```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                      2 ************************************************************************
                                                      3 *
                                                      4 *Testcase IEEE CONVERT FROM FIXED 64
                                                      5 *  Test case capability includes ieee exceptions trappable and
                                                      6 *  otherwise.  Test result, FPCR flags, and DXC saved for all tests.
                                                      7 *  Convert From Fixed does not set the condition code.
                                                      8 *
                                                      9 *
                                                     10 *                        ********************
                                                     11 *                        **   IMPORTANT!   **
                                                     12 *                        ********************
                                                     13 *
                                                     14 *        This test uses the Hercules Diagnose X'008' interface
                                                     15 *        to display messages and thus your .tst runtest script
                                                     16 *        MUST contain a "DIAG8CMD ENABLE" statement within it!
                                                     17 *
                                                     18 *
                                                     19 ************************************************************************



                                                     21 ************************************************************************
                                                     22 *
                                                     23 *                      bfp-011-cvtfrfix64.asm
                                                     24 *
                                                     25 *        This assembly-language source file is part of the
                                                     26 *        Hercules Binary Floating Point Validation Package
                                                     27 *                           by Stephen R. Orso
                                                     28 *
                                                     29 * Copyright 2016 by Stephen R Orso.
                                                     30 * Runtest *Compare dependency removed by Fish on 2022-08-16
                                                     31 * PADCSECT macro/usage removed by Fish on 2022-08-16
                                                     32 *
                                                     33 * Redistribution and use in source and binary forms, with or without
                                                     34 * modification, are permitted provided that the following conditions
                                                     35 * are met:
                                                     36 *
                                                     37 * 1. Redistributions of source code must retain the above copyright
                                                     38 *    notice, this list of conditions and the following disclaimer.
                                                     39 *
                                                     40 * 2. Redistributions in binary form must reproduce the above copyright
                                                     41 *    notice, this list of conditions and the following disclaimer in
                                                     42 *    the documentation and/or other materials provided  with the
                                                     43 *    distribution.
                                                     44 *
                                                     45 * 3. The name of the author may not be used to endorse or promote
                                                     46 *    products derived from this software without specific prior written
                                                     47 *    permission.
                                                     48 *
                                                     49 * DISCLAMER: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER "AS IS"
                                                     50 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
                                                     51 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
                                                     52 * PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT
                                                     53 * HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
                                                     54 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
                                                     55 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
                                                     56 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

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



                                                        63 ********************************************************************************
                                                        64 *
                                                        65 * Tests the following six conversion instructions
                                                        66 *   CONVERT FROM FIXED (64 to short BFP, RRE)
                                                        67 *   CONVERT FROM FIXED (64 to long BFP, RRE)
                                                        68 *   CONVERT FROM FIXED (64 to extended BFP, RRE)
                                                        69 *   CONVERT FROM FIXED (64 to short BFP, RRF-e)
                                                        70 *   CONVERT FROM FIXED (64 to long BFP, RRF-e)
                                                        71 *   CONVERT FROM FIXED (64 to extended BFP, RRF-e)
                                                        72 *
                                                        73 * Test data is compiled into this program.  The test script that runs
                                                        74 * this program can provide alternative test data through Hercules R
                                                        75 * commands.
                                                        76 *
                                                        77 * Test Case Order
                                                        78 * 1) Int-32 to Short BFP
                                                        79 * 2) Int-32 to Short BFP with all rounding modes
                                                        80 * 3) Int-32 to Long BFP
                                                        81 * 4) Int-32 to Long BFP with all rounding modes
                                                        82 * 5) Int-32 to Extended BFP
                                                        83 *
                                                        84 * Provided test data:
                                                        85 *      1, 2, 4, -2,
                                                        86 *      9 223 372 036 854 775 807   (0x7FFFFFFFFFFF)
                                                        87 *      -9 223 372 036 854 775 807  (0x800000000000)
                                                        88 *   The last two values trigger inexact exceptions when converted
                                                        89 *   to long or short BFP and are used to exhaustively test
                                                        90 *   operation of the various rounding modes.  Int-64 to extended
                                                        91 *   BFP is always exact.
                                                        92 *
                                                        93 * Also tests the following floating point support instructions
                                                        94 *   LOAD  (Short)
                                                        95 *   LOAD  (Long)
                                                        96 *   LOAD FPC
                                                        97 *   SET BFP ROUNDING MODE 2-BIT
                                                        98 *   SET BFP ROUNDING MODE 3-BIT
                                                        99 *   STORE (Short)
                                                       100 *   STORE (Long)
                                                       101 *   STORE FPC
                                                       102 *
                                                       103 ********************************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                   105 *
                                                   106 *  Note: for compatibility with the z/CMS test rig, do not change
                                                   107 *  or use R11, R14, or R15.  Everything else is fair game.
                                                   108 *
                               00000000  00006133  109 BFPCVTFF START 0
                               00000000  00000001  110 STRTLABL EQU   *
                               00000000  00000001  111 R0       EQU   0                   Work register for cc extraction
                               00000001  00000001  112 R1       EQU   1
                               00000002  00000001  113 R2       EQU   2                   Holds count of test input values
                               00000003  00000001  114 R3       EQU   3                   Points to next test input value(s)
                               00000004  00000001  115 R4       EQU   4                   Available
                               00000005  00000001  116 R5       EQU   5                   Available
                               00000006  00000001  117 R6       EQU   6                   Available
                               00000007  00000001  118 R7       EQU   7                   Pointer to next result value(s)
                               00000008  00000001  119 R8       EQU   8                   Pointer to next FPCR result
                               00000009  00000001  120 R9       EQU   9                   Rounding tests top of outer loop
                               0000000A  00000001  121 R10      EQU   10                  Pointer to test address list
                               0000000B  00000001  122 R11      EQU   11                  **Reserved for z/CMS test rig
                               0000000C  00000001  123 R12      EQU   12                  Holds number of test cases in set
                               0000000D  00000001  124 R13      EQU   13                  Mainline return address
                               0000000E  00000001  125 R14      EQU   14                  **Return address for z/CMS test rig
                               0000000F  00000001  126 R15      EQU   15                  **Base register on z/CMS or Hyperion
                                                   127 *
                                                   128 * Floating Point Register equates to keep the cross reference clean
                                                   129 *
                               00000000  00000001  130 FPR0     EQU   0
                               00000001  00000001  131 FPR1     EQU   1
                               00000002  00000001  132 FPR2     EQU   2
                               00000003  00000001  133 FPR3     EQU   3
                               00000004  00000001  134 FPR4     EQU   4
                               00000005  00000001  135 FPR5     EQU   5
                               00000006  00000001  136 FPR6     EQU   6
                               00000007  00000001  137 FPR7     EQU   7
                               00000008  00000001  138 FPR8     EQU   8
                               00000009  00000001  139 FPR9     EQU   9
                               0000000A  00000001  140 FPR10    EQU   10
                               0000000B  00000001  141 FPR11    EQU   11
                               0000000C  00000001  142 FPR12    EQU   12
                               0000000D  00000001  143 FPR13    EQU   13
                               0000000E  00000001  144 FPR14    EQU   14
                               0000000F  00000001  145 FPR15    EQU   15
                                                   146 *
00000000                       00000000            147          USING *,R15
00000000                       00005D80            148          USING HELPERS,R12
                                                   149 *
                                                   150 * Above works on real iron (R15=0 after sysclear)
                                                   151 * and in z/CMS (R15 points to start of load module)
                                                   152 *


                                                   154 ***********************************************************************
                                                   155 *
                                                   156 * Low core definitions, Restart PSW, and Program Check Routine.
                                                   157 *
                                                   158 ***********************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000000                     00000000  0000008E   160          ORG   STRTLABL+X'8E'      Program check interrution code
0000008E  0000                                   161 PCINTCD  DS    H
                                                 162 *
                             00000150  00000001  163 PCOLDPSW EQU   STRTLABL+X'150'     z/Arch Program check old PSW
                                                 164 *
00000090                     00000090  000001A0  165          ORG   STRTLABL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                      166          DC    X'0000000180000000',AD(START)
                                                 167 *
000001B0                     000001B0  000001D0  168          ORG   STRTLABL+X'1D0'     z/Arch Program check NEW PSW
000001D0  00000000 00000000                      169          DC    X'0000000000000000',AD(PROGCHK)
                                                 170 *
                                                 171 * Program check routine.  If Data Exception, continue execution at
                                                 172 * the instruction following the program check.  Otherwise, hard wait.
                                                 173 * No need to collect data.  All interesting DXC stuff is captured
                                                 174 * in the FPCR.
                                                 175 *
000001E0                     000001E0  00000200  176          ORG   STRTLABL+X'200'
00000200                                         177 PROGCHK  DS    0H              Program check occured...
00000200  9507 F08F                    0000008F  178          CLI   PCINTCD+1,X'07'  Data Exception?
00000204  A774 0004                    0000020C  179          JNE   PCNOTDTA       ..no, hardwait (not sure if R15 is ok)
00000208  B2B2 F150                    00000150  180          LPSWE PCOLDPSW       ..yes, resume program execution

0000020C  900F F23C                    0000023C  182 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                    0000027C  183          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                    00005D80  184          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                    0000023C  185          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                   187          LTR   R14,R14        Return address provided?
0000021E  077E                                   188          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                    00000228  189          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                      190 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2D8                    000002D8  191 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                      192 SAVEREGS DC    16F'0'          Registers save area
0000027C  00005D80                               193 AHELPERS DC    A(HELPERS)      Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    195 ************************************************************************
                                                    196 *
                                                    197 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                    198 *
00000280  B600 F2E8                        000002E8 199 START    STCTL R0,R0,CTLR0    Store CR0 to enable AFP
00000284  9604 F2E9                        000002E9 200          OI    CTLR0+1,X'04'  Turn on AFP bit
00000288  B700 F2E8                        000002E8 201          LCTL  R0,R0,CTLR0    Reload updated CR0
                                                    202 *
0000028C  41A0 F2F4                        000002F4 203          LA    R10,SHORTS     Point to int-64 test inputs
00000290  4DD0 F344                        00000344 204          BAS   R13,CEGBR      Convert values from fixed to short BFP
00000294  41A0 F324                        00000324 205          LA    R10,RMSHORTS   Point to inputs for rounding mode tests
00000298  4DD0 F388                        00000388 206          BAS   R13,CEGBRA     Convert to short BFP using rm options
                                                    207 *
0000029C  41A0 F304                        00000304 208          LA    R10,LONGS      Point to int-64 test inputs
000002A0  4DD0 F45C                        0000045C 209          BAS   R13,CDGBR      Convert values from fixed to long BFP
000002A4  41A0 F334                        00000334 210          LA    R10,RMLONGS    Point to inputs for rounding mode tests
000002A8  4DD0 F4A0                        000004A0 211          BAS   R13,CDGBRA     Convert to long BFP using rm options
                                                    212 *
000002AC  41A0 F314                        00000314 213          LA    R10,EXTDS      Point to int-64 test inputs
000002B0  4DD0 F574                        00000574 214          BAS   R13,CXGBR      Convert values from fixed to extended
                                                    215 *
                                                    216 ************************************************************************
                                                    217 *                    Verify test results...
                                                    218 ************************************************************************
                                                    219 *
000002B4  58C0 F27C                        0000027C 220          L     R12,AHELPERS     Get address of helper subroutines
000002B8  4DD0 C0A0                        00005E20 221          BAS   R13,VERISUB      Go verify results
000002BC  12EE                                      222          LTR   R14,R14          Was return address provided?
000002BE  077E                                      223          BNZR  R14              Yes, return to z/CMS test rig.
000002C0  B2B2 F2C8                        000002C8 224          LPSWE GOODPSW          Load SUCCESS PSW
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000002C8                                          226          DS    0D            Ensure correct alignment for PSW
000002C8  00020000 00000000                       227 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002D8  00020000 00000000                       228 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                  229 *
000002E8  00000000                                230 CTLR0    DS    F
000002EC  00000000                                231 FPCREGNT DC    X'00000000'    FPCR Reg IEEE exceptions Not Trappable
000002F0  F8000000                                232 FPCREGTR DC    X'F8000000'    FPCR Reg IEEE exceptions TRappable
                                                  233 *
                                                  234 * Input values parameter list, four fullwords:
                                                  235 *      1) Count,
                                                  236 *      2) Address of inputs,
                                                  237 *      3) Address to place results, and
                                                  238 *      4) Address to place DXC/Flags/cc values.
                                                  239 *
000002F4                                          240 SHORTS   DS    0F          int-64 inputs for short BFP testing
000002F4  0000000A                                241          DC    A(INTCOUNT)
000002F8  000005C0                                242          DC    A(INTIN)
000002FC  00001000                                243          DC    A(SBFPOUT)
00000300  00001100                                244          DC    A(SBFPFLGS)
                                                  245 *
00000304                                          246 LONGS    DS    0F          int-64 inputs for long BFP testing
00000304  0000000A                                247          DC    A(INTCOUNT)
00000308  000005C0                                248          DC    A(INTIN)
0000030C  00002000                                249          DC    A(LBFPOUT)
00000310  00002100                                250          DC    A(LBFPFLGS)
                                                  251 *
00000314                                          252 EXTDS    DS    0F          int-64 inputs for Extended BFP testing
00000314  0000000A                                253          DC    A(INTCOUNT)
00000318  000005C0                                254          DC    A(INTIN)
0000031C  00003000                                255          DC    A(XBFPOUT)
00000320  00003200                                256          DC    A(XBFPFLGS)
                                                  257 *
00000324                                          258 RMSHORTS DS    0F          int-64's for short BFP rounding mode tests
00000324  00000006                                259          DC    A(SINTRMCT)
00000328  00000610                                260          DC    A(SINTRMIN) Last two int-64 are only concerns
0000032C  00001200                                261          DC    A(SBFPRMO)  Space for rounding mode results
00000330  00001500                                262          DC    A(SBFPRMOF) Space for rounding mode FPCR contents
                                                  263 *
00000334                                          264 RMLONGS  DS    0F          int-64's for long BFP rounding mode tests
00000334  00000006                                265          DC    A(LINTRMCT)
00000338  00000640                                266          DC    A(LINTRMIN) Last two int-64 are only concerns
0000033C  00002200                                267          DC    A(LBFPRMO)  Space for rounding mode results
00000340  00002700                                268          DC    A(LBFPRMOF) Space for rounding mode FPCR contents
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    270 ************************************************************************
                                                    271 *
                                                    272 * Convert int-64 to short BFP format.  A pair of results is generated
                                                    273 * for each input: one with all exceptions non-trappable, and the second
                                                    274 * with all exceptions trappable.   The FPCR is stored for each result.
                                                    275 *
                                                    276 ************************************************************************


                                                    278
00000344  9823 A000                 00000000    279 CEGBR    LM    R2,R3,0(R10)  Get count and address of test input values
00000348  1222                                  280          LTR   R2,R2         Any test cases?
0000034A  078D                                  281          BZR   R13           ..No, return to caller
0000034C  9878 A008                 00000008    282          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000350  0DC0                                  283          BASR  R12,0         Set top of loop
                                                    284 *
00000352  E310 3000 0004            00000000    285          LG    R1,0(,R3)     Get integer test value
00000358  B29D F2EC                 000002EC    286          LFPC  FPCREGNT      Set exceptions non-trappable
0000035C  B3A4 0081                             287          CEGBR FPR8,R1       Cvt Int in GPR1 to float in FPFPR8
00000360  7080 7000                 00000000    288          STE   FPR8,0(,R7)   Store short BFP result
00000364  B29C 8000                 00000000    289          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                    290 *
00000368  B29D F2F0                 000002F0    291          LFPC  FPCREGTR      Set exceptions trappable
0000036C  B3A4 0081                             292          CEGBR FPR8,R1       Cvt Int in GPR1 to float in FPFPR8
00000370  7080 7004                 00000004    293          STE   FPR8,4(,R7)   Store short BFP result
00000374  B29C 8004                 00000004    294          STFPC 4(R8)         Store resulting FPCR flags and DXC
00000378  4130 3008                 00000008    295          LA    R3,8(,R3)     Point to next input values
0000037C  4170 7008                 00000008    296          LA    R7,8(,R7)     Point to next short BFP converted values
00000380  4180 8008                 00000008    297          LA    R8,8(,R8)     Point to next FPCR/CC result area
00000384  062C                                  298          BCTR  R2,R12        Convert next input value.
00000386  07FD                                  299          BR    R13           All converted; return.
                                                    300 *
                                                    301 * Convert int-64 to short BFP format using each possible rounding mode.
                                                    302 * Ten test results are generated for each input.  A 48-byte test result
                                                    303 * section is used to keep results sets aligned on a quad-double word.
                                                    304 *
                                                    305 * The first four tests use rounding modes specified in the FPCR with
                                                    306 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for the
                                                    307 * first two FPCR-controlled tests and SRNMB (3-bit) is used for the
                                                    308 * last two to get full coverage of that instruction pair.
                                                    309 *
                                                    310 * The next six results use instruction-specified rounding modes.
                                                    311 *
                                                    312 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    313 * prior tests used the default rounding mode.  RNTE is tested
                                                    314 * explicitly as a rounding mode in this section.
                                                    315 *
00000388  9823 A000                 00000000    316 CEGBRA   LM    R2,R3,0(R10)  Get count and address of test input values
0000038C  1222                                  317          LTR   R2,R2         Any test cases?
0000038E  078D                                  318          BZR   R13           ..No, return to caller
00000390  9878 A008                 00000008    319          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000394  0DC0                                  320          BASR  R12,0         Set top of loop
                                                    321 *
00000396  E310 3000 0004            00000000    322          LG    R1,0(,R3)     Get int-64 test value
                                                    323 *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    324 * Convert the Int-64 in GPR1 to float-64 in FPR8 using the rounding
                                                    325 * mode specified in the FPCR.  Mask inexact exceptions.
                                                    326 *
0000039C  B29D F2EC                       000002EC  327          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003A0  B299 0001                       00000001  328          SRNM  1             SET FPCR to RZ, towards zero.
000003A4  B3A4 0481                                 329          CEGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003A8  6080 7000                       00000000  330          STD   FPR8,0*4(,R7) Store short BFP result
000003AC  B29C 8000                       00000000  331          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                    332 *
000003B0  B29D F2EC                       000002EC  333          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003B4  B299 0002                       00000002  334          SRNM  2             SET FPCR to RP, to +infinity
000003B8  B3A4 0481                                 335          CEGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003BC  6080 7004                       00000004  336          STD   FPR8,1*4(,R7) Store short BFP result
000003C0  B29C 8004                       00000004  337          STFPC 1*4(R8)       Store resulting FPCR flags and DXC
                                                    338 *
000003C4  B29D F2EC                       000002EC  339          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003C8  B2B8 0003                       00000003  340          SRNMB 3             SET FPCR to RM, to -infinity
000003CC  B3A4 0481                                 341          CEGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003D0  6080 7008                       00000008  342          STD   FPR8,2*4(,R7) Store short BFP result
000003D4  B29C 8008                       00000008  343          STFPC 2*4(R8)       Store resulting FPCR flags and DXC
                                                    344 *
000003D8  B29D F2EC                       000002EC  345          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003DC  B2B8 0007                       00000007  346          SRNMB 7             RPS, Prepare for Shorter Precision
000003E0  B3A4 0481                                 347          CEGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000003E4  6080 700C                       0000000C  348          STD   FPR8,3*4(,R7) Store short BFP result
000003E8  B29C 800C                       0000000C  349          STFPC 3*4(R8)       Store resulting FPCR flags and DXC
                                                    350 *
                                                    351 * Convert the Int-64 in GPR1 to float-64 in FPFPR8 using the rounding
                                                    352 * mode specified in the M3 field of the instruction.
                                                    353 *
000003EC  B29D F2EC                       000002EC  354          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003F0  B3A4 1081                                 355          CEGBRA FPR8,1,R1,B'0000'  RNTA, to nearest, ties away
000003F4  7080 7010                       00000010  356          STE   FPR8,4*4(,R7) Store short BFP result
000003F8  B29C 8010                       00000010  357          STFPC 4*4(R8)       Store resulting FPCR flags and DXC
                                                    358 *
000003FC  B29D F2EC                       000002EC  359          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000400  B3A4 3081                                 360          CEGBRA FPR8,3,R1,B'0000'  RPS, prepare for shorter precision
00000404  7080 7014                       00000014  361          STE   FPR8,5*4(,R7) Store short BFP result
00000408  B29C 8014                       00000014  362          STFPC 5*4(R8)       Store resulting FPCR flags and DXC
                                                    363 *
0000040C  B29D F2EC                       000002EC  364          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000410  B3A4 4081                                 365          CEGBRA FPR8,4,R1,B'0000'  RNTE, to nearest, ties to even
00000414  7080 7018                       00000018  366          STE   FPR8,6*4(,R7) Store short BFP result
00000418  B29C 8018                       00000018  367          STFPC 6*4(R8)       Store resulting FPCR flags and DXC
                                                    368 *
0000041C  B29D F2EC                       000002EC  369          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000420  B3A4 5081                                 370          CEGBRA FPR8,5,R1,B'0000'  RZ, toward zero
00000424  7080 701C                       0000001C  371          STE   FPR8,7*4(,R7)  Store short BFP result
00000428  B29C 801C                       0000001C  372          STFPC 7*4(R8)       Store resulting FPCR flags and DXC
                                                    373 *
0000042C  B29D F2EC                       000002EC  374          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000430  B3A4 6081                                 375          CEGBRA FPR8,6,R1,B'0000'  RP, to +inf
00000434  7080 7020                       00000020  376          STE   FPR8,8*4(,R7) Store short BFP result
00000438  B29C 8020                       00000020  377          STFPC 8*4(R8)       Store resulting FPCR flags and DXC
                                                    378 *
0000043C  B29D F2EC                       000002EC  379          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00000440  B3A4 7081                          380          CEGBRA FPR8,7,R1,B'0000'  RM, to -inf
00000444  7080 7024                 00000024 381          STE   FPR8,9*4(,R7) Store short BFP result
00000448  B29C 8024                 00000024 382          STFPC 9*4(R8)       Store resulting FPCR flags and DXC
                                             383 *
0000044C  4130 3008                 00000008 384          LA    R3,8(,R3)     Point to next input value
00000450  4170 7030                 00000030 385          LA    R7,12*4(,R7)  Point to next short BFP rounded set
00000454  4180 8030                 00000030 386          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
00000458  062C                               387          BCTR  R2,R12        Convert next input value.
0000045A  07FD                               388          BR    R13           All converted; return.
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                     390 ************************************************************************
                                                     391 *
                                                     392 * Convert int-64 to long BFP format.  A pair of results is generated
                                                     393 * for each input: one with all exceptions non-trappable, and the second
                                                     394 * with all exceptions trappable.   The FPCR is stored for each result.
                                                     395 * Conversion of a 32-bit integer to long is always exact; no exceptions
                                                     396 * are expected
                                                     397 *
                                                     398 ************************************************************************


0000045C  9823 A000                 00000000    400 CDGBR    LM    R2,R3,0(R10)  Get count and address of test input values
00000460  9878 A008                 00000008    401          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000464  1222                                  402          LTR   R2,R2         Any test cases?
00000466  078D                                  403          BZR   R13           ..No, return to caller
00000468  0DC0                                  404          BASR  R12,0         Set top of loop
                                                405 *
0000046A  E310 3000 0004            00000000    406          LG    R1,0(,R3)    Get integer test value
00000470  B29D F2EC                 000002EC    407          LFPC  FPCREGNT      Set exceptions non-trappable
00000474  B3A5 0081                             408          CDGBR FPR8,R1         Cvt Int in GPR1 to float in FPFPR8
00000478  6080 7000                 00000000    409          STD   FPR8,0(,R7)    Store long BFP result
0000047C  B29C 8000                 00000000    410          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                411 *
00000480  B29D F2F0                 000002F0    412          LFPC  FPCREGTR      Set exceptions trappable
00000484  B3A5 0081                             413          CDGBR FPR8,R1         Cvt Int in GPR1 to float in FPFPR8
00000488  6080 7008                 00000008    414          STD   FPR8,8(,R7)    Store long BFP result
0000048C  B29C 8004                 00000004    415          STFPC 4(R8)         Store resulting FPCR flags and DXC
00000490  4130 3008                 00000008    416          LA    R3,8(,R3)    point to next input value
00000494  4170 7010                 00000010    417          LA    R7,16(,R7)   Point to next long BFP converted value
00000498  4180 8008                 00000008    418          LA    R8,8(,R8)    Point to next FPCR/CC result area
0000049C  062C                                  419          BCTR  R2,R12        Convert next input value.
0000049E  07FD                                  420          BR    R13           All converted; return.
                                                421 *
                                                422 * Convert int-64 to long BFP format using each possible rounding mode.
                                                423 * Ten test results are generated for each input.  A 48-byte test result
                                                424 * section is used to keep results sets aligned on a quad-double word.
                                                425 *
                                                426 * The first four tests use rounding modes specified in the FPCR with
                                                427 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for the
                                                428 * first two FPCR-controlled tests and SRNMB (3-bit) is used for the
                                                429 * last two to get full coverage of that instruction pair.
                                                430 *
                                                431 * The next six results use instruction-specified rounding modes.
                                                432 *
                                                433 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                434 * prior tests used the default rounding mode.  RNTE is tested
                                                435 * explicitly as a rounding mode in this section.
                                                436 *
000004A0  9823 A000                 00000000    437 CDGBRA   LM    R2,R3,0(R10)  Get count and address of test input values
000004A4  9878 A008                 00000008    438          LM    R7,R8,8(R10)  Get address of result area and flag area.
000004A8  1222                                  439          LTR   R2,R2         Any test cases?
000004AA  078D                                  440          BZR   R13           ..No, return to caller
000004AC  0DC0                                  441          BASR  R12,0         Set top of loop
                                                442 *
000004AE  E310 3000 0004            00000000    443          LG    R1,0(,R3)     Get int-64 test value
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    444 *
                                                    445 * Convert the Int-64 in GPR1 to float-64 in FPR8 using the rounding
                                                    446 * mode specified in the FPCR.  Mask inexact exceptions.
                                                    447 *
000004B4  B29D F2EC                    000002EC    448          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004B8  B299 0001                    00000001    449          SRNM  1             SET FPCR to RZ, towards zero.
000004BC  B3A5 0481                                450          CDGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004C0  6080 7000                    00000000    451          STD   FPR8,0*8(,R7) Store long BFP result
000004C4  B29C 8000                    00000000    452          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                    453 *
000004C8  B29D F2EC                    000002EC    454          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004CC  B299 0002                    00000002    455          SRNM  2             SET FPCR to RP, to +infinity
000004D0  B3A5 0481                                456          CDGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004D4  6080 7008                    00000008    457          STD   FPR8,1*8(,R7) Store long BFP result
000004D8  B29C 8004                    00000004    458          STFPC 1*4(R8)       Store resulting FPCR flags and DXC
                                                    459 *
000004DC  B29D F2EC                    000002EC    460          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004E0  B2B8 0003                    00000003    461          SRNMB 3             SET FPCR to RM, to -infinity
000004E4  B3A5 0481                                462          CDGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004E8  6080 7010                    00000010    463          STD   FPR8,2*8(,R7) Store long BFP result
000004EC  B29C 8008                    00000008    464          STFPC 2*4(R8)       Store resulting FPCR flags and DXC
                                                    465 *
000004F0  B29D F2EC                    000002EC    466          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004F4  B2B8 0007                    00000007    467          SRNMB 7             RPS, Prepare for Shorter Precision
000004F8  B3A5 0481                                468          CDGBRA FPR8,0,R1,B'0100'  FPCR ctl'd rounding, inexact masked
000004FC  6080 7018                    00000018    469          STD   FPR8,3*8(,R7) Store long BFP result
00000500  B29C 800C                    0000000C    470          STFPC 3*4(R8)       Store resulting FPCR flags and DXC
                                                    471 *
                                                    472 * Convert the Int-64 in GPR1 to float-64 in FPFPR8 using the rounding
                                                    473 * mode specified in the M3 field of the instruction.
                                                    474 *
00000504  B29D F2EC                    000002EC    475          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000508  B3A5 1081                                476          CDGBRA FPR8,1,R1,B'0000'  RNTA, to nearest, ties away
0000050C  6080 7020                    00000020    477          STD   FPR8,4*8(,R7) Store long BFP result
00000510  B29C 8010                    00000010    478          STFPC 4*4(R8)       Store resulting FPCR flags and DXC
                                                    479 *
00000514  B29D F2EC                    000002EC    480          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000518  B3A5 3081                                481          CDGBRA FPR8,3,R1,B'0000'  RPS, prepare for shorter precision
0000051C  6080 7028                    00000028    482          STD   FPR8,5*8(,R7) Store long BFP result
00000520  B29C 8014                    00000014    483          STFPC 5*4(R8)       Store resulting FPCR flags and DXC
                                                    484 *
00000524  B29D F2EC                    000002EC    485          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000528  B3A5 4081                                486          CDGBRA FPR8,4,R1,B'0000'  RNTE, to nearest, ties to even
0000052C  6080 7030                    00000030    487          STD   FPR8,6*8(,R7) Store long BFP result
00000530  B29C 8018                    00000018    488          STFPC 6*4(R8)       Store resulting FPCR flags and DXC
                                                    489 *
00000534  B29D F2EC                    000002EC    490          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000538  B3A5 5081                                491          CDGBRA FPR8,5,R1,B'0000'  RZ, toward zero
0000053C  6080 7038                    00000038    492          STD   FPR8,7*8(,R7) Store long BFP result
00000540  B29C 801C                    0000001C    493          STFPC 7*4(R8)       Store resulting FPCR flags and DXC
                                                    494 *
00000544  B29D F2EC                    000002EC    495          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000548  B3A5 6081                                496          CDGBRA FPR8,6,R1,B'0000'  RP, to +inf
0000054C  6080 7040                    00000040    497          STD   FPR8,8*8(,R7) Store long BFP result
00000550  B29C 8020                    00000020    498          STFPC 8*4(R8)       Store resulting FPCR flags and DXC
                                                    499 *
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00000554  B29D F2EC                      000002EC   500          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000558  B3A5 7081                                 501          CDGBRA FPR8,7,R1,B'0000'  RM, to -inf
0000055C  6080 7048                      00000048   502          STD   FPR8,9*8(,R7) Store long BFP result
00000560  B29C 8024                      00000024   503          STFPC 9*4(R8)       Store resulting FPCR flags and DXC
                                                    504 *
00000564  4130 3008                      00000008   505          LA    R3,8(,R3)     Point to next input value
00000568  4170 7050                      00000050   506          LA    R7,10*8(,R7)  Point to next long BFP rounded value set
0000056C  4180 8030                      00000030   507          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
00000570  062C                                      508          BCTR  R2,R12        Convert next input value.
00000572  07FD                                      509          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    511 ************************************************************************
                                                    512 *
                                                    513 * Convert int-64 to extended BFP format.  A pair of results is
                                                    514 * generated * for each input: one with all exceptions non-trappable,
                                                    515 * and the second with all exceptions trappable.   The FPCR is stored
                                                    516 * for each result.  Conversion of an int-64to extended is always exact;
                                                    517 * no exceptions are expected.
                                                    518 *
                                                    519 ************************************************************************


00000574  9823 A000                 00000000   521 CXGBR    LM    R2,R3,0(R10)  Get count and address of test input values
00000578  9878 A008                 00000008   522          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000057C  1222                                  523          LTR   R2,R2         Any test cases?
0000057E  078D                                  524          BZR   R13           ..No, return to caller
00000580  0DC0                                  525          BASR  R12,0         Set top of loop
                                                526 *
00000582  E310 3000 0004            00000000   527          LG    R1,0(,R3)     Get integer test value
00000588  B29D F2EC                 000002EC   528          LFPC  FPCREGNT      Set exceptions non-trappable
0000058C  B3A6 0081                            529          CXGBR FPR8,R1       Cvt Int in GPR1 to float in FPR8-FPR10
00000590  6080 7000                 00000000   530          STD   FPR8,0(,R7)   Store extended BFP result part 1
00000594  60A0 7008                 00000008   531          STD   FPR10,8(,R7)  Store extended BFP result part 2
00000598  B29C 8000                 00000000   532          STFPC 0(R8)         Store resulting FPCR flags and DXC
                                                533 *
0000059C  B29D F2F0                 000002F0   534          LFPC  FPCREGTR      Set exceptions trappable
000005A0  B3A6 0081                            535          CXGBR FPR8,R1       Cvt Int in GPR1 to float in FPR8-FPR10
000005A4  6080 7010                 00000010   536          STD   FPR8,16(,R7)  Store extended BFP result part 1
000005A8  60A0 7018                 00000018   537          STD   FPR10,24(,R7) Store extended BFP result part 2
000005AC  B29C 8004                 00000004   538          STFPC 4(R8)         Store resulting FPCR flags and DXC
000005B0  4130 3008                 00000008   539          LA    R3,8(,R3)     Point to next input value
000005B4  4170 7020                 00000020   540          LA    R7,32(,R7)    Point to next extended BFP converted value
000005B8  4180 8008                 00000008   541          LA    R8,8(,R8)     Point to next FPCR/CC result area
000005BC  062C                                  542          BCTR  R2,R12        Convert next input value.
000005BE  07FD                                  543          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    545 ************************************************************************
                                                    546 *
                                                    547 * Int-64 inputs for Convert From Fixed testing.  The same set of
                                                    548 * inputs are used for short, long, and extended formats.  The last two
                                                    549 * values are used for rounding mode tests for short and long only;
                                                    550 * conversion of int-64 to extended is always exact.
                                                    551 *
                                                    552 ************************************************************************


000005C0                                            554 INTIN    DS    0D
000005C0  00000000 00000001                        555          DC    FD'1'
000005C8  00000000 00000002                        556          DC    FD'2'
000005D0  00000000 00000004                        557          DC    FD'4'
000005D8  FFFFFFFF FFFFFFFE                        558          DC    FD'-2'                    X'FFFFFFFF FFFFFFFE'
                                                    559 *
                                                    560 *                                 Below inexact and incre. for short & long
000005E0  7FFFFFFF FFFFFFFF                        561          DC    FD'9223372036854775807'   X'7FFFFFFF FFFFFFFF'
000005E8  80000000 00000001                        562          DC    FD'-9223372036854775807'  X'80000000 00000001'
                                                    563 *
                                                    564 *                                 Below exact for all
000005F0  7FFFFF80 00000000                        565          DC    FD'9223371487098961920'   X'7FFFFF80 00000000'
000005F8  80000080 00000000                        566          DC    FD'-9223371487098961920'  X'80000080 00000000'
                                                    567 *
                                                    568 *                                 Below exact for long and extended
00000600  7FFFFFFF FFFFFC00                        569          DC    FD'9223372036854774784'   X'7FFFFFFF FFFFFC00'
00000608  80000000 00000400                        570          DC    FD'-9223372036854774784'  X'80000000 00000400'
                            0000000A  00000001    571 INTCOUNT EQU   (*-INTIN)/8    Count of int-64 input values
                                                    572 *
                                                    573 * int-64 inputs for exhaustive short BFP rounding mode tests
                                                    574 *
00000610                                            575 SINTRMIN DS    0D            Values for short BFP rounding tests
                                                    576 *                             Below rounds nearest away from zero
00000610  7FFFFFE0 00000000                        577          DC    FD'9223371899415822336'   X'7FFFFFE0 00000000'
00000618  80000020 00000000                        578          DC    FD'-9223371899415822336'  X'80000020 00000000'
                                                    579 *                             Below rounds nearest tie
00000620  7FFFFFC0 00000000                        580          DC    FD'9223371761976868864'   X'7FFFFFC0 00000000'
00000628  80000040 00000000                        581          DC    FD'-9223371761976868864'  X'80000040 00000000'
                                                    582 *                             Below rounds nearest toward zero
00000630  7FFFFFA0 00000000                        583          DC    FD'9223371624537915392'   X'7FFFFFA0 00000000'
00000638  80000060 00000000                        584          DC    FD'-9223371624537915392'  X'80000060 00000000'
                            00000006  00000001    585 SINTRMCT EQU   (*-SINTRMIN)/8  Count of int-64 for rounding tests
                                                    586 *
                                                    587 * int-64 inputs for exhaustive long BFP rounding mode tests
                                                    588 *
00000640                                            589 LINTRMIN DS    0D            Values for short BFP rounding mode tests
                                                    590 *                             Below rounds nearest away from zero
00000640  7FFFFFFF FFFFFF00                        591          DC    FD'9223372036854775552'   X'7FFFFFFF FFFFFF00'
00000648  80000000 00000100                        592          DC    FD'-9223372036854775552'  X'80000000 00000100'
                                                    593 *                             Below rounds nearest tie
00000650  7FFFFFFF FFFFFE00                        594          DC    FD'9223372036854775296'   X'7FFFFFFF FFFFFE00'
00000658  80000000 00000200                        595          DC    FD'-9223372036854775296'  X'80000000 00000200'
                                                    596 *                             Below rounds nearest toward zero
00000660  7FFFFFFF FFFFFD00                        597          DC    FD'9223372036854775040'   X'7FFFFFFF FFFFFD00'
00000668  80000000 00000300                        598          DC    FD'-9223372036854775040'  X'80000000 00000300'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                00000006  00000001   599 LINTRMCT EQU   (*-LINTRMIN)/8  Count of int-64 for rounding tests
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          601 ********************************************************************************
                                          602 *                    ACTUAL results saved here
                                          603 ********************************************************************************
                                          604 *
                                          605 *                 Locations for ACTUAL results
                                          606 *
                                          607 *
                          00001000  00000001  608 SBFPOUT  EQU   STRTLABL+X'1000'    Int-64 results from short BFP inputs
                                          609 *                                    ..6 pairs used, room for 32 pairs
                          00001100  00000001  610 SBFPFLGS EQU   STRTLABL+X'1100'    FPCR flags and DXC from short BFP
                                          611 *                                    ..6 pairs used, room for 32 pairs
                          00001200  00000001  612 SBFPRMO  EQU   STRTLABL+X'1200'    Short BFP rounding mode results
                                          613 *                                    ..2 sets used, room for 16
                          00001500  00000001  614 SBFPRMOF EQU   STRTLABL+X'1500'    Short BFP rndg mode FPCR contents
                                          615 *                                    ..2 sets used, room for 32
                                          616 *                                    ..next available location X'1600'
                                          617 *
                          00002000  00000001  618 LBFPOUT  EQU   STRTLABL+X'2000'    Int-64 results from long BFP inputs
                                          619 *                                    ..6 pairs used, room for 16 pairs
                          00002100  00000001  620 LBFPFLGS EQU   STRTLABL+X'2100'    Long BFP FPCR contents
                                          621 *                                    ..6 pairs used, room for 32 pairs
                          00002200  00000001  622 LBFPRMO  EQU   STRTLABL+X'2200'    Long BFP rounding mode results
                                          623 *                                    ..2 sets used, room for 16 sets
                          00002700  00000001  624 LBFPRMOF EQU   STRTLABL+X'2700'    Long BFP rndg mode FPCR contents
                                          625 *                                    ..2 sets used, room for 16 sets
                                          626 *                                    ..next available location X'2A00'
                                          627 *
                          00003000  00000001  628 XBFPOUT  EQU   STRTLABL+X'3000'    Int-64 results from extended BFP
                                          629 *                                    ..6 pairs used, room for 16 pairs
                          00003200  00000001  630 XBFPFLGS EQU   STRTLABL+X'3200'    CLXBR restulting FPCR contents
                                          631 *                                    ..6 pairs used, room for 16 pairs
                                          632 *                                    ..next available location X'3300'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    634 ********************************************************************************
                                                    635 *                       EXPECTED results
                                                    636 ********************************************************************************
                                                    637 *
00000670                      00000670  00004000    638          ORG   STRTLABL+X'4000'   (past end of actual results)
                                                    639 *
                              00004000  00000001    640 SBFPOUT_GOOD EQU *
00004000  C3C5C7C2 D9409985                         641  DC CL48'CEGBR result pairs 1-2'
00004030  3F800000 3F800000                         642  DC XL16'3F8000003F800004000000040000000'
00004040  C3C5C7C2 D9409985                         643  DC CL48'CEGBR result pairs 3-4'
00004070  40800000 40800000                         644  DC XL16'40800000408000C0000000C0000000'
00004080  C3C5C7C2 D9409985                         645  DC CL48'CEGBR result pairs 5-6'
000040B0  5F000000 5F000000                         646  DC XL16'5F0000005F000000DF000000DF000000'
000040C0  C3C5C7C2 D9409985                         647  DC CL48'CEGBR result pairs 7-8'
000040F0  5EFFFFFF 5EFFFFFF                         648  DC XL16'5EFFFFFF5EFFFFFFDEFFFFFFDEFFFFFF'
00004100  C3C5C7C2 D9409985                         649  DC CL48'CEGBR result pairs 9-10'
00004130  5F000000 5F000000                         650  DC XL16'5F0000005F000000DF000000DF000000'
                              00000005  00000001    651 SBFPOUT_NUM EQU (*-SBFPOUT_GOOD)/64
                                                    652 *
                                                    653 *
                              00004140  00000001    654 SBFPFLGS_GOOD EQU *
00004140  C3C5C7C2 D940C6D7                         655  DC CL48'CEGBR FPCR pairs 1-2'
00004170  00000000 F8000000                         656  DC XL16'00000000F8000000000000000F8000000'
00004180  C3C5C7C2 D940C6D7                         657  DC CL48'CEGBR FPCR pairs 3-4'
000041B0  00000000 F8000000                         658  DC XL16'00000000F8000000000000000F8000000'
000041C0  C3C5C7C2 D940C6D7                         659  DC CL48'CEGBR FPCR pairs 5-6'
000041F0  00080000 F8000C00                         660  DC XL16'00080000F8000C0000080000F8000C00'
00004200  C3C5C7C2 D940C6D7                         661  DC CL48'CEGBR FPCR pairs 7-8'
00004230  00000000 F8000000                         662  DC XL16'00000000F8000000000000000F8000000'
00004240  C3C5C7C2 D940C6D7                         663  DC CL48'CEGBR FPCR pairs 9-10'
00004270  00080000 F8000C00                         664  DC XL16'00080000F8000C0000080000F8000C00'
                              00000005  00000001    665 SBFPFLGS_NUM EQU (*-SBFPFLGS_GOOD)/64
                                                    666 *
                                                    667 *
                              00004280  00000001    668 SBFPRMO_GOOD EQU *
00004280  C3C5C7C2 D9C1404E                         669  DC CL48'CEGBRA +away result FPCRmodes 1-3, 7'
000042B0  5EFFFFFF 5F000000                         670  DC XL16'5EFFFFFF5F0000005EFFFFFF5EFFFFFF'
000042C0  C3C5C7C2 D9C1404E                         671  DC CL48'CEGBRA +away result M3 modes 1, 3-5'
000042F0  5F000000 5EFFFFFF                         672  DC XL16'5F0000005EFFFFFF5F0000005EFFFFFF'
00004300  C3C5C7C2 D9C1404E                         673  DC CL48'CEGBRA +away result M3 modes 6, 7'
00004330  5F000000 5EFFFFFF                         674  DC XL16'5F0000005EFFFFFF0000000000000000'
00004340  C3C5C7C2 D9C14060                         675  DC CL48'CEGBRA -away result FPCRmodes 1-3, 7'
00004370  DEFFFFFF DEFFFFFF                         676  DC XL16'DEFFFFFFDEFFFFFFDF000000DEFFFFFF'
00004380  C3C5C7C2 D9C14060                         677  DC CL48'CEGBRA -away result M3 modes 1, 3-5'
000043B0  DF000000 DEFFFFFF                         678  DC XL16'DF000000DEFFFFFFDF000000DEFFFFFF'
000043C0  C3C5C7C2 D9C14060                         679  DC CL48'CEGBRA -away result M3 modes 6, 7'
000043F0  DEFFFFFF DF000000                         680  DC XL16'DEFFFFFFDF000000000000000000000000'
00004400  C3C5C7C2 D9C1404E                         681  DC CL48'CEGBRA +tie result FPCRmodes 1-3, 7'
00004430  5EFFFFFF 5F000000                         682  DC XL16'5EFFFFFF5F0000005EFFFFFF5EFFFFFF'
00004440  C3C5C7C2 D9C1404E                         683  DC CL48'CEGBRA +tie result M3 modes 1, 3-5'
00004470  5F000000 5EFFFFFF                         684  DC XL16'5F0000005EFFFFFF5F0000005EFFFFFF'
00004480  C3C5C7C2 D9C1404E                         685  DC CL48'CEGBRA +tie result M3 modes 6, 7'
000044B0  5F000000 5EFFFFFF                         686  DC XL16'5F0000005EFFFFFF0000000000000000'
000044C0  C3C5C7C2 D9C14060                         687  DC CL48'CEGBRA -tie result FPCRmodes 1-3, 7'
000044F0  DEFFFFFF DEFFFFFF                         688  DC XL16'DEFFFFFFDEFFFFFFDF000000DEFFFFFF'
00004500  C3C5C7C2 D9C14060                         689  DC CL48'CEGBRA -tie result M3 modes 1, 3-5'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00004530  DF000000 DEFFFFFF                          690  DC XL16'DF000000DEFFFFFFDF000000DEFFFFFF'
00004540  C3C5C7C2 D9C14060                          691  DC CL48'CEGBRA -tie result M3 modes 6, 7'
00004570  DEFFFFFF DF000000                          692  DC XL16'DEFFFFFFDF000000000000000000000000'
00004580  C3C5C7C2 D9C1404E                          693  DC CL48'CEGBRA +tozero result FPCRmodes 1-3, 7'
000045B0  5EFFFFFF 5F000000                          694  DC XL16'5EFFFFFF5F0000005EFFFFFF5EFFFFFF'
000045C0  C3C5C7C2 D9C1404E                          695  DC CL48'CEGBRA +tozero result M3 modes 1, 3-5'
000045F0  5EFFFFFF 5EFFFFFF                          696  DC XL16'5EFFFFFF5EFFFFFF5EFFFFFF5EFFFFFF'
00004600  C3C5C7C2 D9C1404E                          697  DC CL48'CEGBRA +tozero result M3 modes 6, 7'
00004630  5F000000 5EFFFFFF                          698  DC XL16'5F0000005EFFFFFF0000000000000000'
00004640  C3C5C7C2 D9C14060                          699  DC CL48'CEGBRA -tozero result FPCRmodes 1-3, 7'
00004670  DEFFFFFF DEFFFFFF                          700  DC XL16'DEFFFFFFDEFFFFFFDF000000DEFFFFFF'
00004680  C3C5C7C2 D9C14060                          701  DC CL48'CEGBRA -tozero result M3 modes 1, 3-5'
000046B0  DEFFFFFF DEFFFFFF                          702  DC XL16'DEFFFFFFDEFFFFFFDEFFFFFFDEFFFFFF'
000046C0  C3C5C7C2 D9C14060                          703  DC CL48'CEGBRA -tozero result M3 modes 6, 7'
000046F0  DEFFFFFF DF000000                          704  DC XL16'DEFFFFFFDF000000000000000000000000'
                             00000012  00000001    705 SBFPRMO_NUM EQU (*-SBFPRMO_GOOD)/64
                                                  706 *
                                                  707 *
                             00004700  00000001    708 SBFPRMOF_GOOD EQU *
00004700  C3C5C7C2 D9C1404E                          709  DC CL48'CEGBRA +away FPCRmodes 1-3, 7 FPCR'
00004730  00000001 00000002                          710  DC XL16'00000001000000020000000300000007'
00004740  C3C5C7C2 D9C1404E                          711  DC CL48'CEGBRA +away M3 modes 1, 3-5 FPCR'
00004770  00080000 00080000                          712  DC XL16'00080000000800000008000000080000'
00004780  C3C5C7C2 D9C1404E                          713  DC CL48'CEGBRA +away M3 modes 6, 7 FPCR'
000047B0  00080000 00080000                          714  DC XL16'00080000000800000000000000000000'
000047C0  C3C5C7C2 D9C14060                          715  DC CL48'CEGBRA -away FPCRmodes 1-3, 7 FPCR'
000047F0  00000001 00000002                          716  DC XL16'00000001000000020000000300000007'
00004800  C3C5C7C2 D9C14060                          717  DC CL48'CEGBRA -away M3 modes 1, 3-5 FPCR'
00004830  00080000 00080000                          718  DC XL16'00080000000800000008000000080000'
00004840  C3C5C7C2 D9C14060                          719  DC CL48'CEGBRA -away M3 modes 6, 7 FPCR'
00004870  00080000 00080000                          720  DC XL16'00080000000800000000000000000000'
00004880  C3C5C7C2 D9C1404E                          721  DC CL48'CEGBRA +tie FPCRmodes 1-3, 7 FPCR'
000048B0  00000001 00000002                          722  DC XL16'00000001000000020000000300000007'
000048C0  C3C5C7C2 D9C1404E                          723  DC CL48'CEGBRA +tie M3 modes 1, 3-5 FPCR'
000048F0  00080000 00080000                          724  DC XL16'00080000000800000008000000080000'
00004900  C3C5C7C2 D9C1404E                          725  DC CL48'CEGBRA +tie M3 modes 6, 7 FPCR'
00004930  00080000 00080000                          726  DC XL16'00080000000800000000000000000000'
00004940  C3C5C7C2 D9C14060                          727  DC CL48'CEGBRA -tie FPCRmodes 1-3, 7 FPCR'
00004970  00000001 00000002                          728  DC XL16'00000001000000020000000300000007'
00004980  C3C5C7C2 D9C14060                          729  DC CL48'CEGBRA -tie M3 modes 1, 3-5 FPCR'
000049B0  00080000 00080000                          730  DC XL16'00080000000800000008000000080000'
000049C0  C3C5C7C2 D9C14060                          731  DC CL48'CEGBRA -tie M3 modes 6, 7 FPCR'
000049F0  00080000 00080000                          732  DC XL16'00080000000800000000000000000000'
00004A00  C3C5C7C2 D9C1404E                          733  DC CL48'CEGBRA +tozero FPCRmodes 1-3, 7 FPCR'
00004A30  00000001 00000002                          734  DC XL16'00000001000000020000000300000007'
00004A40  C3C5C7C2 D9C1404E                          735  DC CL48'CEGBRA +tozero M3 modes 1, 3-5 FPCR'
00004A70  00080000 00080000                          736  DC XL16'00080000000800000008000000080000'
00004A80  C3C5C7C2 D9C1404E                          737  DC CL48'CEGBRA +tozero M3 modes 6, 7 FPCR'
00004AB0  00080000 00080000                          738  DC XL16'00080000000800000000000000000000'
00004AC0  C3C5C7C2 D9C14060                          739  DC CL48'CEGBRA -tozero FPCRmodes 1-3, 7 FPCR'
00004AF0  00000001 00000002                          740  DC XL16'00000001000000020000000300000007'
00004B00  C3C5C7C2 D9C14060                          741  DC CL48'CEGBRA -tozero M3 modes 1, 3-5 FPCR'
00004B30  00080000 00080000                          742  DC XL16'00080000000800000008000000080000'
00004B40  C3C5C7C2 D9C14060                          743  DC CL48'CEGBRA -tozero M3 modes 6, 7 FPCR'
00004B70  00080000 00080000                          744  DC XL16'00080000000800000000000000000000'
                             00000012  00000001    745 SBFPRMOF_NUM EQU (*-SBFPRMOF_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                     746 *
                                                     747 *
                              00004B80  00000001   748 LBFPOUT_GOOD EQU *
00004B80  C3C4C7C2 D9409985                          749  DC CL48'CDGBR result pair 1'
00004BB0  3FF00000 00000000                          750  DC XL16'3FF00000000000003FF0000000000000'
00004BC0  C3C4C7C2 D9409985                          751  DC CL48'CDGBR result pair 2'
00004BF0  40000000 00000000                          752  DC XL16'40000000000000004000000000000000'
00004C00  C3C4C7C2 D9409985                          753  DC CL48'CDGBR result pair 3'
00004C30  40100000 00000000                          754  DC XL16'40100000000000004010000000000000'
00004C40  C3C4C7C2 D9409985                          755  DC CL48'CDGBR result pair 4'
00004C70  C0000000 00000000                          756  DC XL16'C0000000000000004000000000000000'
00004C80  C3C4C7C2 D9409985                          757  DC CL48'CDGBR result pair 5'
00004CB0  43E00000 00000000                          758  DC XL16'43E0000000000000043E0000000000000'
00004CC0  C3C4C7C2 D9409985                          759  DC CL48'CDGBR result pair 6'
00004CF0  C3E00000 00000000                          760  DC XL16'C3E0000000000000C3E0000000000000'
                              00000006  00000001   761 LBFPOUT_NUM EQU (*-LBFPOUT_GOOD)/64
                                                     762 *
                                                     763 *
                              00004D00  00000001   764 LBFPFLGS_GOOD EQU *
00004D00  C3C4C7C2 D940C6D7                          765  DC CL48'CDGBR FPCR pairs 1-2'
00004D30  00000000 F8000000                          766  DC XL16'00000000F8000000000000000F8000000'
00004D40  C3C4C7C2 D940C6D7                          767  DC CL48'CDGBR FPCR pairs 3-4'
00004D70  00000000 F8000000                          768  DC XL16'00000000F8000000000000000F8000000'
00004D80  C3C4C7C2 D940C6D7                          769  DC CL48'CDGBR FPCR pairs 5-6'
00004DB0  00080000 F8000C00                          770  DC XL16'00080000F8000C0000080000F8000C00'
                              00000003  00000001   771 LBFPFLGS_NUM EQU (*-LBFPFLGS_GOOD)/64
                                                     772 *
                                                     773 *
                              00004DC0  00000001   774 LBFPRMO_GOOD EQU *
00004DC0  C3C4C7C2 D9C1404E                          775  DC CL48'CDGBRA +away FPCRmodes 1, 2'
00004DF0  43DFFFFF FFFFFFFF                          776  DC XL16'43DFFFFFFFFFFFFF43E0000000000000'
00004E00  C3C4C7C2 D9C1404E                          777  DC CL48'CDGBRA +away FPCRmodes 3, 7'
00004E30  43DFFFFF FFFFFFFF                          778  DC XL16'43DFFFFFFFFFFFFF43DFFFFFFFFFFFFF'
00004E40  C3C4C7C2 D9C1404E                          779  DC CL48'CDGBRA +away M3 modes 1, 3'
00004E70  43E00000 00000000                          780  DC XL16'43E0000000000000043DFFFFFFFFFFFFF'
00004E80  C3C4C7C2 D9C1404E                          781  DC CL48'CDGBRA +away M3 modes 4, 5'
00004EB0  43E00000 00000000                          782  DC XL16'43E0000000000000043DFFFFFFFFFFFFF'
00004EC0  C3C4C7C2 D9C1404E                          783  DC CL48'CDGBRA +away M3 modes 6, 7'
00004EF0  43E00000 00000000                          784  DC XL16'43E0000000000000043DFFFFFFFFFFFFF'
00004F00  C3C4C7C2 D9C14060                          785  DC CL48'CDGBRA -away FPCRmodes 1, 2'
00004F30  C3DFFFFF FFFFFFFF                          786  DC XL16'C3DFFFFFFFFFFFFFC3DFFFFFFFFFFFFF'
00004F40  C3C4C7C2 D9C14060                          787  DC CL48'CDGBRA -away FPCRmodes 3, 7'
00004F70  C3E00000 00000000                          788  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00004F80  C3C4C7C2 D9C14060                          789  DC CL48'CDGBRA -away M3 modes 1, 3'
00004FB0  C3E00000 00000000                          790  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00004FC0  C3C4C7C2 D9C14060                          791  DC CL48'CDGBRA -away M3 modes 4, 5'
00004FF0  C3E00000 00000000                          792  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00005000  C3C4C7C2 D9C14060                          793  DC CL48'CDGBRA -away M3 modes 6, 7'
00005030  C3DFFFFF FFFFFFFF                          794  DC XL16'C3DFFFFFFFFFFFFFC3E0000000000000'
00005040  C3C4C7C2 D9C1404E                          795  DC CL48'CDGBRA +tie FPCRmodes 1, 2'
00005070  43DFFFFF FFFFFFFF                          796  DC XL16'43DFFFFFFFFFFFFF43E0000000000000'
00005080  C3C4C7C2 D9C1404E                          797  DC CL48'CDGBRA +tie FPCRmodes 3, 7'
000050B0  43DFFFFF FFFFFFFF                          798  DC XL16'43DFFFFFFFFFFFFF43DFFFFFFFFFFFFF'
000050C0  C3C4C7C2 D9C1404E                          799  DC CL48'CDGBRA +tie M3 modes 1, 3'
000050F0  43E00000 00000000                          800  DC XL16'43E0000000000000043DFFFFFFFFFFFFF'
00005100  C3C4C7C2 D9C1404E                          801  DC CL48'CDGBRA +tie M3 modes 4, 5'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00005130  43E00000 00000000                          802  DC XL16'43E000000000000043DFFFFFFFFFFFFF'
00005140  C3C4C7C2 D9C1404E                          803  DC CL48'CDGBRA +tie M3 modes 6, 7'
00005170  43E00000 00000000                          804  DC XL16'43E000000000000043DFFFFFFFFFFFFF'
00005180  C3C4C7C2 D9C14060                          805  DC CL48'CDGBRA -tie FPCRmodes 1, 2'
000051B0  C3DFFFFF FFFFFFFF                          806  DC XL16'C3DFFFFFFFFFFFFFC3DFFFFFFFFFFFFF'
000051C0  C3C4C7C2 D9C14060                          807  DC CL48'CDGBRA -tie FPCRmodes 3, 7'
000051F0  C3E00000 00000000                          808  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00005200  C3C4C7C2 D9C14060                          809  DC CL48'CDGBRA -tie M3 modes 1, 3'
00005230  C3E00000 00000000                          810  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00005240  C3C4C7C2 D9C14060                          811  DC CL48'CDGBRA -tie M3 modes 4, 5'
00005270  C3E00000 00000000                          812  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00005280  C3C4C7C2 D9C14060                          813  DC CL48'CDGBRA -tie M3 modes 6, 7'
000052B0  C3DFFFFF FFFFFFFF                          814  DC XL16'C3DFFFFFFFFFFFFFC3E0000000000000'
000052C0  C3C4C7C2 D9C1404E                          815  DC CL48'CDGBRA +tozero FPCRmodes 1, 2'
000052F0  43DFFFFF FFFFFFFF                          816  DC XL16'43DFFFFFFFFFFFFF43E0000000000000'
00005300  C3C4C7C2 D9C1404E                          817  DC CL48'CDGBRA +tozero FPCRmodes 3, 7'
00005330  43DFFFFF FFFFFFFF                          818  DC XL16'43DFFFFFFFFFFFFF43DFFFFFFFFFFFFF'
00005340  C3C4C7C2 D9C1404E                          819  DC CL48'CDGBRA +tozero M3 modes 1, 3'
00005370  43DFFFFF FFFFFFFF                          820  DC XL16'43DFFFFFFFFFFFFF43DFFFFFFFFFFFFF'
00005380  C3C4C7C2 D9C1404E                          821  DC CL48'CDGBRA +tozero M3 modes 4, 5'
000053B0  43DFFFFF FFFFFFFF                          822  DC XL16'43DFFFFFFFFFFFFF43DFFFFFFFFFFFFF'
000053C0  C3C4C7C2 D9C1404E                          823  DC CL48'CDGBRA +tozero M3 modes 6, 7'
000053F0  43E00000 00000000                          824  DC XL16'43E000000000000043DFFFFFFFFFFFFF'
00005400  C3C4C7C2 D9C14060                          825  DC CL48'CDGBRA -tozero FPCRmodes 1, 2'
00005430  C3DFFFFF FFFFFFFF                          826  DC XL16'C3DFFFFFFFFFFFFFC3DFFFFFFFFFFFFF'
00005440  C3C4C7C2 D9C14060                          827  DC CL48'CDGBRA -tozero FPCRmodes 3, 7'
00005470  C3E00000 00000000                          828  DC XL16'C3E0000000000000C3DFFFFFFFFFFFFF'
00005480  C3C4C7C2 D9C14060                          829  DC CL48'CDGBRA -tozero M3 modes 1, 3'
000054B0  C3DFFFFF FFFFFFFF                          830  DC XL16'C3DFFFFFFFFFFFFFC3DFFFFFFFFFFFFF'
000054C0  C3C4C7C2 D9C14060                          831  DC CL48'CDGBRA -tozero M3 modes 4, 5'
000054F0  C3DFFFFF FFFFFFFF                          832  DC XL16'C3DFFFFFFFFFFFFFC3DFFFFFFFFFFFFF'
00005500  C3C4C7C2 D9C14060                          833  DC CL48'CDGBRA -tozero M3 modes 6, 7'
00005530  C3DFFFFF FFFFFFFF                          834  DC XL16'C3DFFFFFFFFFFFFFC3E0000000000000'
                            0000001E  00000001     835 LBFPRMO_NUM EQU (*-LBFPRMO_GOOD)/64
                                                   836 *
                                                   837 *
                            00005540  00000001     838 LBFPRMOF_GOOD EQU *
00005540  C3C4C7C2 D9C1404E                          839  DC CL48'CDGBRA +away FPCRmodes 1-3, 7 FPCR'
00005570  00000001 00000002                          840  DC XL16'00000001000000020000000300000007'
00005580  C3C4C7C2 D9C1404E                          841  DC CL48'CDGBRA +away M3 modes 1, 3-5 FPCR'
000055B0  00080000 00080000                          842  DC XL16'00080000000800000008000000080000'
000055C0  C3C4C7C2 D9C1404E                          843  DC CL48'CDGBRA +away M3 modes 6, 7 FPCR'
000055F0  00080000 00080000                          844  DC XL16'00080000000800000000000000000000'
00005600  C3C4C7C2 D9C14060                          845  DC CL48'CDGBRA -away FPCRmodes 1-3, 7 FPCR'
00005630  00000001 00000002                          846  DC XL16'00000001000000020000000300000007'
00005640  C3C4C7C2 D9C14060                          847  DC CL48'CDGBRA -away M3 modes 1, 3-5 FPCR'
00005670  00080000 00080000                          848  DC XL16'00080000000800000008000000080000'
00005680  C3C4C7C2 D9C14060                          849  DC CL48'CDGBRA -away M3 modes 6, 7 FPCR'
000056B0  00080000 00080000                          850  DC XL16'00080000000800000000000000000000'
000056C0  C3C4C7C2 D9C1404E                          851  DC CL48'CDGBRA +tie FPCRmodes 1-3, 7 FPCR'
000056F0  00000001 00000002                          852  DC XL16'00000001000000020000000300000007'
00005700  C3C4C7C2 D9C1404E                          853  DC CL48'CDGBRA +tie M3 modes 1, 3-5 FPCR'
00005730  00080000 00080000                          854  DC XL16'00080000000800000008000000080000'
00005740  C3C4C7C2 D9C1404E                          855  DC CL48'CDGBRA +tie M3 modes 6, 7 FPCR'
00005770  00080000 00080000                          856  DC XL16'00080000000800000000000000000000'
00005780  C3C4C7C2 D9C14060                          857  DC CL48'CDGBRA -tie FPCRmodes 1-3, 7 FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000057B0  00000001 00000002                          858  DC XL16'00000001000000020000000300000007'
000057C0  C3C4C7C2 D9C14060                          859  DC CL48'CDGBRA -tie M3 modes 1, 3-5 FPCR'
000057F0  00080000 00080000                          860  DC XL16'00080000000800000008000000080000'
00005800  C3C4C7C2 D9C14060                          861  DC CL48'CDGBRA -tie M3 modes 6, 7 FPCR'
00005830  00080000 00080000                          862  DC XL16'00080000000800000000000000000000'
00005840  C3C4C7C2 D9C1404E                          863  DC CL48'CDGBRA +tozero FPCRmodes 1-3, 7 FPCR'
00005870  00000001 00000002                          864  DC XL16'00000001000000020000000300000007'
00005880  C3C4C7C2 D9C1404E                          865  DC CL48'CDGBRA +tozero M3 modes 1, 3-5 FPCR'
000058B0  00080000 00080000                          866  DC XL16'00080000000800000008000000080000'
000058C0  C3C4C7C2 D9C1404E                          867  DC CL48'CDGBRA +tozero M3 modes 6, 7 FPCR'
000058F0  00080000 00080000                          868  DC XL16'00080000000800000000000000000000'
00005900  C3C4C7C2 D9C14060                          869  DC CL48'CDGBRA -tozero FPCRmodes 1-3, 7 FPCR'
00005930  00000001 00000002                          870  DC XL16'00000001000000020000000300000007'
00005940  C3C4C7C2 D9C14060                          871  DC CL48'CDGBRA -tozero M3 modes 1, 3-5 FPCR'
00005970  00080000 00080000                          872  DC XL16'00080000000800000008000000080000'
00005980  C3C4C7C2 D9C14060                          873  DC CL48'CDGBRA -tozero M3 modes 6, 7 FPCR'
000059B0  00080000 00080000                          874  DC XL16'00080000000800000000000000000000'
                            00000012  00000001    875 LBFPRMOF_NUM EQU (*-LBFPRMOF_GOOD)/64
                                                  876 *
                                                  877 *
                            000059C0  00000001    878 XBFPOUT_GOOD EQU *
000059C0  C3E7C7C2 D9409985                          879  DC CL48'CXGBR result 1a'
000059F0  3FFF0000 00000000                          880  DC XL16'3FFF0000000000000000000000000000'
00005A00  C3E7C7C2 D9409985                          881  DC CL48'CXGBR result 1b'
00005A30  3FFF0000 00000000                          882  DC XL16'3FFF0000000000000000000000000000'
00005A40  C3E7C7C2 D9409985                          883  DC CL48'CXGBR result 2a'
00005A70  40000000 00000000                          884  DC XL16'40000000000000000000000000000000'
00005A80  C3E7C7C2 D9409985                          885  DC CL48'CXGBR result 2b'
00005AB0  40000000 00000000                          886  DC XL16'40000000000000000000000000000000'
00005AC0  C3E7C7C2 D9409985                          887  DC CL48'CXGBR result 3a'
00005AF0  40010000 00000000                          888  DC XL16'40010000000000000000000000000000'
00005B00  C3E7C7C2 D9409985                          889  DC CL48'CXGBR result 3b'
00005B30  40010000 00000000                          890  DC XL16'40010000000000000000000000000000'
00005B40  C3E7C7C2 D9409985                          891  DC CL48'CXGBR result 4a'
00005B70  C0000000 00000000                          892  DC XL16'C0000000000000000000000000000000'
00005B80  C3E7C7C2 D9409985                          893  DC CL48'CXGBR result 4b'
00005BB0  C0000000 00000000                          894  DC XL16'C0000000000000000000000000000000'
00005BC0  C3E7C7C2 D9409985                          895  DC CL48'CXGBR result 5a'
00005BF0  403DFFFF FFFFFFFF                          896  DC XL16'403DFFFFFFFFFFFFFFC000000000000'
00005C00  C3E7C7C2 D9409985                          897  DC CL48'CXGBR result 5b'
00005C30  403DFFFF FFFFFFFF                          898  DC XL16'403DFFFFFFFFFFFFFFC000000000000'
00005C40  C3E7C7C2 D9409985                          899  DC CL48'CXGBR result 6a'
00005C70  C03DFFFF FFFFFFFF                          900  DC XL16'C03DFFFFFFFFFFFFFFC000000000000'
00005C80  C3E7C7C2 D9409985                          901  DC CL48'CXGBR result 6b'
00005CB0  C03DFFFF FFFFFFFF                          902  DC XL16'C03DFFFFFFFFFFFFFFC000000000000'
                            0000000C  00000001    903 XBFPOUT_NUM EQU (*-XBFPOUT_GOOD)/64
                                                  904 *
                                                  905 *
                            00005CC0  00000001    906 XBFPFLGS_GOOD EQU *
00005CC0  C3E7C7C2 D940C6D7                          907  DC CL48'CXGBR FPCRpairs 1-2'
00005CF0  00000000 F8000000                          908  DC XL16'00000000F800000000000000F8000000'
00005D00  C3E7C7C2 D940C6D7                          909  DC CL48'CXGBR FPCRpairs 3-4'
00005D30  00000000 F8000000                          910  DC XL16'00000000F800000000000000F8000000'
00005D40  C3E7C7C2 D940C6D7                          911  DC CL48'CXGBR FPCRpairs 5-6'
00005D70  00000000 F8000000                          912  DC XL16'00000000F800000000000000F8000000'
                            00000003  00000001    913 XBFPFLGS_NUM EQU (*-XBFPFLGS_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00005D80                                              915 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                      917 ********************************************************************
                                                      918 *                 REPORT UNEXPECTED PROGRAM CHECK
                                                      919 ********************************************************************

00005D80                                              921 PGMCK    DS    0H
00005D80  F342 C072 F08E      00005DF2  0000008E      922          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
00005D86  926B C076                     00005DF6      923          MVI   PGMCOMMA,C','
00005D8A  DC03 C072 C178      00005DF2  00005EF8      924          TR    PROGCODE,HEXTRTAB

00005D90  F384 C07C F150      00005DFC  00000150      926          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
00005D96  9240 C084                     00005E04      927          MVI   PGMPSW+(0*9)+8,C' '
00005D9A  DC07 C07C C178      00005DFC  00005EF8      928          TR    PGMPSW+(0*9)(8),HEXTRTAB

00005DA0  F384 C085 F154      00005E05  00000154      930          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
00005DA6  9240 C08D                     00005E0D      931          MVI   PGMPSW+(1*9)+8,C' '
00005DAA  DC07 C085 C178      00005E05  00005EF8      932          TR    PGMPSW+(1*9)(8),HEXTRTAB

00005DB0  F384 C08E F158      00005E0E  00000158      934          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
00005DB6  9240 C096                     00005E16      935          MVI   PGMPSW+(2*9)+8,C' '
00005DBA  DC07 C08E C178      00005E0E  00005EF8      936          TR    PGMPSW+(2*9)(8),HEXTRTAB

00005DC0  F384 C097 F15C      00005E17  0000015C      938          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
00005DC6  9240 C09F                     00005E1F      939          MVI   PGMPSW+(3*9)+8,C' '
00005DCA  DC07 C097 C178      00005E17  00005EF8      940          TR    PGMPSW+(3*9)(8),HEXTRTAB

00005DD0  4100 0042                     00000042      942          LA    R0,L'PROGMSG     R0 <== length of message
00005DD4  4110 C05E                     00005DDE      943          LA    R1,PROGMSG       R1 --> the message text itself
00005DD8  4520 C27A                     00005FFA      944          BAL   R2,MSG           Go display this message
                                                      945
00005DDC  07FD                                        946          BR    R13              Return to caller

00005DDE                                              948 PROGMSG  DS   0CL66
00005DDE  D7D9D6C7 D9C1D440                           949          DC    CL20'PROGRAM CHECK! CODE '
00005DF2  88888888                                    950 PROGCODE DC    CL4'hhhh'
00005DF6  6B                                          951 PGMCOMMA DC    CL1','
00005DF7  40D7E2E6 40                                 952          DC    CL5' PSW '
00005DFC  88888888 88888888                           953 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                      955 ********************************************************************************
                                                      956 *                        VERIFICATION ROUTINE
                                                      957 ********************************************************************************

00005E20                                              959 VERISUB  DS    0H
                                                      960 *
                                                      961 **       Loop through the VERIFY TABLE...
                                                      962 *

00005E20  4110 C32C                   000060AC        964          LA    R1,VERIFTAB      R1 --> Verify table
00005E24  4120 000A                   0000000A        965          LA    R2,VERIFLEN      R2 <== Number of entries
00005E28  0D30                                        966          BASR  R3,0             Set top of loop

00005E2A  9846 1000                   00000000        968          LM    R4,R6,0(R1)      Load verify table values
00005E2E  4D70 C0C2                   00005E42        969          BAS   R7,VERIFY        Verify results
00005E32  4110 100C                   0000000C        970          LA    R1,12(,R1)       Next verify table entry
00005E36  0623                                        971          BCTR  R2,R3            Loop through verify table

00005E38  9500 C278                   00005FF8        973          CLI   FAILFLAG,X'00'   Did all tests verify okay?
00005E3C  078D                                        974          BER   R13              Yes, return to caller
00005E3E  47F0 F238                   00000238        975          B     FAIL             No, load FAILURE disabled wait PSW


                                                      977 *
                                                      978 **       Loop through the ACTUAL / EXPECTED results...
                                                      979 *

00005E42  0D80                                        981 VERIFY   BASR  R8,0             Set top of loop

00005E44  D50F 4000 5030      00000000  00000030    983          CLC   0(16,R4),48(R5)  Actual results == Expected results?
00005E4A  4770 C0DA                     00005E5A    984          BNE   VERIFAIL         No, show failure
00005E4E  4140 4010                     00000010    985 VERINEXT LA    R4,16(,R4)       Next actual result
00005E52  4150 5040                     00000040    986          LA    R5,64(,R5)       Next expected result
00005E56  0668                                      987          BCTR  R6,R8            Loop through results

00005E58  07F7                                      989          BR    R7               Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    991 ************************************************************************
                                                    992 *                      Report the failure...
                                                    993 ************************************************************************

00005E5A  9005 C250                    00005FD0    995 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
00005E5E  92FF C278                    00005FF8    996          MVI   FAILFLAG,X'FF'   Remember verification failure
                                                    997 *
                                                    998 **       First, show them the description...
                                                    999 *
00005E62  D22F C1E0 5000      00005F60  00000000   1000          MVC   FAILDESC,0(R5)   Save results/test description
00005E68  4100 0044                    00000044   1001          LA    R0,L'FAILMSG1    R0 <== length of message
00005E6C  4110 C1CC                    00005F4C   1002          LA    R1,FAILMSG1      R1 --> the message text itself
00005E70  4520 C27A                    00005FFA   1003          BAL   R2,MSG           Go display this message
                                                   1004 *
                                                   1005 **       Save address of actual and expected results
                                                   1006 *
00005E74  5040 C24C                    00005FCC   1007          ST    R4,AACTUAL       Save A(actual results)
00005E78  4150 5030                    00000030   1008          LA    R5,48(,R5)       R5 ==> expected results
00005E7C  5050 C248                    00005FC8   1009          ST    R5,AEXPECT       Save A(expected results)
                                                   1010 *
                                                   1011 **       Format and show them the EXPECTED ("Want") results...
                                                   1012 *
00005E80  D205 C210 C3A8      00005F90  00006128   1013          MVC   WANTGOT,=CL6'Want: '
00005E86  F384 C216 C248      00005F96  00005FC8   1014          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
00005E8C  9240 C21E                    00005F9E   1015          MVI   BLANKEQ,C' '
00005E90  DC07 C216 C178      00005F96  00005EF8   1016          TR    FAILADR,HEXTRTAB

00005E96  F384 C221 5000      00005FA1  00000000   1018          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
00005E9C  9240 C229                    00005FA9   1019          MVI   FAILVALS+(0*9)+8,C' '
00005EA0  DC07 C221 C178      00005FA1  00005EF8   1020          TR    FAILVALS+(0*9)(8),HEXTRTAB

00005EA6  F384 C22A 5004      00005FAA  00000004   1022          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
00005EAC  9240 C232                    00005FB2   1023          MVI   FAILVALS+(1*9)+8,C' '
00005EB0  DC07 C22A C178      00005FAA  00005EF8   1024          TR    FAILVALS+(1*9)(8),HEXTRTAB

00005EB6  F384 C233 5008      00005FB3  00000008   1026          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
00005EBC  9240 C23B                    00005FBB   1027          MVI   FAILVALS+(2*9)+8,C' '
00005EC0  DC07 C233 C178      00005FB3  00005EF8   1028          TR    FAILVALS+(2*9)(8),HEXTRTAB

00005EC6  F384 C23C 500C      00005FBC  0000000C   1030          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
00005ECC  9240 C244                    00005FC4   1031          MVI   FAILVALS+(3*9)+8,C' '
00005ED0  DC07 C23C C178      00005FBC  00005EF8   1032          TR    FAILVALS+(3*9)(8),HEXTRTAB

00005ED6  4100 0035                    00000035   1034          LA    R0,L'FAILMSG2    R0 <== length of message
00005EDA  4110 C210                    00005F90   1035          LA    R1,FAILMSG2      R1 --> the message text itself
00005EDE  4520 C27A                    00005FFA   1036          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                          1038 *
                                          1039 **       Format and show them the ACTUAL ("Got") results...
                                          1040 *
00005EE2  D205 C210 C3AE      00005F90  0000612E  1041          MVC   WANTGOT,=CL6'Got:  '
00005EE8  F384 C216 C24C      00005F96  00005FCC  1042          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
00005EEE  9240 C21E                     00005F9E  1043          MVI   BLANKEQ,C' '
00005EF2  DC07 C216 C178      00005F96  00005EF8  1044          TR    FAILADR,HEXTRTAB

00005EF8  F384 C221 4000      00005FA1  00000000  1046          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
00005EFE  9240 C229                     00005FA9  1047          MVI   FAILVALS+(0*9)+8,C' '
00005F02  DC07 C221 C178      00005FA1  00005EF8  1048          TR    FAILVALS+(0*9)(8),HEXTRTAB

00005F08  F384 C22A 4004      00005FAA  00000004  1050          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
00005F0E  9240 C232                     00005FB2  1051          MVI   FAILVALS+(1*9)+8,C' '
00005F12  DC07 C22A C178      00005FAA  00005EF8  1052          TR    FAILVALS+(1*9)(8),HEXTRTAB

00005F18  F384 C233 4008      00005FB3  00000008  1054          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
00005F1E  9240 C23B                     00005FBB  1055          MVI   FAILVALS+(2*9)+8,C' '
00005F22  DC07 C233 C178      00005FB3  00005EF8  1056          TR    FAILVALS+(2*9)(8),HEXTRTAB

00005F28  F384 C23C 400C      00005FBC  0000000C  1058          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
00005F2E  9240 C244                     00005FC4  1059          MVI   FAILVALS+(3*9)+8,C' '
00005F32  DC07 C23C C178      00005FBC  00005EF8  1060          TR    FAILVALS+(3*9)(8),HEXTRTAB

00005F38  4100 0035                     00000035  1062          LA    R0,L'FAILMSG2    R0 <== length of message
00005F3C  4110 C210                     00005F90  1063          LA    R1,FAILMSG2      R1 --> the message text itself
00005F40  4520 C27A                     00005FFA  1064          BAL   R2,MSG           Go display this message

00005F44  9805 C250                     00005FD0  1066          LM    R0,R5,SAVER0R5   Restore registers
00005F48  47F0 C0CE                     00005E4E  1067          B     VERINEXT         Continue with verification...



00005F4C                                          1069 FAILMSG1 DS    0CL68
00005F4C  C3D6D4D7 C1D9C9E2                       1070          DC    CL20'COMPARISON FAILURE! '
00005F60  4D8485A2 83998997                       1071 FAILDESC DC    CL48'(description)'


00005F90                                          1073 FAILMSG2 DS    0CL53
00005F90  40404040 4040                           1074 WANTGOT  DC    CL6' '           'Want: ' -or- 'Got:  '
00005F96  C1C1C1C1 C1C1C1C1                       1075 FAILADR  DC    CL8'AAAAAAAA'
00005F9E  407E40                                  1076 BLANKEQ  DC    CL3' = '
00005FA1  88888888 88888888                       1077 FAILVALS DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '

00005FC8  00000000                                1079 AEXPECT  DC    F'0'             ==> Expected ("Want") results
00005FCC  00000000                                1080 AACTUAL  DC    F'0'             ==> Actual ("Got") results
00005FD0  00000000 00000000                       1081 SAVER0R5 DC    6F'0'            Registers R0 - R5 save area
00005FE8  F0F1F2F3 F4F5F6F7                       1082 CHARHEX  DC    CL16'0123456789ABCDEF'
                            00005EF8  00000010    1083 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
00005FF8  00                                      1084 FAILFLAG DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          1086 ********************************************************************************
                                          1087 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                          1088 ********************************************************************************

00005FFA  4900 C3A4                 00006124  1090 MSG      CH    R0,=H'0'                Do we even HAVE a message?
00005FFE  07D2                              1091          BNHR  R2                      No, ignore

00006000  9002 C2B0                 00006030  1093          STM   R0,R2,MSGSAVE           Save registers

00006004  4900 C3A6                 00006126  1095          CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00006008  47D0 C290                 00006010  1096          BNH   MSGOK                   Yes, continue
0000600C  4100 005F                 0000005F  1097          LA    R0,L'MSGMSG             No, set to maximum

00006010  1820                              1099 MSGOK    LR    R2,R0                   Copy length to work register
00006012  0620                              1100          BCTR  R2,0                    Minus-1 for execute
00006014  4420 C2BC                 0000603C  1101          EX    R2,MSGMVC               Copy message to O/P buffer

00006018  4120 200A                 0000000A  1103          LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
0000601C  4110 C2C2                 00006042  1104          LA    R1,MSGCMD               Point to true command

00006020  83120008                          1106          DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
00006024  4780 C2AA                 0000602A  1107          BZ    MSGRET                  Return if successful
00006028  0000                              1108          DC    H'0'                    CRASH for debugging purposes

0000602A  9802 C2B0                 00006030  1110 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
0000602E  07F2                              1111          BR    R2                      Return to caller
```

```
00006030  00000000 00000000                 1113 MSGSAVE  DC    3F'0'                   Registers save area
0000603C  D200 C2CB 1000      0000604B  00000000  1114 MSGMVC   MVC   MSGMSG(0),0(R1)      Executed instruction

00006042  D4E2C7D5 D6C8405C                 1116 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
0000604B  40404040 40404040                 1117 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   1119 ********************************************************************************
                                                   1120 *                              VERIFY TABLE
                                                   1121 ********************************************************************************
                                                   1122 *
                                                   1123 *        A(actual results), A(expected results), A(#of results)
                                                   1124 *
                                                   1125 ********************************************************************************

000060AC                                           1127 VERIFTAB DC    0F'0'
000060AC  00001000                                 1128          DC    A(SBFPOUT)
000060B0  00004000                                 1129          DC    A(SBFPOUT_GOOD)
000060B4  00000005                                 1130          DC    A(SBFPOUT_NUM)
                                                   1131 *
000060B8  00001100                                 1132          DC    A(SBFPFLGS)
000060BC  00004140                                 1133          DC    A(SBFPFLGS_GOOD)
000060C0  00000005                                 1134          DC    A(SBFPFLGS_NUM)
                                                   1135 *
000060C4  00001200                                 1136          DC    A(SBFPRMO)
000060C8  00004280                                 1137          DC    A(SBFPRMO_GOOD)
000060CC  00000012                                 1138          DC    A(SBFPRMO_NUM)
                                                   1139 *
000060D0  00001500                                 1140          DC    A(SBFPRMOF)
000060D4  00004700                                 1141          DC    A(SBFPRMOF_GOOD)
000060D8  00000012                                 1142          DC    A(SBFPRMOF_NUM)
                                                   1143 *
000060DC  00002000                                 1144          DC    A(LBFPOUT)
000060E0  00004B80                                 1145          DC    A(LBFPOUT_GOOD)
000060E4  00000006                                 1146          DC    A(LBFPOUT_NUM)
                                                   1147 *
000060E8  00002100                                 1148          DC    A(LBFPFLGS)
000060EC  00004D00                                 1149          DC    A(LBFPFLGS_GOOD)
000060F0  00000003                                 1150          DC    A(LBFPFLGS_NUM)
                                                   1151 *
000060F4  00002200                                 1152          DC    A(LBFPRMO)
000060F8  00004DC0                                 1153          DC    A(LBFPRMO_GOOD)
000060FC  0000001E                                 1154          DC    A(LBFPRMO_NUM)
                                                   1155 *
00006100  00002700                                 1156          DC    A(LBFPRMOF)
00006104  00005540                                 1157          DC    A(LBFPRMOF_GOOD)
00006108  00000012                                 1158          DC    A(LBFPRMOF_NUM)
                                                   1159 *
0000610C  00003000                                 1160          DC    A(XBFPOUT)
00006110  000059C0                                 1161          DC    A(XBFPOUT_GOOD)
00006114  0000000C                                 1162          DC    A(XBFPOUT_NUM)
                                                   1163 *
00006118  00003200                                 1164          DC    A(XBFPFLGS)
0000611C  00005CC0                                 1165          DC    A(XBFPFLGS_GOOD)
00006120  00000003                                 1166          DC    A(XBFPFLGS_NUM)
                                                   1167 *
                           0000000A  00000001      1168 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00006124                                            1170          END
00006124  0000                                      1171                =H'0'
00006126  005F                                      1172                =AL2(L'MSGMSG)
00006128  E68195A3 7A40                             1173                =CL6'Want: '
0000612E  C796A37A 4040                             1174                =CL6'Got:  '
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AACTUAL	F	005FCC	4	1080	1007 1042
AEXPECT	F	005FC8	4	1079	1009 1014
AHELPERS	A	00027C	4	193	183 220
BFPCVTFF	J	000000	24884	109	
BLANKEQ	C	005F9E	3	1076	1015 1043
CDGBR	I	00045C	4	400	209
CDGBRA	I	0004A0	4	437	211
CEGBR	I	000344	4	279	204
CEGBRA	I	000388	4	316	206
CHARHEX	C	005FE8	16	1082	1083
CTLR0	F	0002E8	4	230	199 200 201
CXGBR	I	000574	4	521	214
EXTDS	F	000314	4	252	213
FAIL	I	000238	4	191	975
FAILADR	C	005F96	8	1075	1014 1016 1042 1044
FAILDESC	C	005F60	48	1071	1000
FAILFLAG	X	005FF8	1	1084	973 996
FAILMSG1	C	005F4C	68	1069	1001 1002
FAILMSG2	C	005F90	53	1073	1034 1035 1062 1063
FAILPSW	X	0002D8	8	228	191
FAILVALS	C	005FA1	36	1077	1018 1019 1020 1022 1023 1024 1026 1027 1028 1030 1031 1032 1046 1047 1048 1050 1051 1052 1054 1055 1056 1058 1059 1060
FPCREGNT	X	0002EC	4	231	286 327 333 339 345 354 359 364 369 374 379 407 448 454 460 466 475 480 485 490 495 500 528
FPCREGTR	X	0002F0	4	232	291 412 534
FPR0	U	000000	1	130	
FPR1	U	000001	1	131	
FPR10	U	00000A	1	140	531 537
FPR11	U	00000B	1	141	
FPR12	U	00000C	1	142	
FPR13	U	00000D	1	143	
FPR14	U	00000E	1	144	
FPR15	U	00000F	1	145	
FPR2	U	000002	1	132	
FPR3	U	000003	1	133	
FPR4	U	000004	1	134	
FPR5	U	000005	1	135	
FPR6	U	000006	1	136	
FPR7	U	000007	1	137	
FPR8	U	000008	1	138	287 288 292 293 329 330 335 336 341 342 347 348 355 356 360 361 365 366 370 371 375 376 380 381 408 409 413 414 450 451 456 457 462 463 468 469 476 477 481 482 486 487 491 492 496 497 501 502 529 530 535 536
FPR9	U	000009	1	139	
GOODPSW	X	0002C8	8	227	224
HELPERS	H	005D80	2	915	148 193
HEXTRTAB	U	005EF8	16	1083	924 928 932 936 940 1016 1020 1024 1028 1032 1044 1048 1052 1056 1060
IMAGE	1	000000	24884	0	
INTCOUNT	U	00000A	1	571	241 247 253
INTIN	D	0005C0	8	554	571 242 248 254
LBFPFLGS	U	002100	1	620	250 1148
LBFPFLGS_GOOD	U	004D00	1	764	771 1149
LBFPFLGS_NUM	U	000003	1	771	1150
LBFPOUT	U	002000	1	618	249 1144
LBFPOUT_GOOD	U	004B80	1	748	761 1145

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

LBFPOUT_NUM          U    000006       1   761  1146
LBFPRMO              U    002200       1   622   267  1152
LBFPRMOF             U    002700       1   624   268  1156
LBFPRMOF_GOOD        U    005540       1   838   875  1157
LBFPRMOF_NUM         U    000012       1   875  1158
LBFPRMO_GOOD         U    004DC0       1   774   835  1153
LBFPRMO_NUM          U    00001E       1   835  1154
LINTRMCT             U    000006       1   599   265
LINTRMIN             D    000640       8   589   599   266
LONGS                F    000304       4   246   208
MSG                  I    005FFA       4  1090   944  1003  1036  1064
MSGCMD               C    006042       9  1116  1103  1104
MSGMSG               C    00604B      95  1117  1097  1114  1095
MSGMVC               I    00603C       6  1114  1101
MSGOK                I    006010       2  1099  1096
MSGRET               I    00602A       4  1110  1107
MSGSAVE              F    006030       4  1113  1093  1110
PCINTCD              H    00008E       2   161   178   922
PCNOTDTA             I    00020C       4   182   179
PCOLDPSW             U    000150       1   163   180   926   930   934   938
PGMCK                H    005D80       2   921   184
PGMCOMMA             C    005DF6       1   951   923
PGMPSW               C    005DFC      36   953   926   927   928   930   931   932   934   935   936   938   939   940
PROGCHK              H    000200       2   177   169
PROGCODE             C    005DF2       4   950   922   924
PROGMSG              C    005DDE      66   948   942   943
PROGPSW              D    000228       8   190   189
R0                   U    000000       1   111   182   185   199   201   942   995  1001  1034  1062  1066  1090  1093  1095  1097
                                                1099  1110
R1                   U    000001       1   112   285   287   292   322   329   335   341   347   355   360   365   370   375   380
                                                406   408   413   443   450   456   462   468   476   481   486   491   496   501
                                                527   529   535   943   964   968   970  1002  1035  1063  1104  1114
R10                  U    00000A       1   121   203   205   208   210   213   279   282   316   319   400   401   437   438   521
                                                522
R11                  U    00000B       1   122
R12                  U    00000C       1   123   148   183   220   283   298   320   387   404   419   441   508   525   542
R13                  U    00000D       1   124   184   204   206   209   211   214   221   281   299   318   388   403   420   440
                                                509   524   543   946   974
R14                  U    00000E       1   125   187   188   222   223
R15                  U    00000F       1   126   147   182   185
R2                   U    000002       1   113   279   280   298   316   317   387   400   402   419   437   439   508   521   523
                                                542   944   965   971  1003  1036  1064  1091  1093  1099  1100  1101  1103  1110
                                               1111
R3                   U    000003       1   114   279   285   295   316   322   384   400   406   416   437   443   505   521   527
                                                539   966   971
R4                   U    000004       1   115   968   983   985  1007  1046  1050  1054  1058
R5                   U    000005       1   116   983   986   995  1000  1008  1009  1018  1022  1026  1030  1066
R6                   U    000006       1   117   968   987
R7                   U    000007       1   118   282   288   293   296   319   330   336   342   348   356   361   366   371   376
                                                381   385   401   409   414   417   438   451   457   463   469   477   482   487
                                                492   497   502   506   522   530   531   536   537   540   969   989
R8                   U    000008       1   119   282   289   294   297   319   331   337   343   349   357   362   367   372   377
                                                382   386   401   410   415   418   438   452   458   464   470   478   483   488
                                                493   498   503   507   522   532   538   541   981   987
R9                   U    000009       1   120
RMLONGS              F    000334       4   264   210
```

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

RMSHORTS             F    000324       4   258   205
SAVER0R5             F    005FD0       4  1081   995  1066
SAVEREGS             F    00023C       4   192   182   185
SBFPFLGS             U    001100       1   610   244  1132
SBFPFLGS_GOOD        U    004140       1   654   665  1133
SBFPFLGS_NUM         U    000005       1   665  1134
SBFPOUT              U    001000       1   608   243  1128
SBFPOUT_GOOD         U    004000       1   640   651  1129
SBFPOUT_NUM          U    000005       1   651  1130
SBFPRMO              U    001200       1   612   261  1136
SBFPRMOF             U    001500       1   614   262  1140
SBFPRMOF_GOOD        U    004700       1   708   745  1141
SBFPRMOF_NUM         U    000012       1   745  1142
SBFPRMO_GOOD         U    004280       1   668   705  1137
SBFPRMO_NUM          U    000012       1   705  1138
SHORTS               F    0002F4       4   240   203
SINTRMCT             U    000006       1   585   259
SINTRMIN             D    000610       8   575   585   260
START                I    000280       4   199   166
STRTLABL             U    000000       1   110   160   163   165   168   176   608   610   612   614   618   620   622   624   628
                                                630   638
VERIFAIL             I    005E5A       4   995   984
VERIFLEN             U    00000A       1  1168   965
VERIFTAB             F    0060AC       4  1127  1168   964
VERIFY               I    005E42       2   981   969
VERINEXT             I    005E4E       4   985  1067
VERISUB              H    005E20       2   959   221
WANTGOT              C    005F90       6  1074  1013  1041
XBFPFLGS             U    003200       1   630   256  1164
XBFPFLGS_GOOD        U    005CC0       1   906   913  1165
XBFPFLGS_NUM         U    000003       1   913  1166
XBFPOUT              U    003000       1   628   255  1160
XBFPOUT_GOOD         U    0059C0       1   878   903  1161
XBFPOUT_NUM          U    00000C       1   903  1162
=AL2(L'MSGMSG)       R    006126       2  1172  1095
=CL6'Got:  '         C    00612E       6  1174  1041
=CL6'Want: '         C    006128       6  1173  1013
=H'0'                H    006124       2  1171  1090
```

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 0

Image       IMAGE      24884  0000-6133  0000-6133
  Region                24884  0000-6133  0000-6133
    CSECT   BFPCVTFF   24884  0000-6133  0000-6133
```

```
   STMT                                                    FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-011-cvtfrfix64\bfp-011-cvtfrfix64.asm

** NO ERRORS FOUND **
```