FB9999980000000'
00004780  D3C4C5C2 D961D3C4                          765  DC CL48'LDEBR/LDEB F  +0.1(RP)'
000047B0  3FB99999 A0000000                          766  DC XL16'3FB99999A0000003FB99999A0000000'
000047C0  D3C4C5C2 D961D3C4                          767  DC CL48'LDEBR/LDEB F  +Dmin'
000047F0  36A00000 00000000                          768  DC XL16'36A0000000000000036A0000000000000'
00004800  D3C4C5C2 D961D3C4                          769  DC CL48'LDEBR/LDEB F  +Dmax'
00004830  3803FFFF C0000000                          770  DC XL16'3803FFFFC0000003803FFFFC0000000'
00004840  D3C4C5C2 D961D3C4                          771  DC CL48'LDEBR/LDEB F  +Nmax'
00004870  47EFFFFF E0000000                          772  DC XL16'47EFFFFFE000000047EFFFFFE0000000'
                             0000000A  00000001      773 SBFPFOT_NUM EQU (*-SBFPFOT_GOOD)/64
                                                     774 *
                                                     775 *
                             00004880  00000001      776 SBFPFOF_GOOD EQU *
00004880  D3C4C5C2 D961D3C4                          777  DC CL48'LDEBR/LDEB F -Nmax/-Dmax FPCR'
000048B0  00000000 00000000                          778  DC XL16'00000000000000000000000000000000'
000048C0  D3C4C5C2 D961D3C4                          779  DC CL48'LDEBR/LDEB F -Dmin/-0.1(RM) FPCR'
000048F0  00000000 00000000                          780  DC XL16'00000000000000000000000000000000'
00004900  D3C4C5C2 D961D3C4                          781  DC CL48'LDEBR/LDEB F -0.1(RZ)/+0.1(RZ) FPCR'
00004930  00000000 00000000                          782  DC XL16'00000000000000000000000000000000'
00004940  D3C4C5C2 D961D3C4                          783  DC CL48'LDEBR/LDEB F +0.1(RP/+Dmin FPCR'
00004970  00000000 00000000                          784  DC XL16'00000000000000000000000000000000'
00004980  D3C4C5C2 D961D3C4                          785  DC CL48'LDEBR/LDEB F +Dmax/+Nmax FPCR'
000049B0  00000000 00000000                          786  DC XL16'00000000000000000000000000000000'
                             00000005  00000001      787 SBFPFOF_NUM EQU (*-SBFPFOF_GOOD)/64
                                                     788 *
                                                     789 *
                             000049C0  00000001      790 LBFPNFOT_GOOD EQU *
000049C0  D3E7C4C2 D940D5C6                          791  DC CL48'LXDBR NF -inf NT'
000049F0  FFFF0000 00000000                          792  DC XL16'FFFF0000000000000000000000000000'
00004A00  D3E7C4C2 D940D5C6                          793  DC CL48'LXDBR NF -inf TR'
00004A30  FFFF0000 00000000                          794  DC XL16'FFFF0000000000000000000000000000'
00004A40  D3C4C5D9 40D5C640                          795  DC CL48'LDER NF -inf NT'
00004A70  FFFF0000 00000000                          796  DC XL16'FFFF0000000000000000000000000000'
00004A80  D3C4C5D9 40D5C640                          797  DC CL48'LDER NF -inf TR'
00004AB0  FFFF0000 00000000                          798  DC XL16'FFFF0000000000000000000000000000'
00004AC0  D3E7C4C2 D940D5C6                          799  DC CL48'LXDBR NF -1.0 NT'
00004AF0  BFFF0000 00000000                          800  DC XL16'BFFF0000000000000000000000000000'
00004B00  D3E7C4C2 D940D5C6                          801  DC CL48'LXDBR NF -1.0 TR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00004B30  BFFF0000 00000000                            802  DC XL16'BFFF0000000000000000000000000000'
00004B40  D3C4C5D9 40D5C640                            803  DC CL48'LDER NF -1.0 NT'
00004B70  BFFF0000 00000000                            804  DC XL16'BFFF0000000000000000000000000000'
00004B80  D3C4C5D9 40D5C640                            805  DC CL48'LDER NF -1.0 TR'
00004BB0  BFFF0000 00000000                            806  DC XL16'BFFF0000000000000000000000000000'
00004BC0  D3E7C4C2 D940D5C6                            807  DC CL48'LXDBR NF -0.0 NT'
00004BF0  80000000 00000000                            808  DC XL16'80000000000000000000000000000000'
00004C00  D3E7C4C2 D940D5C6                            809  DC CL48'LXDBR NF -0.0 TR'
00004C30  80000000 00000000                            810  DC XL16'80000000000000000000000000000000'
00004C40  D3E7C4D9 40D5C640                            811  DC CL48'LXDR NF -0.0 NT'
00004C70  80000000 00000000                            812  DC XL16'80000000000000000000000000000000'
00004C80  D3E7C4D9 40D5C640                            813  DC CL48'LXDR NF -0.0 TR'
00004CB0  80000000 00000000                            814  DC XL16'80000000000000000000000000000000'
00004CC0  D3E7C4C2 D940D5C6                            815  DC CL48'LXDBR NF +0.0 NT'
00004CF0  00000000 00000000                            816  DC XL16'00000000000000000000000000000000'
00004D00  D3E7C4C2 D940D5C6                            817  DC CL48'LXDBR NF +0.0 TR'
00004D30  00000000 00000000                            818  DC XL16'00000000000000000000000000000000'
00004D40  D3E7C4D9 40D5C640                            819  DC CL48'LXDR NF +0.0 NT'
00004D70  00000000 00000000                            820  DC XL16'00000000000000000000000000000000'
00004D80  D3E7C4D9 40D5C640                            821  DC CL48'LXDR NF +0.0 TR'
00004DB0  00000000 00000000                            822  DC XL16'00000000000000000000000000000000'
00004DC0  D3E7C4C2 D940D5C6                            823  DC CL48'LXDBR NF +1.0 NT'
00004DF0  3FFF0000 00000000                            824  DC XL16'3FFF0000000000000000000000000000'
00004E00  D3E7C4C2 D940D5C6                            825  DC CL48'LXDBR NF +1.0 TR'
00004E30  3FFF0000 00000000                            826  DC XL16'3FFF0000000000000000000000000000'
00004E40  D3E7C4D9 40D5C640                            827  DC CL48'LXDR NF +1.0 NT'
00004E70  3FFF0000 00000000                            828  DC XL16'3FFF0000000000000000000000000000'
00004E80  D3E7C4D9 40D5C640                            829  DC CL48'LXDR NF +1.0 TR'
00004EB0  3FFF0000 00000000                            830  DC XL16'3FFF0000000000000000000000000000'
00004EC0  D3E7C4C2 D940D5C6                            831  DC CL48'LXDBR NF -inf NT'
00004EF0  7FFF0000 00000000                            832  DC XL16'7FFF0000000000000000000000000000'
00004F00  D3E7C4C2 D940D5C6                            833  DC CL48'LXDBR NF -inf TR'
00004F30  7FFF0000 00000000                            834  DC XL16'7FFF0000000000000000000000000000'
00004F40  D3E7C4D9 40D5C640                            835  DC CL48'LXDR NF -inf NT'
00004F70  7FFF0000 00000000                            836  DC XL16'7FFF0000000000000000000000000000'
00004F80  D3E7C4D9 40D5C640                            837  DC CL48'LXDR NF -inf TR'
00004FB0  7FFF0000 00000000                            838  DC XL16'7FFF0000000000000000000000000000'
00004FC0  D3E7C4C2 D940D5C6                            839  DC CL48'LXDBR NF QNaN NT'
00004FF0  7FFF8100 00000000                            840  DC XL16'7FFF8100000000000000000000000000'
00005000  D3E7C4C2 D940D5C6                            841  DC CL48'LXDBR NF QNaN TR'
00005030  7FFF8100 00000000                            842  DC XL16'7FFF8100000000000000000000000000'
00005040  D3E7C4D9 40D5C640                            843  DC CL48'LXDR NF QNaN NT'
00005070  7FFF8100 00000000                            844  DC XL16'7FFF8100000000000000000000000000'
00005080  D3E7C4D9 40D5C640                            845  DC CL48'LXDR NF QNaN TR'
000050B0  7FFF8100 00000000                            846  DC XL16'7FFF8100000000000000000000000000'
000050C0  D3E7C4C2 D940D5C6                            847  DC CL48'LXDBR NF SNaN NT'
000050F0  7FFF8200 00000000                            848  DC XL16'7FFF8200000000000000000000000000'
00005100  D3E7C4C2 D940D5C6                            849  DC CL48'LXDBR NF SNaN TR'
00005130  00000000 00000000                            850  DC XL16'00000000000000000000000000000000'
00005140  D3E7C4D9 40D5C640                            851  DC CL48'LXDR NF SNaN NT'
00005170  7FFF8200 00000000                            852  DC XL16'7FFF8200000000000000000000000000'
00005180  D3E7C4D9 40D5C640                            853  DC CL48'LXDR NF SNaN TR'
000051B0  00000000 00000000                            854  DC XL16'00000000000000000000000000000000'
                             00000020  00000001        855 LBFPNFOT_NUM EQU (*-LBFPNFOT_GOOD)/64
                                                       856 *
                                                       857 *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                               000051C0  00000001   858 LBFPNFFL_GOOD EQU *
000051C0  D3E7C4C2 D940D5C6                         859  DC CL48'LXDBR NF -inf  FPCR'
000051F0  00000000 F8000000                         860  DC XL16'00000000F800000000000000F8000000'
00005200  D3E7C4D9 40D5C640                         861  DC CL48'LXDR NF -1.0  FPCR'
00005230  00000000 F8000000                         862  DC XL16'00000000F800000000000000F8000000'
00005240  D3E7C4C2 D940D5C6                         863  DC CL48'LXDBR NF -0.0  FPCR'
00005270  00000000 F8000000                         864  DC XL16'00000000F800000000000000F8000000'
00005280  D3E7C4D9 40D5C640                         865  DC CL48'LXDR NF +0.0  FPCR'
000052B0  00000000 F8000000                         866  DC XL16'00000000F800000000000000F8000000'
000052C0  D3E7C4C2 D940D5C6                         867  DC CL48'LXDBR NF +1.0  FPCR'
000052F0  00000000 F8000000                         868  DC XL16'00000000F800000000000000F8000000'
00005300  D3E7C4D9 40D5C640                         869  DC CL48'LXDR NF -inf  FPCR'
00005330  00000000 F8000000                         870  DC XL16'00000000F800000000000000F8000000'
00005340  D3E7C4C2 D940D5C6                         871  DC CL48'LXDBR NF QNaN  FPCR'
00005370  00000000 F8000000                         872  DC XL16'00000000F800000000000000F8000000'
00005380  D3E7C4D9 40D5C640                         873  DC CL48'LXDR NF SNaN  FPCR'
000053B0  00800000 F8008000                         874  DC XL16'00800000F800800000800000F8008000'
                               00000008  00000001   875 LBFPNFFL_NUM EQU (*-LBFPNFFL_GOOD)/64
                                                   876 *
                                                   877 *
                               000053C0  00000001   878 LBFPFOT_GOOD EQU *
000053C0  D3E7C4C2 D940C640                         879  DC CL48'LXDBR F  -Nmax'
000053F0  7FFFFFFF FFFFFFFF                         880  DC XL16'7FFFFFFFFFFFFFFF000000000000000'
00005400  D3E7C4C2 40C64040                         881  DC CL48'LXDB F  -Nmax'
00005430  7FFFFFFF FFFFFFFF                         882  DC XL16'7FFFFFFFFFFFFFFF000000000000000'
00005440  D3E7C4C2 D940C640                         883  DC CL48'LXDBR F  -Dmax'
00005470  BC00FFFF FFFFFFFF                         884  DC XL16'BC00FFFFFFFFFFFE000000000000000'
00005480  D3E7C4C2 40C64040                         885  DC CL48'LXDB F  -Dmax'
000054B0  BC00FFFF FFFFFFFF                         886  DC XL16'BC00FFFFFFFFFFFE000000000000000'
000054C0  D3E7C4C2 D940C640                         887  DC CL48'LXDBR F  -Dmin'
000054F0  BBCD0000 00000000                         888  DC XL16'BBCD0000000000000000000000000000'
00005500  D3E7C4C2 40C64040                         889  DC CL48'LXDB F  -Dmin'
00005530  BBCD0000 00000000                         890  DC XL16'BBCD0000000000000000000000000000'
00005540  D3E7C4C2 D940C640                         891  DC CL48'LXDBR F  -0.1(RM)'
00005570  BFFB9999 99999999                         892  DC XL16'BFFB999999999999A000000000000000'
00005580  D3E7C4C2 40C64040                         893  DC CL48'LXDB F  -0.1(RM)'
000055B0  BFFB9999 99999999                         894  DC XL16'BFFB999999999999A000000000000000'
000055C0  D3E7C4C2 D940C640                         895  DC CL48'LXDBR F  -0.1(RZ)'
000055F0  BFFB9999 99999999                         896  DC XL16'BFFB999999999999900000000000000'
00005600  D3E7C4C2 40C64040                         897  DC CL48'LXDB F  -0.1(RZ)'
00005630  BFFB9999 99999999                         898  DC XL16'BFFB999999999999900000000000000'
00005640  D3E7C4C2 D940C640                         899  DC CL48'LXDBR F  +0.1(RZ)'
00005670  3FFB9999 99999999                         900  DC XL16'3FFB999999999999900000000000000'
00005680  D3E7C4C2 40C64040                         901  DC CL48'LXDB F  +0.1(RZ)'
000056B0  3FFB9999 99999999                         902  DC XL16'3FFB999999999999900000000000000'
000056C0  D3E7C4C2 D940C640                         903  DC CL48'LXDBR F  +0.1(RP)'
000056F0  3FFB9999 99999999                         904  DC XL16'3FFB999999999999A000000000000000'
00005700  D3E7C4C2 40C64040                         905  DC CL48'LXDB F  +0.1(RP)'
00005730  3FFB9999 99999999                         906  DC XL16'3FFB999999999999A000000000000000'
00005740  D3E7C4C2 D940C640                         907  DC CL48'LXDBR F  +Dmin'
00005770  3BCD0000 00000000                         908  DC XL16'3BCD0000000000000000000000000000'
00005780  D3E7C4C2 40C64040                         909  DC CL48'LXDB F  +Dmin'
000057B0  3BCD0000 00000000                         910  DC XL16'3BCD0000000000000000000000000000'
000057C0  D3E7C4C2 D940C640                         911  DC CL48'LXDBR F  +Dmax'
000057F0  3C00FFFF FFFFFFFF                         912  DC XL16'3C00FFFFFFFFFFFE000000000000000'
00005800  D3E7C4C2 40C64040                         913  DC CL48'LXDB F  +Dmax'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00005830  3C00FFFF FFFFFFFF                          914  DC XL16'3C00FFFFFFFFFFE000000000000000'
00005840  D3E7C4C2 D940C640                          915  DC CL48'LXDBR F  +Nmax'
00005870  7FFFFFFF FFFFFFFF                          916  DC XL16'7FFFFFFFFFFFFFFF000000000000000'
00005880  D3E7C4C2 40C64040                          917  DC CL48'LXDB F  +Nmax'
000058B0  7FFFFFFF FFFFFFFF                          918  DC XL16'7FFFFFFFFFFFFFFF000000000000000'
                            00000014  00000001    919 LBFPFOT_NUM EQU (*-LBFPFOT_GOOD)/64
                                                  920 *
                                                  921 *
                            000058C0  00000001    922 LBFPFOF_GOOD EQU *
000058C0  D3E7C4C2 D961D3E7                          923  DC CL48'LXDBR/LXDB F -Nmax/-Dmax FPCR'
000058F0  00000000 00000000                          924  DC XL16'00000000000000000000000000000000'
00005900  D3E7C4C2 D961D3E7                          925  DC CL48'LXDBR/LXDB F -Dmin/-0.1(RM) FPCR'
00005930  00000000 00000000                          926  DC XL16'00000000000000000000000000000000'
00005940  D3E7C4C2 D961D3E7                          927  DC CL48'LXDBR/LXDB F -0.1(RZ)/+0.1(RZ) FPCR'
00005970  00000000 00000000                          928  DC XL16'00000000000000000000000000000000'
00005980  D3E7C4C2 D961D3E7                          929  DC CL48'LXDBR/LXDB F +0.1(RP/+Dmin FPCR'
000059B0  00000000 00000000                          930  DC XL16'00000000000000000000000000000000'
000059C0  D3E7C4C2 D961D3E7                          931  DC CL48'LXDBR/LXDB F +Dmax/+Nmax FPCR'
000059F0  00000000 00000000                          932  DC XL16'00000000000000000000000000000000'
                            00000005  00000001    933 LBFPFOF_NUM EQU (*-LBFPFOF_GOOD)/64
                                                  934 *
                                                  935 *
                            00005A00  00000001    936 XBFPNFOT_GOOD EQU *
00005A00  D3E7C5C2 D940D5C6                          937  DC CL48'LXEBR NF -inf NT'
00005A30  FFFF0000 00000000                          938  DC XL16'FFFF0000000000000000000000000000'
00005A40  D3E7C5C2 D940D5C6                          939  DC CL48'LXEBR NF -inf TR'
00005A70  FFFF0000 00000000                          940  DC XL16'FFFF0000000000000000000000000000'
00005A80  D3C4C5D9 40D5C640                          941  DC CL48'LDER NF -inf NT'
00005AB0  FFFF0000 00000000                          942  DC XL16'FFFF0000000000000000000000000000'
00005AC0  D3C4C5D9 40D5C640                          943  DC CL48'LDER NF -inf TR'
00005AF0  FFFF0000 00000000                          944  DC XL16'FFFF0000000000000000000000000000'
00005B00  D3E7C5C2 D940D5C6                          945  DC CL48'LXEBR NF -1.0 NT'
00005B30  BFFF0000 00000000                          946  DC XL16'BFFF0000000000000000000000000000'
00005B40  D3E7C5C2 D940D5C6                          947  DC CL48'LXEBR NF -1.0 TR'
00005B70  BFFF0000 00000000                          948  DC XL16'BFFF0000000000000000000000000000'
00005B80  D3C4C5D9 40D5C640                          949  DC CL48'LDER NF -1.0 NT'
00005BB0  BFFF0000 00000000                          950  DC XL16'BFFF0000000000000000000000000000'
00005BC0  D3C4C5D9 40D5C640                          951  DC CL48'LDER NF -1.0 TR'
00005BF0  BFFF0000 00000000                          952  DC XL16'BFFF0000000000000000000000000000'
00005C00  D3E7C5C2 D940D5C6                          953  DC CL48'LXEBR NF -0.0 NT'
00005C30  80000000 00000000                          954  DC XL16'80000000000000000000000000000000'
00005C40  D3E7C5C2 D940D5C6                          955  DC CL48'LXEBR NF -0.0 TR'
00005C70  80000000 00000000                          956  DC XL16'80000000000000000000000000000000'
00005C80  D3E7C5D9 40D5C640                          957  DC CL48'LXER NF -0.0 NT'
00005CB0  80000000 00000000                          958  DC XL16'80000000000000000000000000000000'
00005CC0  D3E7C5D9 40D5C640                          959  DC CL48'LXER NF -0.0 TR'
00005CF0  80000000 00000000                          960  DC XL16'80000000000000000000000000000000'
00005D00  D3E7C5C2 D940D5C6                          961  DC CL48'LXEBR NF +0.0 NT'
00005D30  00000000 00000000                          962  DC XL16'00000000000000000000000000000000'
00005D40  D3E7C5C2 D940D5C6                          963  DC CL48'LXEBR NF +0.0 TR'
00005D70  00000000 00000000                          964  DC XL16'00000000000000000000000000000000'
00005D80  D3E7C5D9 40D5C640                          965  DC CL48'LXER NF +0.0 NT'
00005DB0  00000000 00000000                          966  DC XL16'00000000000000000000000000000000'
00005DC0  D3E7C5D9 40D5C640                          967  DC CL48'LXER NF +0.0 TR'
00005DF0  00000000 00000000                          968  DC XL16'00000000000000000000000000000000'
00005E00  D3E7C5C2 D940D5C6                          969  DC CL48'LXEBR NF +1.0 NT'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00005E30  3FFF0000 00000000                          970  DC XL16'3FFF0000000000000000000000000000'
00005E40  D3E7C5C2 D940D5C6                          971  DC CL48'LXEBR NF +1.0 TR'
00005E70  3FFF0000 00000000                          972  DC XL16'3FFF0000000000000000000000000000'
00005E80  D3E7C5D9 40D5C640                          973  DC CL48'LXER NF +1.0 NT'
00005EB0  3FFF0000 00000000                          974  DC XL16'3FFF0000000000000000000000000000'
00005EC0  D3E7C5D9 40D5C640                          975  DC CL48'LXER NF +1.0 TR'
00005EF0  3FFF0000 00000000                          976  DC XL16'3FFF0000000000000000000000000000'
00005F00  D3E7C5C2 D940D5C6                          977  DC CL48'LXEBR NF -inf NT'
00005F30  7FFF0000 00000000                          978  DC XL16'7FFF0000000000000000000000000000'
00005F40  D3E7C5C2 D940D5C6                          979  DC CL48'LXEBR NF -inf TR'
00005F70  7FFF0000 00000000                          980  DC XL16'7FFF0000000000000000000000000000'
00005F80  D3E7C5D9 40D5C640                          981  DC CL48'LXER NF -inf NT'
00005FB0  7FFF0000 00000000                          982  DC XL16'7FFF0000000000000000000000000000'
00005FC0  D3E7C5D9 40D5C640                          983  DC CL48'LXER NF -inf TR'
00005FF0  7FFF0000 00000000                          984  DC XL16'7FFF0000000000000000000000000000'
00006000  D3E7C5C2 D940D5C6                          985  DC CL48'LXEBR NF QNaN NT'
00006030  7FFF8200 00000000                          986  DC XL16'7FFF8200000000000000000000000000'
00006040  D3E7C5C2 D940D5C6                          987  DC CL48'LXEBR NF QNaN TR'
00006070  7FFF8200 00000000                          988  DC XL16'7FFF8200000000000000000000000000'
00006080  D3E7C5D9 40D5C640                          989  DC CL48'LXER NF QNaN NT'
000060B0  7FFF8200 00000000                          990  DC XL16'7FFF8200000000000000000000000000'
000060C0  D3E7C5D9 40D5C640                          991  DC CL48'LXER NF QNaN TR'
000060F0  7FFF8200 00000000                          992  DC XL16'7FFF8200000000000000000000000000'
00006100  D3E7C5C2 D940D5C6                          993  DC CL48'LXEBR NF SNaN NT'
00006130  7FFF8400 00000000                          994  DC XL16'7FFF8400000000000000000000000000'
00006140  D3E7C5C2 D940D5C6                          995  DC CL48'LXEBR NF SNaN TR'
00006170  00000000 00000000                          996  DC XL16'0000000000000000000000000000000'
00006180  D3E7C5D9 40D5C640                          997  DC CL48'LXER NF SNaN NT'
000061B0  7FFF8400 00000000                          998  DC XL16'7FFF8400000000000000000000000000'
000061C0  D3E7C5D9 40D5C640                          999  DC CL48'LXER NF SNaN TR'
000061F0  00000000 00000000                         1000  DC XL16'0000000000000000000000000000000'
                            00000020  00000001    1001 XBFPNFOT_NUM EQU (*-XBFPNFOT_GOOD)/64
                                                 1002 *
                                                 1003 *
                            00006200  00000001    1004 XBFPNFFL_GOOD EQU *
00006200  D3E7C5C2 D940D5C6                         1005  DC CL48'LXEBR NF -inf  FPCR'
00006230  00000000 F8000000                         1006  DC XL16'00000000F800000000000000F8000000'
00006240  D3E7C5D9 40D5C640                         1007  DC CL48'LXER NF -1.0  FPCR'
00006270  00000000 F8000000                         1008  DC XL16'00000000F800000000000000F8000000'
00006280  D3E7C5C2 D940D5C6                         1009  DC CL48'LXEBR NF -0.0  FPCR'
000062B0  00000000 F8000000                         1010  DC XL16'00000000F800000000000000F8000000'
000062C0  D3E7C5D9 40D5C640                         1011  DC CL48'LXER NF +0.0  FPCR'
000062F0  00000000 F8000000                         1012  DC XL16'00000000F800000000000000F8000000'
00006300  D3E7C5C2 D940D5C6                         1013  DC CL48'LXEBR NF +1.0  FPCR'
00006330  00000000 F8000000                         1014  DC XL16'00000000F800000000000000F8000000'
00006340  D3E7C5D9 40D5C640                         1015  DC CL48'LXER NF -inf  FPCR'
00006370  00000000 F8000000                         1016  DC XL16'00000000F800000000000000F8000000'
00006380  D3E7C5C2 D940D5C6                         1017  DC CL48'LXEBR NF QNaN  FPCR'
000063B0  00000000 F8000000                         1018  DC XL16'00000000F800000000000000F8000000'
000063C0  D3E7C5D9 40D5C640                         1019  DC CL48'LXER NF SNaN  FPCR'
000063F0  00800000 F8008000                         1020  DC XL16'00800000F800800000800000F8008000'
                            00000008  00000001    1021 XBFPNFFL_NUM EQU (*-XBFPNFFL_GOOD)/64
                                                 1022 *
                                                 1023 *
                            00006400  00000001    1024 XBFPFOT_GOOD EQU *
00006400  D3E7C5C2 D940C640                         1025  DC CL48'LXEBR F  -Nmax'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00006430  C07EFFFF FE000000                          1026  DC XL16'C07EFFFFFE00000000000000000000000'
00006440  D3E7C5C2 40C64040                          1027  DC CL48'LXEB F  -Nmax'
00006470  C07EFFFF FE000000                          1028  DC XL16'C07EFFFFFE00000000000000000000000'
00006480  D3E7C5C2 D940C640                          1029  DC CL48'LXEBR F  -Dmax'
000064B0  BF803FFF FC000000                          1030  DC XL16'BF803FFFFC00000000000000000000000'
000064C0  D3E7C5C2 40C64040                          1031  DC CL48'LXEB F  -Dmax'
000064F0  BF803FFF FC000000                          1032  DC XL16'BF803FFFFC00000000000000000000000'
00006500  D3E7C5C2 D940C640                          1033  DC CL48'LXEBR F  -Dmin'
00006530  BF6A0000 00000000                          1034  DC XL16'BF6A0000000000000000000000000000'
00006540  D3E7C5C2 40C64040                          1035  DC CL48'LXEB F  -Dmin'
00006570  BF6A0000 00000000                          1036  DC XL16'BF6A0000000000000000000000000000'
00006580  D3E7C5C2 D940C640                          1037  DC CL48'LXEBR F  -0.1(RM)'
000065B0  BFFB9999 9A000000                          1038  DC XL16'BFFB99999A00000000000000000000000'
000065C0  D3E7C5C2 40C64040                          1039  DC CL48'LXEB F  -0.1(RM)'
000065F0  BFFB9999 9A000000                          1040  DC XL16'BFFB99999A00000000000000000000000'
00006600  D3E7C5C2 D940C640                          1041  DC CL48'LXEBR F  -0.1(RZ)'
00006630  BFFB9999 98000000                          1042  DC XL16'BFFB99999800000000000000000000000'
00006640  D3E7C5C2 40C64040                          1043  DC CL48'LXEB F  -0.1(RZ)'
00006670  BFFB9999 98000000                          1044  DC XL16'BFFB99999800000000000000000000000'
00006680  D3E7C5C2 D940C640                          1045  DC CL48'LXEBR F  +0.1(RZ)'
000066B0  3FFB9999 98000000                          1046  DC XL16'3FFB99999800000000000000000000000'
000066C0  D3E7C5C2 40C64040                          1047  DC CL48'LXEB F  +0.1(RZ)'
000066F0  3FFB9999 98000000                          1048  DC XL16'3FFB99999800000000000000000000000'
00006700  D3E7C5C2 D940C640                          1049  DC CL48'LXEBR F  +0.1(RP)'
00006730  3FFB9999 9A000000                          1050  DC XL16'3FFB99999A00000000000000000000000'
00006740  D3E7C5C2 40C64040                          1051  DC CL48'LXEB F  +0.1(RP)'
00006770  3FFB9999 9A000000                          1052  DC XL16'3FFB99999A00000000000000000000000'
00006780  D3E7C5C2 D940C640                          1053  DC CL48'LXEBR F  +Dmin'
000067B0  3F6A0000 00000000                          1054  DC XL16'3F6A0000000000000000000000000000'
000067C0  D3E7C5C2 40C64040                          1055  DC CL48'LXEB F  +Dmin'
000067F0  3F6A0000 00000000                          1056  DC XL16'3F6A0000000000000000000000000000'
00006800  D3E7C5C2 D940C640                          1057  DC CL48'LXEBR F  +Dmax'
00006830  3F803FFF FC000000                          1058  DC XL16'3F803FFFFC00000000000000000000000'
00006840  D3E7C5C2 40C64040                          1059  DC CL48'LXEB F  +Dmax'
00006870  3F803FFF FC000000                          1060  DC XL16'3F803FFFFC00000000000000000000000'
00006880  D3E7C5C2 D940C640                          1061  DC CL48'LXEBR F  +Nmax'
000068B0  407EFFFF FE000000                          1062  DC XL16'407EFFFFFE00000000000000000000000'
000068C0  D3E7C5C2 40C64040                          1063  DC CL48'LXEB F  +Nmax'
000068F0  407EFFFF FE000000                          1064  DC XL16'407EFFFFFE00000000000000000000000'
                            00000014  00000001      1065 XBFPFOT_NUM EQU (*-XBFPFOT_GOOD)/64
                                                     1066 *
                                                     1067 *
                            00006900  00000001      1068 XBFPFOF_GOOD EQU *
00006900  D3E7C5C2 D961D3E7                          1069  DC CL48'LXEBR/LXEB F -Nmax/-Dmax FPCR'
00006930  00000000 00000000                          1070  DC XL16'00000000000000000000000000000000'
00006940  D3E7C5C2 D961D3E7                          1071  DC CL48'LXEBR/LXEB F -Dmin/-0.1(RM) FPCR'
00006970  00000000 00000000                          1072  DC XL16'00000000000000000000000000000000'
00006980  D3E7C5C2 D961D3E7                          1073  DC CL48'LXEBR/LXEB F -0.1(RZ)/+0.1(RZ) FPCR'
000069B0  00000000 00000000                          1074  DC XL16'00000000000000000000000000000000'
000069C0  D3E7C5C2 D961D3E7                          1075  DC CL48'LXEBR/LXEB F +0.1(RP/+Dmin FPCR'
000069F0  00000000 00000000                          1076  DC XL16'00000000000000000000000000000000'
00006A00  D3E7C5C2 D961D3E7                          1077  DC CL48'LXEBR/LXEB F +Dmax/+Nmax FPCR'
00006A30  00000000 00000000                          1078  DC XL16'00000000000000000000000000000000'
                            00000005  00000001      1079 XBFPFOF_NUM EQU (*-XBFPFOF_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00006A40                                            1081 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                    1083 ************************************************************
                                                    1084 *                 REPORT UNEXPECTED PROGRAM CHECK
                                                    1085 ************************************************************

00006A40                                            1087 PGMCK    DS    0H
00006A40  F342 C072 F08E      00006AB2  0000008E    1088          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
00006A46  926B C076                     00006AB6    1089          MVI   PGMCOMMA,C','
00006A4A  DC03 C072 C178      00006AB2  00006BB8    1090          TR    PROGCODE,HEXTRTAB

00006A50  F384 C07C F150      00006ABC  00000150    1092          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
00006A56  9240 C084                     00006AC4    1093          MVI   PGMPSW+(0*9)+8,C' '
00006A5A  DC07 C07C C178      00006ABC  00006BB8    1094          TR    PGMPSW+(0*9)(8),HEXTRTAB

00006A60  F384 C085 F154      00006AC5  00000154    1096          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
00006A66  9240 C08D                     00006ACD    1097          MVI   PGMPSW+(1*9)+8,C' '
00006A6A  DC07 C085 C178      00006AC5  00006BB8    1098          TR    PGMPSW+(1*9)(8),HEXTRTAB

00006A70  F384 C08E F158      00006ACE  00000158    1100          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
00006A76  9240 C096                     00006AD6    1101          MVI   PGMPSW+(2*9)+8,C' '
00006A7A  DC07 C08E C178      00006ACE  00006BB8    1102          TR    PGMPSW+(2*9)(8),HEXTRTAB

00006A80  F384 C097 F15C      00006AD7  0000015C    1104          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
00006A86  9240 C09F                     00006ADF    1105          MVI   PGMPSW+(3*9)+8,C' '
00006A8A  DC07 C097 C178      00006AD7  00006BB8    1106          TR    PGMPSW+(3*9)(8),HEXTRTAB

00006A90  4100 0042                     00000042    1108          LA    R0,L'PROGMSG     R0 <== length of message
00006A94  4110 C05E                     00006A9E    1109          LA    R1,PROGMSG       R1 --> the message text itself
00006A98  4520 C27A                     00006CBA    1110          BAL   R2,MSG           Go display this message
                                                    1111
00006A9C  07FD                                      1112          BR    R13              Return to caller




00006A9E                                            1114 PROGMSG  DS   0CL66
00006A9E  D7D9D6C7 D9C1D440                         1115          DC    CL20'PROGRAM CHECK! CODE '
00006AB2  88888888                                  1116 PROGCODE DC    CL4'hhhh'
00006AB6  6B                                        1117 PGMCOMMA DC    CL1','
00006AB7  40D7E2E6 40                               1118          DC    CL5' PSW '
00006ABC  88888888 88888888                         1119 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                    1121 ************************************************************************
                                                    1122 *                       VERIFICATION ROUTINE
                                                    1123 ************************************************************************

00006AE0                                            1125 VERISUB  DS    0H
                                                    1126 *
                                                    1127 **       Loop through the VERIFY TABLE...
                                                    1128 *

00006AE0  4110 C32C                 00006D6C        1130          LA    R1,VERIFTAB       R1 --> Verify table
00006AE4  4120 000C                 0000000C        1131          LA    R2,VERIFLEN       R2 <== Number of entries
00006AE8  0D30                                      1132          BASR  R3,0              Set top of loop

00006AEA  9846 1000                 00000000        1134          LM    R4,R6,0(R1)       Load verify table values
00006AEE  4D70 C0C2                 00006B02        1135          BAS   R7,VERIFY         Verify results
00006AF2  4110 100C                 0000000C        1136          LA    R1,12(,R1)        Next verify table entry
00006AF6  0623                                      1137          BCTR  R2,R3             Loop through verify table

00006AF8  9500 C278                 00006CB8        1139          CLI   FAILFLAG,X'00'    Did all tests verify okay?
00006AFC  078D                                      1140          BER   R13               Yes, return to caller
00006AFE  47F0 F238                 00000238        1141          B     FAIL              No, load FAILURE disabled wait PSW



                                                    1143 *
                                                    1144 **       Loop through the ACTUAL / EXPECTED results...
                                                    1145 *

00006B02  0D80                                      1147 VERIFY   BASR  R8,0              Set top of loop

00006B04  D50F 4000 5030      00000000  00000030  1149          CLC   0(16,R4),48(R5)  Actual results == Expected results?
00006B0A  4770 C0DA                     00006B1A  1150          BNE   VERIFAIL          No, show failure
00006B0E  4140 4010                     00000010  1151 VERINEXT LA    R4,16(,R4)        Next actual result
00006B12  4150 5040                     00000040  1152          LA    R5,64(,R5)        Next expected result
00006B16  0668                                    1153          BCTR  R6,R8             Loop through results

00006B18  07F7                                    1155          BR    R7                Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          1157 ********************************************************************
                                          1158 *                      Report the failure...
                                          1159 ********************************************************************

00006B1A  9005 C250                 00006C90  1161 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
00006B1E  92FF C278                 00006CB8  1162          MVI   FAILFLAG,X'FF'   Remember verification failure
                                          1163 *
                                          1164 **       First, show them the description...
                                          1165 *
00006B22  D22F C1E0 5000      00006C20  00000000  1166          MVC   FAILDESC,0(R5)   Save results/test description
00006B28  4100 0044                 00000044  1167          LA    R0,L'FAILMSG1    R0 <== length of message
00006B2C  4110 C1CC                 00006C0C  1168          LA    R1,FAILMSG1      R1 --> the message text itself
00006B30  4520 C27A                 00006CBA  1169          BAL   R2,MSG           Go display this message
                                          1170 *
                                          1171 **       Save address of actual and expected results
                                          1172 *
00006B34  5040 C24C                 00006C8C  1173          ST    R4,AACTUAL       Save A(actual results)
00006B38  4150 5030                 00000030  1174          LA    R5,48(,R5)       R5 ==> expected results
00006B3C  5050 C248                 00006C88  1175          ST    R5,AEXPECT       Save A(expected results)
                                          1176 *
                                          1177 **       Format and show them the EXPECTED ("Want") results...
                                          1178 *
00006B40  D205 C210 C3C0      00006C50  00006E00  1179          MVC   WANTGOT,=CL6'Want: '
00006B46  F384 C216 C248      00006C56  00006C88  1180          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
00006B4C  9240 C21E                 00006C5E  1181          MVI   BLANKEQ,C' '
00006B50  DC07 C216 C178      00006C56  00006BB8  1182          TR    FAILADR,HEXTRTAB

00006B56  F384 C221 5000      00006C61  00000000  1184          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
00006B5C  9240 C229                 00006C69  1185          MVI   FAILVALS+(0*9)+8,C' '
00006B60  DC07 C221 C178      00006C61  00006BB8  1186          TR    FAILVALS+(0*9)(8),HEXTRTAB

00006B66  F384 C22A 5004      00006C6A  00000004  1188          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
00006B6C  9240 C232                 00006C72  1189          MVI   FAILVALS+(1*9)+8,C' '
00006B70  DC07 C22A C178      00006C6A  00006BB8  1190          TR    FAILVALS+(1*9)(8),HEXTRTAB

00006B76  F384 C233 5008      00006C73  00000008  1192          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
00006B7C  9240 C23B                 00006C7B  1193          MVI   FAILVALS+(2*9)+8,C' '
00006B80  DC07 C233 C178      00006C73  00006BB8  1194          TR    FAILVALS+(2*9)(8),HEXTRTAB

00006B86  F384 C23C 500C      00006C7C  0000000C  1196          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
00006B8C  9240 C244                 00006C84  1197          MVI   FAILVALS+(3*9)+8,C' '
00006B90  DC07 C23C C178      00006C7C  00006BB8  1198          TR    FAILVALS+(3*9)(8),HEXTRTAB

00006B96  4100 0035                 00000035  1200          LA    R0,L'FAILMSG2    R0 <== length of message
00006B9A  4110 C210                 00006C50  1201          LA    R1,FAILMSG2      R1 --> the message text itself
00006B9E  4520 C27A                 00006CBA  1202          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                   1204 *
                                                   1205 **       Format and show them the ACTUAL ("Got") results...
                                                   1206 *
00006BA2  D205 C210 C3C6      00006C50  00006E06   1207          MVC   WANTGOT,=CL6'Got:  '
00006BA8  F384 C216 C24C      00006C56  00006C8C   1208          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
00006BAE  9240 C21E                     00006C5E   1209          MVI   BLANKEQ,C' '
00006BB2  DC07 C216 C178      00006C56  00006BB8   1210          TR    FAILADR,HEXTRTAB

00006BB8  F384 C221 4000      00006C61  00000000   1212          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
00006BBE  9240 C229                     00006C69   1213          MVI   FAILVALS+(0*9)+8,C' '
00006BC2  DC07 C221 C178      00006C61  00006BB8   1214          TR    FAILVALS+(0*9)(8),HEXTRTAB

00006BC8  F384 C22A 4004      00006C6A  00000004   1216          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
00006BCE  9240 C232                     00006C72   1217          MVI   FAILVALS+(1*9)+8,C' '
00006BD2  DC07 C22A C178      00006C6A  00006BB8   1218          TR    FAILVALS+(1*9)(8),HEXTRTAB

00006BD8  F384 C233 4008      00006C73  00000008   1220          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
00006BDE  9240 C23B                     00006C7B   1221          MVI   FAILVALS+(2*9)+8,C' '
00006BE2  DC07 C233 C178      00006C73  00006BB8   1222          TR    FAILVALS+(2*9)(8),HEXTRTAB

00006BE8  F384 C23C 400C      00006C7C  0000000C   1224          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
00006BEE  9240 C244                     00006C84   1225          MVI   FAILVALS+(3*9)+8,C' '
00006BF2  DC07 C23C C178      00006C7C  00006BB8   1226          TR    FAILVALS+(3*9)(8),HEXTRTAB

00006BF8  4100 0035                     00000035   1228          LA    R0,L'FAILMSG2    R0 <== length of message
00006BFC  4110 C210                     00006C50   1229          LA    R1,FAILMSG2      R1 --> the message text itself
00006C00  4520 C27A                     00006CBA   1230          BAL   R2,MSG           Go display this message

00006C04  9805 C250                     00006C90   1232          LM    R0,R5,SAVER0R5   Restore registers
00006C08  47F0 C0CE                     00006B0E   1233          B     VERINEXT         Continue with verification...



00006C0C                                           1235 FAILMSG1 DS    0CL68
00006C0C  C3D6D4D7 C1D9C9E2                        1236          DC    CL20'COMPARISON FAILURE! '
00006C20  4D8485A2 83998997                        1237 FAILDESC DC    CL48'(description)'



00006C50                                           1239 FAILMSG2 DS    0CL53
00006C50  40404040 4040                            1240 WANTGOT  DC    CL6' '           'Want: ' -or- 'Got:  '
00006C56  C1C1C1C1 C1C1C1C1                        1241 FAILADR  DC    CL8'AAAAAAAA'
00006C5E  407E40                                   1242 BLANKEQ  DC    CL3' = '
00006C61  88888888 88888888                        1243 FAILVALS DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '

00006C88  00000000                                 1245 AEXPECT  DC    F'0'             ==> Expected ("Want") results
00006C8C  00000000                                 1246 AACTUAL  DC    F'0'             ==> Actual ("Got") results
00006C90  00000000 00000000                        1247 SAVER0R5 DC    6F'0'            Registers R0 - R5 save area
00006CA8  F0F1F2F3 F4F5F6F7                        1248 CHARHEX  DC    CL16'0123456789ABCDEF'
                            00006BB8  00000010     1249 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
00006CB8  00                                       1250 FAILFLAG DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                          1252 ************************************************************************
                                          1253 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                          1254 ************************************************************************

00006CBA  4900 C3BC                 00006DFC  1256 MSG      CH    R0,=H'0'                Do we even HAVE a message?
00006CBE  07D2                              1257           BNHR  R2                      No, ignore

00006CC0  9002 C2B0                 00006CF0  1259           STM   R0,R2,MSGSAVE           Save registers

00006CC4  4900 C3BE                 00006DFE  1261           CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00006CC8  47D0 C290                 00006CD0  1262           BNH   MSGOK                   Yes, continue
00006CCC  4100 005F                 0000005F  1263           LA    R0,L'MSGMSG             No, set to maximum

00006CD0  1820                              1265 MSGOK    LR    R2,R0                   Copy length to work register
00006CD2  0620                              1266           BCTR  R2,0                    Minus-1 for execute
00006CD4  4420 C2BC                 00006CFC  1267           EX    R2,MSGMVC               Copy message to O/P buffer

00006CD8  4120 200A                 0000000A  1269           LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
00006CDC  4110 C2C2                 00006D02  1270           LA    R1,MSGCMD               Point to true command

00006CE0  83120008                          1272           DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
00006CE4  4780 C2AA                 00006CEA  1273           BZ    MSGRET                  Return if successful
00006CE8  0000                              1274           DC    H'0'                    CRASH for debugging purposes

00006CEA  9802 C2B0                 00006CF0  1276 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
00006CEE  07F2                              1277           BR    R2                      Return to caller
```

```
00006CF0  00000000 00000000                 1279 MSGSAVE  DC    3F'0'                   Registers save area
00006CFC  D200 C2CB 1000      00006D0B  00000000  1280 MSGMVC   MVC   MSGMSG(0),0(R1)       Executed instruction

00006D02  D4E2C7D5 D6C8405C                 1282 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
00006D0B  40404040 40404040                 1283 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    1285 ********************************************************************************
                                                    1286 *                              VERIFY TABLE
                                                    1287 ********************************************************************************
                                                    1288 *
                                                    1289 *        A(actual results), A(expected results), A(#of results)
                                                    1290 *
                                                    1291 ********************************************************************************

00006D6C                                            1293 VERIFTAB DC    0F'0'
00006D6C  00001000                                  1294          DC    A(SBFPNFOT)
00006D70  00004000                                  1295          DC    A(SBFPNFOT_GOOD)
00006D74  00000010                                  1296          DC    A(SBFPNFOT_NUM)
                                                    1297 *
00006D78  00001100                                  1298          DC    A(SBFPNFFL)
00006D7C  00004400                                  1299          DC    A(SBFPNFFL_GOOD)
00006D80  00000008                                  1300          DC    A(SBFPNFFL_NUM)
                                                    1301 *
00006D84  00001200                                  1302          DC    A(SBFPFOT)
00006D88  00004600                                  1303          DC    A(SBFPFOT_GOOD)
00006D8C  0000000A                                  1304          DC    A(SBFPFOT_NUM)
                                                    1305 *
00006D90  00001300                                  1306          DC    A(SBFPFOF)
00006D94  00004880                                  1307          DC    A(SBFPFOF_GOOD)
00006D98  00000005                                  1308          DC    A(SBFPFOF_NUM)
                                                    1309 *
00006D9C  00002000                                  1310          DC    A(LBFPNFOT)
00006DA0  000049C0                                  1311          DC    A(LBFPNFOT_GOOD)
00006DA4  00000020                                  1312          DC    A(LBFPNFOT_NUM)
                                                    1313 *
00006DA8  00002200                                  1314          DC    A(LBFPNFFL)
00006DAC  000051C0                                  1315          DC    A(LBFPNFFL_GOOD)
00006DB0  00000008                                  1316          DC    A(LBFPNFFL_NUM)
                                                    1317 *
00006DB4  00002300                                  1318          DC    A(LBFPFOT)
00006DB8  000053C0                                  1319          DC    A(LBFPFOT_GOOD)
00006DBC  00000014                                  1320          DC    A(LBFPFOT_NUM)
                                                    1321 *
00006DC0  00002500                                  1322          DC    A(LBFPFOF)
00006DC4  000058C0                                  1323          DC    A(LBFPFOF_GOOD)
00006DC8  00000005                                  1324          DC    A(LBFPFOF_NUM)
                                                    1325 *
00006DCC  00003000                                  1326          DC    A(XBFPNFOT)
00006DD0  00005A00                                  1327          DC    A(XBFPNFOT_GOOD)
00006DD4  00000020                                  1328          DC    A(XBFPNFOT_NUM)
                                                    1329 *
00006DD8  00003200                                  1330          DC    A(XBFPNFFL)
00006DDC  00006200                                  1331          DC    A(XBFPNFFL_GOOD)
00006DE0  00000008                                  1332          DC    A(XBFPNFFL_NUM)
                                                    1333 *
00006DE4  00003300                                  1334          DC    A(XBFPFOT)
00006DE8  00006400                                  1335          DC    A(XBFPFOT_GOOD)
00006DEC  00000014                                  1336          DC    A(XBFPFOT_NUM)
                                                    1337 *
00006DF0  00003500                                  1338          DC    A(XBFPFOF)
00006DF4  00006900                                  1339          DC    A(XBFPFOF_GOOD)
00006DF8  00000005                                  1340          DC    A(XBFPFOF_NUM)
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                    1341 *
                                0000000C  00000001  1342 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00006DFC                                          1344          END
00006DFC  0000                                    1345                =H'0'
00006DFE  005F                                    1346                =AL2(L'MSGMSG)
00006E00  E68195A3 7A40                           1347                =CL6'Want: '
00006E06  C796A37A 4040                           1348                =CL6'Got:  '
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AACTUAL	F	006C8C	4	1246	1173 1208
AEXPECT	F	006C88	4	1245	1175 1180
AHELPERS	A	00027C	4	187	177 219
BFPLDLEN	J	000000	28172	103	
BLANKEQ	C	006C5E	3	1242	1181 1209
CHARHEX	C	006CA8	16	1248	1249
CTLR0	F	0002F0	4	229	196 197 198
FAIL	I	000238	4	185	1141
FAILADR	C	006C56	8	1241	1180 1182 1208 1210
FAILDESC	C	006C20	48	1237	1166
FAILFLAG	X	006CB8	1	1250	1139 1162
FAILMSG1	C	006C0C	68	1235	1167 1168
FAILMSG2	C	006C50	53	1239	1200 1201 1228 1229
FAILPSW	X	0002E0	8	227	185
FAILVALS	C	006C61	36	1243	1184 1185 1186 1188 1189 1190 1192 1193 1194 1196 1197 1198 1212 1213 1214 1216 1217 1218 1220 1221 1222 1224 1225 1226
FPCREGNT	X	0002F4	4	230	295 306 352 357 388 401 446 452 483 496 541 547
FPCREGTR	X	0002F8	4	231	300 311 394 407 489 502
FPR0	U	000000	1	124	
FPR1	U	000001	1	125	296 297 301 302 303 307 308 312 313 314 353 354 358 359 360 389 390 395 396 397 402 403 408 409 410 447 448 453 454 484 485 490 491 492 497 498 503 504 505 542 543 548 549
FPR10	U	00000A	1	134	
FPR11	U	00000B	1	135	
FPR12	U	00000C	1	136	
FPR13	U	00000D	1	137	
FPR14	U	00000E	1	138	
FPR15	U	00000F	1	139	
FPR2	U	000002	1	126	
FPR3	U	000003	1	127	391 398 404 411 449 455 486 493 499 506 544 550
FPR4	U	000004	1	128	
FPR5	U	000005	1	129	
FPR6	U	000006	1	130	
FPR7	U	000007	1	131	
FPR8	U	000008	1	132	294 296 302 350 353 387 389 396 445 447 482 484 491 540 542
FPR9	U	000009	1	133	
GOODPSW	X	0002D0	8	226	223
HELPERS	H	006A40	2	1081	142 187
HEXTRTAB	U	006BB8	16	1249	1090 1094 1098 1102 1106 1182 1186 1190 1194 1198 1210 1214 1218 1222 1226
IMAGE	1	000000	28172	0	
LBFPFCT	U	00000A	1	645	258
LBFPFIN	F	000630	4	634	645 259
LBFPFOF	U	002500	1	674	261 1322
LBFPFOF_GOOD	U	0058C0	1	922	933 1323
LBFPFOF_NUM	U	000005	1	933	1324
LBFPFOT	U	002300	1	672	260 1318
LBFPFOT_GOOD	U	0053C0	1	878	919 1319
LBFPFOT_NUM	U	000014	1	919	1320
LBFPNFCT	U	000008	1	623	252
LBFPNFFL	U	002200	1	670	255 1314
LBFPNFFL_GOOD	U	0051C0	1	858	875 1315
LBFPNFFL_NUM	U	000008	1	875	1316
LBFPNFIN	D	0005F0	8	614	623 253

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
LBFPNFOT	U	002000	1	668	254 1310
LBFPNFOT_GOOD	U	0049C0	1	790	855 1311
LBFPNFOT_NUM	U	000020	1	855	1312
LDEBRF	I	0003CA	4	344	203
LDEBRNF	H	00035C	2	287	201
LONGF	F	00032C	4	257	207
LONGNF	F	00031C	4	251	205
LXDBRF	I	000490	4	439	208
LXDBRNF	I	000412	4	381	206
LXEBRF	I	00055A	4	534	213
LXEBRNF	I	0004DC	4	476	211
MSG	I	006CBA	4	1256	1110 1169 1202 1230
MSGCMD	C	006D02	9	1282	1269 1270
MSGMSG	C	006D0B	95	1283	1263 1280 1261
MSGMVC	I	006CFC	6	1280	1267
MSGOK	I	006CD0	2	1265	1262
MSGRET	I	006CEA	4	1276	1273
MSGSAVE	F	006CF0	4	1279	1259 1276
PCINTCD	H	00008E	2	155	172 1088
PCNOTDTA	I	00020C	4	176	173
PCOLDPSW	U	000150	1	157	174 1092 1096 1100 1104
PGMCK	H	006A40	2	1087	178
PGMCOMMA	C	006AB6	1	1117	1089
PGMPSW	C	006ABC	36	1119	1092 1093 1094 1096 1097 1098 1100 1101 1102 1104 1105 1106
PROGCHK	H	000200	2	171	163
PROGCODE	C	006AB2	4	1116	1088 1090
PROGMSG	C	006A9E	66	1114	1108 1109
PROGPSW	D	000228	8	184	183
R0	U	000000	1	105	176 179 196 198 1108 1161 1167 1200 1228 1232 1256 1259 1261 1263 1265 1276
R1	U	000001	1	106	1109 1130 1134 1136 1168 1201 1229 1270 1280
R10	U	00000A	1	115	200 202 205 207 210 212 288 289 344 345 381 382 439 440 476 477 534 535
R11	U	00000B	1	116	
R12	U	00000C	1	117	142 177 219 292 320 348 366 385 417 443 461 480 512 538 556
R13	U	00000D	1	118	178 201 203 206 208 211 213 220 291 321 347 367 384 418 442 462 479 513 537 557 1112 1140
R14	U	00000E	1	119	181 182 221 222
R15	U	00000F	1	120	141 176 179
R2	U	000002	1	107	288 290 320 344 346 366 381 383 417 439 441 461 476 478 512 534 536 556 1110 1131 1137 1169 1202 1230 1257 1259 1265 1266 1267 1269 1276 1277
R3	U	000003	1	108	288 294 307 313 317 344 350 359 363 381 387 402 409 414 439 445 453 458 476 482 497 504 509 534 540 548 553 1132 1137
R4	U	000004	1	109	1134 1149 1151 1173 1212 1216 1220 1224
R5	U	000005	1	110	1149 1152 1161 1166 1174 1175 1184 1188 1192 1196 1232
R6	U	000006	1	111	1134 1153
R7	U	000007	1	112	289 297 303 308 314 318 345 354 360 364 382 390 391 397 398 403 404 410 411 415 440 448 449 454 455 459 477 485 486 492 493 498 499 505 506 510 535 543 544 549 550 554 1135 1155
R8	U	000008	1	113	289 298 304 309 315 319 345 355 361 365 382 392 399 405 412 416 440 450 456 460 477 487 494 500 507 511 535 545 551 555 1147 1153

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

R9                   U    000009       1   114
SAVER0R5             F    006C90       4  1247  1161  1232
SAVEREGS             F    00023C       4   186   176   179
SBFPFCT              U    00000A       1   603   246   270
SBFPFIN              F    0005C8       4   592   603   247   271
SBFPFOF              U    001300       1   662   249  1306
SBFPFOF_GOOD         U    004880       1   776   787  1307
SBFPFOF_NUM          U    000005       1   787  1308
SBFPFOT              U    001200       1   660   248  1302
SBFPFOT_GOOD         U    004600       1   752   773  1303
SBFPFOT_NUM          U    00000A       1   773  1304
SBFPNFCT             U    000008       1   579   240   264
SBFPNFFL             U    001100       1   658   243  1298
SBFPNFFL_GOOD        U    004400       1   732   749  1299
SBFPNFFL_NUM         U    000008       1   749  1300
SBFPNFIN             F    0005A8       4   570   579   241   265
SBFPNFOT             U    001000       1   656   242  1294
SBFPNFOT_GOOD        U    004000       1   696   729  1295
SBFPNFOT_NUM         U    000010       1   729  1296
SHORTF               F    00030C       4   245   202
SHORTNF              F    0002FC       4   239   200
START                H    000280       2   195   160
STRTLABL             U    000000       1   104   154   157   159   162   170   656   658   660   662   668   670   672   674   680
                                              682   684   686   694
VERIFAIL             I    006B1A       4  1161  1150
VERIFLEN             U    00000C       1  1342  1131
VERIFTAB             F    006D6C       4  1293  1342  1130
VERIFY               I    006B02       2  1147  1135
VERINEXT             I    006B0E       4  1151  1233
VERISUB              H    006AE0       2  1125   220
WANTGOT              C    006C50       6  1240  1179  1207
XBFPFOF              U    003500       1   686   273  1338
XBFPFOF_GOOD         U    006900       1  1068  1079  1339
XBFPFOF_NUM          U    000005       1  1079  1340
XBFPFOT              U    003300       1   684   272  1334
XBFPFOT_GOOD         U    006400       1  1024  1065  1335
XBFPFOT_NUM          U    000014       1  1065  1336
XBFPNFFL             U    003200       1   682   267  1330
XBFPNFFL_GOOD        U    006200       1  1004  1021  1331
XBFPNFFL_NUM         U    000008       1  1021  1332
XBFPNFOT             U    003000       1   680   266  1326
XBFPNFOT_GOOD        U    005A00       1   936  1001  1327
XBFPNFOT_NUM         U    000020       1  1001  1328
XTNDF                F    00034C       4   269   212
XTNDNF               F    00033C       4   263   210
=AL2(L'MSGMSG)       R    006DFE       2  1346  1261
=CL6'Got:  '         C    006E06       6  1348  1207
=CL6'Want: '         C    006E00       6  1347  1179
=H'0'                H    006DFC       2  1345  1256
```

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 0

Image       IMAGE      28172  0000-6E0B  0000-6E0B
  Region               28172  0000-6E0B  0000-6E0B
    CSECT   BFPLDLEN   28172  0000-6E0B  0000-6E0B
```

```
   STMT                                                FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-017-loadl\bfp-017-loadl.asm

** NO ERRORS FOUND **
```