```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                     2 ***********************************************************************
                                                     3 *
                                                     4 *Testcase IEEE SQUARE ROOT
                                                     5 *  Test case capability includes IEEE exceptions trappable and
                                                     6 *  otherwise. Test results, FPCR flags, and any DXC are saved for all
                                                     7 *  tests.
                                                     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-015-sqrt.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 *
                                                    66 * Tests five square root instructions:
                                                    67 *   SQUARE ROOT (extended BFP, RRE)
                                                    68 *   SQUARE ROOT (long BFP, RRE)
                                                    69 *   SQUARE ROOT (long BFP, RXE)
                                                    70 *   SQUARE ROOT (short BFP, RRE)
                                                    71 *   SQUARE ROOT (short 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 BFP basic tests, including traps and NaN propagation
                                                    79 * 2) Short BFP FPC-controlled rounding mode exhaustive tests
                                                    80 * 3) Long BFP basic tests, including traps and NaN propagation
                                                    81 * 4) Long BFP FPC-controlled rounding mode exhaustive tests
                                                    82 * 5) Extended BFP basic tests, including traps and NaN propagation
                                                    83 * 6) Extended BFP FPC-controlled rounding mode exhaustive tests
                                                    84 *
                                                    85 * Two input test sets are provided each for short, long, and extended
                                                    86 * BFP inputs.  The first set covers non-finites, negatives, NaNs, and
                                                    87 * traps on inexact and inexact-incremented.  The second more limited
                                                    88 * set exhaustively tests Square Root with each rounding mode that can
                                                    89 * be specified in the FPC.  Test values are the same for each
                                                    90 * precision.  Interestingly, the square root of 3 is nearer to a lower
                                                    91 * magnitude for all three precisions, while the square root of 5 is
                                                    92 * nearer to a larger magnitude for all three precisions.  The square
                                                    93 * root of 2 does not have this property.
                                                    94 *
                                                    95 * Note: Square Root recognizes only the IEEE exceptions Invalid and
                                                    96 * Inexact.  Neither overflow nor underflow can occur with Square Root.
                                                    97 * For values greater than 1, the result from Square Root is smaller
                                                    98 * than the input.  For values less than 1 and greater than zero, the
                                                    99 * result from square root is larger than the input value.
                                                   100 *
                                                   101 * Also tests the following floating point support instructions
                                                   102 *   LOAD  (Short)
                                                   103 *   LOAD  (Long)
                                                   104 *   LFPC  (Load Floating Point Control Register)
                                                   105 *   SRNMB (Set BFP Rounding Mode 3-bit)
                                                   106 *   STORE (Short)
                                                   107 *   STORE (Long)
                                                   108 *   STFPC (Store Floating Point Control Register)
                                                   109 *
                                                   110 *
                                                   111 ***********************************************************************
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   113 *
                                                   114 *  Note: for compatibility with the z/CMS test rig, do not change
                                                   115 *  or use R11, R14, or R15.  Everything else is fair game.
                                                   116 *
                              00000000  0000844B   117 BFPSQRTS START 0
                              00000000  00000001   118 STRTLABL EQU   *
                              00000000  00000001   119 R0       EQU   0                   Work register for cc extraction
                              00000001  00000001   120 R1       EQU   1
                              00000002  00000001   121 R2       EQU   2                   Holds count of test input values
                              00000003  00000001   122 R3       EQU   3                   Points to next test input value(s)
                              00000004  00000001   123 R4       EQU   4                   Rounding tests inner loop control
                              00000005  00000001   124 R5       EQU   5                   Rounding tests outer loop control
                              00000006  00000001   125 R6       EQU   6                   Rounding tests top of inner loop
                              00000007  00000001   126 R7       EQU   7                   Pointer to next result value(s)
                              00000008  00000001   127 R8       EQU   8                   Pointer to next FPCR result
                              00000009  00000001   128 R9       EQU   9                   Rounding tests top of inner loop
                              0000000A  00000001   129 R10      EQU   10                  Pointer to test address list
                              0000000B  00000001   130 R11      EQU   11                  **Reserved for z/CMS test rig
                              0000000C  00000001   131 R12      EQU   12                  Top of outer loop address in tests
                              0000000D  00000001   132 R13      EQU   13                  Mainline return address
                              0000000E  00000001   133 R14      EQU   14                  **Return address for z/CMS test rig
                              0000000F  00000001   134 R15      EQU   15                  **Base register on z/CMS or Hyperion
                                                   135 *
                                                   136 * Floating Point Register equates to keep the cross reference clean
                                                   137 *
                              00000000  00000001   138 FPR0     EQU   0
                              00000001  00000001   139 FPR1     EQU   1             Input value for RRE instructions
                              00000002  00000001   140 FPR2     EQU   2
                              00000003  00000001   141 FPR3     EQU   3             ..Paired with FPR1 for RRE extended
                              00000004  00000001   142 FPR4     EQU   4
                              00000005  00000001   143 FPR5     EQU   5
                              00000006  00000001   144 FPR6     EQU   6
                              00000007  00000001   145 FPR7     EQU   7
                              00000008  00000001   146 FPR8     EQU   8             Result value from Square Root
                              00000009  00000001   147 FPR9     EQU   9
                              0000000A  00000001   148 FPR10    EQU   10            ..Paired with FPR8 for RRE extended
                              0000000B  00000001   149 FPR11    EQU   11
                              0000000C  00000001   150 FPR12    EQU   12
                              0000000D  00000001   151 FPR13    EQU   13
                              0000000E  00000001   152 FPR14    EQU   14
                              0000000F  00000001   153 FPR15    EQU   15
                                                   154 *
00000000                      00000000             155          USING *,R15
00000000                      00008080             156          USING HELPERS,R12
                                                   157 *
                                                   158 * Above works on real iron (R15=0 after sysclear)
                                                   159 * and in z/CMS (R15 points to start of load module)
                                                   160 *

                                                   162 ***********************************************************************
                                                   163 *
                                                   164 * Low core definitions, Restart PSW, and Program Check Routine.
                                                   165 *
                                                   166 ***********************************************************************
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000000                      00000000  0000008E   168          ORG   STRTLABL+X'8E'      Program check interrution code
0000008E  0000                                     169 PCINTCD  DS    H
                                                   170 *
                              00000150  00000001   171 PCOLDPSW EQU   STRTLABL+X'150'     z/Arch Program check old PSW
                                                   172 *
00000090                      00000090  000001A0   173          ORG   STRTLABL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                        174          DC    X'0000000180000000',AD(START)
                                                   175 *
000001B0                      000001B0  000001D0   176          ORG   STRTLABL+X'1D0'     z/Arch Program check NEW PSW
000001D0  00000000 00000000                        177          DC    X'0000000000000000',AD(PROGCHK)
                                                   178 *
                                                   179 * Program check routine.  If Data Exception, continue execution at
                                                   180 * the instruction following the program check.  Otherwise, hard wait.
                                                   181 * No need to collect data.  All interesting DXC stuff is captured
                                                   182 * in the FPCR.
                                                   183 *
000001E0                      000001E0  00000200   184          ORG   STRTLABL+X'200'
00000200                                           185 PROGCHK  DS    0H             Program check occured...
00000200  9507 F08F                     0000008F   186          CLI   PCINTCD+1,X'07'  Data Exception?
00000204  A774 0004                     0000020C   187          JNE   PCNOTDTA       ..no, hardwait (not sure if R15 is ok)
00000208  B2B2 F150                     00000150   188          LPSWE PCOLDPSW       ..yes, resume program execution

0000020C  900F F23C                     0000023C   190 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                     0000027C   191          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                     00008080   192          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                     0000023C   193          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                     195          LTR   R14,R14        Return address provided?
0000021E  077E                                     196          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                     00000228   197          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                        198 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2E0                     000002E0   199 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                        200 SAVEREGS DC    16F'0'         Registers save area
0000027C  00008080                                 201 AHELPERS DC    A(HELPERS)     Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   203 ***********************************************************************
                                                   204 *
                                                   205 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                   206 *
                                                   207 ***********************************************************************

00000280                                           209 START    DS    0H
00000280  B600 F2F0                     000002F0   210          STCTL R0,R0,CTLR0    Store CR0 to enable AFP
00000284  9604 F2F1                     000002F1   211          OI    CTLR0+1,X'04'  Turn on AFP bit
00000288  B700 F2F0                     000002F0   212          LCTL  R0,R0,CTLR0    Reload updated CR0
                                                   213 *
0000028C  41A0 F2FC                     000002FC   214          LA    R10,SHORTB    Point to short BFP inputs
00000290  4DD0 F35C                     0000035C   215          BAS   R13,SBFPB     Take square root of short BFP values
00000294  41A0 F30C                     0000030C   216          LA    R10,RMSHORTS  Point to short BFP rounding mode tests
00000298  4DD0 F3D2                     000003D2   217          BAS   R13,SBFPRM    Take sqrt of short BFP for rounding tests
                                                   218 *
0000029C  41A0 F31C                     0000031C   219          LA    R10,LONGB     Point to long BFP inputs
000002A0  4DD0 F434                     00000434   220          BAS   R13,LBFPB     Take square root of long BFP values
000002A4  41A0 F32C                     0000032C   221          LA    R10,RMLONGS   Point to long  BFP rounding mode tests
000002A8  4DD0 F4AA                     000004AA   222          BAS   R13,LBFPRM    Take sqrt of long BFP for rounding tests
                                                   223 *
000002AC  41A0 F33C                     0000033C   224          LA    R10,XTNDB     Point to extended BFP inputs
000002B0  4DD0 F508                     00000508   225          BAS   R13,XBFPB    Take square root of extended BFP values
000002B4  41A0 F34C                     0000034C   226          LA    R10,RMXTNDS   Point to ext'd BFP rounding mode tests
000002B8  4DD0 F55E                     0000055E   227          BAS   R13,XBFPRM    Take sqrt of ext'd BFP for rounding tests
                                                   228 *
                                                   229 ***********************************************************************
                                                   230 *                   Verify test results...
                                                   231 ***********************************************************************
                                                   232 *
000002BC  58C0 F27C                     0000027C   233          L     R12,AHELPERS     Get address of helper subroutines
000002C0  4DD0 C0A0                     00008120   234          BAS   R13,VERISUB      Go verify results
000002C4  12EE                                     235          LTR   R14,R14          Was return address provided?
000002C6  077E                                     236          BNZR  R14              Yes, return to z/CMS test rig.
000002C8  B2B2 F2D0                     000002D0   237          LPSWE GOODPSW          Load SUCCESS PSW
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000002D0                                           239          DS    0D            Ensure correct alignment for PSW
000002D0  00020000 00000000                        240 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002E0  00020000 00000000                        241 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                   242 *
000002F0  00000000                                 243 CTLR0    DS    F
000002F4  00000000                                 244 FPCREGNT DC    X'00000000'  FPCR, trap all IEEE exceptions, zero flags
000002F8  F8000000                                 245 FPCREGTR DC    X'F8000000'  FPCR, trap no IEEE exceptions, zero flags
                                                   246 *
                                                   247 * Input values parameter list, four fullwords for each test data set
                                                   248 *      1) Count,
                                                   249 *      2) Address of inputs,
                                                   250 *      3) Address to place results, and
                                                   251 *      4) Address to place DXC/Flags/cc values.
                                                   252 *
000002FC                                           253 SHORTB   DS    0F           Input pairs for short BFP basic tests
000002FC  0000000B                                 254          DC    A(SBFPBCT)
00000300  000005B4                                 255          DC    A(SBFPBIN)
00000304  00001000                                 256          DC    A(SBFPBOT)
00000308  00001100                                 257          DC    A(SBFPBFL)
                                                   258 *
0000030C                                           259 RMSHORTS DS    0F           Input pairs for short BFP rounding testing
0000030C  00000002                                 260          DC    A(SBFPRMCT)
00000310  000005E0                                 261          DC    A(SBFPINRM)
00000314  00001200                                 262          DC    A(SBFPRMO)
00000318  00001400                                 263          DC    A(SBFPRMOF)
                                                   264 *
0000031C                                           265 LONGB    DS    0F           Input pairs for long BFP basic testing
0000031C  0000000B                                 266          DC    A(LBFPBCT)
00000320  000005E8                                 267          DC    A(LBFPBIN)
00000324  00003000                                 268          DC    A(LBFPBOT)
00000328  00003200                                 269          DC    A(LBFPBFL)
                                                   270 *
0000032C                                           271 RMLONGS  DS    0F           Input pairs for long BFP rounding testing
0000032C  00000002                                 272          DC    A(LBFPRMCT)
00000330  00000640                                 273          DC    A(LBFPINRM)
00000334  00003400                                 274          DC    A(LBFPRMO)
00000338  00003700                                 275          DC    A(LBFPRMOF)
                                                   276 *
0000033C                                           277 XTNDB    DS    0F           Inputs for extended BFP basic testing
0000033C  0000000B                                 278          DC    A(XBFPBCT)
00000340  00000650                                 279          DC    A(XBFPBIN)
00000344  00005000                                 280          DC    A(XBFPBOT)
00000348  00005400                                 281          DC    A(XBFPBFL)
                                                   282 *
0000034C                                           283 RMXTNDS  DS    0F           Inputs for ext'd BFP non-finite testing
0000034C  00000002                                 284          DC    A(XBFPRMCT)
00000350  00000700                                 285          DC    A(XBFPINRM)
00000354  00005500                                 286          DC    A(XBFPRMO)
00000358  00005A00                                 287          DC    A(XBFPRMOF)
                                                   288 *
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   290 ****************************************************************************
                                                   291 *
                                                   292 * Take square roots using provided short BFP inputs.  This set of tests
                                                   293 * checks NaN propagation, operations on values that are not finite
                                                   294 * numbers, and other basic tests.  This set generates results that can
                                                   295 * be validated against Figure 19-17 on page 19-21 of SA22-7832-10.
                                                   296 *
                                                   297 * Four results are generated for each input: one RRE with all
                                                   298 * exceptions non-trappable, a second RRE with all exceptions trappable,
                                                   299 * a third RXE with all exceptions non-trappable, a fourth RXE with all
                                                   300 * exceptions trappable,
                                                   301 *
                                                   302 * The square root and FPCR are stored for each result.
                                                   303 *
                                                   304 ****************************************************************************

0000035C                                           306 SBFPB    DS    0H            BFP Short basic tests
0000035C  9823 A000                     00000000   307          LM    R2,R3,0(R10)  Get count and address of dividendd values
00000360  9878 A008                     00000008   308          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000364  1222                                     309          LTR   R2,R2         Any test cases?
00000366  078D                                     310          BZR   R13           ..No, return to caller
00000368  0DC0                                     311          BASR  R12,0         Set top of loop
                                                   312 *
0000036A  B374 0080                                313          LZER  FPR8          Zero result register
0000036E  7810 3000                     00000000   314          LE    FPR1,0(,R3)   Get short BFP input
00000372  B29D F2F4                     000002F4   315          LFPC  FPCREGNT      Set exceptions non-trappable
00000376  B314 0081                                316          SQEBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
0000037A  7080 7000                     00000000   317          STE   FPR8,0(,R7)   Store short BFP square root
0000037E  B29C 8000                     00000000   318          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                   319 *
00000382  B374 0080                                320          LZER  FPR8          Zero result register
00000386  B29D F2F8                     000002F8   321          LFPC  FPCREGTR      Set exceptions trappable
0000038A  B314 0081                                322          SQEBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
0000038E  7080 7004                     00000004   323          STE   FPR8,4(,R7)   Store short BFP square root
00000392  B29C 8004                     00000004   324          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                                   325 *
00000396  B374 0080                                326          LZER  FPR8          Zero result register
0000039A  B29D F2F4                     000002F4   327          LFPC  FPCREGNT      Set exceptions non-trappable
0000039E  ED80 3000 0014                00000000   328          SQEB  FPR8,0(,R3)   Take square root, place in FPR8 RXE
000003A4  7080 7008                     00000008   329          STE   FPR8,8(,R7)   Store short BFP square root
000003A8  B29C 8008                     00000008   330          STFPC 8(R8)         Store resulting FPCR flags and DXC
                                                   331 *
000003AC  B374 0080                                332          LZER  FPR8          Zero result register
000003B0  B29D F2F8                     000002F8   333          LFPC  FPCREGTR      Set exceptions trappable
000003B4  ED80 3000 0014                00000000   334          SQEB  FPR8,0(,R3)   Take square root, place in FPR8 RXE
000003BA  7080 700C                     0000000C   335          STE   FPR8,12(,R7)  Store short BFP square root
000003BE  B29C 800C                     0000000C   336          STFPC 12(R8)        Store resulting FPCR flags and DXC
                                                   337 *
000003C2  4170 7010                     00000010   338          LA    R7,16(,R7)    Point to next Square Root result area
000003C6  4180 8010                     00000010   339          LA    R8,16(,R8)    Point to next Square Root FPCR area
000003CA  4130 3004                     00000004   340          LA    R3,4(,R3)     Point to next input value
000003CE  062C                                     341          BCTR  R2,R12        Convert next input value.
                                                   342 *
000003D0  07FD                                     343          BR    R13           All converted; return.
                                                   344 *
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   346 ***********************************************************************
                                                   347 *
                                                   348 * Perform Square Root using provided short BFP inputs.  This set of
                                                   349 * tests exhaustively tests all rounding modes available for Square
                                                   350 * Root. The rounding mode can only be specified in the FPC.
                                                   351 *
                                                   352 * All five FPC rounding modes are tested because the preceeding tests,
                                                   353 * using rounding mode RNTE, do not often create results that require
                                                   354 * rounding.
                                                   355 *
                                                   356 * Two results are generated for each input and rounding mode: one RRE
                                                   357 * and one RXE.  Traps are disabled for all rounding mode tests.
                                                   358 *
                                                   359 * The quotient and FPCR contents are stored for each test.
                                                   360 *
                                                   361 ***********************************************************************


000003D2  9823 A000                 00000000   363 SBFPRM   LM    R2,R3,0(R10)  Get count and address of test input values
000003D6  9878 A008                 00000008   364          LM    R7,R8,8(R10)  Get address of result area and flag area.
000003DA  1222                                 365          LTR   R2,R2         Any test cases?
000003DC  078D                                 366          BZR   R13           ..No, return to caller
000003DE  1711                                 367          XR    R1,R1         Zero register 1 for use in IC/STC/indexing
000003E0  0DC0                                 368          BASR  R12,0         Set top of test case loop
                                               369
000003E2  4150 0005                 00000005   370          LA    R5,FPCMCT     Get count of FPC modes to be tested
000003E6  0D90                                 371          BASR  R9,0          Set top of rounding mode outer loop
                                               372 *
000003E8  4315 F5AD                 000005AD   373          IC    R1,FPCMODES-L'FPCMODES(R5)  Get next FPC mode
                                               374 *
000003EC  B29D F2F4                 000002F4   375          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003F0  B2B8 1000                 00000000   376          SRNMB 0(R1)         Set FPC Rounding Mode
000003F4  7810 3000                 00000000   377          LE    FPR1,0(,R3)   Get short BFP input value
000003F8  B314 0081                            378          SQEBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
000003FC  7080 7000                 00000000   379          STE   FPR8,0(,R7)   Store short BFP quotient
00000400  B29C 8000                 00000000   380          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                               381 *
00000404  B29D F2F4                 000002F4   382          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000408  B2B8 1000                 00000000   383          SRNMB 0(R1)         Set FPC Rounding Mode
0000040C  ED80 3000 0014            00000000   384          SQEB  FPR8,0(,R3)   Take square root of value into FPR8 RXE
00000412  7080 7004                 00000004   385          STE   FPR8,4(,R7)   Store short BFP quotient
00000416  B29C 8004                 00000004   386          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                               387 *
0000041A  4170 7008                 00000008   388          LA    R7,8(,R7)     Point to next square root result
0000041E  4180 8008                 00000008   389          LA    R8,8(,R8)     Point to next FPCR result area
                                               390 *
00000422  0659                                 391          BCTR  R5,R9         Iterate to next FPC mode
                                               392 *
                                               393 * End of FPC modes to be tested.  Advance to next test case.
                                               394 *
00000424  4130 3004                 00000004   395          LA    R3,4(,R3)     Point to next input value
00000428  4170 7008                 00000008   396          LA    R7,8(,R7)     Skip to start of next result area
0000042C  4180 8008                 00000008   397          LA    R8,8(,R8)     Skip to start of next FPCR result area
00000430  062C                                 398          BCTR  R2,R12        Divide next input value lots of times
                                               399 *
00000432  07FD                                 400          BR    R13           All converted; return.
```

```
  LOC       OBJECT CODE      ADDR1     ADDR2    STMT

                                                   402 ********************************************************************
                                                   403 *
                                                   404 * Take square roots using provided long BFP inputs.  This set of tests
                                                   405 * checks NaN propagation, operations on values that are not finite
                                                   406 * numbers, and other basic tests.  This set generates results that can
                                                   407 * be validated against Figure 19-17 on page 19-21 of SA22-7832-10.
                                                   408 *
                                                   409 * Four results are generated for each input: one RRE with all
                                                   410 * exceptions non-trappable, a second RRE with all exceptions trappable,
                                                   411 * a third RXE with all exceptions non-trappable, a fourth RXE with all
                                                   412 * exceptions trappable,
                                                   413 *
                                                   414 * The square root and FPCR are stored for each result.
                                                   415 *
                                                   416 ********************************************************************

00000434                                           418 LBFPB    DS    0H            BFP long basic tests
00000434  9823 A000                     00000000   419          LM    R2,R3,0(R10)  Get count and address of dividendd values
00000438  9878 A008                     00000008   420          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000043C  1222                                     421          LTR   R2,R2         Any test cases?
0000043E  078D                                     422          BZR   R13           ..No, return to caller
00000440  0DC0                                     423          BASR  R12,0         Set top of loop
                                                   424 *
00000442  B375 0080                                425          LZDR  FPR8          Zero result register
00000446  6810 3000                     00000000   426          LD    FPR1,0(,R3)   Get long BFP input
0000044A  B29D F2F4                     000002F4   427          LFPC  FPCREGNT      Set exceptions non-trappable
0000044E  B315 0081                                428          SQDBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
00000452  6080 7000                     00000000   429          STD   FPR8,0(,R7)   Store long BFP square root
00000456  B29C 8000                     00000000   430          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                   431 *
0000045A  B375 0080                                432          LZDR  FPR8          Zero result register
0000045E  B29D F2F8                     000002F8   433          LFPC  FPCREGTR      Set exceptions trappable
00000462  B315 0081                                434          SQDBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
00000466  6080 7008                     00000008   435          STD   FPR8,8(,R7)   Store long BFP square root
0000046A  B29C 8004                     00000004   436          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                                   437 *
0000046E  B375 0080                                438          LZDR  FPR8          Zero result register
00000472  B29D F2F4                     000002F4   439          LFPC  FPCREGNT      Set exceptions non-trappable
00000476  ED80 3000 0015                00000000   440          SQDB  FPR8,0(,R3)   Take square root, place in FPR8 RXE
0000047C  6080 7010                     00000010   441          STD   FPR8,16(,R7)  Store long BFP square root
00000480  B29C 8008                     00000008   442          STFPC 8(R8)         Store resulting FPCR flags and DXC
                                                   443 *
00000484  B375 0080                                444          LZDR  FPR8          Zero result register
00000488  B29D F2F8                     000002F8   445          LFPC  FPCREGTR      Set exceptions trappable
0000048C  ED80 3000 0015                00000000   446          SQDB  FPR8,0(,R3)   Take square root, place in FPR8 RXE
00000492  6080 7018                     00000018   447          STD   FPR8,24(,R7)  Store short BFP square root
00000496  B29C 800C                     0000000C   448          STFPC 12(R8)        Store resulting FPCR flags and DXC
                                                   449 *
0000049A  4170 7020                     00000020   450          LA    R7,32(,R7)    Point to next Square Root result area
0000049E  4180 8010                     00000010   451          LA    R8,16(,R8)    Point to next Square Root FPCR area
000004A2  4130 3008                     00000008   452          LA    R3,8(,R3)     Point to next input value
000004A6  062C                                     453          BCTR  R2,R12        Convert next input value.
                                                   454 *
000004A8  07FD                                     455          BR    R13           All converted; return.
                                                   456 *
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   458 ***********************************************************************
                                                   459 *
                                                   460 * Perform Square Root using provided long BFP inputs.  This set of
                                                   461 * tests exhaustively tests all rounding modes available for Square
                                                   462 * Root. The rounding mode can only be specified in the FPC.
                                                   463 *
                                                   464 * All five FPC rounding modes are tested because the preceeding tests,
                                                   465 * using rounding mode RNTE, do not often create results that require
                                                   466 * rounding.
                                                   467 *
                                                   468 * Two results are generated for each input and rounding mode: one RRE
                                                   469 * and one RXE.  Traps are disabled for all rounding mode tests.
                                                   470 *
                                                   471 * The quotient and FPCR contents are stored for each test.
                                                   472 *
                                                   473 ***********************************************************************

000004AA  9823 A000                     00000000   475 LBFPRM   LM    R2,R3,0(R10)  Get count and address of test input values
000004AE  9878 A008                     00000008   476          LM    R7,R8,8(R10)  Get address of result area and flag area.
000004B2  1222                                     477          LTR   R2,R2         Any test cases?
000004B4  078D                                     478          BZR   R13           ..No, return to caller
000004B6  1711                                     479          XR    R1,R1         Zero register 1 for use in IC/STC/indexing
000004B8  0DC0                                     480          BASR  R12,0         Set top of test case loop
                                                   481
000004BA  4150 0005                     00000005   482          LA    R5,FPCMCT     Get count of FPC modes to be tested
000004BE  0D90                                     483          BASR  R9,0          Set top of rounding mode outer loop
                                                   484 *
000004C0  4315 F5AD                     000005AD   485          IC    R1,FPCMODES-L'FPCMODES(R5)  Get next FPC mode
                                                   486 *
000004C4  B29D F2F4                     000002F4   487          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004C8  B2B8 1000                     00000000   488          SRNMB 0(R1)         Set FPC Rounding Mode
000004CC  6810 3000                     00000000   489          LD    FPR1,0(,R3)   Get long BFP input value
000004D0  B315 0081                                490          SQDBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
000004D4  6080 7000                     00000000   491          STD   FPR8,0(,R7)   Store long BFP quotient
000004D8  B29C 8000                     00000000   492          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                   493 *
000004DC  B29D F2F4                     000002F4   494          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004E0  B2B8 1000                     00000000   495          SRNMB 0(R1)         Set FPC Rounding Mode
000004E4  ED80 3000 0015                00000000   496          SQDB  FPR8,0(,R3)   Take square root of value into FPR8 RXE
000004EA  6080 7008                     00000008   497          STD   FPR8,8(,R7)   Store short BFP quotient
000004EE  B29C 8004                     00000004   498          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                                   499 *
000004F2  4170 7010                     00000010   500          LA    R7,16(,R7)    Point to next square root result
000004F6  4180 8008                     00000008   501          LA    R8,8(,R8)     Point to next FPCR result area
                                                   502 *
000004FA  0659                                     503          BCTR  R5,R9         Iterate to next FPC mode
                                                   504 *
                                                   505 * End of FPC modes to be tested.  Advance to next test case.
                                                   506 *
000004FC  4130 3008                     00000008   507          LA    R3,8(,R3)     Point to next input value
00000500  4180 8008                     00000008   508          LA    R8,8(,R8)     Skip to start of next FPCR result area
00000504  062C                                     509          BCTR  R2,R12        Divide next input value lots of times
                                                   510 *
00000506  07FD                                     511          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   513 ***********************************************************************
                                                   514 *
                                                   515 * Take square roots using provided extended BFP inputs.  This set of
                                                   516 * tests checks NaN propagation, operations on values that are not
                                                   517 * finite numbers, and other basic tests.  This set generates results
                                                   518 * that can be validated against Figure 19-17 on page 19-21 of
                                                   519 *  SA22-7832-10.
                                                   520 *
                                                   521 * Four results are generated for each input: one RRE with all
                                                   522 * exceptions non-trappable, a second RRE with all exceptions trappable,
                                                   523 * a third RXE with all exceptions non-trappable, a fourth RXE with all
                                                   524 * exceptions trappable,
                                                   525 *
                                                   526 * The square root and FPCR are stored for each result.
                                                   527 *
                                                   528 ***********************************************************************

00000508                                           530 XBFPB    DS    0H            BFP extended basic tests
00000508  9823 A000            00000000            531          LM    R2,R3,0(R10)  Get count and address of dividendd values
0000050C  9878 A008            00000008            532          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000510  1222                                     533          LTR   R2,R2         Any test cases?
00000512  078D                                     534          BZR   R13           ..No, return to caller
00000514  0DC0                                     535          BASR  R12,0         Set top of loop
                                                   536 *
00000516  B376 0080                                537          LZXR  FPR8          Zero result register
0000051A  6810 3000            00000000            538          LD    FPR1,0(,R3)   Get extended BFP input part 1
0000051E  6830 3008            00000008            539          LD    FPR3,8(,R3)   Get extended BFP input part 2
00000522  B29D F2F4            000002F4            540          LFPC  FPCREGNT      Set exceptions non-trappable
00000526  B316 0081                                541          SQXBR FPR8,FPR1     Take square root of FPR1 into FPR8-10 RRE
0000052A  6080 7000            00000000            542          STD   FPR8,0(,R7)   Store extended BFP square root part 1
0000052E  60A0 7008            00000008            543          STD   FPR10,8(,R7)  Store extended BFP square root part 2
00000532  B29C 8000            00000000            544          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                   545 *
00000536  B376 0080                                546          LZXR  FPR8          Zero result register
0000053A  B29D F2F8            000002F8            547          LFPC  FPCREGTR      Set exceptions trappable
0000053E  B316 0081                                548          SQXBR FPR8,FPR1     Take square root of FPR1 into FPR8-10 RRE
00000542  6080 7010            00000010            549          STD   FPR8,16(,R7)  Store extended BFP square root part 1
00000546  60A0 7018            00000018            550          STD   FPR10,24(,R7) Store extended BFP square root part 2
0000054A  B29C 8004            00000004            551          STFPC 4(R8)         Store resulting FPCR flags and DXC
                                                   552 *
0000054E  4170 7020            00000020            553          LA    R7,32(,R7)    Point to next Square Root result area
00000552  4180 8010            00000010            554          LA    R8,16(,R8)    Point to next Square Root FPCR area
00000556  4130 3010            00000010            555          LA    R3,16(,R3)    Point to next input value
0000055A  062C                                     556          BCTR  R2,R12        Convert next input value.
                                                   557 *
0000055C  07FD                                     558          BR    R13           All converted; return.
                                                   559 *
  LOC       OBJECT CODE       ADDR1     ADDR2    STMT

                                                   561 ***********************************************************************
                                                   562 *
                                                   563 * Perform Square Root using provided extended BFP inputs.  This set of
                                                   564 * tests exhaustively tests all rounding modes available for Square
                                                   565 * Root. The rounding mode can only be specified in the FPC.
                                                   566 *
                                                   567 * All five FPC rounding modes are tested because the preceeding tests,
                                                   568 * using rounding mode RNTE, do not often create results that require
                                                   569 * rounding.
                                                   570 *
                                                   571 * Two results are generated for each input and rounding mode: one RRE
                                                   572 * and one RXE.  Traps are disabled for all rounding mode tests.
                                                   573 *
                                                   574 * The quotient and FPCR contents are stored for each test.
                                                   575 *
                                                   576 ***********************************************************************

0000055E  9823 A000                     00000000   578 XBFPRM   LM    R2,R3,0(R10)  Get count and address of test input values
00000562  9878 A008                     00000008   579          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000566  1222                                     580          LTR   R2,R2         Any test cases?
00000568  078D                                     581          BZR   R13           ..No, return to caller
0000056A  1711                                     582          XR    R1,R1         Zero register 1 for use in IC/STC/indexing
0000056C  0DC0                                     583          BASR  R12,0         Set top of test case loop
                                                   584
0000056E  4150 0005                     00000005   585          LA    R5,FPCMCT     Get count of FPC modes to be tested
00000572  0D90                                     586          BASR  R9,0          Set top of rounding mode outer loop
                                                   587 *
00000574  4315 F5AD                     000005AD   588          IC    R1,FPCMODES-L'FPCMODES(R5)  Get next FPC mode
                                                   589 *
00000578  B29D F2F4                     000002F4   590          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000057C  B2B8 1000                     00000000   591          SRNMB 0(R1)         Set FPC Rounding Mode
00000580  6810 3000                     00000000   592          LD    FPR1,0(,R3)   Get long BFP input value part 1
00000584  6830 3008                     00000008   593          LD    FPR3,8(,R3)   Get long BFP input value part 2
00000588  B316 0081                                594          SQXBR FPR8,FPR1     Take square root of FPR1 into FPR8 RRE
0000058C  6080 7000                     00000000   595          STD   FPR8,0(,R7)   Store extended BFP root part 1
00000590  60A0 7008                     00000008   596          STD   FPR10,8(,R7)  Store extended BFP root part 2
00000594  B29C 8000                     00000000   597          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                   598 *
00000598  4170 7010                     00000010   599          LA    R7,16(,R7)    Point to next square root result
0000059C  4180 8004                     00000004   600          LA    R8,4(,R8)     Point to next FPCR result area
                                                   601 *
000005A0  0659                                     602          BCTR  R5,R9         Iterate to next FPC mode
                                                   603 *
                                                   604 * End of FPC modes to be tested.  Advance to next test case.
                                                   605 *
000005A2  4130 3010                     00000010   606          LA    R3,16(,R3)     Point to next input value
000005A6  4180 800C                     0000000C   607          LA    R8,12(,R8)    Skip to start of next FPCR result area
000005AA  062C                                     608          BCTR  R2,R12        Divide next input value lots of times
                                                   609 *
000005AC  07FD                                     610          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   612 ***********************************************************************
                                                   613 *
                                                   614 * Table of FPC rounding modes to test quotient rounding modes.
                                                   615 *
                                                   616 * The Set BFP Rounding Mode does allow specification of the FPC
                                                   617 * rounding mode as an address, so we shall index into a table of
                                                   618 * BFP rounding modes without bothering with Execute.
                                                   619 *
                                                   620 ***********************************************************************


                                                   622 *
                                                   623 * Rounding modes that may be set in the FPCR.  The FPCR controls
                                                   624 * rounding of the quotient.
                                                   625 *
                                                   626 * These are indexed directly by the loop counter, which counts down.
                                                   627 * So the modes are listed in reverse order here.
                                                   628 *
000005AE                                           629 FPCMODES DS    0C
000005AE  07                                       630          DC    AL1(7)              RFS, Round for shorter precision
000005AF  03                                       631          DC    AL1(3)              RM, Round to -infinity
000005B0  02                                       632          DC    AL1(2)              RP, Round to +infinity
000005B1  01                                       633          DC    AL1(1)              RZ, Round to zero
000005B2  00                                       634          DC    AL1(0)              RNTE, Round to Nearest, ties to even
                              00000005  00000001   635 FPCMCT   EQU   *-FPCMODES          Count of FPC Modes to be tested
                                                   636 *
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   638 ************************************************************************
                                                   639 *
                                                   640 * Short BFP test data sets for Square Root testing.
                                                   641 *
                                                   642 * The first test data set is used for tests of basic functionality,
                                                   643 * NaN propagation, and results from operations involving other than
                                                   644 * finite numbers.
                                                   645 *
                                                   646 * The second test data set is used for testing boundary conditions
                                                   647 * using finite non-zero values.  Each possible type of result (normal,
                                                   648 * scaled, etc) is created by members of this test data set.
                                                   649 *
                                                   650 * The third test data set is used for exhaustive testing of final
                                                   651 * results across the five rounding modes available for the Square Root
                                                   652 * instruction.
                                                   653 *
                                                   654 ************************************************************************

                                                   656 ************************************************************************
                                                   657 *
                                                   658 * First input test data set, to test operations using non-finite or
                                                   659 * zero inputs.  Member values chosen to validate part 1 of Figure 19-17
                                                   660 * on page 19-21 of SA22-7832-10.
                                                   661 *
                                                   662 ************************************************************************

000005B4                                           664 SBFPBIN  DS    0F                Inputs for short BFP basic tests
000005B4  FF800000                                 665          DC    X'FF800000'         -inf
000005B8  C0800000                                 666          DC    X'C0800000'         -4.0
000005BC  80000000                                 667          DC    X'80000000'         -0
000005C0  00000000                                 668          DC    X'00000000'         +0
000005C4  40800000                                 669          DC    X'40800000'         +4.0
000005C8  7F800000                                 670          DC    X'7F800000'         +inf
000005CC  FFCB0000                                 671          DC    X'FFCB0000'         -QNaN
000005D0  7F8A0000                                 672          DC    X'7F8A0000'         +SNaN
000005D4  40400000                                 673          DC    X'40400000'         +3.0    Inexact, truncated
000005D8  40A00000                                 674          DC    X'40A00000'         +5.0    Inexact, incremented
000005DC  3D800000                                 675          DC    X'3D800000'         +0.0625 exact, expect 0.25
                              0000000B  00000001   676 SBFPBCT  EQU   (*-SBFPBIN)/4     Count of short BFP in list

                                                   678 ************************************************************************
                                                   679 *
                                                   680 * Second input test data set.  These are finite positive values
                                                   681 * intended to test all combinations of rounding mode for a given
                                                   682 * result.  Values are chosen to create a requirement to round to the
                                                   683 * target precision after the computation
                                                   684 *
                                                   685 ************************************************************************

000005E0                                           687 SBFPINRM DS    0F                Inputs for short BFP rounding testing
                                                   688 *
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000005E0  40400000                                 689          DC    X'40400000'         +3.0    Inexact, truncated
000005E4  40A00000                                 690          DC    X'40A00000'         +5.0    Inexact, incremented
                                                   691 *
                              00000002  00000001   692 SBFPRMCT EQU   (*-SBFPINRM)/4    Count of short BFP rounding tests
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   694 ***********************************************************************
                                                   695 *
                                                   696 * Long BFP test data sets for Divide testing.
                                                   697 *
                                                   698 * The first test data set is used for tests of basic functionality,
                                                   699 * NaN propagation, and results from operations involving other than
                                                   700 * finite numbers.
                                                   701 *
                                                   702 * The second test data set is used for testing boundary conditions
                                                   703 * using finite non-zero values.  Each possible type of result (normal,
                                                   704 * scaled, etc) is created by members of this test data set.
                                                   705 *
                                                   706 * The third test data set is used for exhaustive testing of final
                                                   707 * results across the five rounding modes available for the Square Root
                                                   708 * instruction.
                                                   709 *
                                                   710 ***********************************************************************


                                                   712 ***********************************************************************
                                                   713 *
                                                   714 * First input test data set, to test operations using non-finite or
                                                   715 * zero inputs.  Member values chosen to validate part 1 of Figure 19-17
                                                   716 * on page 19-21 of SA22-7832-10.
                                                   717 *
                                                   718 ***********************************************************************


000005E8                                           720 LBFPBIN  DS    0D                   Inputs for long BFP testing
000005E8  FFF00000 00000000                        721          DC    X'FFF0000000000000'         -inf
000005F0  C0100000 00000000                        722          DC    X'C010000000000000'         -4.0
000005F8  80000000 00000000                        723          DC    X'8000000000000000'         -0
00000600  00000000 00000000                        724          DC    X'0000000000000000'         +0
00000608  40100000 00000000                        725          DC    X'4010000000000000'         +4.0
00000610  7FF00000 00000000                        726          DC    X'7FF0000000000000'         +inf
00000618  FFF8B000 00000000                        727          DC    X'FFF8B00000000000'         -QNaN
00000620  7FF0A000 00000000                        728          DC    X'7FF0A00000000000'         +SNaN
00000628  40080000 00000000                        729          DC    X'4008000000000000'         +3.0 Inexact, truncated
00000630  40140000 00000000                        730          DC    X'4014000000000000'         +5.0 Inexact, incremented
00000638  3FB00000 00000000                        731          DC    X'3FB0000000000000'         +0.0625 exact, expect 0.25
                              0000000B  00000001   732 LBFPBCT  EQU   (*-LBFPBIN)/8        Count of long BFP in list


                                                   734 ***********************************************************************
                                                   735 *
                                                   736 * Second input test data set.  These are finite positive values
                                                   737 * intended to test all combinations of rounding mode for a given
                                                   738 * result.  Values are chosen to create a requirement to round to the
                                                   739 * target precision after the computation
                                                   740 *
                                                   741 ***********************************************************************

00000640                                           743 LBFPINRM DS    0F
                                                   744 *
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000640  40080000 00000000                        745          DC    X'4008000000000000'         +3.0 Inexact, truncated
00000648  40140000 00000000                        746          DC    X'4014000000000000'         +5.0 Inexact, incremented
                                                   747 *
                              00000002  00000001   748 LBFPRMCT EQU   (*-LBFPINRM)/8    Count of long BFP rounding tests * 8
```

```
                                                   750 *********************************************************************
                                                   751 *
                                                   752 * Extended BFP test data sets for Divide testing.
                                                   753 *
                                                   754 * The first test data set is used for tests of basic functionality,
                                                   755 * NaN propagation, and results from operations involving other than
                                                   756 * finite numbers.
                                                   757 *
                                                   758 * The second test data set is used for testing boundary conditions
                                                   759 * using finite non-zero values.  Each possible type of result (normal,
                                                   760 * scaled, etc) is created by members of this test data set.
                                                   761 *
                                                   762 * The third test data set is used for exhaustive testing of final
                                                   763 * results across the five rounding modes available for the Square Root
                                                   764 * instruction.
                                                   765 *
                                                   766 *********************************************************************


                                                   768 *********************************************************************
                                                   769 *
                                                   770 * First input test data set, to test operations using non-finite or
                                                   771 * zero inputs.  Member values chosen to validate part 1 of Figure 19-17
                                                   772 * on page 19-21 of SA22-7832-10.
                                                   773 *
                                                   774 *********************************************************************


00000650                                           776 XBFPBIN  DS    0D                Inputs for extended BFP testing
00000650  FFFF0000 00000000                        777          DC    X'FFFF0000000000000000000000000000' -inf
00000660  C0010000 00000000                        778          DC    X'C0010000000000000000000000000000' -4.0
00000670  80000000 00000000                        779          DC    X'80000000000000000000000000000000' -0
00000680  00000000 00000000                        780          DC    X'00000000000000000000000000000000' +0
00000690  40010000 00000000                        781          DC    X'40010000000000000000000000000000' +4.0
000006A0  7FFF0000 00000000                        782          DC    X'7FFF0000000000000000000000000000' +inf
000006B0  FFFF8B00 00000000                        783          DC    X'FFFF8B00000000000000000000000000' -QNaN
000006C0  7FFF0A00 00000000                        784          DC    X'7FFF0A00000000000000000000000000' +SNaN
000006D0  40008000 00000000                        785          DC    X'40008000000000000000000000000000' +3.0 Inexact, trunc.
000006E0  40014000 00000000                        786          DC    X'40014000000000000000000000000000' +5.0 Inexact, incre.
000006F0  3FFB0000 00000000                        787          DC    X'3FFB0000000000000000000000000000' +0.0625 expect 0.25
                              0000000B  00000001   788 XBFPBCT  EQU   (*-XBFPBIN)/16    Count of extended BFP in list


                                                   790 *********************************************************************
                                                   791 *
                                                   792 * Second input test data set.  These are finite positive values
                                                   793 * intended to test all combinations of rounding mode for a given
                                                   794 * result.  Values are chosen to create a requirement to round to the
                                                   795 * target precision after the computation
                                                   796 *
                                                   797 *********************************************************************

00000700                                           799 XBFPINRM DS    0D
                                                   800 *
```

```
00000700  40008000 00000000                        801          DC    X'40008000000000000000000000000000' +3.0 Inexact, trunc.
00000710  40014000 00000000                        802          DC    X'40014000000000000000000000000000' +5.0 Inexact, incre.
                                                   803 *
                              00000002  00000001   804 XBFPRMCT EQU   (*-XBFPINRM)/16    Count of long BFP rounding tests
```

```
  LOC       OBJECT CODE       ADDR1     ADDR2    STMT

                                                   806 ***********************************************************************
                                                   807 *                 ACTUAL results saved here
                                                   808 ***********************************************************************
                                                   809 *
                                                   810 *               Locations for ACTUAL results
                                                   811 *
                                                   812 *
                              00001000  00000001   813 SBFPBOT  EQU   STRTLABL+X'1000'    Integer short non-finite BFP results
                                                   814 *                                  ..room for 16 tests, 8 used
                              00001100  00000001   815 SBFPBFL  EQU   STRTLABL+X'1100'    FPCR flags and DXC from short BFP
                                                   816 *                                  ..room for 16 tests, 8 used
                                                   817 *
                              00001200  00000001   818 SBFPRMO  EQU   STRTLABL+X'1200'    Short BFP rounding mode test results
                                                   819 *                                  ..Room for 16, 2 used.
                              00001400  00000001   820 SBFPRMOF EQU   STRTLABL+X'1400'    Short BFP rounding mode FPCR results
                                                   821 *                                  ..Room for 16, 2 used.
                                                   822 *                                  ..next location starts at X'1600'
                                                   823 *
                              00003000  00000001   824 LBFPBOT  EQU   STRTLABL+X'3000'    Integer long non-finite BFP results
                                                   825 *                                  ..room for 16 tests, 8 used
                              00003200  00000001   826 LBFPBFL  EQU   STRTLABL+X'3200'    FPCR flags and DXC from long BFP
                                                   827 *                                  ..room for 16 tests, 8 used
                                                   828 *
                              00003400  00000001   829 LBFPRMO  EQU   STRTLABL+X'3400'    Long BFP rounding mode test results
                                                   830 *                                  ..Room for 16, 4 used.
                              00003700  00000001   831 LBFPRMOF EQU   STRTLABL+X'3700'    Long BFP rounding mode FPCR results
                                                   832 *                                  ..Room for 16, 4 used.
                                                   833 *                                  ..next location starts at X'3800'
                                                   834 *
                              00005000  00000001   835 XBFPBOT  EQU   STRTLABL+X'5000'    Integer ext'd non-finite BFP results
                                                   836 *                                  ..room for 16 tests, 8 used
                              00005400  00000001   837 XBFPBFL  EQU   STRTLABL+X'5400'    FPCR flags and DXC from ext'd BFP
                                                   838 *                                  ..room for 16 tests, 8 used
                                                   839 *
                              00005500  00000001   840 XBFPRMO  EQU   STRTLABL+X'5500'    Ext'd BFP rounding mode test results
                                                   841 *                                  ..Room for 16, 4 used.
                              00005A00  00000001   842 XBFPRMOF EQU   STRTLABL+X'5A00'    Ext'd BFP rounding mode FPCR results
                                                   843 *                                  ..Room for 16, 4 used.
                                                   844 *                                  ..next location starts at X'5B00'
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   846 ***********************************************************************
                                                   847 *                    EXPECTED results
                                                   848 ***********************************************************************
                                                   849 *
00000720                      00000720  00006000   850         ORG   STRTLABL+X'6000'   (past end of actual results)
                                                   851 *
                              00006000  00000001   852 SBFPBOT_GOOD EQU *
00006000  E2D8C5C2 D961E2D8                        853  DC CL48'SQEBR/SQEB -inf'
00006030  7FC00000 00000000                        854  DC XL16'7FC00000000000007FC0000000000000'
00006040  E2D8C5C2 D961E2D8                        855  DC CL48'SQEBR/SQEB -4'
00006070  7FC00000 00000000                        856  DC XL16'7FC00000000000007FC0000000000000'
00006080  E2D8C5C2 D961E2D8                        857  DC CL48'SQEBR/SQEB -0'
000060B0  80000000 80000000                        858  DC XL16'80000000800000008000000080000000'
000060C0  E2D8C5C2 D961E2D8                        859  DC CL48'SQEBR/SQEB +0'
000060F0  00000000 00000000                        860  DC XL16'00000000000000000000000000000000'
00006100  E2D8C5C2 D961E2D8                        861  DC CL48'SQEBR/SQEB +4'
00006130  40000000 40000000                        862  DC XL16'40000000400000004000000040000000'
00006140  E2D8C5C2 D961E2D8                        863  DC CL48'SQEBR/SQEB +inf'
00006170  7F800000 7F800000                        864  DC XL16'7F8000007F8000007F8000007F800000'
00006180  E2D8C5C2 D961E2D8                        865  DC CL48'SQEBR/SQEB -QNaN'
000061B0  FFCB0000 FFCB0000                        866  DC XL16'FFCB0000FFCB0000FFCB0000FFCB0000'
000061C0  E2D8C5C2 D961E2D8                        867  DC CL48'SQEBR/SQEB +SNaN'
000061F0  7FCA0000 00000000                        868  DC XL16'7FCA0000000000007FCA000000000000'
00006200  E2D8C5C2 D961E2D8                        869  DC CL48'SQEBR/SQEB +3'
00006230  3FDDB3D7 3FDDB3D7                        870  DC XL16'3FDDB3D73FDDB3D73FDDB3D73FDDB3D7'
00006240  E2D8C5C2 D961E2D8                        871  DC CL48'SQEBR/SQEB +5'
00006270  400F1BBD 400F1BBD                        872  DC XL16'400F1BBD400F1BBD400F1BBD400F1BBD'
00006280  E2D8C5C2 D961E2D8                        873  DC CL48'SQEBR/SQEB +0.0625'
000062B0  3E800000 3E800000                        874  DC XL16'3E8000003E8000003E8000003E800000'
                              0000000B  00000001   875 SBFPBOT_NUM EQU (*-SBFPBOT_GOOD)/64
                                                   876 *
                                                   877 *
                              000062C0  00000001   878 SBFPBFL_GOOD EQU *
000062C0  E2D8C5C2 D961E2D8                        879  DC CL48'SQEBR/SQEB -inf FPCR'
000062F0  00800000 F8008000                        880  DC XL16'00800000F800800000800000F8008000'
00006300  E2D8C5C2 D961E2D8                        881  DC CL48'SQEBR/SQEB -4 FPCR'
00006330  00800000 F8008000                        882  DC XL16'00800000F800800000800000F8008000'
00006340  E2D8C5C2 D961E2D8                        883  DC CL48'SQEBR/SQEB -0 FPCR'
00006370  00000000 F8000000                        884  DC XL16'00000000F800000000000000F8000000'
00006380  E2D8C5C2 D961E2D8                        885  DC CL48'SQEBR/SQEB +0 FPCR'
000063B0  00000000 F8000000                        886  DC XL16'00000000F800000000000000F8000000'
000063C0  E2D8C5C2 D961E2D8                        887  DC CL48'SQEBR/SQEB +4 FPCR'
000063F0  00000000 F8000000                        888  DC XL16'00000000F800000000000000F8000000'
00006400  E2D8C5C2 D961E2D8                        889  DC CL48'SQEBR/SQEB +inf FPCR'
00006430  00000000 F8000000                        890  DC XL16'00000000F800000000000000F8000000'
00006440  E2D8C5C2 D961E2D8                        891  DC CL48'SQEBR/SQEB -QNaN FPCR'
00006470  00000000 F8000000                        892  DC XL16'00000000F800000000000000F8000000'
00006480  E2D8C5C2 D961E2D8                        893  DC CL48'SQEBR/SQEB +SNaN FPCR'
000064B0  00800000 F8008000                        894  DC XL16'00800000F800800000800000F8008000'
000064C0  E2D8C5C2 D961E2D8                        895  DC CL48'SQEBR/SQEB +3 FPCR'
000064F0  00080000 F8000800                        896  DC XL16'00080000F800080000080000F8000800'
00006500  E2D8C5C2 D961E2D8                        897  DC CL48'SQEBR/SQEB +5 FPCR'
00006530  00080000 F8000C00                        898  DC XL16'00080000F8000C0000080000F8000C00'
00006540  E2D8C5C2 D961E2D8                        899  DC CL48'SQEBR/SQEB +0.0625 FPCR'
00006570  00000000 F8000000                        900  DC XL16'00000000F800000000000000F8000000'
                              0000000B  00000001   901 SBFPBFL_NUM EQU (*-SBFPBFL_GOOD)/64
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   902 *
                                                   903 *
                              00006580  00000001   904 SBFPRMO_GOOD EQU *
00006580  E2D8C5C2 D961E2D8                        905  DC CL48'SQEBR/SQEB RM RNTE,RZ +3'
000065B0  3FDDB3D7 3FDDB3D7                        906  DC XL16'3FDDB3D73FDDB3D73FDDB3D73FDDB3D7'
000065C0  E2D8C5C2 D961E2D8                        907  DC CL48'SQEBR/SQEB RM RP,RM +3'
000065F0  3FDDB3D8 3FDDB3D8                        908  DC XL16'3FDDB3D83FDDB3D83FDDB3D73FDDB3D7'
00006600  E2D8C5C2 D961E2D8                        909  DC CL48'SQEBR/SQEB RM RFS +3'
00006630  3FDDB3D7 3FDDB3D7                        910  DC XL16'3FDDB3D73FDDB3D70000000000000000'
00006640  E2D8C5C2 D961E2D8                        911  DC CL48'SQEBR/SQEB RM RNTE,RZ +5'
00006670  400F1BBD 400F1BBD                        912  DC XL16'400F1BBD400F1BBD400F1BBC400F1BBC'
00006680  E2D8C5C2 D961E2D8                        913  DC CL48'SQEBR/SQEB RM RP,RM +5'
000066B0  400F1BBD 400F1BBD                        914  DC XL16'400F1BBD400F1BBD400F1BBC400F1BBC'
000066C0  E2D8C5C2 D961E2D8                        915  DC CL48'SQEBR/SQEB RM RFS +5'
000066F0  400F1BBD 400F1BBD                        916  DC XL16'400F1BBD400F1BBD0000000000000000'
                              00000006  00000001   917 SBFPRMO_NUM EQU (*-SBFPRMO_GOOD)/64
                                                   918 *
                                                   919 *
                              00006700  00000001   920 SBFPRMOF_GOOD EQU *
00006700  E2D8C5C2 D961E2D8                        921  DC CL48'SQEBR/SQEB RM RNTE,RZ +3 FPCR'
00006730  00080000 00080000                        922  DC XL16'00080000000800000008000100080001'
00006740  E2D8C5C2 D961E2D8                        923  DC CL48'SQEBR/SQEB RM RP,RM +3 FPCR'
00006770  00080002 00080002                        924  DC XL16'00080002000800020008000300080003'
00006780  E2D8C5C2 D961E2D8                        925  DC CL48'SQEBR/SQEB RM RFS +3 FPCR'
000067B0  00080007 00080007                        926  DC XL16'00080007000800070000000000000000'
000067C0  E2D8C5C2 D961E2D8                        927  DC CL48'SQEBR/SQEB RM RNTE,RZ +5 FPCR'
000067F0  00080000 00080000                        928  DC XL16'00080000000800000008000100080001'
00006800  E2D8C5C2 D961E2D8                        929  DC CL48'SQEBR/SQEB RM RP,RM +5 FPCR'
00006830  00080002 00080002                        930  DC XL16'00080002000800020008000300080003'
00006840  E2D8C5C2 D961E2D8                        931  DC CL48'SQEBR/SQEB RM RFS +5 FPCR'
00006870  00080007 00080007                        932  DC XL16'00080007000800070000000000000000'
                              00000006  00000001   933 SBFPRMOF_NUM EQU (*-SBFPRMOF_GOOD)/64
                                                   934 *
                                                   935 *
                              00006880  00000001   936 LBFPBOT_GOOD EQU *
00006880  E2D8C4C2 D9406089                        937  DC CL48'SQDBR -inf'
000068B0  7FF80000 00000000                        938  DC XL16'7FF80000000000000000000000000000'
000068C0  E2D8C4C2 40406089                        939  DC CL48'SQDB  -inf'
000068F0  7FF80000 00000000                        940  DC XL16'7FF80000000000000000000000000000'
00006900  E2D8C4C2 D94060F4                        941  DC CL48'SQDBR -4'
00006930  7FF80000 00000000                        942  DC XL16'7FF80000000000000000000000000000'
00006940  E2D8C4C2 404060F4                        943  DC CL48'SQDB  -4'
00006970  7FF80000 00000000                        944  DC XL16'7FF80000000000000000000000000000'
00006980  E2D8C4C2 D94060F0                        945  DC CL48'SQDBR -0'
000069B0  80000000 00000000                        946  DC XL16'80000000000000008000000000000000'
000069C0  E2D8C4C2 404060F0                        947  DC CL48'SQDB  -0'
000069F0  80000000 00000000                        948  DC XL16'80000000000000008000000000000000'
00006A00  E2D8C4C2 D9404EF0                        949  DC CL48'SQDBR +0'
00006A30  00000000 00000000                        950  DC XL16'00000000000000000000000000000000'
00006A40  E2D8C4C2 40404EF0                        951  DC CL48'SQDB  +0'
00006A70  00000000 00000000                        952  DC XL16'00000000000000000000000000000000'
00006A80  E2D8C4C2 D9404EF4                        953  DC CL48'SQDBR +4'
00006AB0  40000000 00000000                        954  DC XL16'40000000000000004000000000000000'
00006AC0  E2D8C4C2 40404EF4                        955  DC CL48'SQDB  +4'
00006AF0  40000000 00000000                        956  DC XL16'40000000000000004000000000000000'
00006B00  E2D8C4C2 D9404E89                        957  DC CL48'SQDBR +inf'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00006B30  7FF00000 00000000                        958  DC XL16'7FF00000000000007FF0000000000000'
00006B40  E2D8C4C2 40404E89                        959  DC CL48'SQDB  +inf'
00006B70  7FF00000 00000000                        960  DC XL16'7FF00000000000007FF0000000000000'
00006B80  E2D8C4C2 D94060D8                        961  DC CL48'SQDBR -QNaN'
00006BB0  FFF8B000 00000000                        962  DC XL16'FFF8B00000000000FFF8B00000000000'
00006BC0  E2D8C4C2 404060D8                        963  DC CL48'SQDB  -QNaN'
00006BF0  FFF8B000 00000000                        964  DC XL16'FFF8B00000000000FFF8B00000000000'
00006C00  E2D8C4C2 D9404EE2                        965  DC CL48'SQDBR +SNaN'
00006C30  7FF8A000 00000000                        966  DC XL16'7FF8A000000000000000000000000000'
00006C40  E2D8C4C2 40404EE2                        967  DC CL48'SQDB  +SNaN'
00006C70  7FF8A000 00000000                        968  DC XL16'7FF8A000000000000000000000000000'
00006C80  E2D8C4C2 D9404EF3                        969  DC CL48'SQDBR +3'
00006CB0  3FFBB67A E8584CAA                        970  DC XL16'3FFBB67AE8584CAA3FFBB67AE8584CAA'
00006CC0  E2D8C4C2 40404EF3                        971  DC CL48'SQDB  +3'
00006CF0  3FFBB67A E8584CAA                        972  DC XL16'3FFBB67AE8584CAA3FFBB67AE8584CAA'
00006D00  E2D8C4C2 D9404EF5                        973  DC CL48'SQDBR +5'
00006D30  4001E377 9B97F4A8                        974  DC XL16'4001E3779B97F4A84001E3779B97F4A8'
00006D40  E2D8C4C2 40404EF5                        975  DC CL48'SQDB  +5'
00006D70  4001E377 9B97F4A8                        976  DC XL16'4001E3779B97F4A84001E3779B97F4A8'
00006D80  E2D8C4C2 D940F04B                        977  DC CL48'SQDBR 0.0625'
00006DB0  3FD00000 00000000                        978  DC XL16'3FD00000000000003FD0000000000000'
00006DC0  E2D8C4C2 4040F04B                        979  DC CL48'SQDB  0.0625'
00006DF0  3FD00000 00000000                        980  DC XL16'3FD00000000000003FD0000000000000'
                              00000016  00000001   981 LBFPBOT_NUM EQU (*-LBFPBOT_GOOD)/64
                                                   982 *
                                                   983 *
                              00006E00  00000001   984 LBFPBFL_GOOD EQU *
00006E00  E2D8C4C2 D961E2D8                        985  DC CL48'SQDBR/SQDB -inf FPCR'
00006E30  00800000 F8008000                        986  DC XL16'00800000F800800000800000F8008000'
00006E40  E2D8C4C2 D961E2D8                        987  DC CL48'SQDBR/SQDB -4 FPCR'
00006E70  00800000 F8008000                        988  DC XL16'00800000F800800000800000F8008000'
00006E80  E2D8C4C2 D961E2D8                        989  DC CL48'SQDBR/SQDB -0 FPCR'
00006EB0  00000000 F8000000                        990  DC XL16'00000000F800000000000000F8000000'
00006EC0  E2D8C4C2 D961E2D8                        991  DC CL48'SQDBR/SQDB +0 FPCR'
00006EF0  00000000 F8000000                        992  DC XL16'00000000F800000000000000F8000000'
00006F00  E2D8C4C2 D961E2D8                        993  DC CL48'SQDBR/SQDB +4 FPCR'
00006F30  00000000 F8000000                        994  DC XL16'00000000F800000000000000F8000000'
00006F40  E2D8C4C2 D961E2D8                        995  DC CL48'SQDBR/SQDB +inf FPCR'
00006F70  00000000 F8000000                        996  DC XL16'00000000F800000000000000F8000000'
00006F80  E2D8C4C2 D961E2D8                        997  DC CL48'SQDBR/SQDB -QNaN FPCR'
00006FB0  00000000 F8000000                        998  DC XL16'00000000F800000000000000F8000000'
00006FC0  E2D8C4C2 D961E2D8                        999  DC CL48'SQDBR/SQDB +SNaN FPCR'
00006FF0  00800000 F8008000                       1000  DC XL16'00800000F800800000800000F8008000'
00007000  E2D8C4C2 D961E2D8                       1001  DC CL48'SQDBR/SQDB +3 FPCR'
00007030  00080000 F8000800                       1002  DC XL16'00080000F800080000080000F8000800'
00007040  E2D8C4C2 D961E2D8                       1003  DC CL48'SQDBR/SQDB +5 FPCR'
00007070  00080000 F8000C00                       1004  DC XL16'00080000F8000C0000080000F8000C00'
00007080  E2D8C4C2 D961E2D8                       1005  DC CL48'SQDBR/SQDB +0.0625 FPCR'
000070B0  00000000 F8000000                       1006  DC XL16'00000000F800000000000000F8000000'
                              0000000B  00000001  1007 LBFPBFL_NUM EQU (*-LBFPBFL_GOOD)/64
                                                  1008 *
                                                  1009 *
                              000070C0  00000001  1010 LBFPRMO_GOOD EQU *
000070C0  E2D8C4C2 D961E2D8                       1011  DC CL48'SQDBR/SQDB RM RNTE +3'
000070F0  3FFBB67A E8584CAA                       1012  DC XL16'3FFBB67AE8584CAA3FFBB67AE8584CAA'
00007100  E2D8C4C2 D961E2D8                       1013  DC CL48'SQDBR/SQDB RM RZ +3'
```

```
  LOC        OBJECT CODE        ADDR1     ADDR2    STMT

00007130  3FFBB67A E8584CAA                       1014  DC XL16'3FFBB67AE8584CAA3FFBB67AE8584CAA'
00007140  E2D8C4C2 D961E2D8                       1015  DC CL48'SQDBR/SQDB RM RP +3'
00007170  3FFBB67A E8584CAB                       1016  DC XL16'3FFBB67AE8584CAB3FFBB67AE8584CAB'
00007180  E2D8C4C2 D961E2D8                       1017  DC CL48'SQDBR/SQDB RM RM +3'
000071B0  3FFBB67A E8584CAA                       1018  DC XL16'3FFBB67AE8584CAA3FFBB67AE8584CAA'
000071C0  E2D8C4C2 D961E2D8                       1019  DC CL48'SQDBR/SQDB RM RFS +3'
000071F0  3FFBB67A E8584CAB                       1020  DC XL16'3FFBB67AE8584CAB3FFBB67AE8584CAB'
00007200  E2D8C4C2 D961E2D8                       1021  DC CL48'SQDBR/SQDB RM RNTE +5'
00007230  4001E377 9B97F4A8                       1022  DC XL16'4001E3779B97F4A84001E3779B97F4A8'
00007240  E2D8C4C2 D961E2D8                       1023  DC CL48'SQDBR/SQDB RM RZ +5'
00007270  4001E377 9B97F4A7                       1024  DC XL16'4001E3779B97F4A74001E3779B97F4A7'
00007280  E2D8C4C2 D961E2D8                       1025  DC CL48'SQDBR/SQDB RM RP +5'
000072B0  4001E377 9B97F4A8                       1026  DC XL16'4001E3779B97F4A84001E3779B97F4A8'
000072C0  E2D8C4C2 D961E2D8                       1027  DC CL48'SQDBR/SQDB RM RM +5'
000072F0  4001E377 9B97F4A7                       1028  DC XL16'4001E3779B97F4A74001E3779B97F4A7'
00007300  E2D8C4C2 D961E2D8                       1029  DC CL48'SQDBR/SQDB RM RFS +5'
00007330  4001E377 9B97F4A7                       1030  DC XL16'4001E3779B97F4A74001E3779B97F4A7'
                              0000000A  00000001  1031 LBFPRMO_NUM EQU (*-LBFPRMO_GOOD)/64
                                                  1032 *
                                                  1033 *
                              00007340  00000001  1034 LBFPRMOF_GOOD EQU *
00007340  E2D8C5C2 D961E2D8                       1035  DC CL48'SQEBR/SQEB RM RNTE,RZ +3 FPCR'
00007370  00080000 00080000                       1036  DC XL16'00080000000800000008000100080001'
00007380  E2D8C5C2 D961E2D8                       1037  DC CL48'SQEBR/SQEB RM RP,RM +3 FPCR'
000073B0  00080002 00080002                       1038  DC XL16'00080002000800020008000300080003'
000073C0  E2D8C5C2 D961E2D8                       1039  DC CL48'SQEBR/SQEB RM RFS +3 FPCR'
000073F0  00080007 00080007                       1040  DC XL16'00080007000800070000000000000000'
00007400  E2D8C5C2 D961E2D8                       1041  DC CL48'SQEBR/SQEB RM RNTE,RZ +5 FPCR'
00007430  00080000 00080000                       1042  DC XL16'00080000000800000008000100080001'
00007440  E2D8C5C2 D961E2D8                       1043  DC CL48'SQEBR/SQEB RM RP,RM +5 FPCR'
00007470  00080002 00080002                       1044  DC XL16'00080002000800020008000300080003'
00007480  E2D8C5C2 D961E2D8                       1045  DC CL48'SQEBR/SQEB RM RFS +5 FPCR'
000074B0  00080007 00080007                       1046  DC XL16'00080007000800070000000000000000'
                              00000006  00000001  1047 LBFPRMOF_NUM EQU (*-LBFPRMOF_GOOD)/64
                                                  1048 *
                                                  1049 *
                              000074C0  00000001  1050 XBFPBOT_GOOD EQU *
000074C0  E2D8E7C2 D940D5E3                       1051  DC CL48'SQXBR NT -inf'
000074F0  7FFF8000 00000000                       1052  DC XL16'7FFF8000000000000000000000000000'
00007500  E2D8E7C2 D940E399                       1053  DC CL48'SQXBR Tr -inf'
00007530  00000000 00000000                       1054  DC XL16'00000000000000000000000000000000'
00007540  E2D8E7C2 D940D5E3                       1055  DC CL48'SQXBR NT -4'
00007570  7FFF8000 00000000                       1056  DC XL16'7FFF8000000000000000000000000000'
00007580  E2D8E7C2 D940E399                       1057  DC CL48'SQXBR Tr -4'
000075B0  00000000 00000000                       1058  DC XL16'00000000000000000000000000000000'
000075C0  E2D8E7C2 D940D5E3                       1059  DC CL48'SQXBR NT -0'
000075F0  80000000 00000000                       1060  DC XL16'80000000000000000000000000000000'
00007600  E2D8E7C2 D940E399                       1061  DC CL48'SQXBR Tr -0'
00007630  80000000 00000000                       1062  DC XL16'80000000000000000000000000000000'
00007640  E2D8E7C2 D940D5E3                       1063  DC CL48'SQXBR NT +0'
00007670  00000000 00000000                       1064  DC XL16'00000000000000000000000000000000'
00007680  E2D8E7C2 D940E399                       1065  DC CL48'SQXBR Tr +0'
000076B0  00000000 00000000                       1066  DC XL16'00000000000000000000000000000000'
000076C0  E2D8E7C2 D940D5E3                       1067  DC CL48'SQXBR NT +4'
000076F0  40000000 00000000                       1068  DC XL16'40000000000000000000000000000000'
00007700  E2D8E7C2 D940E399                       1069  DC CL48'SQXBR Tr +4'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00007730  40000000 00000000                       1070  DC XL16'40000000000000000000000000000000'
00007740  E2D8E7C2 D940D5E3                       1071  DC CL48'SQXBR NT +inf'
00007770  7FFF0000 00000000                       1072  DC XL16'7FFF0000000000000000000000000000'
00007780  E2D8E7C2 D940E399                       1073  DC CL48'SQXBR Tr +inf'
000077B0  7FFF0000 00000000                       1074  DC XL16'7FFF0000000000000000000000000000'
000077C0  E2D8E7C2 D940D5E3                       1075  DC CL48'SQXBR NT -QNaN'
000077F0  FFFF8B00 00000000                       1076  DC XL16'FFFF8B00000000000000000000000000'
00007800  E2D8E7C2 D940E399                       1077  DC CL48'SQXBR Tr -QNaN'
00007830  FFFF8B00 00000000                       1078  DC XL16'FFFF8B00000000000000000000000000'
00007840  E2D8E7C2 D940D5E3                       1079  DC CL48'SQXBR NT +SNaN'
00007870  7FFF8A00 00000000                       1080  DC XL16'7FFF8A00000000000000000000000000'
00007880  E2D8E7C2 D940E399                       1081  DC CL48'SQXBR Tr +SNaN'
000078B0  00000000 00000000                       1082  DC XL16'00000000000000000000000000000000'
000078C0  E2D8E7C2 D940D5E3                       1083  DC CL48'SQXBR NT +3'
000078F0  3FFFBB67 AE8584CA                       1084  DC XL16'3FFFBB67AE8584CAA73B25742D7078B8'
00007900  E2D8E7C2 D940E399                       1085  DC CL48'SQXBR Tr +3'
00007930  3FFFBB67 AE8584CA                       1086  DC XL16'3FFFBB67AE8584CAA73B25742D7078B8'
00007940  E2D8E7C2 D940D5E3                       1087  DC CL48'SQXBR NT +5'
00007970  40001E37 79B97F4A                       1088  DC XL16'40001E3779B97F4A7C15F39CC0605CEE'
00007980  E2D8E7C2 D940E399                       1089  DC CL48'SQXBR Tr +5'
000079B0  40001E37 79B97F4A                       1090  DC XL16'40001E3779B97F4A7C15F39CC0605CEE'
000079C0  E2D8E7C2 D940D5E3                       1091  DC CL48'SQXBR NT 0.0625'
000079F0  3FFD0000 00000000                       1092  DC XL16'3FFD0000000000000000000000000000'
00007A00  E2D8E7C2 D940E399                       1093  DC CL48'SQXBR Tr 0.0625'
00007A30  3FFD0000 00000000                       1094  DC XL16'3FFD0000000000000000000000000000'
                              00000016  00000001  1095 XBFPBOT_NUM EQU (*-XBFPBOT_GOOD)/64
                                                  1096 *
                                                  1097 *
                              00007A40  00000001  1098 XBFPBFL_GOOD EQU *
00007A40  E2D8E7C2 D9406089                       1099  DC CL48'SQXBR -inf FPCR'
00007A70  00800000 F8008000                       1100  DC XL16'00800000F80080000000000000000000'
00007A80  E2D8E7C2 D94060F4                       1101  DC CL48'SQXBR -4 FPCR'
00007AB0  00800000 F8008000                       1102  DC XL16'00800000F80080000000000000000000'
00007AC0  E2D8E7C2 D94060F0                       1103  DC CL48'SQXBR -0 FPCR'
00007AF0  00000000 F8000000                       1104  DC XL16'00000000F80000000000000000000000'
00007B00  E2D8E7C2 D9404EF0                       1105  DC CL48'SQXBR +0 FPCR'
00007B30  00000000 F8000000                       1106  DC XL16'00000000F80000000000000000000000'
00007B40  E2D8E7C2 D9404EF4                       1107  DC CL48'SQXBR +4 FPCR'
00007B70  00000000 F8000000                       1108  DC XL16'00000000F80000000000000000000000'
00007B80  E2D8E7C2 D9404E89                       1109  DC CL48'SQXBR +inf FPCR'
00007BB0  00000000 F8000000                       1110  DC XL16'00000000F80000000000000000000000'
00007BC0  E2D8E7C2 D94060D8                       1111  DC CL48'SQXBR -QNaN FPCR'
00007BF0  00000000 F8000000                       1112  DC XL16'00000000F80000000000000000000000'
00007C00  E2D8E7C2 D9404EE2                       1113  DC CL48'SQXBR +SNaN FPCR'
00007C30  00800000 F8008000                       1114  DC XL16'00800000F80080000000000000000000'
00007C40  E2D8E7C2 D9404EF3                       1115  DC CL48'SQXBR +3 FPCR'
00007C70  00080000 F8000800                       1116  DC XL16'00080000F80008000000000000000000'
00007C80  E2D8E7C2 D9404EF5                       1117  DC CL48'SQXBR +5 FPCR'
00007CB0  00080000 F8000C00                       1118  DC XL16'00080000F8000C000000000000000000'
00007CC0  E2D8E7C2 D9404EF0                       1119  DC CL48'SQXBR +0.0625 FPCR'
00007CF0  00000000 F8000000                       1120  DC XL16'00000000F80000000000000000000000'
                              0000000B  00000001  1121 XBFPBFL_NUM EQU (*-XBFPBFL_GOOD)/64
                                                  1122 *
                                                  1123 *
                              00007D00  00000001  1124 XBFPRMO_GOOD EQU *
00007D00  E2D8E7C2 D940D9D4                       1125  DC CL48'SQXBR RM RNTE +3'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00007D30  3FFFBB67 AE8584CA                       1126  DC XL16'3FFFBB67AE8584CAA73B25742D7078B8'
00007D40  E2D8E7C2 D940D9D4                       1127  DC CL48'SQXBR RM RZ +3'
00007D70  3FFFBB67 AE8584CA                       1128  DC XL16'3FFFBB67AE8584CAA73B25742D7078B8'
00007D80  E2D8E7C2 D940D9D4                       1129  DC CL48'SQXBR RM RP +3'
00007DB0  3FFFBB67 AE8584CA                       1130  DC XL16'3FFFBB67AE8584CAA73B25742D7078B9'
00007DC0  E2D8E7C2 D940D9D4                       1131  DC CL48'SQXBR RM RM +3'
00007DF0  3FFFBB67 AE8584CA                       1132  DC XL16'3FFFBB67AE8584CAA73B25742D7078B8'
00007E00  E2D8E7C2 D940D9D4                       1133  DC CL48'SQXBR RM RFS +3'
00007E30  3FFFBB67 AE8584CA                       1134  DC XL16'3FFFBB67AE8584CAA73B25742D7078B9'
00007E40  E2D8E7C2 D940D9D4                       1135  DC CL48'SQXBR RM RNTE +5'
00007E70  40001E37 79B97F4A                       1136  DC XL16'40001E3779B97F4A7C15F39CC0605CEE'
00007E80  E2D8E7C2 D940D9D4                       1137  DC CL48'SQXBR RM RZ +5'
00007EB0  40001E37 79B97F4A                       1138  DC XL16'40001E3779B97F4A7C15F39CC0605CED'
00007EC0  E2D8E7C2 D940D9D4                       1139  DC CL48'SQXBR RM RP +5'
00007EF0  40001E37 79B97F4A                       1140  DC XL16'40001E3779B97F4A7C15F39CC0605CEE'
00007F00  E2D8E7C2 D940D9D4                       1141  DC CL48'SQXBR RM RM +5'
00007F30  40001E37 79B97F4A                       1142  DC XL16'40001E3779B97F4A7C15F39CC0605CED'
00007F40  E2D8E7C2 D940D9D4                       1143  DC CL48'SQXBR RM RFS +5'
00007F70  40001E37 79B97F4A                       1144  DC XL16'40001E3779B97F4A7C15F39CC0605CED'
                              0000000A  00000001  1145 XBFPRMO_NUM EQU (*-XBFPRMO_GOOD)/64
                                                  1146 *
                                                  1147 *
                              00007F80  00000001  1148 XBFPRMOF_GOOD EQU *
00007F80  E2D8E7C2 D940D9D4                       1149  DC CL48'SQXBR RM RTNE,RZ,RP,RM +3 FPCR'
00007FB0  00080000 00080001                       1150  DC XL16'00080000000800010008000200080003'
00007FC0  E2D8E7C2 D940D9D4                       1151  DC CL48'SQXBR RM RFS +3 FPCR'
00007FF0  00080007 00000000                       1152  DC XL16'00080007000000000000000000000000'
00008000  E2D8E7C2 D940D9D4                       1153  DC CL48'SQXBR RM RTNE,RZ,RP,RM +5 FPCR'
00008030  00080000 00080001                       1154  DC XL16'00080000000800010008000200080003'
00008040  E2D8E7C2 D940D9D4                       1155  DC CL48'SQXBR RM RFS +5 FPCR'
00008070  00080007 00000000                       1156  DC XL16'00080007000000000000000000000000'
                              00000004  00000001  1157 XBFPRMOF_NUM EQU (*-XBFPRMOF_GOOD)/64
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00008080                                          1159 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                  1161 ***********************************************************************
                                                  1162 *               REPORT UNEXPECTED PROGRAM CHECK
                                                  1163 ***********************************************************************

00008080                                          1165 PGMCK    DS    0H
00008080  F342 C072 F08E      000080F2  0000008E  1166          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
00008086  926B C076                     000080F6  1167          MVI   PGMCOMMA,C','
0000808A  DC03 C072 C178      000080F2  000081F8  1168          TR    PROGCODE,HEXTRTAB

00008090  F384 C07C F150      000080FC  00000150  1170          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
00008096  9240 C084                     00008104  1171          MVI   PGMPSW+(0*9)+8,C' '
0000809A  DC07 C07C C178      000080FC  000081F8  1172          TR    PGMPSW+(0*9)(8),HEXTRTAB

000080A0  F384 C085 F154      00008105  00000154  1174          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
000080A6  9240 C08D                     0000810D  1175          MVI   PGMPSW+(1*9)+8,C' '
000080AA  DC07 C085 C178      00008105  000081F8  1176          TR    PGMPSW+(1*9)(8),HEXTRTAB

000080B0  F384 C08E F158      0000810E  00000158  1178          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
000080B6  9240 C096                     00008116  1179          MVI   PGMPSW+(2*9)+8,C' '
000080BA  DC07 C08E C178      0000810E  000081F8  1180          TR    PGMPSW+(2*9)(8),HEXTRTAB

000080C0  F384 C097 F15C      00008117  0000015C  1182          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
000080C6  9240 C09F                     0000811F  1183          MVI   PGMPSW+(3*9)+8,C' '
000080CA  DC07 C097 C178      00008117  000081F8  1184          TR    PGMPSW+(3*9)(8),HEXTRTAB

000080D0  4100 0042                     00000042  1186          LA    R0,L'PROGMSG     R0 <== length of message
000080D4  4110 C05E                     000080DE  1187          LA    R1,PROGMSG       R1 --> the message text itself
000080D8  4520 C27A                     000082FA  1188          BAL   R2,MSG           Go display this message
                                                  1189
000080DC  07FD                                    1190          BR    R13              Return to caller



000080DE                                          1192 PROGMSG  DS   0CL66
000080DE  D7D9D6C7 D9C1D440                       1193          DC    CL20'PROGRAM CHECK! CODE '
000080F2  88888888                                1194 PROGCODE DC    CL4'hhhh'
000080F6  6B                                      1195 PGMCOMMA DC    CL1','
000080F7  40D7E2E6 40                             1196          DC    CL5' PSW '
000080FC  88888888 88888888                       1197 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1199 ***********************************************************************
                                                  1200 *                    VERIFICATION ROUTINE
                                                  1201 ***********************************************************************

00008120                                          1203 VERISUB  DS    0H
                                                  1204 *
                                                  1205 **       Loop through the VERIFY TABLE...
                                                  1206 *

00008120  4110 C32C                     000083AC  1208          LA    R1,VERIFTAB      R1 --> Verify table
00008124  4120 000C                     0000000C  1209          LA    R2,VERIFLEN      R2 <== Number of entries
00008128  0D30                                    1210          BASR  R3,0             Set top of loop

0000812A  9846 1000                     00000000  1212          LM    R4,R6,0(R1)      Load verify table values
0000812E  4D70 C0C2                     00008142  1213          BAS   R7,VERIFY        Verify results
00008132  4110 100C                     0000000C  1214          LA    R1,12(,R1)       Next verify table entry
00008136  0623                                    1215          BCTR  R2,R3            Loop through verify table

00008138  9500 C278                     000082F8  1217          CLI   FAILFLAG,X'00'   Did all tests verify okay?
0000813C  078D                                    1218          BER   R13              Yes, return to caller
0000813E  47F0 F238                     00000238  1219          B     FAIL             No, load FAILURE disabled wait PSW






                                                  1221 *
                                                  1222 **       Loop through the ACTUAL / EXPECTED results...
                                                  1223 *

00008142  0D80                                    1225 VERIFY   BASR  R8,0             Set top of loop

00008144  D50F 4000 5030      00000000  00000030  1227          CLC   0(16,R4),48(R5)  Actual results == Expected results?
0000814A  4770 C0DA                     0000815A  1228          BNE   VERIFAIL         No, show failure
0000814E  4140 4010                     00000010  1229 VERINEXT LA    R4,16(,R4)       Next actual result
00008152  4150 5040                     00000040  1230          LA    R5,64(,R5)       Next expected result
00008156  0668                                    1231          BCTR  R6,R8            Loop through results

00008158  07F7                                    1233          BR    R7               Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1235 ***********************************************************************
                                                  1236 *                    Report the failure...
                                                  1237 ***********************************************************************

0000815A  9005 C250                     000082D0  1239 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
0000815E  92FF C278                     000082F8  1240          MVI   FAILFLAG,X'FF'   Remember verification failure
                                                  1241 *
                                                  1242 **       First, show them the description...
                                                  1243 *
00008162  D22F C1E0 5000      00008260  00000000  1244          MVC   FAILDESC,0(R5)   Save results/test description
00008168  4100 0044                     00000044  1245          LA    R0,L'FAILMSG1    R0 <== length of message
0000816C  4110 C1CC                     0000824C  1246          LA    R1,FAILMSG1      R1 --> the message text itself
00008170  4520 C27A                     000082FA  1247          BAL   R2,MSG           Go display this message
                                                  1248 *
                                                  1249 **       Save address of actual and expected results
                                                  1250 *
00008174  5040 C24C                     000082CC  1251          ST    R4,AACTUAL       Save A(actual results)
00008178  4150 5030                     00000030  1252          LA    R5,48(,R5)       R5 ==> expected results
0000817C  5050 C248                     000082C8  1253          ST    R5,AEXPECT       Save A(expected results)
                                                  1254 *
                                                  1255 **       Format and show them the EXPECTED ("Want") results...
                                                  1256 *
00008180  D205 C210 C3C0      00008290  00008440  1257          MVC   WANTGOT,=CL6'Want: '
00008186  F384 C216 C248      00008296  000082C8  1258          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
0000818C  9240 C21E                     0000829E  1259          MVI   BLANKEQ,C' '
00008190  DC07 C216 C178      00008296  000081F8  1260          TR    FAILADR,HEXTRTAB

00008196  F384 C221 5000      000082A1  00000000  1262          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
0000819C  9240 C229                     000082A9  1263          MVI   FAILVALS+(0*9)+8,C' '
000081A0  DC07 C221 C178      000082A1  000081F8  1264          TR    FAILVALS+(0*9)(8),HEXTRTAB

000081A6  F384 C22A 5004      000082AA  00000004  1266          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
000081AC  9240 C232                     000082B2  1267          MVI   FAILVALS+(1*9)+8,C' '
000081B0  DC07 C22A C178      000082AA  000081F8  1268          TR    FAILVALS+(1*9)(8),HEXTRTAB

000081B6  F384 C233 5008      000082B3  00000008  1270          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
000081BC  9240 C23B                     000082BB  1271          MVI   FAILVALS+(2*9)+8,C' '
000081C0  DC07 C233 C178      000082B3  000081F8  1272          TR    FAILVALS+(2*9)(8),HEXTRTAB

000081C6  F384 C23C 500C      000082BC  0000000C  1274          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
000081CC  9240 C244                     000082C4  1275          MVI   FAILVALS+(3*9)+8,C' '
000081D0  DC07 C23C C178      000082BC  000081F8  1276          TR    FAILVALS+(3*9)(8),HEXTRTAB

000081D6  4100 0035                     00000035  1278          LA    R0,L'FAILMSG2    R0 <== length of message
000081DA  4110 C210                     00008290  1279          LA    R1,FAILMSG2      R1 --> the message text itself
000081DE  4520 C27A                     000082FA  1280          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1282 *
                                                  1283 **       Format and show them the ACTUAL ("Got") results...
                                                  1284 *
000081E2  D205 C210 C3C6      00008290  00008446  1285          MVC   WANTGOT,=CL6'Got:  '
000081E8  F384 C216 C24C      00008296  000082CC  1286          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
000081EE  9240 C21E                     0000829E  1287          MVI   BLANKEQ,C' '
000081F2  DC07 C216 C178      00008296  000081F8  1288          TR    FAILADR,HEXTRTAB

000081F8  F384 C221 4000      000082A1  00000000  1290          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
000081FE  9240 C229                     000082A9  1291          MVI   FAILVALS+(0*9)+8,C' '
00008202  DC07 C221 C178      000082A1  000081F8  1292          TR    FAILVALS+(0*9)(8),HEXTRTAB

00008208  F384 C22A 4004      000082AA  00000004  1294          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
0000820E  9240 C232                     000082B2  1295          MVI   FAILVALS+(1*9)+8,C' '
00008212  DC07 C22A C178      000082AA  000081F8  1296          TR    FAILVALS+(1*9)(8),HEXTRTAB

00008218  F384 C233 4008      000082B3  00000008  1298          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
0000821E  9240 C23B                     000082BB  1299          MVI   FAILVALS+(2*9)+8,C' '
00008222  DC07 C233 C178      000082B3  000081F8  1300          TR    FAILVALS+(2*9)(8),HEXTRTAB

00008228  F384 C23C 400C      000082BC  0000000C  1302          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
0000822E  9240 C244                     000082C4  1303          MVI   FAILVALS+(3*9)+8,C' '
00008232  DC07 C23C C178      000082BC  000081F8  1304          TR    FAILVALS+(3*9)(8),HEXTRTAB

00008238  4100 0035                     00000035  1306          LA    R0,L'FAILMSG2    R0 <== length of message
0000823C  4110 C210                     00008290  1307          LA    R1,FAILMSG2      R1 --> the message text itself
00008240  4520 C27A                     000082FA  1308          BAL   R2,MSG           Go display this message

00008244  9805 C250                     000082D0  1310          LM    R0,R5,SAVER0R5   Restore registers
00008248  47F0 C0CE                     0000814E  1311          B     VERINEXT         Continue with verification...



0000824C                                          1313 FAILMSG1  DS   0CL68
0000824C  C3D6D4D7 C1D9C9E2                       1314           DC    CL20'COMPARISON FAILURE! '
00008260  4D8485A2 83998997                       1315 FAILDESC  DC    CL48'(description)'


00008290                                          1317 FAILMSG2  DS   0CL53
00008290  40404040 4040                           1318 WANTGOT   DC    CL6' '           'Want: ' -or- 'Got:  '
00008296  C1C1C1C1 C1C1C1C1                       1319 FAILADR   DC    CL8'AAAAAAAA'
0000829E  407E40                                  1320 BLANKEQ   DC    CL3' = '
000082A1  88888888 88888888                       1321 FAILVALS  DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '


000082C8  00000000                                1323 AEXPECT   DC    F'0'             ==> Expected ("Want") results
000082CC  00000000                                1324 AACTUAL   DC    F'0'             ==> Actual ("Got") results
000082D0  00000000 00000000                       1325 SAVER0R5  DC    6F'0'            Registers R0 - R5 save area
000082E8  F0F1F2F3 F4F5F6F7                       1326 CHARHEX   DC    CL16'0123456789ABCDEF'
                              000081F8  00000010  1327 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
000082F8  00                                      1328 FAILFLAG  DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1330 ***********************************************************************
                                                  1331 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                                  1332 ***********************************************************************

000082FA  4900 C3BC                     0000843C  1334 MSG      CH    R0,=H'0'               Do we even HAVE a message?
000082FE  07D2                                    1335          BNHR  R2                     No, ignore

00008300  9002 C2B0                     00008330  1337          STM   R0,R2,MSGSAVE          Save registers

00008304  4900 C3BE                     0000843E  1339          CH    R0,=AL2(L'MSGMSG)      Message length within limits?
00008308  47D0 C290                     00008310  1340          BNH   MSGOK                  Yes, continue
0000830C  4100 005F                     0000005F  1341          LA    R0,L'MSGMSG            No, set to maximum

00008310  1820                                    1343 MSGOK    LR    R2,R0                  Copy length to work register
00008312  0620                                    1344          BCTR  R2,0                   Minus-1 for execute
00008314  4420 C2BC                     0000833C  1345          EX    R2,MSGMVC              Copy message to O/P buffer

00008318  4120 200A                     0000000A  1347          LA    R2,1+L'MSGCMD(,R2)     Calculate true command length
0000831C  4110 C2C2                     00008342  1348          LA    R1,MSGCMD              Point to true command

00008320  83120008                                1350          DC    X'83',X'12',X'0008'    Issue Hercules Diagnose X'008'
00008324  4780 C2AA                     0000832A  1351          BZ    MSGRET                 Return if successful
00008328  0000                                    1352          DC    H'0'                   CRASH for debugging purposes

0000832A  9802 C2B0                     00008330  1354 MSGRET   LM    R0,R2,MSGSAVE          Restore registers
0000832E  07F2                                    1355          BR    R2                     Return to caller






00008330  00000000 00000000                       1357 MSGSAVE  DC    3F'0'                  Registers save area
0000833C  D200 C2CB 1000      0000834B  00000000  1358 MSGMVC   MVC   MSGMSG(0),0(R1)        Executed instruction


00008342  D4E2C7D5 D6C8405C                       1360 MSGCMD   DC    C'MSGNOH * '           *** HERCULES MESSAGE COMMAND ***
0000834B  40404040 40404040                       1361 MSGMSG   DC    CL95' '                The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1363 ***********************************************************************
                                                  1364 *                         VERIFY TABLE
                                                  1365 ***********************************************************************
                                                  1366 *
                                                  1367 *        A(actual results), A(expected results), A(#of results)
                                                  1368 *
                                                  1369 ***********************************************************************

000083AC                                          1371 VERIFTAB DC    0F'0'
000083AC  00001000                                1372          DC    A(SBFPBOT)
000083B0  00006000                                1373          DC    A(SBFPBOT_GOOD)
000083B4  0000000B                                1374          DC    A(SBFPBOT_NUM)
                                                  1375 *
000083B8  00001100                                1376          DC    A(SBFPBFL)
000083BC  000062C0                                1377          DC    A(SBFPBFL_GOOD)
000083C0  0000000B                                1378          DC    A(SBFPBFL_NUM)
                                                  1379 *
000083C4  00001200                                1380          DC    A(SBFPRMO)
000083C8  00006580                                1381          DC    A(SBFPRMO_GOOD)
000083CC  00000006                                1382          DC    A(SBFPRMO_NUM)
                                                  1383 *
000083D0  00001400                                1384          DC    A(SBFPRMOF)
000083D4  00006700                                1385          DC    A(SBFPRMOF_GOOD)
000083D8  00000006                                1386          DC    A(SBFPRMOF_NUM)
                                                  1387 *
000083DC  00003000                                1388          DC    A(LBFPBOT)
000083E0  00006880                                1389          DC    A(LBFPBOT_GOOD)
000083E4  00000016                                1390          DC    A(LBFPBOT_NUM)
                                                  1391 *
000083E8  00003200                                1392          DC    A(LBFPBFL)
000083EC  00006E00                                1393          DC    A(LBFPBFL_GOOD)
000083F0  0000000B                                1394          DC    A(LBFPBFL_NUM)
                                                  1395 *
000083F4  00003400                                1396          DC    A(LBFPRMO)
000083F8  000070C0                                1397          DC    A(LBFPRMO_GOOD)
000083FC  0000000A                                1398          DC    A(LBFPRMO_NUM)
                                                  1399 *
00008400  00003700                                1400          DC    A(LBFPRMOF)
00008404  00007340                                1401          DC    A(LBFPRMOF_GOOD)
00008408  00000006                                1402          DC    A(LBFPRMOF_NUM)
                                                  1403 *
0000840C  00005000                                1404          DC    A(XBFPBOT)
00008410  000074C0                                1405          DC    A(XBFPBOT_GOOD)
00008414  00000016                                1406          DC    A(XBFPBOT_NUM)
                                                  1407 *
00008418  00005400                                1408          DC    A(XBFPBFL)
0000841C  00007A40                                1409          DC    A(XBFPBFL_GOOD)
00008420  0000000B                                1410          DC    A(XBFPBFL_NUM)
                                                  1411 *
00008424  00005500                                1412          DC    A(XBFPRMO)
00008428  00007D00                                1413          DC    A(XBFPRMO_GOOD)
0000842C  0000000A                                1414          DC    A(XBFPRMO_NUM)
                                                  1415 *
00008430  00005A00                                1416          DC    A(XBFPRMOF)
00008434  00007F80                                1417          DC    A(XBFPRMOF_GOOD)
00008438  00000004                                1418          DC    A(XBFPRMOF_NUM)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1419 *
                             0000000C  00000001  1420 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

0000843C                                          1422          END
0000843C  0000                                    1423                =H'0'
0000843E  005F                                    1424                =AL2(L'MSGMSG)
00008440  E68195A3 7A40                           1425                =CL6'Want: '
00008446  C796A37A 4040                           1426                =CL6'Got:  '
```

```
     SYMBOL        TYPE  VALUE   LENGTH  DEFN  REFERENCES

AACTUAL             F    0082CC       4  1324  1251  1286
AEXPECT             F    0082C8       4  1323  1253  1258
AHELPERS            A    00027C       4   201   191   233
BFPSQRTS            J    000000   33868   117
BLANKEQ             C    00829E       3  1320  1259  1287
CHARHEX             C    0082E8      16  1326  1327
CTLR0               F    0002F0       4   243   210   211   212
FAIL                I    000238       4   199  1219
FAILADR             C    008296       8  1319  1258  1260  1286  1288
FAILDESC            C    008260      48  1315  1244
FAILFLAG            X    0082F8       1  1328  1217  1240
FAILMSG1            C    00824C      68  1313  1245  1246
FAILMSG2            C    008290      53  1317  1278  1279  1306  1307
FAILPSW             X    0002E0       8   241   199
FAILVALS            C    0082A1      36  1321  1262  1263  1264  1266  1267  1268  1270  1271  1272  1274  1275  1276  1290  1291
                                               1292  1294  1295  1296  1298  1299  1300  1302  1303  1304
FPCMCT              U    000005       1   635   370   482   585
FPCMODES            C    0005AE       1   629   635   373   485   588
FPCREGNT            X    0002F4       4   244   315   327   375   382   427   439   487   494   540   590
FPCREGTR            X    0002F8       4   245   321   333   433   445   547
FPR0                U    000000       1   138
FPR1                U    000001       1   139   314   316   322   377   378   426   428   434   489   490   538   541   548   592
                                                594
FPR10               U    00000A       1   148   543   550   596
FPR11               U    00000B       1   149
FPR12               U    00000C       1   150
FPR13               U    00000D       1   151
FPR14               U    00000E       1   152
FPR15               U    00000F       1   153
FPR2                U    000002       1   140
FPR3                U    000003       1   141   539   593
FPR4                U    000004       1   142
FPR5                U    000005       1   143
FPR6                U    000006       1   144
FPR7                U    000007       1   145
FPR8                U    000008       1   146   313   316   317   320   322   323   326   328   329   332   334   335   378   379
                                                384   385   425   428   429   432   434   435   438   440   441   444   446   447
                                                490   491   496   497   537   541   542   546   548   549   594   595
FPR9                U    000009       1   147
GOODPSW             X    0002D0       8   240   237
HELPERS             H    008080       2  1159   156   201
HEXTRTAB            U    0081F8      16  1327  1168  1172  1176  1180  1184  1260  1264  1268  1272  1276  1288  1292  1296  1300
                                               1304
IMAGE               1    000000   33868     0
LBFPB               H    000434       2   418   220
LBFPBCT             U    00000B       1   732   266
LBFPBFL             U    003200       1   826   269  1392
LBFPBFL_GOOD        U    006E00       1   984  1007  1393
LBFPBFL_NUM         U    00000B       1  1007  1394
LBFPBIN             D    0005E8       8   720   732   267
LBFPBOT             U    003000       1   824   268  1388
LBFPBOT_GOOD        U    006880       1   936   981  1389
LBFPBOT_NUM         U    000016       1   981  1390
LBFPINRM            F    000640       4   743   748   273
LBFPRM              I    0004AA       4   475   222
LBFPRMCT            U    000002       1   748   272
```

```
     SYMBOL        TYPE  VALUE   LENGTH  DEFN  REFERENCES

LBFPRMO             U    003400       1   829   274  1396
LBFPRMOF            U    003700       1   831   275  1400
LBFPRMOF_GOOD       U    007340       1  1034  1047  1401
LBFPRMOF_NUM        U    000006       1  1047  1402
LBFPRMO_GOOD        U    0070C0       1  1010  1031  1397
LBFPRMO_NUM         U    00000A       1  1031  1398
LONGB               F    00031C       4   265   219
MSG                 I    0082FA       4  1334  1188  1247  1280  1308
MSGCMD              C    008342       9  1360  1347  1348
MSGMSG              C    00834B      95  1361  1341  1358  1339
MSGMVC              I    00833C       6  1358  1345
MSGOK               I    008310       2  1343  1340
MSGRET              I    00832A       4  1354  1351
MSGSAVE             F    008330       4  1357  1337  1354
PCINTCD             H    00008E       2   169   186  1166
PCNOTDTA            I    00020C       4   190   187
PCOLDPSW            U    000150       1   171   188  1170  1174  1178  1182
PGMCK               H    008080       2  1165   192
PGMCOMMA            C    0080F6       1  1195  1167
PGMPSW              C    0080FC      36  1197  1170  1171  1172  1174  1175  1176  1178  1179  1180  1182  1183  1184
PROGCHK             H    000200       2   185   177
PROGCODE            C    0080F2       4  1194  1166  1168
PROGMSG             C    0080DE      66  1192  1186  1187
PROGPSW             D    000228       8   198   197
R0                  U    000000       1   119   190   193   210   212  1186  1239  1245  1278  1306  1310  1334  1337  1339  1341
                                                 1343  1354
R1                  U    000001       1   120   367   373   376   383   479   485   488   495   582   588   591  1187  1208  1212
                                                 1214  1246  1279  1307  1348  1358
R10                 U    00000A       1   129   214   216   219   221   224   226   307   308   363   364   419   420   475   476
                                                  531   532   578   579
R11                 U    00000B       1   130
R12                 U    00000C       1   131   156   191   233   311   341   368   398   423   453   480   509   535   556   583
                                                  608
R13                 U    00000D       1   132   192   215   217   220   222   225   227   234   310   343   366   400   422   455
                                                  478   511   534   558   581   610  1190  1218
R14                 U    00000E       1   133   195   196   235   236
R15                 U    00000F       1   134   155   190   193
R2                  U    000002       1   121   307   309   341   363   365   398   419   421   453   475   477   509   531   533
                                                  556   578   580   608  1188  1209  1215  1247  1280  1308  1335  1337  1343  1344
                                                 1345  1347  1354  1355
R3                  U    000003       1   122   307   314   328   334   340   363   377   384   395   419   426   440   446   452
                                                  475   489   496   507   531   538   539   555   578   592   593   606  1210  1215
R4                  U    000004       1   123  1212  1227  1229  1251  1290  1294  1298  1302
R5                  U    000005       1   124   370   373   391   482   485   503   585   588   602  1227  1230  1239  1244  1252
                                                 1253  1262  1266  1270  1274  1310
R6                  U    000006       1   125  1212  1231
R7                  U    000007       1   126   308   317   323   329   335   338   364   379   385   388   396   420   429   435
                                                  441   447   450   476   491   497   500   532   542   543   549   550   553   579
                                                  595   596   599  1213  1233
R8                  U    000008       1   127   308   318   324   330   336   339   364   380   386   389   397   420   430   436
                                                  442   448   451   476   492   498   501   508   532   544   551   554   579   597
                                                  600   607  1225  1231
R9                  U    000009       1   128   371   391   483   503   586   602
RMLONGS             F    00032C       4   271   221
RMSHORTS            F    00030C       4   259   216
RMXTNDS             F    00034C       4   283   226
```

```
     SYMBOL        TYPE  VALUE   LENGTH  DEFN  REFERENCES

SAVER0R5            F    0082D0       4  1325  1239  1310
SAVEREGS            F    00023C       4   200   190   193
SBFPB               H    00035C       2   306   215
SBFPBCT             U    00000B       1   676   254
SBFPBFL             U    001100       1   815   257  1376
SBFPBFL_GOOD        U    0062C0       1   878   901  1377
SBFPBFL_NUM         U    00000B       1   901  1378
SBFPBIN             F    0005B4       4   664   676   255
SBFPBOT             U    001000       1   813   256  1372
SBFPBOT_GOOD        U    006000       1   852   875  1373
SBFPBOT_NUM         U    00000B       1   875  1374
SBFPINRM            F    0005E0       4   687   692   261
SBFPRM              I    0003D2       4   363   217
SBFPRMCT            U    000002       1   692   260
SBFPRMO             U    001200       1   818   262  1380
SBFPRMOF            U    001400       1   820   263  1384
SBFPRMOF_GOOD       U    006700       1   920   933  1385
SBFPRMOF_NUM        U    000006       1   933  1386
SBFPRMO_GOOD        U    006580       1   904   917  1381
SBFPRMO_NUM         U    000006       1   917  1382
SHORTB              F    0002FC       4   253   214
START               H    000280       2   209   174
STRTLABL            U    000000       1   118   168   171   173   176   184   813   815   818   820   824   826   829   831   835
                                                837   840   842   850
VERIFAIL            I    00815A       4  1239  1228
VERIFLEN            U    00000C       1  1420  1209
VERIFTAB            F    0083AC       4  1371  1420  1208
VERIFY              I    008142       2  1225  1213
VERINEXT            I    00814E       4  1229  1311
VERISUB             H    008120       2  1203   234
WANTGOT             C    008290       6  1318  1257  1285
XBFPB               H    000508       2   530   225
XBFPBCT             U    00000B       1   788   278
XBFPBFL             U    005400       1   837   281  1408
XBFPBFL_GOOD        U    007A40       1  1098  1121  1409
XBFPBFL_NUM         U    00000B       1  1121  1410
XBFPBIN             D    000650       8   776   788   279
XBFPBOT             U    005000       1   835   280  1404
XBFPBOT_GOOD        U    0074C0       1  1050  1095  1405
XBFPBOT_NUM         U    000016       1  1095  1406
XBFPINRM            D    000700       8   799   804   285
XBFPRM              I    00055E       4   578   227
XBFPRMCT            U    000002       1   804   284
XBFPRMO             U    005500       1   840   286  1412
XBFPRMOF            U    005A00       1   842   287  1416
XBFPRMOF_GOOD       U    007F80       1  1148  1157  1417
XBFPRMOF_NUM        U    000004       1  1157  1418
XBFPRMO_GOOD        U    007D00       1  1124  1145  1413
XBFPRMO_NUM         U    00000A       1  1145  1414
XTNDB               F    00033C       4   277   224
=AL2(L'MSGMSG)      R    00843E       2  1424  1339
=CL6'Got:  '        C    008446       6  1426  1285
=CL6'Want: '        C    008440       6  1425  1257
=H'0'               H    00843C       2  1423  1334
```

```
MACRO  DEFN  REFERENCES

No defined macros
```

```
   DESC     SYMBOL    SIZE     POS        ADDR

Entry: 0

Image      IMAGE     33868  0000-844B  0000-844B
  Region             33868  0000-844B  0000-844B
    CSECT  BFPSQRTS  33868  0000-844B  0000-844B
   STMT                                                FILE NAME

1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-015-sqrt\bfp-015-sqrt.asm

** NO ERRORS FOUND **
```