```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    2 ************************************************************************
                                                    3 *
                                                    4 *Testcase IEEE CONVERT TO LOGICAL 64
                                                    5 *  Test case capability includes ieee exceptions trappable and
                                                    6 *  otherwise.  Test results, FPCR flags, DXC, and condition codes are
                                                    7 *  saved for all tests.
                                                    8 *
                                                    9 *
                                                   10 *                      ********************
                                                   11 *                      **   IMPORTANT!   **
                                                   12 *                      ********************
                                                   13 *
                                                   14 *       This test uses the Hercules Diagnose X'008' interface
                                                   15 *       to display messages and thus your .tst runtest script
                                                   16 *       MUST contain a "DIAG8CMD ENABLE" statement within it!
                                                   17 *
                                                   18 *
                                                   19 ************************************************************************


                                                   21 ************************************************************************
                                                   22 *
                                                   23 *                     bfp-005-cvttolog64.asm
                                                   24 *
                                                   25 *       This assembly-language source file is part of the
                                                   26 *       Hercules Binary Floating Point Validation Package
                                                   27 *                          by Stephen R. Orso
                                                   28 *
                                                   29 * Copyright 2016 by Stephen R Orso.
                                                   30 * Runtest *Compare dependency removed by Fish on 2022-08-16
                                                   31 * PADCSECT macro/usage removed by Fish on 2022-08-16
                                                   32 *
                                                   33 * Redistribution and use in source and binary forms, with or without
                                                   34 * modification, are permitted provided that the following conditions
                                                   35 * are met:
                                                   36 *
                                                   37 * 1. Redistributions of source code must retain the above copyright
                                                   38 *    notice, this list of conditions and the following disclaimer.
                                                   39 *
                                                   40 * 2. Redistributions in binary form must reproduce the above copyright
                                                   41 *    notice, this list of conditions and the following disclaimer in
                                                   42 *    the documentation and/or other materials provided  with the
                                                   43 *    distribution.
                                                   44 *
                                                   45 * 3. The name of the author may not be used to endorse or promote
                                                   46 *    products derived from this software without specific prior written
                                                   47 *    permission.
                                                   48 *
                                                   49 * DISCLAMER: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER "AS IS"
                                                   50 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
                                                   51 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
                                                   52 * PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT
                                                   53 * HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
                                                   54 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
                                                   55 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
                                                   56 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT

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



                                                    63 ********************************************************************************
                                                    64 *
                                                    65 * Tests the following three conversion instructions
                                                    66 *   CONVERT TO LOGICAL (short BFP to uint-64, RRF-e)
                                                    67 *   CONVERT TO LOGICAL (long BFP to uint-64, RRF-e)
                                                    68 *   CONVERT TO LOGICAL (extended BFP to uint-64, RRF-e)
                                                    69 *
                                                    70 * Test data is compiled into this program.  The test script that runs
                                                    71 * this program can provide alternative test data through Hercules R
                                                    72 * commands.
                                                    73 *
                                                    74 * Test Case Order
                                                    75 * 1) Short BFP to uint-64
                                                    76 * 2) Short BFP to uint-64 with all rounding modes
                                                    77 * 3) Long BFP uint-64
                                                    78 * 3) Long BFP uint-64 with all rounding modes
                                                    79 * 4) Extended BFP to uint-64
                                                    80 * 4) Extended BFP to uint-64 with all rounding modes
                                                    81 *
                                                    82 * Provided test data is:
                                                    83 *      1, 2, 4, -2, QNaN, SNaN, max uint-64 + 1
                                                    84 *   The last value will trigger inexact exceptions when converted
                                                    85 *   to uint-64.
                                                    86 * The same values are provided in each of the three input formats
                                                    87 *   except for the last input.  This is rounded up to the nearest
                                                    88 *   value that can be represented in the input format.  Extended
                                                    89 *   BFP is the only format with an exact representation.
                                                    90 *     Extended BFP: 403F0000000000000000000000000000 =>
                                                    91 *                                   18 446 744 073 709 551 616 (exact)
                                                    92 *     Long BFP      43F0000000000001 =>
                                                    93 *                                   18 446 744 073 709 555 712
                                                    94 *     Short BFP:    5F800001 => 18 446 746 272 732 807 168
                                                    95 * Provided test data for rounding tests:
                                                    96 *   -1.5, -0.5, +0.5, +1.5, +2.5, +5.5, +9.5, max uint-64
                                                    97 *   This data is taken from Table 9-11 on page 9-16 of SA22-7832-10.
                                                    98 *   While the table illustrates LOAD FP INTEGER, the same results
                                                    99 *   should be generated when creating a uint-32 or uint-64 integer
                                                   100 *   from a floating point value.  The last value, max uint-64,
                                                   101 *   is rounded down (truncated) to the input format.  Extended is
                                                   102 *   the only format with an exact representation.
                                                   103 *     Extended BFP: 403EFFFFFFFFFFFFFFF000000000000 =>
                                                   104 *                                18 446 744 073 709 551 615.5 (exact)
                                                   105 *     Long BFP      43EFFFFFFFFFFFFF =>
                                                   106 *                                18 446 744 073 709 549 568
                                                   107 *     Short BFP:    5F7FFFFF => 18 446 742 974 197 923 840
                                                   108 *   These values are used so that rounding mode determines whether
                                                   109 *   the result fits in a uint-64.
                                                   110 *
                                                   111 * Also tests the following floating point support instructions
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                     112 *   LOAD  (Short)
                                                     113 *   LOAD  (Long)
                                                     114 *   LOAD FPC
                                                     115 *   SET BFP ROUNDING MODE 2-bit
                                                     116 *   SET BFP ROUNDING MODE 3-bit
                                                     117 *   STORE (Short)
                                                     118 *   STORE (Long)
                                                     119 *   STORE FPC
                                                     120 *
                                                     121 ********************************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    123 *
                                                    124 *  Note: for compatibility with the z/CMS test rig, do not change
                                                    125 *  or use R11, R14, or R15.  Everything else is fair game.
                                                    126 *
                               00000000  0000968B   127 BFPCVTTL START 0
                               00000000  00000001   128 R0       EQU   0                   Work register for cc extraction
                               00000001  00000001   129 R1       EQU   1
                               00000002  00000001   130 R2       EQU   2                   Holds count of test input values
                               00000003  00000001   131 R3       EQU   3                   Points to next test input value(s)
                               00000004  00000001   132 R4       EQU   4                   Rounding tests inner loop control
                               00000005  00000001   133 R5       EQU   5                   Rounding tests outer loop control
                               00000006  00000001   134 R6       EQU   6                   Rounding tests top of inner loop
                               00000007  00000001   135 R7       EQU   7                   Pointer to next result value(s)
                               00000008  00000001   136 R8       EQU   8                   Pointer to next FPCR result
                               00000009  00000001   137 R9       EQU   9                   Rounding tests top of outer loop
                               0000000A  00000001   138 R10      EQU   10                  Pointer to test address list
                               0000000B  00000001   139 R11      EQU   11                  **Reserved for z/CMS test rig
                               0000000C  00000001   140 R12      EQU   12                  Holds number of test cases in set
                               0000000D  00000001   141 R13      EQU   13                  Mainline return address
                               0000000E  00000001   142 R14      EQU   14                  **Return address for z/CMS test rig
                               0000000F  00000001   143 R15      EQU   15                  **Base register on z/CMS or Hyperion
                                                    144 *
                                                    145 * Floating Point Register equates to keep the cross reference clean
                                                    146 *
                               00000000  00000001   147 FPR0     EQU   0
                               00000001  00000001   148 FPR1     EQU   1
                               00000002  00000001   149 FPR2     EQU   2
                               00000003  00000001   150 FPR3     EQU   3
                               00000004  00000001   151 FPR4     EQU   4
                               00000005  00000001   152 FPR5     EQU   5
                               00000006  00000001   153 FPR6     EQU   6
                               00000007  00000001   154 FPR7     EQU   7
                               00000008  00000001   155 FPR8     EQU   8
                               00000009  00000001   156 FPR9     EQU   9
                               0000000A  00000001   157 FPR10    EQU   10
                               0000000B  00000001   158 FPR11    EQU   11
                               0000000C  00000001   159 FPR12    EQU   12
                               0000000D  00000001   160 FPR13    EQU   13
                               0000000E  00000001   161 FPR14    EQU   14
                               0000000F  00000001   162 FPR15    EQU   15
                                                    163 *
00000000                       00000000             164          USING *,R15
00000000                       000092C0             165          USING HELPERS,R12
                                                    166 *
                                                    167 * Above works on real iron (R15=0 after sysclear)
                                                    168 * and in z/CMS (R15 points to start of load module)
                                                    169 *


                                                    171 ***************************************************************
                                                    172 *
                                                    173 * Low core definitions, Restart PSW, and Program Check Routine.
                                                    174 *
                                                    175 ***************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000000                     00000000  0000008E   177          ORG   BFPCVTTL+X'8E'      Program check interrution code
0000008E  0000                                   178 PCINTCD  DS    H
                                                 179 *
                             00000150  00000000  180 PCOLDPSW EQU   BFPCVTTL+X'150'     z/Arch Program check old PSW
                                                 181 *
00000090                     00000090  000001A0  182          ORG   BFPCVTTL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                      183          DC    X'0000000180000000',AD(START)
                                                 184 *
000001B0                     000001B0  000001D0  185          ORG   BFPCVTTL+X'1D0'     z/Arch Program check NEW PSW
000001D0  00000000 00000000                      186          DC    X'0000000000000000',AD(PROGCHK)
                                                 187 *
                                                 188 * Program check routine.  If Data Exception, continue execution at
                                                 189 * the instruction following the program check.  Otherwise, hard wait.
                                                 190 * No need to collect data.  All interesting DXC stuff is captured
                                                 191 * in the FPCR.
                                                 192 *
000001E0                     000001E0  00000200  193          ORG   BFPCVTTL+X'200'
00000200                                         194 PROGCHK  DS    0H              Program check occured...
00000200  9507 F08F                    0000008F  195          CLI   PCINTCD+1,X'07'  Data Exception?
00000204  A774 0004                    0000020C  196          JNE   PCNOTDTA       ..no, hardwait (not sure if R15 is ok)
00000208  B2B2 F150                    00000150  197          LPSWE PCOLDPSW       ..yes, resume program execution

0000020C  900F F23C                    0000023C  199 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                    0000027C  200          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                    000092C0  201          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                    0000023C  202          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                   204          LTR   R14,R14        Return address provided?
0000021E  077E                                   205          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                    00000228  206          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                      207 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2E0                    000002E0  208 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                      209 SAVEREGS DC    16F'0'          Registers save area
0000027C  000092C0                               210 AHELPERS DC    A(HELPERS)      Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    212 ********************************************************************************
                                                    213 *
                                                    214 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                    215 *
                                                    216 ********************************************************************************


00000280  B600 F2F0                        000002F0  218 START    STCTL R0,R0,CTLR0    Store CR0 to enable AFP
00000284  9604 F2F1                        000002F1  219          OI    CTLR0+1,X'04'  Turn on AFP bit
00000288  B700 F2F0                        000002F0  220          LCTL  R0,R0,CTLR0    Reload updated CR0
                                                    221 *
                                                    222 * Short BFP Input testing
                                                    223 *
0000028C  41A0 F2FC                        000002FC  224           LA    R10,SHORTS     Point to short BFP test inputs
00000290  4DD0 F35C                        0000035C  225           BAS   R13,CLGEBR     Convert values to uint-64 from short BFP
00000294  41A0 F32C                        0000032C  226           LA    R10,RMSHORTS   Point to inputs for rounding mode tests
00000298  4DD0 F3C0                        000003C0  227           BAS   R13,CLGEBRA    Convert using all rounding mode options
                                                    228 *
                                                    229 * Short BFP Input testing
                                                    230 *
0000029C  41A0 F30C                        0000030C  231           LA    R10,LONGS      Point to long BFP test inputs
000002A0  4DD0 F51E                        0000051E  232           BAS   R13,CLGDBR     Convert values to uint-64 from long BFP
000002A4  41A0 F33C                        0000033C  233           LA    R10,RMLONGS    Point to inputs for rounding mode tests
000002A8  4DD0 F582                        00000582  234           BAS   R13,CLGDBRA    Convert using all rounding mode options
                                                    235 *
                                                    236 * Short BFP Input testing
                                                    237 *
000002AC  41A0 F31C                        0000031C  238           LA    R10,EXTDS      Point to extended BFP test inputs
000002B0  4DD0 F6E0                        000006E0  239           BAS   R13,CLGXBR     Convert values to uint-64 from extended
000002B4  41A0 F34C                        0000034C  240           LA    R10,RMEXTDS    Point to inputs for rounding mode tests
000002B8  4DD0 F748                        00000748  241           BAS   R13,CLGXBRA    Convert using all rounding mode options
                                                    242 *
                                                    243 ********************************************************************************
                                                    244 *                     Verify test results...
                                                    245 ********************************************************************************
                                                    246 *
000002BC  58C0 F27C                        0000027C  247           L     R12,AHELPERS     Get address of helper subroutines
000002C0  4DD0 C0A0                        00009360  248           BAS   R13,VERISUB      Go verify results
000002C4  12EE                                       249           LTR   R14,R14          Was return address provided?
000002C6  077E                                       250           BNZR  R14              Yes, return to z/CMS test rig.
000002C8  B2B2 F2D0                        000002D0  251           LPSWE GOODPSW          Load SUCCESS PSW
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000002D0                                          253          DS    0D             Ensure correct alignment for PSW
000002D0  00020000 00000000                       254 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002E0  00020000 00000000                       255 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                  256 *
000002F0  00000000                                257 CTLR0    DS    F
000002F4  00000000                                258 FPCREGNT DC    X'00000000'  FPCR, trap all IEEE exceptions, zero flags
000002F8  F8000000                                259 FPCREGTR DC    X'F8000000'  FPCR, trap no IEEE exceptions, zero flags
                                                  260 *
                                                  261 * Input values parameter list, four fullwords:
                                                  262 *      1) Count,
                                                  263 *      2) Address of inputs,
                                                  264 *      3) Address to place results, and
                                                  265 *      4) Address to place DXC/Flags/cc values.
                                                  266 *
000002FC                                          267 SHORTS   DS    0F           Inputs for short BFP testing
000002FC  00000009                                268          DC    A(SBFPCT/4)
00000300  000008AC                                269          DC    A(SBFPIN)
00000304  00001000                                270          DC    A(SINTOUT)
00000308  00001200                                271          DC    A(SINTFLGS)
                                                  272 *
0000030C                                          273 LONGS    DS    0F           Inputs for long BFP testing
0000030C  00000009                                274          DC    A(LBFPCT/8)
00000310  000008F8                                275          DC    A(LBFPIN)
00000314  00002000                                276          DC    A(LINTOUT)
00000318  00002200                                277          DC    A(LINTFLGS)
                                                  278 *
0000031C                                          279 EXTDS    DS    0F           Inputs for Extended BFP testing
0000031C  0000000A                                280          DC    A(XBFPCT/16)
00000320  00000988                                281          DC    A(XBFPIN)
00000324  00003000                                282          DC    A(XINTOUT)
00000328  00003200                                283          DC    A(XINTFLGS)
                                                  284 *
0000032C                                          285 RMSHORTS DS    0F           Inputs for short BFP rounding testing
0000032C  0000000A                                286          DC    A(SBFPRMCT/4)
00000330  000008D0                                287          DC    A(SBFPINRM)
00000334  00001300                                288          DC    A(SINTRMO)
00000338  00001800                                289          DC    A(SINTRMOF)
                                                  290 *
0000033C                                          291 RMLONGS  DS    0F           Inputs for long BFP rounding testing
0000033C  00000009                                292          DC    A(LBFPRMCT/8)
00000340  00000940                                293          DC    A(LBFPINRM)
00000344  00002300                                294          DC    A(LINTRMO)
00000348  00002800                                295          DC    A(LINTRMOF)
                                                  296 *
0000034C                                          297 RMEXTDS  DS    0F           Inputs for extd BFP rounding testing
0000034C  0000000A                                298          DC    A(XBFPRMCT/16)
00000350  00000A28                                299          DC    A(XBFPINRM)
00000354  00003300                                300          DC    A(XINTRMO)
00000358  00003800                                301          DC    A(XINTRMOF)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    303 ********************************************************************************
                                                    304 *
                                                    305 * Convert short BFP to uint-64 format.  A pair of results is generated
                                                    306 * for each input: one with all exceptions non-trappable, and the second
                                                    307 * with all exceptions trappable.   The FPCR and condition code is
                                                    308 * stored for each result.  Rounding mode RNTE, round to nearest, ties
                                                    309 * to even is used for each of these tests
                                                    310 *
                                                    311 ********************************************************************************

0000035C  9823 A000                 00000000   313 CLGEBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000360  9878 A008                 00000008   314          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000364  1222                                 315          LTR   R2,R2         Any test cases?
00000366  078D                                 316          BZR   R13           ..No, return to caller
00000368  0DC0                                 317          BASR  R12,0         Set top of loop
                                               318 *
0000036A  7800 3000                 00000000   319          LE    FPR0,0(,R3)   Get short BFP test value
0000036E  B29D F2F4                 000002F4   320          LFPC  FPCREGNT      Set exceptions non-trappable
00000372  B3AC 0010                            321          CLGEBR R1,0,FPR0,0  Cvt float in FPR0 to uint-64 in GPR1
00000376  E310 7000 0024            00000000   322          STG   R1,0(,R7)     Store uint-64 result
0000037C  B29C 8000                 00000000   323          STFPC 0*4(R8)       Store resulting FPC flags and DXC
00000380  B222 0010                            324          IPM   R1            Get condition code and program mask
00000384  8810 001C                 0000001C   325          SRL   R1,28         Isolate CC in low order byte
00000388  4210 8003                 00000003   326          STC   R1,(0*4)+3(,R8)  Save CC as low byte of FPCR
                                               327 *
0000038C  B29D F2F8                 000002F8   328          LFPC  FPCREGTR      Set exceptions trappable
00000390  B982 0011                            329          XGR   R1,R1         Clear any residual result in R1
00000394  0410                                 330          SPM   R1            Clear out any residual nz condition code
00000396  B3AC 0010                            331          CLGEBR R1,0,FPR0,0  Cvt float in FPR0 to uint-64 in GPR1
0000039A  E310 7008 0024            00000008   332          STG   R1,8(,R7)     Store short BFP result
000003A0  B29C 8004                 00000004   333          STFPC 4(R8)         Store resulting FPC flags and DXC
000003A4  B222 0010                            334          IPM   R1            Get condition code and program mask
000003A8  8810 001C                 0000001C   335          SRL   R1,28         Isolate CC in low order byte
000003AC  4210 8007                 00000007   336          STC   R1,(1*4)+3(,R8)  Save CC as low byte of FPCR
                                               337 *
000003B0  4130 3004                 00000004   338          LA    R3,4(,R3)     Point to next input value
000003B4  4170 7010                 00000010   339          LA    R7,2*8(,R7)   Point to next uint-64 converted value pair
000003B8  4180 8008                 00000008   340          LA    R8,2*4(,R8)   Point to next FPCR/CC result pair
000003BC  062C                                 341          BCTR  R2,R12        Convert next input value.
000003BE  07FD                                 342          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    344 ********************************************************************************
                                                    345 *
                                                    346 * Convert short BFP to integers using each possible rounding mode.
                                                    347 * Ten test results are generated for each input.  A 48-byte test result
                                                    348 * section is used to keep results sets aligned on a quad-double word.
                                                    349 *
                                                    350 * The first four tests use rounding modes specified in the FPCR with
                                                    351 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    352 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    353 * the last two To get full coverage of that instruction pair.
                                                    354 *
                                                    355 * The next six results use instruction-specified rounding modes.
                                                    356 *
                                                    357 * The default rounding mode (0 for RNTE) is not tested in this
                                                    358 * section; prior tests used the default rounding mode.  RNTE is tested
                                                    359 * explicitly as a rounding mode in this section.
                                                    360 *
                                                    361 ********************************************************************************


000003C0  9823 A000                 00000000   363 CLGEBRA  LM    R2,R3,0(R10)  Get count and address of test input values
000003C4  9878 A008                 00000008   364          LM    R7,R8,8(R10)  Get address of result area and flag area.
000003C8  1222                                 365          LTR   R2,R2         Any test cases?
000003CA  078D                                 366          BZR   R13           ..No, return to caller
000003CC  0DC0                                 367          BASR  R12,0         Set top of loop
                                               368 *
000003CE  7800 3000                 00000000   369          LE    FPR0,0(,R3)   Get short BFP test value
                                               370 *
                                               371 * Test cases using rounding mode specified in the FPCR
                                               372 *
000003D2  B29D F2F4                 000002F4   373          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003D6  B299 0001                 00000001   374          SRNM  1             SET FPC to RZ, towards zero.
000003DA  B3AC 0410                            375          CLGEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000003DE  E310 7000 0024            00000000   376          STG   R1,0*8(,R7)   Store uint-64 result
000003E4  B29C 8000                 00000000   377          STFPC 0*4(R8)       Store resulting FPC flags and DXC
000003E8  B222 0010                            378          IPM   R1            Get condition code and program mask
000003EC  8810 001C                 0000001C   379          SRL   R1,28         Isolate CC in low order byte
000003F0  4210 8003                 00000003   380          STC   R1,(0*4)+3(,R8)  Save CC as low byte of FPCR
                                               381 *
000003F4  B29D F2F4                 000002F4   382          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003F8  B299 0002                 00000002   383          SRNM  2             SET FPC to RP, to +infinity
000003FC  B3AC 0410                            384          CLGEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000400  E310 7008 0024            00000008   385          STG   R1,1*8(,R7)   Store uint-64 result
00000406  B29C 8004                 00000004   386          STFPC 1*4(R8)       Store resulting FPC flags and DXC
0000040A  B222 0010                            387          IPM   R1            Get condition code and program mask
0000040E  8810 001C                 0000001C   388          SRL   R1,28         Isolate CC in low order byte
00000412  4210 8007                 00000007   389          STC   R1,(1*4)+3(,R8)  Save CC as low byte of FPCR
                                               390 *
00000416  B29D F2F4                 000002F4   391          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000041A  B2B8 0003                 00000003   392          SRNMB 3             SET FPC to RM, to -infinity
0000041E  B3AC 0410                            393          CLGEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000422  E310 7010 0024            00000010   394          STG   R1,2*8(,R7)   Store uint-64 result
00000428  B29C 8008                 00000008   395          STFPC 2*4(R8)       Store resulting FPC flags and DXC
0000042C  B222 0010                            396          IPM   R1            Get condition code and program mask
00000430  8810 001C                 0000001C   397          SRL   R1,28         Isolate CC in low order byte
00000434  4210 800B                 0000000B   398          STC   R1,(2*4)+3(,R8)  Save CC as low byte of FPCR
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    399 *
00000438  B29D F2F4                     000002F4   400          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000043C  B2B8 0007                     00000007   401          SRNMB 7             RFS, Prepare for Shorter Precision
00000440  B3AC 0410                                402          CLGEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000444  E310 7018 0024                00000018   403          STG   R1,3*8(,R7)   Store uint-64 result
0000044A  B29C 800C                     0000000C   404          STFPC 3*4(R8)       Store resulting FPC flags and DXC
0000044E  B222 0010                                405          IPM   R1            Get condition code and program mask
00000452  8810 001C                     0000001C   406          SRL   R1,28         Isolate CC in low order byte
00000456  4210 800F                     0000000F   407          STC   R1,(3*4)+3(,R8)  Save CC as low byte of FPCR
                                                    408 *
                                                    409 * Test cases using rounding mode specified in the instruction M3 field
                                                    410 *
0000045A  B29D F2F4                     000002F4   411          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000045E  B3AC 1010                                412          CLGEBR R1,1,FPR0,B'0000'  RNTA, to nearest, ties away
00000462  E310 7020 0024                00000020   413          STG   R1,4*8(,R7)   Store uint-64 result
00000468  B29C 8010                     00000010   414          STFPC 4*4(R8)       Store resulting FPC flags and DXC
0000046C  B222 0010                                415          IPM   R1            Get condition code and program mask
00000470  8810 001C                     0000001C   416          SRL   R1,28         Isolate CC in low order byte
00000474  4210 8013                     00000013   417          STC   R1,(4*4)+3(,R8)  Save CC as low byte of FPCR
                                                    418 *
00000478  B29D F2F4                     000002F4   419          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000047C  B3AC 3010                                420          CLGEBR R1,3,FPR0,B'0000'  RFS, prepare for shorter precision
00000480  E310 7028 0024                00000028   421          STG   R1,5*8(,R7)   Store uint-64 result
00000486  B29C 8014                     00000014   422          STFPC 5*4(R8)       Store resulting FPC flags and DXC
0000048A  B222 0010                                423          IPM   R1            Get condition code and program mask
0000048E  8810 001C                     0000001C   424          SRL   R1,28         Isolate CC in low order byte
00000492  4210 8017                     00000017   425          STC   R1,(5*4)+3(,R8)  Save CC as low byte of FPCR
                                                    426 *
00000496  B29D F2F4                     000002F4   427          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000049A  B3AC 4010                                428          CLGEBR R1,4,FPR0,B'0000'  RNTE, to nearest, ties to even
0000049E  E310 7030 0024                00000030   429          STG   R1,6*8(,R7)   Store uint-64 result
000004A4  B29C 8018                     00000018   430          STFPC 6*4(R8)       Store resulting FPC flags and DXC
000004A8  B222 0010                                431          IPM   R1            Get condition code and program mask
000004AC  8810 001C                     0000001C   432          SRL   R1,28         Isolate CC in low order byte
000004B0  4210 801B                     0000001B   433          STC   R1,(6*4)+3(,R8)  Save CC as low byte of FPCR
                                                    434 *
000004B4  B29D F2F4                     000002F4   435          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004B8  B3AC 5010                                436          CLGEBR R1,5,FPR0,B'0000'  RZ, toward zero
000004BC  E310 7038 0024                00000038   437          STG   R1,7*8(,R7)   Store uint-64 result
000004C2  B29C 801C                     0000001C   438          STFPC 7*4(R8)       Store resulting FPC flags and DXC
000004C6  B222 0010                                439          IPM   R1            Get condition code and program mask
000004CA  8810 001C                     0000001C   440          SRL   R1,28         Isolate CC in low order byte
000004CE  4210 801F                     0000001F   441          STC   R1,(7*4)+3(,R8)  Save CC as low byte of FPCR
                                                    442 *
000004D2  B29D F2F4                     000002F4   443          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004D6  B3AC 6010                                444          CLGEBR R1,6,FPR0,B'0000'  RP, to +inf
000004DA  E310 7040 0024                00000040   445          STG   R1,8*8(,R7)   Store uint-64 result
000004E0  B29C 8020                     00000020   446          STFPC 8*4(R8)       Store resulting FPC flags and DXC
000004E4  B222 0010                                447          IPM   R1            Get condition code and program mask
000004E8  8810 001C                     0000001C   448          SRL   R1,28         Isolate CC in low order byte
000004EC  4210 8023                     00000023   449          STC   R1,(8*4)+3(,R8)  Save CC as low byte of FPCR
                                                    450 *
000004F0  B29D F2F4                     000002F4   451          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004F4  B3AC 7010                                452          CLGEBR R1,7,FPR0,B'0000'  RM, to -inf
000004F8  E310 7048 0024                00000048   453          STG   R1,9*8(,R7)   Store uint-64 result
000004FE  B29C 8024                     00000024   454          STFPC 9*4(R8)       Store resulting FPC flags and DXC
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000502  B222 0010                          455          IPM   R1            Get condition code and program mask
00000506  8810 001C                0000001C  456          SRL   R1,28          Isolate CC in low order byte
0000050A  4210 8027                00000027  457          STC   R1,(9*4)+3(,R8)  Save CC as low byte of FPCR
                                             458 *
0000050E  4130 3004                00000004  459          LA    R3,4(,R3)      Point to next input value
00000512  4170 7050                00000050  460          LA    R7,10*8(,R7)   Point to next uint-64 result set
00000516  4180 8030                00000030  461          LA    R8,12*4(,R8)   Point to next FPCR/CC result set
0000051A  062C                             462          BCTR  R2,R12         Convert next input value.
0000051C  07FD                             463          BR    R13            All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    465 ********************************************************************************
                                                    466 *
                                                    467 * Convert long BFP inputs to uint-64.  A pair of results is generated
                                                    468 * for each input: one with all exceptions non-trappable, and the second
                                                    469 * with all exceptions trappable.   The FPCR and condition code is
                                                    470 * stored for each result.
                                                    471 *
                                                    472 ********************************************************************************


0000051E  9823 A000                 00000000   474 CLGDBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000522  9878 A008                 00000008   475          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000526  1222                                 476          LTR   R2,R2         Any test cases?
00000528  078D                                 477          BZR   R13           ..No, return to caller
0000052A  0DC0                                 478          BASR  R12,0         Set top of loop
                                               479 *
0000052C  6800 3000                 00000000   480          LD    FPR0,0(,R3)   Get long BFP test value
00000530  B29D F2F4                 000002F4   481          LFPC  FPCREGNT      Set exceptions non-trappable
00000534  B3AD 0010                            482          CLGDBR R1,0,FPR0,0  Cvt float in FPR0 to uint-64 in GPR1
00000538  E310 7000 0024            00000000   483          STG   R1,0(,R7)     Store long BFP result
0000053E  B29C 8000                 00000000   484          STFPC 0*4(R8)       Store resulting FPC flags and DXC
00000542  B222 0010                            485          IPM   R1            Get condition code and program mask
00000546  8810 001C                 0000001C   486          SRL   R1,28         Isolate CC in low order byte
0000054A  4210 8003                 00000003   487          STC   R1,(0*4)+3(,R8)  Save CC as low byte of FPCR
                                               488 *
0000054E  B29D F2F8                 000002F8   489          LFPC  FPCREGTR      Set exceptions trappable
00000552  B982 0011                            490          XGR   R1,R1         Clear any residual result in R1
00000556  0410                                 491          SPM   R1            Clear out any residual nz condition code
00000558  B3AD 0010                            492          CLGDBR R1,0,FPR0,0  Cvt float in FPR0 to uint-64 in GPR1
0000055C  E310 7008 0024            00000008   493          STG   R1,8(,R7)     Store uint-64 result
00000562  B29C 8004                 00000004   494          STFPC 1*4(R8)       Store resulting FPC flags and DXC
00000566  B222 0010                            495          IPM   R1            Get condition code and program mask
0000056A  8810 001C                 0000001C   496          SRL   R1,28         Isolate CC in low order byte
0000056E  4210 8007                 00000007   497          STC   R1,(1*4)+3(,R8)  Save CC as low byte of FPCR
                                               498 *
00000572  4130 3008                 00000008   499          LA    R3,8(,R3)     Point to next input value
00000576  4170 7010                 00000010   500          LA    R7,16(,R7)    Point to next uint-64 result pair
0000057A  4180 8008                 00000008   501          LA    R8,8(,R8)     Point to next FPCR/CC result pair
0000057E  062C                                 502          BCTR  R2,R12        Convert next input value.
00000580  07FD                                 503          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    505 ************************************************************************
                                                    506 *
                                                    507 * Convert long BFP to integers using each possible rounding mode.
                                                    508 * Ten test results are generated for each input.  A 48-byte test result
                                                    509 * section is used to keep results sets aligned on a quad-double word.
                                                    510 *
                                                    511 * The first four tests use rounding modes specified in the FPCR with
                                                    512 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    513 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    514 * the last two To get full coverage of that instruction pair.
                                                    515 *
                                                    516 * The next six results use instruction-specified rounding modes.
                                                    517 *
                                                    518 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    519 * prior tests used the default rounding mode.  RNTE is tested
                                                    520 * explicitly as a rounding mode in this section.
                                                    521 *
                                                    522 ************************************************************************


00000582  9823 A000                 00000000    524 CLGDBRA  LM    R2,R3,0(R10)  Get count and address of test input values
00000586  9878 A008                 00000008    525          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000058A  1222                                  526          LTR   R2,R2         Any test cases?
0000058C  078D                                  527          BZR   R13           ..No, return to caller
0000058E  0DC0                                  528          BASR  R12,0         Set top of loop
                                                    529 *
00000590  6800 3000                 00000000    530          LD    FPR0,0(,R3)    Get long BFP test value
                                                    531 *
                                                    532 * Test cases using rounding mode specified in the FPCR
                                                    533 *
00000594  B29D F2F4                 000002F4    534          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000598  B299 0001                 00000001    535          SRNM  1             SET FPC to RZ, towards zero.
0000059C  B3AD 0410                             536          CLGDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000005A0  E310 7000 0024            00000000    537          STG   R1,0*8(,R7)   Store uint-64 result
000005A6  B29C 8000                 00000000    538          STFPC 0(R8)         Store resulting FPC flags and DXC
000005AA  B222 0010                             539          IPM   R1            Get condition code and program mask
000005AE  8810 001C                 0000001C    540          SRL   R1,28         Isolate CC in low order byte
000005B2  4210 8003                 00000003    541          STC   R1,3(,R8)     Save CC as low byte of FPCR
                                                    542 *
000005B6  B29D F2F4                 000002F4    543          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000005BA  B299 0002                 00000002    544          SRNM  2             SET FPC to RP, to +infinity
000005BE  B3AD 0410                             545          CLGDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000005C2  E310 7008 0024            00000008    546          STG   R1,1*8(,R7)   Store uint-64 result
000005C8  B29C 8004                 00000004    547          STFPC 1*4(R8)       Store resulting FPC flags and DXC
000005CC  B222 0010                             548          IPM   R1            Get condition code and program mask
000005D0  8810 001C                 0000001C    549          SRL   R1,28         Isolate CC in low order byte
000005D4  4210 8007                 00000007    550          STC   R1,(1*4)+3(,R8)  Save CC as low byte of FPCR
                                                    551 *
000005D8  B29D F2F4                 000002F4    552          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000005DC  B2B8 0003                 00000003    553          SRNMB 3             SET FPC to RM, to -infinity
000005E0  B3AD 0410                             554          CLGDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000005E4  E310 7010 0024            00000010    555          STG   R1,2*8(,R7)   Store uint-64 result
000005EA  B29C 8008                 00000008    556          STFPC 2*4(R8)       Store resulting FPC flags and DXC
000005EE  B222 0010                             557          IPM   R1            Get condition code and program mask
000005F2  8810 001C                 0000001C    558          SRL   R1,28         Isolate CC in low order byte
000005F6  4210 800B                 0000000B    559          STC   R1,(2*4)+3(,R8)  Save CC as low byte of FPCR
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    560 *
000005FA  B29D F2F4                     000002F4   561          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000005FE  B2B8 0007                     00000007   562          SRNMB 7             RFS, Prepare for Shorter Precision
00000602  B3AD 0410                                563          CLGDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000606  E310 7018 0024                00000018   564          STG   R1,3*8(,R7)   Store uint-64 result
0000060C  B29C 800C                     0000000C   565          STFPC 3*4(R8)       Store resulting FPC flags and DXC
00000610  B222 0010                                566          IPM   R1            Get condition code and program mask
00000614  8810 001C                     0000001C   567          SRL   R1,28         Isolate CC in low order byte
00000618  4210 800F                     0000000F   568          STC   R1,(3*4)+3(,R8)  Save CC as low byte of FPCR
                                                    569 *
                                                    570 * Test cases using rounding mode specified in the instruction M3 field
                                                    571 *
0000061C  B29D F2F4                     000002F4   572          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000620  B3AD 1010                                573          CLGDBR R1,1,FPR0,B'0000'  RNTA, to nearest, ties away from 0
00000624  E310 7020 0024                00000020   574          STG   R1,4*8(,R7)   Store uint-64 result
0000062A  B29C 8010                     00000010   575          STFPC 4*4(R8)       Store resulting FPC flags and DXC
0000062E  B222 0010                                576          IPM   R1            Get condition code and program mask
00000632  8810 001C                     0000001C   577          SRL   R1,28         Isolate CC in low order byte
00000636  4210 8013                     00000013   578          STC   R1,(4*4)+3(,R8)  Save CC as low byte of FPCR
                                                    579 *
0000063A  B29D F2F4                     000002F4   580          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000063E  B3AD 3010                                581          CLGDBR R1,3,FPR0,B'0000'  RFS, prepare for shorter precision
00000642  E310 7028 0024                00000028   582          STG   R1,5*8(,R7)   Store uint-64 result
00000648  B29C 8014                     00000014   583          STFPC 5*4(R8)       Store resulting FPC flags and DXC
0000064C  B222 0010                                584          IPM   R1            Get condition code and program mask
00000650  8810 001C                     0000001C   585          SRL   R1,28         Isolate CC in low order byte
00000654  4210 8017                     00000017   586          STC   R1,(5*4)+3(,R8)  Save CC as low byte of FPCR
                                                    587 *
00000658  B29D F2F4                     000002F4   588          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000065C  B3AD 4010                                589          CLGDBR R1,4,FPR0,B'0000'  RNTE, to nearest, ties to even
00000660  E310 7030 0024                00000030   590          STG   R1,6*8(,R7)   Store uint-64 result
00000666  B29C 8018                     00000018   591          STFPC 6*4(R8)       Store resulting FPC flags and DXC
0000066A  B222 0010                                592          IPM   R1            Get condition code and program mask
0000066E  8810 001C                     0000001C   593          SRL   R1,28         Isolate CC in low order byte
00000672  4210 801B                     0000001B   594          STC   R1,(6*4)+3(,R8)  Save CC as low byte of FPCR
                                                    595 *
00000676  B29D F2F4                     000002F4   596          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000067A  B3AD 5010                                597          CLGDBR R1,5,FPR0,B'0000'  RZ, toward zero
0000067E  E310 7038 0024                00000038   598          STG   R1,7*8(,R7)   Store uint-64 result
00000684  B29C 801C                     0000001C   599          STFPC 7*4(R8)       Store resulting FPC flags and DXC
00000688  B222 0010                                600          IPM   R1            Get condition code and program mask
0000068C  8810 001C                     0000001C   601          SRL   R1,28         Isolate CC in low order byte
00000690  4210 801F                     0000001F   602          STC   R1,(7*4)+3(,R8)  Save CC as low byte of FPCR
                                                    603 *
00000694  B29D F2F4                     000002F4   604          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000698  B3AD 6010                                605          CLGDBR R1,6,FPR0,B'0000'  RP, to +inf
0000069C  E310 7040 0024                00000040   606          STG   R1,8*8(,R7)   Store uint-64 result
000006A2  B29C 8020                     00000020   607          STFPC 8*4(R8)       Store resulting FPC flags and DXC
000006A6  B222 0010                                608          IPM   R1            Get condition code and program mask
000006AA  8810 001C                     0000001C   609          SRL   R1,28         Isolate CC in low order byte
000006AE  4210 8023                     00000023   610          STC   R1,(8*4)+3(,R8)  Save CC as low byte of FPCR
                                                    611 *
000006B2  B29D F2F4                     000002F4   612          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000006B6  B3AD 7010                                613          CLGDBR R1,7,FPR0,B'0000'  RM, to -inf
000006BA  E310 7048 0024                00000048   614          STG   R1,9*8(,R7)   Store uint-64 result
000006C0  B29C 8024                     00000024   615          STFPC 9*4(R8)       Store resulting FPC flags and DXC
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000006C4  B222 0010                          616          IPM   R1            Get condition code and program mask
000006C8  8810 001C                 0000001C 617          SRL   R1,28         Isolate CC in low order byte
000006CC  4210 8027                 00000027 618          STC   R1,(9*4)+3(,R8)  Save CC as low byte of FPCR
                                             619 *
000006D0  4130 3008                 00000008 620          LA    R3,8(,R3)     Point to next input value
000006D4  4170 7050                 00000050 621          LA    R7,10*8(,R7)  Point to next uint-64 result set
000006D8  4180 8030                 00000030 622          LA    R8,12*4(,R8)  Point to next FPCR/CC result set
000006DC  062C                               623          BCTR  R2,R12        Convert next input value.
000006DE  07FD                               624          BR    R13           All converted; return.
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                                     626 ************************************************************************
                                                     627 *
                                                     628 * Convert extended BFP to uint-64.  A pair of results is generated
                                                     629 * for each input: one with all exceptions non-trappable, and the
                                                     630 * second with all exceptions trappable.   The FPCR and condition code
                                                     631 * are stored for each result.
                                                     632 *
                                                     633 ************************************************************************


000006E0  9823 A000                  00000000    635 CLGXBR   LM    R2,R3,0(R10)  Get count and address of test input values
000006E4  9878 A008                  00000008    636          LM    R7,R8,8(R10)  Get address of result area and flag area.
000006E8  1222                                   637          LTR   R2,R2         Any test cases?
000006EA  078D                                   638          BZR   R13           ..No, return to caller
000006EC  0DC0                                   639          BASR  R12,0         Set top of loop
                                                 640 *
000006EE  6800 3000                  00000000    641          LD    FPR0,0(,R3)    Get extended BFP test value part 1
000006F2  6820 3008                  00000008    642          LD    R2,8(,R3)     Get extended BFP test value part 1
000006F6  B29D F2F4                  000002F4    643          LFPC  FPCREGNT      Set exceptions non-trappable
000006FA  B3AE 0010                              644          CLGXBR R1,0,FPR0,0  Cvt float in FPR0-FPR2 to uint-64 in GPR1
000006FE  E310 7000 0024             00000000    645          STG   R1,0(,R7)     Store uint-64 result
00000704  B29C 8000                  00000000    646          STFPC (0*4)(R8)     Store resulting FPC flags and DXC
00000708  B222 0010                              647          IPM   R1            Get condition code and program mask
0000070C  8810 001C                  0000001C    648          SRL   R1,28         Isolate CC in low order byte
00000710  4210 8003                  00000003    649          STC   R1,(0*4)+3(,R8)  Save CC as low byte of FPCR
                                                 650 *
00000714  B29D F2F8                  000002F8    651          LFPC  FPCREGTR      Set exceptions trappable
00000718  B982 0011                              652          XGR   R1,R1         Clear any residual result in R1
0000071C  0410                                   653          SPM   R1            Clear out any residual nz condition code
0000071E  B3AE 0010                              654          CLGXBR R1,0,FPR0,0  Cvt float in FPR0-FPR2 to uint-64 in GPR1
00000722  E310 7008 0024             00000008    655          STG   R1,8(,R7)     Store uint-64 result
00000728  B29C 8004                  00000004    656          STFPC (1*4)(R8)     Store resulting FPC flags and DXC
0000072C  B222 0010                              657          IPM   R1            Get condition code and program mask
00000730  8810 001C                  0000001C    658          SRL   R1,28         Isolate CC in low order byte
00000734  4210 8007                  00000007    659          STC   R1,(1*4)+3(,R8)  Save CC as low byte of FPCR
                                                 660 *
00000738  4130 3010                  00000010    661          LA    R3,16(,R3)    Point to next extended BFP input value
0000073C  4170 7010                  00000010    662          LA    R7,16(,R7)    Point to next uint-64 result pair
00000740  4180 8008                  00000008    663          LA    R8,8(,R8)     Point to next FPCR/CC result pair
00000744  062C                                   664          BCTR  R2,R12        Convert next input value.
00000746  07FD                                   665          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    667 ********************************************************************************
                                                    668 *
                                                    669 * Convert extended BFP to integers using each possible rounding mode.
                                                    670 * Ten test results are generated for each input.  A 48-byte test result
                                                    671 * section is used to keep results sets aligned on a quad-double word.
                                                    672 *
                                                    673 * The first four tests use rounding modes specified in the FPCR with
                                                    674 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    675 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    676 * the last two To get full coverage of that instruction pair.
                                                    677 *
                                                    678 * The next six results use instruction-specified rounding modes.
                                                    679 *
                                                    680 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    681 * prior tests used the default rounding mode.  RNTE is tested
                                                    682 * explicitly as a rounding mode in this section.
                                                    683 *
                                                    684 ********************************************************************************

00000748  9823 A000                 00000000    686 CLGXBRA  LM    R2,R3,0(R10)  Get count and address of test input values
0000074C  9878 A008                 00000008    687          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000750  1222                                  688          LTR   R2,R2         Any test cases?
00000752  078D                                  689          BZR   R13           ..No, return to caller
00000754  0DC0                                  690          BASR  R12,0         Set top of loop
                                                    691 *
00000756  6800 3000                 00000000    692          LD    R0,0(,R3)     Get extended BFP test value part 1
0000075A  6820 3008                 00000008    693          LD    R2,8(,R3)     Get extended BFP test value part 2
                                                    694 *
                                                    695 * Test cases using rounding mode specified in the FPCR
                                                    696 *
0000075E  B29D F2F4                 000002F4    697          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000762  B299 0001                 00000001    698          SRNM  1             SET FPC to RZ, towards zero.
00000766  B3AE 0410                             699          CLGXBR R1,0,R0,B'0100'  FPCR ctl'd rounding, inexact masked
0000076A  E310 7000 0024            00000000    700          STG   R1,0*8(,R7)   Store uint-64 result
00000770  B29C 8000                 00000000    701          STFPC 0(R8)         Store resulting FPC flags and DXC
00000774  B222 0010                             702          IPM   R1            Get condition code and program mask
00000778  8810 001C                 0000001C    703          SRL   R1,28         Isolate CC in low order byte
0000077C  4210 8003                 00000003    704          STC   R1,3(,R8)     Save CC as low byte of FPCR
                                                    705 *
00000780  B29D F2F4                 000002F4    706          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000784  B299 0002                 00000002    707          SRNM  2             SET FPC to RP, to +infinity
00000788  B3AE 0410                             708          CLGXBR R1,0,R0,B'0100'  FPCR ctl'd rounding, inexact masked
0000078C  E310 7008 0024            00000008    709          STG   R1,1*8(,R7)   Store uint-64 result
00000792  B29C 8004                 00000004    710          STFPC 1*4(R8)       Store resulting FPC flags and DXC
00000796  B222 0010                             711          IPM   R1            Get condition code and program mask
0000079A  8810 001C                 0000001C    712          SRL   R1,28         Isolate CC in low order byte
0000079E  4210 8007                 00000007    713          STC   R1,(1*4)+3(,R8)  Save CC as low byte of FPCR
                                                    714 *
000007A2  B29D F2F4                 000002F4    715          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007A6  B2B8 0003                 00000003    716          SRNMB 3             SET FPC to RM, to -infinity
000007AA  B3AE 0410                             717          CLGXBR R1,0,R0,B'0100'  FPCR ctl'd rounding, inexact masked
000007AE  E310 7010 0024            00000010    718          STG   R1,2*8(,R7)   Store uint-64 result
000007B4  B29C 8008                 00000008    719          STFPC 2*4(R8)       Store resulting FPC flags and DXC
000007B8  B222 0010                             720          IPM   R1            Get condition code and program mask
000007BC  8810 001C                 0000001C    721          SRL   R1,28         Isolate CC in low order byte
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000007C0  4210 800B                    0000000B   722          STC   R1,(2*4)+3(,R8)  Save CC as low byte of FPCR
                                                  723 *
000007C4  B29D F2F4                    000002F4   724          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007C8  B2B8 0007                    00000007   725          SRNMB 7             RFS, Prepare for Shorter Precision
000007CC  B3AE 0410                               726          CLGXBR R1,0,R0,B'0100'  FPCR ctl'd rounding, inexact masked
000007D0  E310 7018 0024               00000018   727          STG   R1,3*8(,R7)   Store uint-64 result
000007D6  B29C 800C                    0000000C   728          STFPC 3*4(R8)       Store resulting FPC flags and DXC
000007DA  B222 0010                               729          IPM   R1            Get condition code and program mask
000007DE  8810 001C                    0000001C   730          SRL   R1,28         Isolate CC in low order byte
000007E2  4210 800F                    0000000F   731          STC   R1,(3*4)+3(,R8)  Save CC as low byte of FPCR
                                                  732 *
                                                  733 * Test cases using rounding mode specified in the instruction M3 field
                                                  734 *
000007E6  B29D F2F4                    000002F4   735          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007EA  B3AE 1010                               736          CLGXBR R1,1,R0,B'0000'  RNTA, to nearest, ties away from zero
000007EE  E310 7020 0024               00000020   737          STG   R1,4*8(,R7)   Store uint-64 result
000007F4  B29C 8010                    00000010   738          STFPC 4*4(R8)       Store resulting FPC flags and DXC
000007F8  B222 0010                               739          IPM   R1            Get condition code and program mask
000007FC  8810 001C                    0000001C   740          SRL   R1,28         Isolate CC in low order byte
00000800  4210 8013                    00000013   741          STC   R1,(4*4)+3(,R8)  Save CC as low byte of FPCR
                                                  742 *
00000804  B29D F2F4                    000002F4   743          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000808  B3AE 3010                               744          CLGXBR R1,3,R0,B'0000'  RFS, to prepare for shorter precision
0000080C  E310 7028 0024               00000028   745          STG   R1,5*8(,R7)   Store uint-64 result
00000812  B29C 8014                    00000014   746          STFPC 5*4(R8)       Store resulting FPC flags and DXC
00000816  B222 0010                               747          IPM   R1            Get condition code and program mask
0000081A  8810 001C                    0000001C   748          SRL   R1,28         Isolate CC in low order byte
0000081E  4210 8017                    00000017   749          STC   R1,(5*4)+3(,R8)  Save CC as low byte of FPCR
                                                  750 *
00000822  B29D F2F4                    000002F4   751          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000826  B3AE 4010                               752          CLGXBR R1,4,R0,B'0000'  RNTE, to nearest, ties to even
0000082A  E310 7030 0024               00000030   753          STG   R1,6*8(,R7)   Store uint-64 result
00000830  B29C 8018                    00000018   754          STFPC 6*4(R8)       Store resulting FPC flags and DXC
00000834  B222 0010                               755          IPM   R1            Get condition code and program mask
00000838  8810 001C                    0000001C   756          SRL   R1,28         Isolate CC in low order byte
0000083C  4210 801B                    0000001B   757          STC   R1,(6*4)+3(,R8)  Save CC as low byte of FPCR
                                                  758 *
00000840  B29D F2F4                    000002F4   759          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000844  B3AE 5010                               760          CLGXBR R1,5,R0,B'0000'  RZ, toward zero
00000848  E310 7038 0024               00000038   761          STG   R1,7*8(,R7)   Store uint-64 result
0000084E  B29C 801C                    0000001C   762          STFPC 7*4(R8)       Store resulting FPC flags and DXC
00000852  B222 0010                               763          IPM   R1            Get condition code and program mask
00000856  8810 001C                    0000001C   764          SRL   R1,28         Isolate CC in low order byte
0000085A  4210 801F                    0000001F   765          STC   R1,(7*4)+3(,R8)  Save CC as low byte of FPCR
                                                  766 *
0000085E  B29D F2F4                    000002F4   767          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000862  B3AE 6010                               768          CLGXBR R1,6,R0,B'0000'  RP, to +inf
00000866  E310 7040 0024               00000040   769          STG   R1,8*8(,R7)   Store uint-64 result
0000086C  B29C 8020                    00000020   770          STFPC 8*4(R8)       Store resulting FPC flags and DXC
00000870  B222 0010                               771          IPM   R1            Get condition code and program mask
00000874  8810 001C                    0000001C   772          SRL   R1,28         Isolate CC in low order byte
00000878  4210 8023                    00000023   773          STC   R1,(8*4)+3(,R8)  Save CC as low byte of FPCR
                                                  774 *
0000087C  B29D F2F4                    000002F4   775          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000880  B3AE 7010                               776          CLGXBR R1,7,R0,B'0000'  RM, to -inf
00000884  E310 7048 0024               00000048   777          STG   R1,9*8(,R7)   Store uint-64 result
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

0000088A  B29C 8024                    00000024   778          STFPC 9*4(R8)       Store resulting FPC flags and DXC
0000088E  B222 0010                               779          IPM   R1            Get condition code and program mask
00000892  8810 001C                    0000001C   780          SRL   R1,28         Isolate CC in low order byte
00000896  4210 8027                    00000027   781          STC   R1,(9*4)+3(,R8)  Save CC as low byte of FPCR
                                                  782 *
0000089A  4130 3010                    00000010   783          LA    R3,16(,R3)    Point to next input value
0000089E  4170 7050                    00000050   784          LA    R7,10*8(,R7)  Point to next uint-64 result set
000008A2  4180 8030                    00000030   785          LA    R8,12*4(,R8)  Point to next FPCR/CC result pair
000008A6  062C                                    786          BCTR  R2,R12        Convert next input value.
000008A8  07FD                                    787          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    789 ********************************************************************************
                                                    790 *
                                                    791 * BFP inputs for Convert To Logical testing.  The same set of values
                                                    792 * are used for short, long, and extended formats, with the exception
                                                    793 * of the last value, which is rounded to fit the input format and
                                                    794 * for the needs of the test (conversion or rounding).
                                                    795 *
                                                    796 ********************************************************************************


                                                    798 *
                                                    799 * Short integer inputs for Convert From Fixed testing.  The same set of
                                                    800 * inputs are used for short, long, and extended formats.  The last two
                                                    801 * values are used for rounding mode tests for short only; conversion of
                                                    802 * uint-64 to long or extended are always exact.
                                                    803 *
000008AC                                            804 SBFPIN   DS    0F           Inputs for short BFP testing
000008AC  3F800000                                  805         DC    X'3F800000'    +1.0
000008B0  40000000                                  806         DC    X'40000000'    +2.0
000008B4  40800000                                  807         DC    X'40800000'    +4.0
000008B8  7F810000                                  808         DC    X'7F810000'    SNaN
000008BC  7FC10000                                  809         DC    X'7FC10000'    QNaN
000008C0  5F800001                                  810         DC    X'5F800001'    max uint-64 + 1 rounded up to short BFP
                                                    811 *                                    18 446 746 272 732 807 168
                                                    812 *                                Note: above value rounds to max uint-64.
000008C4  5F7FFFFF                                  813         DC    X'5F7FFFFF'    max uint-64 rounded down to short BFP
                                                    814 *                                    18 446 742 974 197 923 840
000008C8  3F400000                                  815         DC    X'3F400000'    +0.75
000008CC  3E800000                                  816         DC    X'3E800000'    +0.25
                                00000024  00000001  817 SBFPCT   EQU   *-SBFPIN     Count of short BFP in list * 4
                                                    818 *
                                                    819 *
000008D0                                            820 SBFPINRM DS    0F           Inputs for short BFP rounding testing
                                                    821 *
                                                    822 * The following values correspond to Figure 9-11 on page 9-16 of the
                                                    823 * z/Arch POP, SA22-7832-10
                                                    824 *
000008D0  BFC00000                                  825         DC    X'BFC00000'    -1.5
000008D4  BF000000                                  826         DC    X'BF000000'    -0.5
000008D8  3F000000                                  827         DC    X'3F000000'    +0.5
000008DC  3FC00000                                  828         DC    X'3FC00000'    +1.5
000008E0  40200000                                  829         DC    X'40200000'    +2.5
000008E4  40B00000                                  830         DC    X'40B00000'    +5.5
000008E8  41180000                                  831         DC    X'41180000'    +9.5
                                                    832 *
                                                    833 * The following values ensure correct rounding for values that
                                                    834 * are not ties.
                                                    835 *
000008EC  5F7FFFFF                                  836         DC    X'5F7FFFFF'    max uint-64 rounded down to short BFP
                                                    837 *                                    18 446 742 974 197 923 840
000008F0  3F400000                                  838         DC    X'3F400000'    +0.75
000008F4  3E800000                                  839         DC    X'3E800000'    +0.25
                                00000028  00000001  840 SBFPRMCT EQU   *-SBFPINRM   Count of rounding mode test short BFP * 4
                                                    841 *
                                                    842 *
000008F8                                            843 LBFPIN   DS    0F            Inputs for long BFP testing
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
000008F8  3FF00000 00000000                          844          DC    X'3FF0000000000000'    +1.0
00000900  40000000 00000000                          845          DC    X'4000000000000000'    +2.0
00000908  40100000 00000000                          846          DC    X'4010000000000000'    +4.0
00000910  7FF01000 00000000                          847          DC    X'7FF0100000000000'    SNaN
00000918  7FF81000 00000000                          848          DC    X'7FF8100000000000'    QNaN
00000920  43F00000 00000000                          849          DC    X'43F0000000000000'   max uint-64 + 1 rounded up
                                                     850 *                                        18 446 744 073 709 555 712
00000928  43EFFFFF FFFFFFFF                          851          DC    X'43EFFFFFFFFFFFFF'   max uint-64 rounded down
                                                     852 *                                        18 446 744 073 709 549 568
00000930  3FE80000 00000000                          853          DC    X'3FE8000000000000'    +0.75
00000938  3FD00000 00000000                          854          DC    X'3FD0000000000000'    +0.25
                            00000048  00000001       855 LBFPCT   EQU   *-LBFPIN     Count of long BFP in list * 8
                                                     856 *
                                                     857 *
00000940                                             858 LBFPINRM DS    0F            Inputs for long BFP rounding testing
                                                     859 *
                                                     860 * The following values correspond to Figure 9-11 on page 9-16 of the
                                                     861 * z/Arch POP, SA22-7832-10
                                                     862 *
00000940  BFF80000 00000000                          863          DC    X'BFF8000000000000'    -1.5
00000948  BFE00000 00000000                          864          DC    X'BFE0000000000000'    -0.5
00000950  3FE00000 00000000                          865          DC    X'3FE0000000000000'    +0.5
00000958  3FF80000 00000000                          866          DC    X'3FF8000000000000'    +1.5
00000960  40040000 00000000                          867          DC    X'4004000000000000'    +2.5
00000968  40160000 00000000                          868          DC    X'4016000000000000'    +5.5
00000970  40230000 00000000                          869          DC    X'4023000000000000'    +9.5
                                                     870 *
                                                     871 * The following values ensure correct rounding for values that
                                                     872 * are not ties.
                                                     873 *
00000978  43EFFFFF FFFFFFFF                          874          DC    X'43EFFFFFFFFFFFFF'   max uint-64 rounded down
                                                     875 *                                       18 446 744 073 709 549 568
                                                     876 *        DC    X'3FE8000000000000'    0.75
00000980  3FD00000 00000000                          877          DC    X'3FD0000000000000'    +0.25
                            00000048  00000001       878 LBFPRMCT EQU   *-LBFPINRM   Count of roundinf test long BFP * 8
                                                     879 *
                                                     880 *
00000988                                             881 XBFPIN   DS    0D            Inputs for extended BFP testing
00000988  3FFF0000 00000000                          882          DC    X'3FFF0000000000000000000000000000'    +1.0
00000998  40000000 00000000                          883          DC    X'4000000000000000000000000000000'    +2.0
000009A8  40010000 00000000                          884          DC    X'4001000000000000000000000000000'    +4.0
000009B8  7FFF0100 00000000                          885          DC    X'7FFF0100000000000000000000000000'    SNaN
000009C8  7FFF8100 00000000                          886          DC    X'7FFF8100000000000000000000000000'    QNaN
000009D8  403F0000 00000000                          887          DC    X'403F0000000000000000000000000000'   max uint-64 + 1
                                                     888 *                                       18 446 744 073 709 551 616 (exact)
000009E8  403EFFFF FFFFFFFF                          889          DC    X'403EFFFFFFFFFFFFFFE000000000000'   max uint-64
                                                     890 *                                       18 446 744 073 709 551 615 (exact)
000009F8  403EFFFF FFFFFFFF                          891          DC    X'403EFFFFFFFFFFFFFFF000000000000'   max uint-64+0.5
                                                     892 *                                       18 446 744 073 709 551 615.5
                                                     893 *    Above is always inexact, and may overflow based on rounding mode
00000A08  3FFE8000 00000000                          894          DC    X'3FFE8000000000000000000000000000'         0.75
00000A18  3FFD0000 00000000                          895          DC    X'3FFD0000000000000000000000000000'         0.25
                            000000A0  00000001       896 XBFPCT   EQU   *-XBFPIN     Count of extended BFP in list * 16
                                                     897 *
                                                     898 *
00000A28                                             899 XBFPINRM DS    0D            Inputs for extended BFP rounding testing
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    900 *
                                                    901 * The following values correspond to Figure 9-11 on page 9-16 of the
                                                    902 * z/Arch POP, SA22-7832-10
                                                    903 *
00000A28  BFFF8000 00000000                         904          DC    X'BFFF8000000000000000000000000000'         -1.5
00000A38  BFFE0000 00000000                         905          DC    X'BFFE0000000000000000000000000000'         -0.5
00000A48  3FFE0000 00000000                         906          DC    X'3FFE0000000000000000000000000000'         +0.5
00000A58  3FFF8000 00000000                         907          DC    X'3FFF8000000000000000000000000000'         +1.5
00000A68  40004000 00000000                         908          DC    X'40004000000000000000000000000000'         +2.5
00000A78  40016000 00000000                         909          DC    X'40016000000000000000000000000000'         +5.5
00000A88  40023000 00000000                         910          DC    X'40023000000000000000000000000000'         +9.5
                                                    911 *
                                                    912 * The following values ensure correct rounding for values that
                                                    913 * are not ties.
                                                    914 *
00000A98  403EFFFF FFFFFFFF                         915          DC    X'403EFFFFFFFFFFFFFFF000000000000'   max uint-64+0.5
                                                    916 *                                  18 446 744 073 709 551 615.5
                                                    917 *    Above is always inexact, and may overflow based on rounding mode
00000AA8  3FFE8000 00000000                         918          DC    X'3FFE8000000000000000000000000000'         0.75
00000AB8  3FFD0000 00000000                         919          DC    X'3FFD0000000000000000000000000000'         0.25
                            000000A0  00000001    920 XBFPRMCT EQU   *-XBFPINRM   Count of rounding test extd BFP * 16
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

                                            922 ********************************************************************************
                                            923 *                    ACTUAL results saved here
                                            924 ********************************************************************************
                                            925 *
                                            926 *                 Locations for ACTUAL results
                                            927 *
                                            928 *
                          00001000  00000000  929 SINTOUT  EQU   BFPCVTTL+X'1000'    Uint-64 values from short BFP
                                            930 *                                  ..6 pairs used, room for 32
                          00001200  00000000  931 SINTFLGS EQU   BFPCVTTL+X'1200'    FPC flags and DXC from short BFP
                                            932 *                                  ..6 pairs used, room for 32
                          00001300  00000000  933 SINTRMO  EQU   BFPCVTTL+X'1300'    Short rounding mode test results
                                            934 *                                  ..8 sets used, room for 16
                          00001800  00000000  935 SINTRMOF EQU   BFPCVTTL+X'1800'    Short rounding mode FPCR contents
                                            936 *                                  ..8 sets used, room for 16
                                            937 *
                          00002000  00000000  938 LINTOUT  EQU   BFPCVTTL+X'2000'    Uint-64 values from long BFP
                                            939 *                                  ..6 pairs used, room for 32
                          00002200  00000000  940 LINTFLGS EQU   BFPCVTTL+X'2200'    FPC flags and DXC from long BFP
                                            941 *                                  ..6 pairs used, room for 32
                          00002300  00000000  942 LINTRMO  EQU   BFPCVTTL+X'2300'    Long rounding mode test results
                                            943 *                                  ..8 sets used, room for 16
                          00002800  00000000  944 LINTRMOF EQU   BFPCVTTL+X'2800'    Long rounding mode FPCR contents
                                            945 *                                  ..8 sets used, room for 16
                                            946 *
                          00003000  00000000  947 XINTOUT  EQU   BFPCVTTL+X'3000'    Uint-64 values from extd BFP
                                            948 *                                  ..6 pairs used, room for 32
                          00003200  00000000  949 XINTFLGS EQU   BFPCVTTL+X'3200'    FPC flags and DXC from extd BFP
                                            950 *                                  ..6 pairs used, room for 32
                          00003300  00000000  951 XINTRMO  EQU   BFPCVTTL+X'3300'    Extended rounding mode test results
                                            952 *                                  ..8 sets used, room for 16
                          00003800  00000000  953 XINTRMOF EQU   BFPCVTTL+X'3800'    Long rounding mode FPCR contents
                                            954 *                                  ..8 sets used, room for 16
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT
                                                     956 ********************************************************************************
                                                     957 *                       EXPECTED results
                                                     958 ********************************************************************************
                                                     959 *
00000AC8                       00000AC8  00005000   960          ORG   BFPCVTTL+X'5000'   (past end of actual results)
                                                     961 *
                                                     962 * I am not satisfied with test case 6. Further investigation needed. It
                                                     963 * would appear that f32_to_uint64 is returning invalid and max uint-64.
                                                     964 *
                               00005000  00000001    965 SINTOUT_GOOD EQU *
00005000  C3D3C7C5 C2D94099                          966  DC CL48'CLGEBR result pair 1'
00005030  00000000 00000001                          967  DC XL16'00000000000000010000000000000001'
00005040  C3D3C7C5 C2D94099                          968  DC CL48'CLGEBR result pair 2'
00005070  00000000 00000002                          969  DC XL16'00000000000000020000000000000002'
00005080  C3D3C7C5 C2D94099                          970  DC CL48'CLGEBR result pair 3'
000050B0  00000000 00000004                          971  DC XL16'00000000000000040000000000000004'
000050C0  C3D3C7C5 C2D94099                          972  DC CL48'CLGEBR result pair 4'
000050F0  00000000 00000000                          973  DC XL16'00000000000000000000000000000000'
00005100  C3D3C7C5 C2D94099                          974  DC CL48'CLGEBR result pair 5'
00005130  00000000 00000000                          975  DC XL16'00000000000000000000000000000000'
00005140  C3D3C7C5 C2D94099                          976  DC CL48'CLGEBR result pair 6'
00005170  FFFFFFFF FFFFFFFF                          977  DC XL16'FFFFFFFFFFFFFFFF0000000000000000'
00005180  C3D3C7C5 C2D94099                          978  DC CL48'CLGEBR result pair 7'
000051B0  FFFFFF00 00000000                          979  DC XL16'FFFFFF0000000000FFFFFF0000000000'
000051C0  C3D3C7C5 C2D94099                          980  DC CL48'CLGEBR result pair 8'
000051F0  00000000 00000001                          981  DC XL16'00000000000000010000000000000001'
00005200  C3D3C7C5 C2D94099                          982  DC CL48'CLGEBR result pair 9'
00005230  00000000 00000000                          983  DC XL16'00000000000000000000000000000000'
                               00000009  00000001    984 SINTOUT_NUM EQU (*-SINTOUT_GOOD)/64
                                                     985 *
                                                     986 *
                               00005240  00000001    987 SINTFLGS_GOOD EQU *
00005240  C3D3C7C5 C2D940C6                          988  DC CL48'CLGEBR FPCR pairs 1-2'
00005270  00000002 F8000002                          989  DC XL16'00000002F800000200000002F8000002'
00005280  C3D3C7C5 C2D940C6                          990  DC CL48'CLGEBR FPCR pairs 3-4'
000052B0  00000002 F8000002                          991  DC XL16'00000002F800000200880003F8008000'
000052C0  C3D3C7C5 C2D940C6                          992  DC CL48'CLGEBR FPCR pairs 5-6'
000052F0  00880003 F8008000                          993  DC XL16'00880003F800800000880003F8008000'
00005300  C3D3C7C5 C2D940C6                          994  DC CL48'CLGEBR FPCR pairs 7-8'
00005330  00000002 F8000002                          995  DC XL16'00000002F800000200080002F8000C02'
00005340  C3D3C7C5 C2D940C6                          996  DC CL48'CLGEBR FPCR pair 9'
00005370  00080002 F8000802                          997  DC XL16'00080002F8000802000000000000000'
                               00000005  00000001    998 SINTFLGS_NUM EQU (*-SINTFLGS_GOOD)/64
                                                     999 *
                                                    1000 *
                               00005380  00000001   1001 SINTRMO_GOOD EQU *
00005380  C3D3C7C5 C2D94060                         1002  DC CL48'CLGEBR -1.5 FPC modes 1, 2'
000053B0  00000000 00000000                         1003  DC XL16'00000000000000000000000000000000'
000053C0  C3D3C7C5 C2D94060                         1004  DC CL48'CLGEBR -1.5 FPC modes 3, 7'
000053F0  00000000 00000000                         1005  DC XL16'00000000000000000000000000000000'
00005400  C3D3C7C5 C2D94060                         1006  DC CL48'CLGEBR -1.5 M3 modes 1, 3'
00005430  00000000 00000000                         1007  DC XL16'00000000000000000000000000000000'
00005440  C3D3C7C5 C2D94060                         1008  DC CL48'CLGEBR -1.5 M3 modes 4, 5'
00005470  00000000 00000000                         1009  DC XL16'00000000000000000000000000000000'
00005480  C3D3C7C5 C2D94060                         1010  DC CL48'CLGEBR -1.5 M3 modes 6, 7'
000054B0  00000000 00000000                         1011  DC XL16'00000000000000000000000000000000'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000054C0  C3D3C7C5 C2D94060                     1012  DC CL48'CLGEBR -0.5 FPC modes 1, 2'
000054F0  00000000 00000000                     1013  DC XL16'00000000000000000000000000000000'
00005500  C3D3C7C5 C2D94060                     1014  DC CL48'CLGEBR -0.5 FPC modes 3, 7'
00005530  00000000 00000000                     1015  DC XL16'00000000000000000000000000000000'
00005540  C3D3C7C5 C2D94060                     1016  DC CL48'CLGEBR -0.5 M3 modes 1, 3'
00005570  00000000 00000000                     1017  DC XL16'00000000000000000000000000000000'
00005580  C3D3C7C5 C2D94060                     1018  DC CL48'CLGEBR -0.5 M3 modes 4, 5'
000055B0  00000000 00000000                     1019  DC XL16'00000000000000000000000000000000'
000055C0  C3D3C7C5 C2D94060                     1020  DC CL48'CLGEBR -0.5 M3 modes 6, 7'
000055F0  00000000 00000000                     1021  DC XL16'00000000000000000000000000000000'
00005600  C3D3C7C5 C2D9404E                     1022  DC CL48'CLGEBR +0.5 FPC modes 1, 2'
00005630  00000000 00000000                     1023  DC XL16'00000000000000000000000000000001'
00005640  C3D3C7C5 C2D9404E                     1024  DC CL48'CLGEBR +0.5 FPC modes 3, 7'
00005670  00000000 00000000                     1025  DC XL16'00000000000000000000000000000001'
00005680  C3D3C7C5 C2D9404E                     1026  DC CL48'CLGEBR +0.5 M3 modes 1, 3'
000056B0  00000000 00000001                     1027  DC XL16'00000000000000010000000000000001'
000056C0  C3D3C7C5 C2D9404E                     1028  DC CL48'CLGEBR +0.5 M3 modes 4, 5'
000056F0  00000000 00000000                     1029  DC XL16'00000000000000000000000000000000'
00005700  C3D3C7C5 C2D9404E                     1030  DC CL48'CLGEBR +0.5 M3 modes 6, 7'
00005730  00000000 00000001                     1031  DC XL16'00000000000000010000000000000000'
00005740  C3D3C7C5 C2D9404E                     1032  DC CL48'CLGEBR +1.5 FPC modes 1, 2'
00005770  00000000 00000001                     1033  DC XL16'00000000000000010000000000000002'
00005780  C3D3C7C5 C2D9404E                     1034  DC CL48'CLGEBR +1.5 FPC modes 3, 7'
000057B0  00000000 00000001                     1035  DC XL16'00000000000000010000000000000001'
000057C0  C3D3C7C5 C2D9404E                     1036  DC CL48'CLGEBR +1.5 M3 modes 1, 3'
000057F0  00000000 00000002                     1037  DC XL16'00000000000000020000000000000001'
00005800  C3D3C7C5 C2D9404E                     1038  DC CL48'CLGEBR +1.5 M3 modes 4, 5'
00005830  00000000 00000002                     1039  DC XL16'00000000000000020000000000000001'
00005840  C3D3C7C5 C2D9404E                     1040  DC CL48'CLGEBR +1.5 M3 modes 6, 7'
00005870  00000000 00000002                     1041  DC XL16'00000000000000020000000000000001'
00005880  C3D3C7C5 C2D9404E                     1042  DC CL48'CLGEBR +2.5 FPC modes 1, 2'
000058B0  00000000 00000002                     1043  DC XL16'00000000000000020000000000000003'
000058C0  C3D3C7C5 C2D9404E                     1044  DC CL48'CLGEBR +2.5 FPC modes 3, 7'
000058F0  00000000 00000002                     1045  DC XL16'00000000000000020000000000000003'
00005900  C3D3C7C5 C2D9404E                     1046  DC CL48'CLGEBR +2.5 M3 modes 1, 3'
00005930  00000000 00000003                     1047  DC XL16'00000000000000030000000000000003'
00005940  C3D3C7C5 C2D9404E                     1048  DC CL48'CLGEBR +2.5 M3 modes 4, 5'
00005970  00000000 00000002                     1049  DC XL16'00000000000000020000000000000002'
00005980  C3D3C7C5 C2D9404E                     1050  DC CL48'CLGEBR +2.5 M3 modes 6, 7'
000059B0  00000000 00000003                     1051  DC XL16'00000000000000030000000000000002'
000059C0  C3D3C7C5 C2D9404E                     1052  DC CL48'CLGEBR +5.5 FPC modes 1, 2'
000059F0  00000000 00000005                     1053  DC XL16'00000000000000050000000000000006'
00005A00  C3D3C7C5 C2D9404E                     1054  DC CL48'CLGEBR +5.5 FPC modes 3, 7'
00005A30  00000000 00000005                     1055  DC XL16'00000000000000050000000000000005'
00005A40  C3D3C7C5 C2D9404E                     1056  DC CL48'CLGEBR +5.5 M3 modes 1, 3'
00005A70  00000000 00000006                     1057  DC XL16'00000000000000060000000000000005'
00005A80  C3D3C7C5 C2D9404E                     1058  DC CL48'CLGEBR +5.5 M3 modes 4, 5'
00005AB0  00000000 00000006                     1059  DC XL16'00000000000000060000000000000005'
00005AC0  C3D3C7C5 C2D9404E                     1060  DC CL48'CLGEBR +5.5 M3 modes 6, 7'
00005AF0  00000000 00000006                     1061  DC XL16'00000000000000060000000000000005'
00005B00  C3D3C7C5 C2D9404E                     1062  DC CL48'CLGEBR +9.5 FPC modes 1, 2'
00005B30  00000000 00000009                     1063  DC XL16'0000000000000009000000000000000A'
00005B40  C3D3C7C5 C2D9404E                     1064  DC CL48'CLGEBR +9.5 FPC modes 3, 7'
00005B70  00000000 00000009                     1065  DC XL16'00000000000000090000000000000009'
00005B80  C3D3C7C5 C2D9404E                     1066  DC CL48'CLGEBR +9.5 M3 modes 1, 3'
00005BB0  00000000 0000000A                     1067  DC XL16'0000000000000000A000000000000009'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00005BC0  C3D3C7C5 C2D9404E                          1068  DC CL48'CLGEBR +9.5 M3 modes 4, 5'
00005BF0  00000000 0000000A                          1069  DC XL16'0000000000000000A0000000000000009'
00005C00  C3D3C7C5 C2D9404E                          1070  DC CL48'CLGEBR +9.5 M3 modes 6, 7'
00005C30  00000000 0000000A                          1071  DC XL16'0000000000000000A0000000000000009'
00005C40  C3D3C7C5 C2D94094                          1072  DC CL48'CLGEBR max FPC modes 1, 2'
00005C70  FFFFFF00 00000000                          1073  DC XL16'FFFFFF0000000000FFFFFF0000000000'
00005C80  C3D3C7C5 C2D94094                          1074  DC CL48'CLGEBR max FPC modes 3, 7'
00005CB0  FFFFFF00 00000000                          1075  DC XL16'FFFFFF0000000000FFFFFF0000000000'
00005CC0  C3D3C7C5 C2D94094                          1076  DC CL48'CLGEBR max M3 modes 1, 3'
00005CF0  FFFFFF00 00000000                          1077  DC XL16'FFFFFF0000000000FFFFFF0000000000'
00005D00  C3D3C7C5 C2D94094                          1078  DC CL48'CLGEBR max M3 modes 4, 5'
00005D30  FFFFFF00 00000000                          1079  DC XL16'FFFFFF0000000000FFFFFF0000000000'
00005D40  C3D3C7C5 C2D94094                          1080  DC CL48'CLGEBR max M3 modes 6, 7'
00005D70  FFFFFF00 00000000                          1081  DC XL16'FFFFFF0000000000FFFFFF0000000000'
00005D80  C3D3C7C5 C2D9404E                          1082  DC CL48'CLGEBR +0.75 FPC modes 1, 2'
00005DB0  00000000 00000000                          1083  DC XL16'00000000000000000000000000000001'
00005DC0  C3D3C7C5 C2D9404E                          1084  DC CL48'CLGEBR +0.75 FPC modes 3, 7'
00005DF0  00000000 00000000                          1085  DC XL16'00000000000000000000000000000001'
00005E00  C3D3C7C5 C2D9404E                          1086  DC CL48'CLGEBR +0.75 M3 modes 1, 3'
00005E30  00000000 00000001                          1087  DC XL16'00000000000000010000000000000001'
00005E40  C3D3C7C5 C2D9404E                          1088  DC CL48'CLGEBR +0.75 M3 modes 4, 5'
00005E70  00000000 00000001                          1089  DC XL16'00000000000000010000000000000000'
00005E80  C3D3C7C5 C2D9404E                          1090  DC CL48'CLGEBR +0.75 M3 modes 6, 7'
00005EB0  00000000 00000001                          1091  DC XL16'00000000000000010000000000000000'
00005EC0  C3D3C7C5 C2D9404E                          1092  DC CL48'CLGEBR +0.25 FPC modes 1, 2'
00005EF0  00000000 00000000                          1093  DC XL16'00000000000000000000000000000001'
00005F00  C3D3C7C5 C2D9404E                          1094  DC CL48'CLGEBR +0.25 FPC modes 3, 7'
00005F30  00000000 00000000                          1095  DC XL16'00000000000000000000000000000001'
00005F40  C3D3C7C5 C2D9404E                          1096  DC CL48'CLGEBR +0.25 M3 modes 1, 3'
00005F70  00000000 00000000                          1097  DC XL16'00000000000000000000000000000001'
00005F80  C3D3C7C5 C2D9404E                          1098  DC CL48'CLGEBR +0.25 M3 modes 4, 5'
00005FB0  00000000 00000000                          1099  DC XL16'00000000000000000000000000000000'
00005FC0  C3D3C7C5 C2D9404E                          1100  DC CL48'CLGEBR +0.25 M3 modes 6, 7'
00005FF0  00000000 00000001                          1101  DC XL16'00000000000000010000000000000000'
                            00000032  00000001       1102 SINTRMO_NUM EQU (*-SINTRMO_GOOD)/64
                                                     1103 *
                                                     1104 *
                            00006000  00000001       1105 SINTRMOF_GOOD EQU *
00006000  C3D3C7C5 C2D94060                          1106  DC CL48'CLGEBR -1.5 FPC modes 1-3, 7 FPCR'
00006030  00800003 00800003                          1107  DC XL16'00800003008000030080000300800003'
00006040  C3D3C7C5 C2D94060                          1108  DC CL48'CLGEBR -1.5 M3 modes 1, 3-5 FPCR'
00006070  00880003 00880003                          1109  DC XL16'00880003008800030088000300880003'
00006080  C3D3C7C5 C2D94060                          1110  DC CL48'CLGEBR -1.5 M3 modes 6, 7 FPCR'
000060B0  00880003 00880003                          1111  DC XL16'00880003008800030000000000000000'
000060C0  C3D3C7C5 C2D94060                          1112  DC CL48'CLGEBR -0.5 FPC modes 1-3, 7 FPCR'
000060F0  00000001 00000001                          1113  DC XL16'00000001000000010080000300800003'
00006100  C3D3C7C5 C2D94060                          1114  DC CL48'CLGEBR -0.5 M3 modes 1, 3-5 FPCR'
00006130  00880003 00880003                          1115  DC XL16'00880003008800030008000100080001'
00006140  C3D3C7C5 C2D94060                          1116  DC CL48'CLGEBR -0.5 M3 modes 6, 7 FPCR'
00006170  00080001 00880003                          1117  DC XL16'00080001008800030000000000000000'
00006180  C3D3C7C5 C2D9404E                          1118  DC CL48'CLGEBR +0.5 FPC modes 1-3, 7 FPCR'
000061B0  00000002 00000002                          1119  DC XL16'00000002000000020000000200000002'
000061C0  C3D3C7C5 C2D9404E                          1120  DC CL48'CLGEBR +0.5 M3 modes 1, 3-5 FPCR'
000061F0  00080002 00080002                          1121  DC XL16'00080002000800020008000200080002'
00006200  C3D3C7C5 C2D9404E                          1122  DC CL48'CLGEBR +0.5 M3 modes 6, 7 FPCR'
00006230  00080002 00080002                          1123  DC XL16'00080002000800020000000000000000'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00006240  C3D3C7C5 C2D9404E                        1124  DC CL48'CLGEBR +1.5 FPC modes 1-3, 7 FPCR'
00006270  00000002 00000002                        1125  DC XL16'00000002000000020000000200000002'
00006280  C3D3C7C5 C2D9404E                        1126  DC CL48'CLGEBR +1.5 M3 modes 1, 3-5 FPCR'
000062B0  00080002 00080002                        1127  DC XL16'00080002000800020008000200080002'
000062C0  C3D3C7C5 C2D9404E                        1128  DC CL48'CLGEBR +1.5 M3 modes 6, 7 FPCR'
000062F0  00080002 00080002                        1129  DC XL16'00080002000800020000000000000000'
00006300  C3D3C7C5 C2D9404E                        1130  DC CL48'CLGEBR +2.5 FPC modes 1-3, 7 FPCR'
00006330  00000002 00000002                        1131  DC XL16'00000002000000020000000200000002'
00006340  C3D3C7C5 C2D9404E                        1132  DC CL48'CLGEBR +2.5 M3 modes 1, 3-5 FPCR'
00006370  00080002 00080002                        1133  DC XL16'00080002000800020008000200080002'
00006380  C3D3C7C5 C2D9404E                        1134  DC CL48'CLGEBR +2.5 M3 modes 6, 7 FPCR'
000063B0  00080002 00080002                        1135  DC XL16'00080002000800020000000000000000'
000063C0  C3D3C7C5 C2D9404E                        1136  DC CL48'CLGEBR +5.5 FPC modes 1-3, 7 FPCR'
000063F0  00000002 00000002                        1137  DC XL16'00000002000000020000000200000002'
00006400  C3D3C7C5 C2D9404E                        1138  DC CL48'CLGEBR +5.5 M3 modes 1, 3-5 FPCR'
00006430  00080002 00080002                        1139  DC XL16'00080002000800020008000200080002'
00006440  C3D3C7C5 C2D9404E                        1140  DC CL48'CLGEBR +5.5 M3 modes 6, 7 FPCR'
00006470  00080002 00080002                        1141  DC XL16'00080002000800020000000000000000'
00006480  C3D3C7C5 C2D9404E                        1142  DC CL48'CLGEBR +9.5 FPC modes 1-3, 7 FPCR'
000064B0  00000002 00000002                        1143  DC XL16'00000002000000020000000200000002'
000064C0  C3D3C7C5 C2D9404E                        1144  DC CL48'CLGEBR +9.5 M3 modes 1, 3-5 FPCR'
000064F0  00080002 00080002                        1145  DC XL16'00080002000800020008000200080002'
00006500  C3D3C7C5 C2D9404E                        1146  DC CL48'CLGEBR +9.5 M3 modes 6, 7 FPCR'
00006530  00080002 00080002                        1147  DC XL16'00080002000800020000000000000000'
00006540  C3D3C7C5 C2D94094                        1148  DC CL48'CLGEBR max+1 FPC modes 1-3, 7 FPCR'
00006570  00000002 00000002                        1149  DC XL16'00000002000000020000000200000002'
00006580  C3D3C7C5 C2D94094                        1150  DC CL48'CLGEBR max+1 M3 modes 1, 3-5 FPCR'
000065B0  00000002 00000002                        1151  DC XL16'00000002000000020000000200000002'
000065C0  C3D3C7C5 C2D94094                        1152  DC CL48'CLGEBR max+1 M3 modes 6, 7 FPCR'
000065F0  00000002 00000002                        1153  DC XL16'00000002000000020000000000000000'
00006600  C3D3C7C5 C2D9404E                        1154  DC CL48'CLGEBR +0.75 FPC modes 1-3, 7 FPCR'
00006630  00000002 00000002                        1155  DC XL16'00000002000000020000000200000002'
00006640  C3D3C7C5 C2D9404E                        1156  DC CL48'CLGEBR +0.75 M3 modes 1, 3-5 FPCR'
00006670  00080002 00080002                        1157  DC XL16'00080002000800020008000200080002'
00006680  C3D3C7C5 C2D9404E                        1158  DC CL48'CLGEBR +0.75 M3 modes 6, 7 FPCR'
000066B0  00080002 00080002                        1159  DC XL16'00080002000800020000000000000000'
000066C0  C3D3C7C5 C2D9404E                        1160  DC CL48'CLGEBR +0.25 FPC modes 1-3, 7 FPCR'
000066F0  00000002 00000002                        1161  DC XL16'00000002000000020000000200000002'
00006700  C3D3C7C5 C2D9404E                        1162  DC CL48'CLGEBR +0.25 M3 modes 1, 3-5 FPCR'
00006730  00080002 00080002                        1163  DC XL16'00080002000800020008000200080002'
00006740  C3D3C7C5 C2D9404E                        1164  DC CL48'CLGEBR +0.5 M3 modes 6, 7 FPCR'
00006770  00080002 00080002                        1165  DC XL16'00080002000800020000000000000000'
                             0000001E  00000001    1166 SINTRMOF_NUM EQU (*-SINTRMOF_GOOD)/64
                                                   1167 *
                                                   1168 *
                             00006780  00000001    1169 LINTOUT_GOOD EQU *
00006780  C3D3C7C4 C2D94099                        1170  DC CL48'CLGDBR result pair 1'
000067B0  00000000 00000001                        1171  DC XL16'00000000000000010000000000000001'
000067C0  C3D3C7C4 C2D94099                        1172  DC CL48'CLGDBR result pair 2'
000067F0  00000000 00000002                        1173  DC XL16'00000000000000020000000000000002'
00006800  C3D3C7C4 C2D94099                        1174  DC CL48'CLGDBR result pair 3'
00006830  00000000 00000004                        1175  DC XL16'00000000000000040000000000000004'
00006840  C3D3C7C4 C2D94099                        1176  DC CL48'CLGDBR result pair 4'
00006870  00000000 00000000                        1177  DC XL16'00000000000000000000000000000000'
00006880  C3D3C7C4 C2D94099                        1178  DC CL48'CLGDBR result pair 5'
000068B0  00000000 00000000                        1179  DC XL16'00000000000000000000000000000000'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000068C0  C3D3C7C4 C2D94099                        1180  DC CL48'CLGDBR result pair 6'
000068F0  FFFFFFFF FFFFFFFF                        1181  DC XL16'FFFFFFFFFFFFFFFF0000000000000000'
00006900  C3D3C7C4 C2D94099                        1182  DC CL48'CLGDBR result pair 7'
00006930  FFFFFFFF FFFFF800                        1183  DC XL16'FFFFFFFFFFFFF800FFFFFFFFFFFFF800'
00006940  C3D3C7C4 C2D94099                        1184  DC CL48'CLGDBR result pair 8'
00006970  00000000 00000001                        1185  DC XL16'00000000000000010000000000000001'
00006980  C3D3C7C4 C2D94099                        1186  DC CL48'CLGDBR result pair 9'
000069B0  00000000 00000000                        1187  DC XL16'00000000000000000000000000000000'
                             00000009  00000001   1188 LINTOUT_NUM EQU (*-LINTOUT_GOOD)/64
                                                  1189 *
                                                  1190 *
                             000069C0  00000001   1191 LINTFLGS_GOOD EQU *
000069C0  C3D3C7C4 C2D940C6                        1192  DC CL48'CLGDBR FPCR pairs 1-2'
000069F0  00000002 F8000002                        1193  DC XL16'00000002F800000200000002F8000002'
00006A00  C3D3C7C4 C2D940C6                        1194  DC CL48'CLGDBR FPCR pairs 3-4'
00006A30  00000002 F8000002                        1195  DC XL16'00000002F800000200880003F8008000'
00006A40  C3D3C7C4 C2D940C6                        1196  DC CL48'CLGDBR FPCR pairs 5-6'
00006A70  00880003 F8008000                        1197  DC XL16'00880003F800800000880003F8008000'
00006A80  C3D3C7C4 C2D940C6                        1198  DC CL48'CLGDBR FPCR pairs 7-8'
00006AB0  00000002 F8000002                        1199  DC XL16'00000002F800000200080002F8000C02'
00006AC0  C3D3C7C4 C2D940C6                        1200  DC CL48'CLGDBR FPCR pair 9'
00006AF0  00080002 F8000802                        1201  DC XL16'00080002F8000802000000000000000'
                             00000005  00000001   1202 LINTFLGS_NUM EQU (*-LINTFLGS_GOOD)/64
                                                  1203 *
                                                  1204 *
                             00006B00  00000001   1205 LINTRMO_GOOD EQU *
00006B00  C3D3C7C4 C2D94060                        1206  DC CL48'CLGDBR -1.5 FPC modes 1, 2'
00006B30  00000000 00000000                        1207  DC XL16'00000000000000000000000000000000'
00006B40  C3D3C7C4 C2D94060                        1208  DC CL48'CLGDBR -1.5 FPC modes 3, 7'
00006B70  00000000 00000000                        1209  DC XL16'00000000000000000000000000000000'
00006B80  C3D3C7C4 C2D94060                        1210  DC CL48'CLGDBR -1.5 M3 modes 1, 3'
00006BB0  00000000 00000000                        1211  DC XL16'00000000000000000000000000000000'
00006BC0  C3D3C7C4 C2D94060                        1212  DC CL48'CLGDBR -1.5 M3 modes 4, 5'
00006BF0  00000000 00000000                        1213  DC XL16'00000000000000000000000000000000'
00006C00  C3D3C7C4 C2D94060                        1214  DC CL48'CLGDBR -1.5 M3 modes 6, 7'
00006C30  00000000 00000000                        1215  DC XL16'00000000000000000000000000000000'
00006C40  C3D3C7C4 C2D94060                        1216  DC CL48'CLGDBR -0.5 FPC modes 1, 2'
00006C70  00000000 00000000                        1217  DC XL16'00000000000000000000000000000000'
00006C80  C3D3C7C4 C2D94060                        1218  DC CL48'CLGDBR -0.5 FPC modes 3, 7'
00006CB0  00000000 00000000                        1219  DC XL16'00000000000000000000000000000000'
00006CC0  C3D3C7C4 C2D94060                        1220  DC CL48'CLGDBR -0.5 M3 modes 1, 3'
00006CF0  00000000 00000000                        1221  DC XL16'00000000000000000000000000000000'
00006D00  C3D3C7C4 C2D94060                        1222  DC CL48'CLGDBR -0.5 M3 modes 4, 5'
00006D30  00000000 00000000                        1223  DC XL16'00000000000000000000000000000000'
00006D40  C3D3C7C4 C2D94060                        1224  DC CL48'CLGDBR -0.5 M3 modes 6, 7'
00006D70  00000000 00000000                        1225  DC XL16'00000000000000000000000000000000'
00006D80  C3D3C7C4 C2D9404E                        1226  DC CL48'CLGDBR +0.5 FPC modes 1, 2'
00006DB0  00000000 00000000                        1227  DC XL16'00000000000000000000000000000001'
00006DC0  C3D3C7C4 C2D9404E                        1228  DC CL48'CLGDBR +0.5 FPC modes 3, 7'
00006DF0  00000000 00000000                        1229  DC XL16'00000000000000000000000000000001'
00006E00  C3D3C7C4 C2D9404E                        1230  DC CL48'CLGDBR +0.5 M3 modes 1, 3'
00006E30  00000000 00000001                        1231  DC XL16'00000000000000010000000000000001'
00006E40  C3D3C7C4 C2D9404E                        1232  DC CL48'CLGDBR +0.5 M3 modes 4, 5'
00006E70  00000000 00000000                        1233  DC XL16'00000000000000000000000000000000'
00006E80  C3D3C7C4 C2D9404E                        1234  DC CL48'CLGDBR +0.5 M3 modes 6, 7'
00006EB0  00000000 00000001                        1235  DC XL16'00000000000000010000000000000000'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00006EC0  C3D3C7C4 C2D9404E                          1236  DC CL48'CLGDBR +1.5 FPC modes 1, 2'
00006EF0  00000000 00000001                          1237  DC XL16'00000000000000010000000000000002'
00006F00  C3D3C7C4 C2D9404E                          1238  DC CL48'CLGDBR +1.5 FPC modes 3, 7'
00006F30  00000000 00000001                          1239  DC XL16'00000000000000010000000000000001'
00006F40  C3D3C7C4 C2D9404E                          1240  DC CL48'CLGDBR +1.5 M3 modes 1, 3'
00006F70  00000000 00000002                          1241  DC XL16'00000000000000020000000000000001'
00006F80  C3D3C7C4 C2D9404E                          1242  DC CL48'CLGDBR +1.5 M3 modes 4, 5'
00006FB0  00000000 00000002                          1243  DC XL16'00000000000000020000000000000001'
00006FC0  C3D3C7C4 C2D9404E                          1244  DC CL48'CLGDBR +1.5 M3 modes 6, 7'
00006FF0  00000000 00000002                          1245  DC XL16'00000000000000020000000000000001'
00007000  C3D3C7C4 C2D9404E                          1246  DC CL48'CLGDBR +2.5 FPC modes 1, 2'
00007030  00000000 00000002                          1247  DC XL16'00000000000000020000000000000003'
00007040  C3D3C7C4 C2D9404E                          1248  DC CL48'CLGDBR +2.5 FPC modes 3, 7'
00007070  00000000 00000002                          1249  DC XL16'00000000000000020000000000000003'
00007080  C3D3C7C4 C2D9404E                          1250  DC CL48'CLGDBR +2.5 M3 modes 1, 3'
000070B0  00000000 00000003                          1251  DC XL16'00000000000000030000000000000003'
000070C0  C3D3C7C4 C2D9404E                          1252  DC CL48'CLGDBR +2.5 M3 modes 4, 5'
000070F0  00000000 00000002                          1253  DC XL16'00000000000000020000000000000002'
00007100  C3D3C7C4 C2D9404E                          1254  DC CL48'CLGDBR +2.5 M3 modes 6, 7'
00007130  00000000 00000003                          1255  DC XL16'00000000000000030000000000000002'
00007140  C3D3C7C4 C2D9404E                          1256  DC CL48'CLGDBR +5.5 FPC modes 1, 2'
00007170  00000000 00000005                          1257  DC XL16'00000000000000050000000000000006'
00007180  C3D3C7C4 C2D9404E                          1258  DC CL48'CLGDBR +5.5 FPC modes 3, 7'
000071B0  00000000 00000005                          1259  DC XL16'00000000000000050000000000000005'
000071C0  C3D3C7C4 C2D9404E                          1260  DC CL48'CLGDBR +5.5 M3 modes 1, 3'
000071F0  00000000 00000006                          1261  DC XL16'00000000000000060000000000000005'
00007200  C3D3C7C4 C2D9404E                          1262  DC CL48'CLGDBR +5.5 M3 modes 4, 5'
00007230  00000000 00000006                          1263  DC XL16'00000000000000060000000000000005'
00007240  C3D3C7C4 C2D9404E                          1264  DC CL48'CLGDBR +5.5 M3 modes 6, 7'
00007270  00000000 00000006                          1265  DC XL16'00000000000000060000000000000005'
00007280  C3D3C7C4 C2D9404E                          1266  DC CL48'CLGDBR +9.5 FPC modes 1, 2'
000072B0  00000000 00000009                          1267  DC XL16'0000000000000009000000000000000A'
000072C0  C3D3C7C4 C2D9404E                          1268  DC CL48'CLGDBR +9.5 FPC modes 3, 7'
000072F0  00000000 00000009                          1269  DC XL16'00000000000000090000000000000009'
00007300  C3D3C7C4 C2D9404E                          1270  DC CL48'CLGDBR +9.5 M3 modes 1, 3'
00007330  00000000 0000000A                          1271  DC XL16'0000000000000000A000000000000009'
00007340  C3D3C7C4 C2D9404E                          1272  DC CL48'CLGDBR +9.5 M3 modes 4, 5'
00007370  00000000 0000000A                          1273  DC XL16'0000000000000000A000000000000009'
00007380  C3D3C7C4 C2D9404E                          1274  DC CL48'CLGDBR +9.5 M3 modes 6, 7'
000073B0  00000000 0000000A                          1275  DC XL16'0000000000000000A000000000000009'
000073C0  C3D3C7C4 C2D94094                          1276  DC CL48'CLGDBR max FPC modes 1, 2'
000073F0  FFFFFFFF FFFFF800                          1277  DC XL16'FFFFFFFFFFFFF800FFFFFFFFFFFFF800'
00007400  C3D3C7C4 C2D94094                          1278  DC CL48'CLGDBR max FPC modes 3, 7'
00007430  FFFFFFFF FFFFF800                          1279  DC XL16'FFFFFFFFFFFFF800FFFFFFFFFFFFF800'
00007440  C3D3C7C4 C2D94094                          1280  DC CL48'CLGDBR max M3 modes 1, 3'
00007470  FFFFFFFF FFFFF800                          1281  DC XL16'FFFFFFFFFFFFF800FFFFFFFFFFFFF800'
00007480  C3D3C7C4 C2D94094                          1282  DC CL48'CLGDBR max M3 modes 4, 5'
000074B0  FFFFFFFF FFFFF800                          1283  DC XL16'FFFFFFFFFFFFF800FFFFFFFFFFFFF800'
000074C0  C3D3C7C4 C2D94094                          1284  DC CL48'CLGDBR max M3 modes 6, 7'
000074F0  FFFFFFFF FFFFF800                          1285  DC XL16'FFFFFFFFFFFFF800FFFFFFFFFFFFF800'
                            00000028  00000001      1286 LINTRMO_NUM EQU (*-LINTRMO_GOOD)/64
                                                     1287 *
                                                     1288 *
                            00007500  00000001      1289 LINTRMOF_GOOD EQU *
00007500  C3D3C7C4 C2D94060                          1290  DC CL48'CLGDBR -1.5 FPC modes 1-3, 7 FPCR'
00007530  00800003 00800003                          1291  DC XL16'00800003008000030080000300800003'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00007540  C3D3C7C4 C2D94060                        1292  DC CL48'CLGDBR -1.5 M3 modes 1, 3-5 FPCR'
00007570  00880003 00880003                        1293  DC XL16'00880003008800030088000300880003'
00007580  C3D3C7C4 C2D94060                        1294  DC CL48'CLGDBR -1.5 M3 modes 6, 7 FPCR'
000075B0  00880003 00880003                        1295  DC XL16'00880003008800030000000000000000'
000075C0  C3D3C7C4 C2D94060                        1296  DC CL48'CLGDBR -0.5 FPC modes 1-3, 7 FPCR'
000075F0  00000001 00000001                        1297  DC XL16'00000001000000010080000300800003'
00007600  C3D3C7C4 C2D94060                        1298  DC CL48'CLGDBR -0.5 M3 modes 1, 3-5 FPCR'
00007630  00880003 00880003                        1299  DC XL16'00880003008800030008000100080001'
00007640  C3D3C7C4 C2D94060                        1300  DC CL48'CLGDBR -0.5 M3 modes 6, 7 FPCR'
00007670  00080001 00880003                        1301  DC XL16'00080001008800030000000000000000'
00007680  C3D3C7C4 C2D9404E                        1302  DC CL48'CLGDBR +0.5 FPC modes 1-3, 7 FPCR'
000076B0  00000002 00000002                        1303  DC XL16'00000002000000020000000200000002'
000076C0  C3D3C7C4 C2D9404E                        1304  DC CL48'CLGDBR +0.5 M3 modes 1, 3-5 FPCR'
000076F0  00080002 00080002                        1305  DC XL16'00080002000800020008000200080002'
00007700  C3D3C7C4 C2D9404E                        1306  DC CL48'CLGDBR +0.5 M3 modes 6, 7 FPCR'
00007730  00080002 00080002                        1307  DC XL16'00080002000800020000000000000000'
00007740  C3D3C7C4 C2D9404E                        1308  DC CL48'CLGDBR +1.5 FPC modes 1-3, 7 FPCR'
00007770  00000002 00000002                        1309  DC XL16'00000002000000020000000200000002'
00007780  C3D3C7C4 C2D9404E                        1310  DC CL48'CLGDBR +1.5 M3 modes 1, 3-5 FPCR'
000077B0  00080002 00080002                        1311  DC XL16'00080002000800020008000200080002'
000077C0  C3D3C7C4 C2D9404E                        1312  DC CL48'CLGDBR +1.5 M3 modes 6, 7 FPCR'
000077F0  00080002 00080002                        1313  DC XL16'00080002000800020000000000000000'
00007800  C3D3C7C4 C2D9404E                        1314  DC CL48'CLGDBR +2.5 FPC modes 1-3, 7 FPCR'
00007830  00000002 00000002                        1315  DC XL16'00000002000000020000000200000002'
00007840  C3D3C7C4 C2D9404E                        1316  DC CL48'CLGDBR +2.5 M3 modes 1, 3-5 FPCR'
00007870  00080002 00080002                        1317  DC XL16'00080002000800020008000200080002'
00007880  C3D3C7C4 C2D9404E                        1318  DC CL48'CLGDBR +2.5 M3 modes 6, 7 FPCR'
000078B0  00080002 00080002                        1319  DC XL16'00080002000800020000000000000000'
000078C0  C3D3C7C4 C2D9404E                        1320  DC CL48'CLGDBR +5.5 FPC modes 1-3, 7 FPCR'
000078F0  00000002 00000002                        1321  DC XL16'00000002000000020000000200000002'
00007900  C3D3C7C4 C2D9404E                        1322  DC CL48'CLGDBR +5.5 M3 modes 1, 3-5 FPCR'
00007930  00080002 00080002                        1323  DC XL16'00080002000800020008000200080002'
00007940  C3D3C7C4 C2D9404E                        1324  DC CL48'CLGDBR +5.5 M3 modes 6, 7 FPCR'
00007970  00080002 00080002                        1325  DC XL16'00080002000800020000000000000000'
00007980  C3D3C7C4 C2D9404E                        1326  DC CL48'CLGDBR +9.5 FPC modes 1-3, 7 FPCR'
000079B0  00000002 00000002                        1327  DC XL16'00000002000000020000000200000002'
000079C0  C3D3C7C4 C2D9404E                        1328  DC CL48'CLGDBR +9.5 M3 modes 1, 3-5 FPCR'
000079F0  00080002 00080002                        1329  DC XL16'00080002000800020008000200080002'
00007A00  C3D3C7C4 C2D9404E                        1330  DC CL48'CLGDBR +9.5 M3 modes 6, 7 FPCR'
00007A30  00080002 00080002                        1331  DC XL16'00080002000800020000000000000000'
00007A40  C3D3C7C4 C2D94094                        1332  DC CL48'CLGDBR max FPC modes 1-3, 7 FPCR'
00007A70  00000002 00000002                        1333  DC XL16'00000002000000020000000200000002'
00007A80  C3D3C7C4 C2D94094                        1334  DC CL48'CLGDBR max M3 modes 1, 3-5 FPCR'
00007AB0  00000002 00000002                        1335  DC XL16'00000002000000020000000200000002'
00007AC0  C3D3C7C4 C2D94094                        1336  DC CL48'CLGDBR max M3 modes 6, 7 FPCR'
00007AF0  00000002 00000002                        1337  DC XL16'00000002000000020000000000000000'
                            00000018  00000001    1338 LINTRMOF_NUM EQU (*-LINTRMOF_GOOD)/64
                                                   1339 *
                                                   1340 *
                            00007B00  00000001    1341 XINTOUT_GOOD EQU *
00007B00  C3D3C7E7 C2D94099                        1342  DC CL48'CLGXBR result pair 1'
00007B30  00000000 00000001                        1343  DC XL16'00000000000000010000000000000001'
00007B40  C3D3C7E7 C2D94099                        1344  DC CL48'CLGXBR result pair 2'
00007B70  00000000 00000002                        1345  DC XL16'00000000000000020000000000000002'
00007B80  C3D3C7E7 C2D94099                        1346  DC CL48'CLGXBR result pair 3'
00007BB0  00000000 00000004                        1347  DC XL16'00000000000000040000000000000004'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00007BC0  C3D3C7E7 C2D94099                        1348  DC CL48'CLGXBR result pair 4'
00007BF0  00000000 00000000                        1349  DC XL16'00000000000000000000000000000000'
00007C00  C3D3C7E7 C2D94099                        1350  DC CL48'CLGXBR result pair 5'
00007C30  00000000 00000000                        1351  DC XL16'00000000000000000000000000000000'
00007C40  C3D3C7E7 C2D94099                        1352  DC CL48'CLGXBR result pair 6'
00007C70  FFFFFFFF FFFFFFFF                        1353  DC XL16'FFFFFFFFFFFFFFFF0000000000000000'
00007C80  C3D3C7E7 C2D94099                        1354  DC CL48'CLGXBR result pair 7'
00007CB0  FFFFFFFF FFFFFFFF                        1355  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00007CC0  C3D3C7E7 C2D94099                        1356  DC CL48'CLGXBR result pair 8'
00007CF0  FFFFFFFF FFFFFFFF                        1357  DC XL16'FFFFFFFFFFFFFFFF0000000000000000'
00007D00  C3D3C7E7 C2D94099                        1358  DC CL48'CLGXBR result pair 9'
00007D30  00000000 00000001                        1359  DC XL16'00000000000000010000000000000001'
00007D40  C3D3C7E7 C2D94099                        1360  DC CL48'CLGXBR result pair 10'
00007D70  00000000 00000000                        1361  DC XL16'00000000000000000000000000000000'
                            0000000A  00000001     1362 XINTOUT_NUM EQU (*-XINTOUT_GOOD)/64
                                                   1363 *
                                                   1364 *
                            00007D80  00000001     1365 XINTFLGS_GOOD EQU *
00007D80  C3D3C7E7 C2D940C6                        1366  DC CL48'CLGXBR FPC pairs 1-2'
00007DB0  00000002 F8000002                        1367  DC XL16'00000002F800000200000002F8000002'
00007DC0  C3D3C7E7 C2D940C6                        1368  DC CL48'CLGXBR FPC pairs 3-4'
00007DF0  00000002 F8000002                        1369  DC XL16'00000002F800000200880003F8008000'
00007E00  C3D3C7E7 C2D940C6                        1370  DC CL48'CLGXBR FPC pairs 5-6'
00007E30  00880003 F8008000                        1371  DC XL16'00880003F800800000880003F8008000'
00007E40  C3D3C7E7 C2D940C6                        1372  DC CL48'CLGXBR FPC pairs 7-8'
00007E70  00000002 F8000002                        1373  DC XL16'00000002F800000200880003F8008000'
00007E80  C3D3C7E7 C2D940C6                        1374  DC CL48'CLGXBR FPC pairs 9-10'
00007EB0  00080002 F8000C02                        1375  DC XL16'00080002F8000C0200080002F8000802'
                            00000005  00000001     1376 XINTFLGS_NUM EQU (*-XINTFLGS_GOOD)/64
                                                   1377 *
                                                   1378 *
                            00007EC0  00000001     1379 XINTRMO_GOOD EQU *
00007EC0  C3D3C7E7 C2D94060                        1380  DC CL48'CLGXBR -1.5 FPC modes 1, 2'
00007EF0  00000000 00000000                        1381  DC XL16'00000000000000000000000000000000'
00007F00  C3D3C7E7 C2D94060                        1382  DC CL48'CLGXBR -1.5 FPC modes 3, 7'
00007F30  00000000 00000000                        1383  DC XL16'00000000000000000000000000000000'
00007F40  C3D3C7E7 C2D94060                        1384  DC CL48'CLGXBR -1.5 M3 modes 1, 3'
00007F70  00000000 00000000                        1385  DC XL16'00000000000000000000000000000000'
00007F80  C3D3C7E7 C2D94060                        1386  DC CL48'CLGXBR -1.5 M3 modes 4, 5'
00007FB0  00000000 00000000                        1387  DC XL16'00000000000000000000000000000000'
00007FC0  C3D3C7E7 C2D94060                        1388  DC CL48'CLGXBR -1.5 M3 modes 6, 7'
00007FF0  00000000 00000000                        1389  DC XL16'00000000000000000000000000000000'
00008000  C3D3C7E7 C2D94060                        1390  DC CL48'CLGXBR -0.5 FPC modes 1, 2'
00008030  00000000 00000000                        1391  DC XL16'00000000000000000000000000000000'
00008040  C3D3C7E7 C2D94060                        1392  DC CL48'CLGXBR -0.5 FPC modes 3, 7'
00008070  00000000 00000000                        1393  DC XL16'00000000000000000000000000000000'
00008080  C3D3C7E7 C2D94060                        1394  DC CL48'CLGXBR -0.5 M3 modes 1, 3'
000080B0  00000000 00000000                        1395  DC XL16'00000000000000000000000000000000'
000080C0  C3D3C7E7 C2D94060                        1396  DC CL48'CLGXBR -0.5 M3 modes 4, 5'
000080F0  00000000 00000000                        1397  DC XL16'00000000000000000000000000000000'
00008100  C3D3C7E7 C2D94060                        1398  DC CL48'CLGXBR -0.5 M3 modes 6, 7'
00008130  00000000 00000000                        1399  DC XL16'00000000000000000000000000000000'
00008140  C3D3C7E7 C2D9404E                        1400  DC CL48'CLGXBR +0.5 FPC modes 1, 2'
00008170  00000000 00000000                        1401  DC XL16'00000000000000000000000000000001'
00008180  C3D3C7E7 C2D9404E                        1402  DC CL48'CLGXBR +0.5 FPC modes 3, 7'
000081B0  00000000 00000000                        1403  DC XL16'00000000000000000000000000000001'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000081C0  C3D3C7E7 C2D9404E                        1404  DC CL48'CLGXBR +0.5 M3 modes 1, 3'
000081F0  00000000 00000001                        1405  DC XL16'00000000000000010000000000000001'
00008200  C3D3C7E7 C2D9404E                        1406  DC CL48'CLGXBR +0.5 M3 modes 4, 5'
00008230  00000000 00000000                        1407  DC XL16'00000000000000000000000000000000'
00008240  C3D3C7E7 C2D9404E                        1408  DC CL48'CLGXBR +0.5 M3 modes 6, 7'
00008270  00000000 00000001                        1409  DC XL16'00000000000000010000000000000000'
00008280  C3D3C7E7 C2D9404E                        1410  DC CL48'CLGXBR +1.5 FPC modes 1, 2'
000082B0  00000000 00000001                        1411  DC XL16'00000000000000010000000000000002'
000082C0  C3D3C7E7 C2D9404E                        1412  DC CL48'CLGXBR +1.5 FPC modes 3, 7'
000082F0  00000000 00000001                        1413  DC XL16'00000000000000010000000000000001'
00008300  C3D3C7E7 C2D9404E                        1414  DC CL48'CLGXBR +1.5 M3 modes 1, 3'
00008330  00000000 00000002                        1415  DC XL16'00000000000000020000000000000001'
00008340  C3D3C7E7 C2D9404E                        1416  DC CL48'CLGXBR +1.5 M3 modes 4, 5'
00008370  00000000 00000002                        1417  DC XL16'00000000000000020000000000000001'
00008380  C3D3C7E7 C2D9404E                        1418  DC CL48'CLGXBR +1.5 M3 modes 6, 7'
000083B0  00000000 00000002                        1419  DC XL16'00000000000000020000000000000001'
000083C0  C3D3C7E7 C2D9404E                        1420  DC CL48'CLGXBR +2.5 FPC modes 1, 2'
000083F0  00000000 00000002                        1421  DC XL16'00000000000000020000000000000003'
00008400  C3D3C7E7 C2D9404E                        1422  DC CL48'CLGXBR +2.5 FPC modes 3, 7'
00008430  00000000 00000002                        1423  DC XL16'00000000000000020000000000000003'
00008440  C3D3C7E7 C2D9404E                        1424  DC CL48'CLGXBR +2.5 M3 modes 1, 3'
00008470  00000000 00000003                        1425  DC XL16'00000000000000030000000000000003'
00008480  C3D3C7E7 C2D9404E                        1426  DC CL48'CLGXBR +2.5 M3 modes 4, 5'
000084B0  00000000 00000002                        1427  DC XL16'00000000000000020000000000000002'
000084C0  C3D3C7E7 C2D9404E                        1428  DC CL48'CLGXBR +2.5 M3 modes 6, 7'
000084F0  00000000 00000003                        1429  DC XL16'00000000000000030000000000000002'
00008500  C3D3C7E7 C2D9404E                        1430  DC CL48'CLGXBR +5.5 FPC modes 1, 2'
00008530  00000000 00000005                        1431  DC XL16'00000000000000050000000000000006'
00008540  C3D3C7E7 C2D9404E                        1432  DC CL48'CLGXBR +5.5 FPC modes 3, 7'
00008570  00000000 00000005                        1433  DC XL16'00000000000000050000000000000005'
00008580  C3D3C7E7 C2D9404E                        1434  DC CL48'CLGXBR +5.5 M3 modes 1, 3'
000085B0  00000000 00000006                        1435  DC XL16'00000000000000060000000000000005'
000085C0  C3D3C7E7 C2D9404E                        1436  DC CL48'CLGXBR +5.5 M3 modes 4, 5'
000085F0  00000000 00000006                        1437  DC XL16'00000000000000060000000000000005'
00008600  C3D3C7E7 C2D9404E                        1438  DC CL48'CLGXBR +5.5 M3 modes 6, 7'
00008630  00000000 00000006                        1439  DC XL16'00000000000000060000000000000005'
00008640  C3D3C7E7 C2D9404E                        1440  DC CL48'CLGXBR +9.5 FPC modes 1, 2'
00008670  00000000 00000009                        1441  DC XL16'0000000000000009000000000000000A'
00008680  C3D3C7E7 C2D9404E                        1442  DC CL48'CLGXBR +9.5 FPC modes 3, 7'
000086B0  00000000 00000009                        1443  DC XL16'00000000000000090000000000000009'
000086C0  C3D3C7E7 C2D9404E                        1444  DC CL48'CLGXBR +9.5 M3 modes 1, 3'
000086F0  00000000 0000000A                        1445  DC XL16'0000000000000000A000000000000009'
00008700  C3D3C7E7 C2D9404E                        1446  DC CL48'CLGXBR +9.5 M3 modes 4, 5'
00008730  00000000 0000000A                        1447  DC XL16'0000000000000000A000000000000009'
00008740  C3D3C7E7 C2D9404E                        1448  DC CL48'CLGXBR +9.5 M3 modes 6, 7'
00008770  00000000 0000000A                        1449  DC XL16'0000000000000000A000000000000009'
00008780  C3D3C7E7 C2D94094                        1450  DC CL48'CLGXBR max+0.5 FPC modes 1, 2'
000087B0  FFFFFFFF FFFFFFFF                        1451  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
000087C0  C3D3C7E7 C2D94094                        1452  DC CL48'CLGXBR max+0.5 FPC modes 3, 7'
000087F0  FFFFFFFF FFFFFFFF                        1453  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00008800  C3D3C7E7 C2D94094                        1454  DC CL48'CLGXBR max+0.5 M3 modes 1, 3'
00008830  FFFFFFFF FFFFFFFF                        1455  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00008840  C3D3C7E7 C2D94094                        1456  DC CL48'CLGXBR max+0.5 M3 modes 4, 5'
00008870  FFFFFFFF FFFFFFFF                        1457  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00008880  C3D3C7E7 C2D94094                        1458  DC CL48'CLGXBR max+0.5 M3 modes 6, 7'
000088B0  FFFFFFFF FFFFFFFF                        1459  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000088C0  C3D3C7E7 C2D9404E                        1460  DC CL48'CLGXBR +0.75 FPC modes 1, 2'
000088F0  00000000 00000000                        1461  DC XL16'00000000000000000000000000000001'
00008900  C3D3C7E7 C2D9404E                        1462  DC CL48'CLGXBR +0.75 FPC modes 3, 7'
00008930  00000000 00000000                        1463  DC XL16'00000000000000000000000000000001'
00008940  C3D3C7E7 C2D9404E                        1464  DC CL48'CLGXBR +0.75 M3 modes 1, 3'
00008970  00000000 00000001                        1465  DC XL16'00000000000000010000000000000001'
00008980  C3D3C7E7 C2D9404E                        1466  DC CL48'CLGXBR +0.75 M3 modes 4, 5'
000089B0  00000000 00000001                        1467  DC XL16'00000000000000010000000000000000'
000089C0  C3D3C7E7 C2D9404E                        1468  DC CL48'CLGXBR +0.75 M3 modes 6, 7'
000089F0  00000000 00000001                        1469  DC XL16'00000000000000010000000000000000'
00008A00  C3D3C7E7 C2D9404E                        1470  DC CL48'CLGXBR +0.25 FPC modes 1, 2'
00008A30  00000000 00000000                        1471  DC XL16'00000000000000000000000000000001'
00008A40  C3D3C7E7 C2D9404E                        1472  DC CL48'CLGXBR +0.25 FPC modes 3, 7'
00008A70  00000000 00000000                        1473  DC XL16'00000000000000000000000000000001'
00008A80  C3D3C7E7 C2D9404E                        1474  DC CL48'CLGXBR +0.25 M3 modes 1, 3'
00008AB0  00000000 00000000                        1475  DC XL16'00000000000000000000000000000001'
00008AC0  C3D3C7E7 C2D9404E                        1476  DC CL48'CLGXBR +0.25 M3 modes 4, 5'
00008AF0  00000000 00000000                        1477  DC XL16'00000000000000000000000000000000'
00008B00  C3D3C7E7 C2D9404E                        1478  DC CL48'CLGXBR +0.25 M3 modes 6, 7'
00008B30  00000000 00000001                        1479  DC XL16'00000000000000010000000000000000'
                            00000032  00000001    1480 XINTRMO_NUM EQU (*-XINTRMO_GOOD)/64
                                                   1481 *
                                                   1482 *
                            00008B40  00000001    1483 XINTRMOF_GOOD EQU *
00008B40  C3D3C7E7 C2D94060                        1484  DC CL48'CLGXBR -1.5 FPC modes 1-3, 7 FPCR'
00008B70  00800003 00800003                        1485  DC XL16'00800003008000030080000300800003'
00008B80  C3D3C7E7 C2D94060                        1486  DC CL48'CLGXBR -1.5 M3 modes 1, 3-5 FPCR'
00008BB0  00880003 00880003                        1487  DC XL16'00880003008800030088000300880003'
00008BC0  C3D3C7E7 C2D94060                        1488  DC CL48'CLGXBR -1.5 M3 modes 6, 7 FPCR'
00008BF0  00880003 00880003                        1489  DC XL16'00880003008800030000000000000000'
00008C00  C3D3C7E7 C2D94060                        1490  DC CL48'CLGXBR -0.5 FPC modes 1-3, 7 FPCR'
00008C30  00000001 00000001                        1491  DC XL16'00000001000000010080000300800003'
00008C40  C3D3C7E7 C2D94060                        1492  DC CL48'CLGXBR -0.5 M3 modes 1, 3-5 FPCR'
00008C70  00880003 00880003                        1493  DC XL16'00880003008800030008000100080001'
00008C80  C3D3C7E7 C2D94060                        1494  DC CL48'CLGXBR -0.5 M3 modes 6, 7 FPCR'
00008CB0  00080001 00880003                        1495  DC XL16'00080001008800030000000000000000'
00008CC0  C3D3C7E7 C2D9404E                        1496  DC CL48'CLGXBR +0.5 FPC modes 1-3, 7 FPCR'
00008CF0  00000002 00000002                        1497  DC XL16'00000002000000020000000200000002'
00008D00  C3D3C7E7 C2D9404E                        1498  DC CL48'CLGXBR +0.5 M3 modes 1, 3-5 FPCR'
00008D30  00080002 00080002                        1499  DC XL16'00080002000800020008000200080002'
00008D40  C3D3C7E7 C2D9404E                        1500  DC CL48'CLGXBR +0.5 M3 modes 6, 7 FPCR'
00008D70  00080002 00080002                        1501  DC XL16'00080002000800020000000000000000'
00008D80  C3D3C7E7 C2D9404E                        1502  DC CL48'CLGXBR +1.5 FPC modes 1-3, 7 FPCR'
00008DB0  00000002 00000002                        1503  DC XL16'00000002000000020000000200000002'
00008DC0  C3D3C7E7 C2D9404E                        1504  DC CL48'CLGXBR +1.5 M3 modes 1, 3-5 FPCR'
00008DF0  00080002 00080002                        1505  DC XL16'00080002000800020008000200080002'
00008E00  C3D3C7E7 C2D9404E                        1506  DC CL48'CLGXBR +1.5 M3 modes 6, 7 FPCR'
00008E30  00080002 00080002                        1507  DC XL16'00080002000800020000000000000000'
00008E40  C3D3C7E7 C2D9404E                        1508  DC CL48'CLGXBR +2.5 FPC modes 1-3, 7 FPCR'
00008E70  00000002 00000002                        1509  DC XL16'00000002000000020000000200000002'
00008E80  C3D3C7E7 C2D9404E                        1510  DC CL48'CLGXBR +2.5 M3 modes 1, 3-5 FPCR'
00008EB0  00080002 00080002                        1511  DC XL16'00080002000800020008000200080002'
00008EC0  C3D3C7E7 C2D9404E                        1512  DC CL48'CLGXBR +2.5 M3 modes 6, 7 FPCR'
00008EF0  00080002 00080002                        1513  DC XL16'00080002000800020000000000000000'
00008F00  C3D3C7E7 C2D9404E                        1514  DC CL48'CLGXBR +5.5 FPC modes 1-3, 7 FPCR'
00008F30  00000002 00000002                        1515  DC XL16'00000002000000020000000200000002'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00008F40  C3D3C7E7 C2D9404E                         1516  DC CL48'CLGXBR +5.5 M3 modes 1, 3-5 FPCR'
00008F70  00080002 00080002                         1517  DC XL16'00080002000800020008000200080002'
00008F80  C3D3C7E7 C2D9404E                         1518  DC CL48'CLGXBR +5.5 M3 modes 6, 7 FPCR'
00008FB0  00080002 00080002                         1519  DC XL16'00080002000800020000000000000000'
00008FC0  C3D3C7E7 C2D9404E                         1520  DC CL48'CLGXBR +9.5 FPC modes 1-3, 7 FPCR'
00008FF0  00000002 00000002                         1521  DC XL16'00000002000000020000000200000002'
00009000  C3D3C7E7 C2D9404E                         1522  DC CL48'CLGXBR +9.5 M3 modes 1, 3-5 FPCR'
00009030  00080002 00080002                         1523  DC XL16'00080002000800020008000200080002'
00009040  C3D3C7E7 C2D9404E                         1524  DC CL48'CLGXBR +9.5 M3 modes 6, 7 FPCR'
00009070  00080002 00080002                         1525  DC XL16'00080002000800020000000000000000'
00009080  C3D3C7E7 C2D94094                         1526  DC CL48'CLGXBR max+0.5 FPC modes 1-3, 7 FPCR'
000090B0  00000002 00800003                         1527  DC XL16'00000002008000030000000200000002'
000090C0  C3D3C7E7 C2D94094                         1528  DC CL48'CLGXBR max+0.5 M3 modes 1, 3-5 FPCR'
000090F0  00880003 00080002                         1529  DC XL16'00880003000800020088000300080002'
00009100  C3D3C7E7 C2D94094                         1530  DC CL48'CLGXBR max+0.5 M3 modes 6, 7 FPCR'
00009130  00880003 00080002                         1531  DC XL16'00880003000800020000000000000000'
00009140  C3D3C7E7 C2D940F0                         1532  DC CL48'CLGXBR 0.75 FPC modes 1-3, 7 FPCR'
00009170  00000002 00000002                         1533  DC XL16'00000002000000020000000200000002'
00009180  C3D3C7E7 C2D940F0                         1534  DC CL48'CLGXBR 0.75 M3 modes 1, 3-5 FPCR'
000091B0  00080002 00080002                         1535  DC XL16'00080002000800020008000200080002'
000091C0  C3D3C7E7 C2D940F0                         1536  DC CL48'CLGXBR 0.75 M3 modes 6, 7 FPCR'
000091F0  00080002 00080002                         1537  DC XL16'00080002000800020000000000000000'
00009200  C3D3C7E7 C2D940F0                         1538  DC CL48'CLGXBR 0.25 FPC modes 1-3, 7 FPCR'
00009230  00000002 00000002                         1539  DC XL16'00000002000000020000000200000002'
00009240  C3D3C7E7 C2D940F0                         1540  DC CL48'CLGXBR 0.25 M3 modes 1, 3-5 FPCR'
00009270  00080002 00080002                         1541  DC XL16'00080002000800020008000200080002'
00009280  C3D3C7E7 C2D940F0                         1542  DC CL48'CLGXBR 0.25 M3 modes 6, 7 FPCR'
000092B0  00080002 00080002                         1543  DC XL16'00080002000800020000000000000000'
                             0000001E  00000001     1544 XINTRMOF_NUM EQU (*-XINTRMOF_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000092C0                                          1546 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                  1548 ************************************************************
                                                  1549 *                 REPORT UNEXPECTED PROGRAM CHECK
                                                  1550 ************************************************************

000092C0                                          1552 PGMCK    DS    0H
000092C0  F342 C072 F08E      00009332  0000008E  1553          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
000092C6  926B C076                     00009336  1554          MVI   PGMCOMMA,C','
000092CA  DC03 C072 C178      00009332  00009438  1555          TR    PROGCODE,HEXTRTAB

000092D0  F384 C07C F150      0000933C  00000150  1557          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
000092D6  9240 C084                     00009344  1558          MVI   PGMPSW+(0*9)+8,C' '
000092DA  DC07 C07C C178      0000933C  00009438  1559          TR    PGMPSW+(0*9)(8),HEXTRTAB

000092E0  F384 C085 F154      00009345  00000154  1561          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
000092E6  9240 C08D                     0000934D  1562          MVI   PGMPSW+(1*9)+8,C' '
000092EA  DC07 C085 C178      00009345  00009438  1563          TR    PGMPSW+(1*9)(8),HEXTRTAB

000092F0  F384 C08E F158      0000934E  00000158  1565          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
000092F6  9240 C096                     00009356  1566          MVI   PGMPSW+(2*9)+8,C' '
000092FA  DC07 C08E C178      0000934E  00009438  1567          TR    PGMPSW+(2*9)(8),HEXTRTAB

00009300  F384 C097 F15C      00009357  0000015C  1569          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
00009306  9240 C09F                     0000935F  1570          MVI   PGMPSW+(3*9)+8,C' '
0000930A  DC07 C097 C178      00009357  00009438  1571          TR    PGMPSW+(3*9)(8),HEXTRTAB

00009310  4100 0042                     00000042  1573          LA    R0,L'PROGMSG     R0 <== length of message
00009314  4110 C05E                     0000931E  1574          LA    R1,PROGMSG       R1 --> the message text itself
00009318  4520 C27A                     0000953A  1575          BAL   R2,MSG           Go display this message
                                                  1576
0000931C  07FD                                    1577          BR    R13              Return to caller




0000931E                                          1579 PROGMSG  DS   0CL66
0000931E  D7D9D6C7 D9C1D440                       1580          DC    CL20'PROGRAM CHECK! CODE '
00009332  88888888                                1581 PROGCODE DC    CL4'hhhh'
00009336  6B                                      1582 PGMCOMMA DC    CL1','
00009337  40D7E2E6 40                             1583          DC    CL5' PSW '
0000933C  88888888 88888888                       1584 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                    1586 ************************************************************************
                                                    1587 *                      VERIFICATION ROUTINE
                                                    1588 ************************************************************************

00009360                                            1590 VERISUB  DS    0H
                                                    1591 *
                                                    1592 **       Loop through the VERIFY TABLE...
                                                    1593 *

00009360  4110 C32C                 000095EC       1595          LA    R1,VERIFTAB      R1 --> Verify table
00009364  4120 000C                 0000000C       1596          LA    R2,VERIFLEN      R2 <== Number of entries
00009368  0D30                                     1597          BASR  R3,0             Set top of loop

0000936A  9846 1000                 00000000       1599          LM    R4,R6,0(R1)      Load verify table values
0000936E  4D70 C0C2                 00009382       1600          BAS   R7,VERIFY        Verify results
00009372  4110 100C                 0000000C       1601          LA    R1,12(,R1)       Next verify table entry
00009376  0623                                     1602          BCTR  R2,R3            Loop through verify table

00009378  9500 C278                 00009538       1604          CLI   FAILFLAG,X'00'   Did all tests verify okay?
0000937C  078D                                     1605          BER   R13              Yes, return to caller
0000937E  47F0 F238                 00000238       1606          B     FAIL             No, load FAILURE disabled wait PSW


                                                    1608 *
                                                    1609 **       Loop through the ACTUAL / EXPECTED results...
                                                    1610 *

00009382  0D80                                     1612 VERIFY   BASR  R8,0             Set top of loop

00009384  D50F 4000 5030      00000000  00000030   1614          CLC   0(16,R4),48(R5)  Actual results == Expected results?
0000938A  4770 C0DA                     0000939A   1615          BNE   VERIFAIL         No, show failure
0000938E  4140 4010                     00000010   1616 VERINEXT LA    R4,16(,R4)       Next actual result
00009392  4150 5040                     00000040   1617          LA    R5,64(,R5)       Next expected result
00009396  0668                                     1618          BCTR  R6,R8            Loop through results

00009398  07F7                                     1620          BR    R7               Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          1622 ********************************************************************
                                          1623 *                     Report the failure...
                                          1624 ********************************************************************

0000939A  9005 C250                 00009510  1626 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
0000939E  92FF C278                 00009538  1627          MVI   FAILFLAG,X'FF'   Remember verification failure
                                          1628 *
                                          1629 **       First, show them the description...
                                          1630 *
000093A2  D22F C1E0 5000      000094A0  00000000  1631          MVC   FAILDESC,0(R5)   Save results/test description
000093A8  4100 0044                 00000044  1632          LA    R0,L'FAILMSG1    R0 <== length of message
000093AC  4110 C1CC                 0000948C  1633          LA    R1,FAILMSG1      R1 --> the message text itself
000093B0  4520 C27A                 0000953A  1634          BAL   R2,MSG           Go display this message
                                          1635 *
                                          1636 **       Save address of actual and expected results
                                          1637 *
000093B4  5040 C24C                 0000950C  1638          ST    R4,AACTUAL       Save A(actual results)
000093B8  4150 5030                 00000030  1639          LA    R5,48(,R5)       R5 ==> expected results
000093BC  5050 C248                 00009508  1640          ST    R5,AEXPECT       Save A(expected results)
                                          1641 *
                                          1642 **       Format and show them the EXPECTED ("Want") results...
                                          1643 *
000093C0  D205 C210 C3C0      000094D0  00009680  1644          MVC   WANTGOT,=CL6'Want: '
000093C6  F384 C216 C248      000094D6  00009508  1645          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
000093CC  9240 C21E                 000094DE  1646          MVI   BLANKEQ,C' '
000093D0  DC07 C216 C178      000094D6  00009438  1647          TR    FAILADR,HEXTRTAB

000093D6  F384 C221 5000      000094E1  00000000  1649          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
000093DC  9240 C229                 000094E9  1650          MVI   FAILVALS+(0*9)+8,C' '
000093E0  DC07 C221 C178      000094E1  00009438  1651          TR    FAILVALS+(0*9)(8),HEXTRTAB

000093E6  F384 C22A 5004      000094EA  00000004  1653          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
000093EC  9240 C232                 000094F2  1654          MVI   FAILVALS+(1*9)+8,C' '
000093F0  DC07 C22A C178      000094EA  00009438  1655          TR    FAILVALS+(1*9)(8),HEXTRTAB

000093F6  F384 C233 5008      000094F3  00000008  1657          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
000093FC  9240 C23B                 000094FB  1658          MVI   FAILVALS+(2*9)+8,C' '
00009400  DC07 C233 C178      000094F3  00009438  1659          TR    FAILVALS+(2*9)(8),HEXTRTAB

00009406  F384 C23C 500C      000094FC  0000000C  1661          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
0000940C  9240 C244                 00009504  1662          MVI   FAILVALS+(3*9)+8,C' '
00009410  DC07 C23C C178      000094FC  00009438  1663          TR    FAILVALS+(3*9)(8),HEXTRTAB

00009416  4100 0035                 00000035  1665          LA    R0,L'FAILMSG2    R0 <== length of message
0000941A  4110 C210                 000094D0  1666          LA    R1,FAILMSG2      R1 --> the message text itself
0000941E  4520 C27A                 0000953A  1667          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                          1669 *
                                          1670 **       Format and show them the ACTUAL ("Got") results...
                                          1671 *
00009422  D205 C210 C3C6      000094D0  00009686  1672          MVC   WANTGOT,=CL6'Got:  '
00009428  F384 C216 C24C      000094D6  0000950C  1673          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
0000942E  9240 C21E                     000094DE  1674          MVI   BLANKEQ,C' '
00009432  DC07 C216 C178      000094D6  00009438  1675          TR    FAILADR,HEXTRTAB

00009438  F384 C221 4000      000094E1  00000000  1677          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
0000943E  9240 C229                     000094E9  1678          MVI   FAILVALS+(0*9)+8,C' '
00009442  DC07 C221 C178      000094E1  00009438  1679          TR    FAILVALS+(0*9)(8),HEXTRTAB

00009448  F384 C22A 4004      000094EA  00000004  1681          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
0000944E  9240 C232                     000094F2  1682          MVI   FAILVALS+(1*9)+8,C' '
00009452  DC07 C22A C178      000094EA  00009438  1683          TR    FAILVALS+(1*9)(8),HEXTRTAB

00009458  F384 C233 4008      000094F3  00000008  1685          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
0000945E  9240 C23B                     000094FB  1686          MVI   FAILVALS+(2*9)+8,C' '
00009462  DC07 C233 C178      000094F3  00009438  1687          TR    FAILVALS+(2*9)(8),HEXTRTAB

00009468  F384 C23C 400C      000094FC  0000000C  1689          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
0000946E  9240 C244                     00009504  1690          MVI   FAILVALS+(3*9)+8,C' '
00009472  DC07 C23C C178      000094FC  00009438  1691          TR    FAILVALS+(3*9)(8),HEXTRTAB

00009478  4100 0035                     00000035  1693          LA    R0,L'FAILMSG2    R0 <== length of message
0000947C  4110 C210                     000094D0  1694          LA    R1,FAILMSG2      R1 --> the message text itself
00009480  4520 C27A                     0000953A  1695          BAL   R2,MSG           Go display this message

00009484  9805 C250                     00009510  1697          LM    R0,R5,SAVER0R5   Restore registers
00009488  47F0 C0CE                     0000938E  1698          B     VERINEXT         Continue with verification...



0000948C                                          1700 FAILMSG1 DS    0CL68
0000948C  C3D6D4D7 C1D9C9E2                       1701          DC    CL20'COMPARISON FAILURE! '
000094A0  4D8485A2 83998997                       1702 FAILDESC DC    CL48'(description)'



000094D0                                          1704 FAILMSG2 DS    0CL53
000094D0  40404040 4040                           1705 WANTGOT  DC    CL6' '           'Want: ' -or- 'Got:  '
000094D6  C1C1C1C1 C1C1C1C1                       1706 FAILADR  DC    CL8'AAAAAAAA'
000094DE  407E40                                  1707 BLANKEQ  DC    CL3' = '
000094E1  88888888 88888888                       1708 FAILVALS DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '

00009508  00000000                                1710 AEXPECT  DC    F'0'             ==> Expected ("Want") results
0000950C  00000000                                1711 AACTUAL  DC    F'0'             ==> Actual ("Got") results
00009510  00000000 00000000                       1712 SAVER0R5 DC    6F'0'            Registers R0 - R5 save area
00009528  F0F1F2F3 F4F5F6F7                       1713 CHARHEX  DC    CL16'0123456789ABCDEF'
                             00009438  00000010  1714 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
00009538  00                                      1715 FAILFLAG DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                          1717 ********************************************************************
                                          1718 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                          1719 ********************************************************************

0000953A  4900 C3BC                 0000967C  1721 MSG      CH    R0,=H'0'                Do we even HAVE a message?
0000953E  07D2                              1722          BNHR  R2                      No, ignore

00009540  9002 C2B0                 00009570  1724          STM   R0,R2,MSGSAVE           Save registers

00009544  4900 C3BE                 0000967E  1726          CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00009548  47D0 C290                 00009550  1727          BNH   MSGOK                   Yes, continue
0000954C  4100 005F                 0000005F  1728          LA    R0,L'MSGMSG             No, set to maximum

00009550  1820                              1730 MSGOK    LR    R2,R0                   Copy length to work register
00009552  0620                              1731          BCTR  R2,0                    Minus-1 for execute
00009554  4420 C2BC                 0000957C  1732          EX    R2,MSGMVC               Copy message to O/P buffer

00009558  4120 200A                 0000000A  1734          LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
0000955C  4110 C2C2                 00009582  1735          LA    R1,MSGCMD               Point to true command

00009560  83120008                          1737          DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
00009564  4780 C2AA                 0000956A  1738          BZ    MSGRET                  Return if successful
00009568  0000                              1739          DC    H'0'                    CRASH for debugging purposes

0000956A  9802 C2B0                 00009570  1741 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
0000956E  07F2                              1742          BR    R2                      Return to caller
```

```
00009570  00000000 00000000                 1744 MSGSAVE  DC    3F'0'                   Registers save area
0000957C  D200 C2CB 1000      0000958B  00000000  1745 MSGMVC   MVC   MSGMSG(0),0(R1)      Executed instruction

00009582  D4E2C7D5 D6C8405C                 1747 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
0000958B  40404040 40404040                 1748 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   1750 ********************************************************************************
                                                   1751 *                            VERIFY TABLE
                                                   1752 ********************************************************************************
                                                   1753 *
                                                   1754 *        A(actual results), A(expected results), A(#of results)
                                                   1755 *
                                                   1756 ********************************************************************************

000095EC                                           1758 VERIFTAB DC    0F'0'
000095EC  00001000                                 1759          DC    A(SINTOUT)
000095F0  00005000                                 1760          DC    A(SINTOUT_GOOD)
000095F4  00000009                                 1761          DC    A(SINTOUT_NUM)
                                                   1762 *
000095F8  00001200                                 1763          DC    A(SINTFLGS)
000095FC  00005240                                 1764          DC    A(SINTFLGS_GOOD)
00009600  00000005                                 1765          DC    A(SINTFLGS_NUM)
                                                   1766 *
00009604  00001300                                 1767          DC    A(SINTRMO)
00009608  00005380                                 1768          DC    A(SINTRMO_GOOD)
0000960C  00000032                                 1769          DC    A(SINTRMO_NUM)
                                                   1770 *
00009610  00001800                                 1771          DC    A(SINTRMOF)
00009614  00006000                                 1772          DC    A(SINTRMOF_GOOD)
00009618  0000001E                                 1773          DC    A(SINTRMOF_NUM)
                                                   1774 *
0000961C  00002000                                 1775          DC    A(LINTOUT)
00009620  00006780                                 1776          DC    A(LINTOUT_GOOD)
00009624  00000009                                 1777          DC    A(LINTOUT_NUM)
                                                   1778 *
00009628  00002200                                 1779          DC    A(LINTFLGS)
0000962C  000069C0                                 1780          DC    A(LINTFLGS_GOOD)
00009630  00000005                                 1781          DC    A(LINTFLGS_NUM)
                                                   1782 *
00009634  00002300                                 1783          DC    A(LINTRMO)
00009638  00006B00                                 1784          DC    A(LINTRMO_GOOD)
0000963C  00000028                                 1785          DC    A(LINTRMO_NUM)
                                                   1786 *
00009640  00002800                                 1787          DC    A(LINTRMOF)
00009644  00007500                                 1788          DC    A(LINTRMOF_GOOD)
00009648  00000018                                 1789          DC    A(LINTRMOF_NUM)
                                                   1790 *
0000964C  00003000                                 1791          DC    A(XINTOUT)
00009650  00007B00                                 1792          DC    A(XINTOUT_GOOD)
00009654  0000000A                                 1793          DC    A(XINTOUT_NUM)
                                                   1794 *
00009658  00003200                                 1795          DC    A(XINTFLGS)
0000965C  00007D80                                 1796          DC    A(XINTFLGS_GOOD)
00009660  00000005                                 1797          DC    A(XINTFLGS_NUM)
                                                   1798 *
00009664  00003300                                 1799          DC    A(XINTRMO)
00009668  00007EC0                                 1800          DC    A(XINTRMO_GOOD)
0000966C  00000032                                 1801          DC    A(XINTRMO_NUM)
                                                   1802 *
00009670  00003800                                 1803          DC    A(XINTRMOF)
00009674  00008B40                                 1804          DC    A(XINTRMOF_GOOD)
00009678  0000001E                                 1805          DC    A(XINTRMOF_NUM)
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                     1806 *
                              0000000C  00000001   1807 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

0000967C                                            1809          END
0000967C  0000                                      1810                =H'0'
0000967E  005F                                      1811                =AL2(L'MSGMSG)
00009680  E68195A3 7A40                             1812                =CL6'Want: '
00009686  C796A37A 4040                             1813                =CL6'Got:  '
```

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

AACTUAL              F    00950C       4  1711  1638  1673
AEXPECT              F    009508       4  1710  1640  1645
AHELPERS             A    00027C       4   210   200   247
BFPCVTTL             J    000000   38540   127   177   180   182   185   193   929   931   933   935   938   940   942   944   947
                                                 949   951   953   960
BLANKEQ              C    0094DE       3  1707  1646  1674
CHARHEX              C    009528      16  1713  1714
CLGDBR               I    00051E       4   474   232
CLGDBRA              I    000582       4   524   234
CLGEBR               I    00035C       4   313   225
CLGEBRA              I    0003C0       4   363   227
CLGXBR               I    0006E0       4   635   239
CLGXBRA              I    000748       4   686   241
CTLR0                F    0002F0       4   257   218   219   220
EXTDS                F    00031C       4   279   238
FAIL                 I    000238       4   208  1606
FAILADR              C    0094D6       8  1706  1645  1647  1673  1675
FAILDESC             C    0094A0      48  1702  1631
FAILFLAG             X    009538       1  1715  1604  1627
FAILMSG1             C    00948C      68  1700  1632  1633
FAILMSG2             C    0094D0      53  1704  1665  1666  1693  1694
FAILPSW              X    0002E0       8   255   208
FAILVALS             C    0094E1      36  1708  1649  1650  1651  1653  1654  1655  1657  1658  1659  1661  1662  1663  1677  1678
                                                 1679  1681  1682  1683  1685  1686  1687  1689  1690  1691
FPCREGNT             X    0002F4       4   258   320   373   382   391   400   411   419   427   435   443   451   481   534   543
                                                 552   561   572   580   588   596   604   612   643   697   706   715   724   735
                                                 743   751   759   767   775
FPCREGTR             X    0002F8       4   259   328   489   651
FPR0                 U    000000       1   147   319   321   331   369   375   384   393   402   412   420   428   436   444   452
                                                 480   482   492   530   536   545   554   563   573   581   589   597   605   613
                                                 641   644   654
FPR1                 U    000001       1   148
FPR10                U    00000A       1   157
FPR11                U    00000B       1   158
FPR12                U    00000C       1   159
FPR13                U    00000D       1   160
FPR14                U    00000E       1   161
FPR15                U    00000F       1   162
FPR2                 U    000002       1   149
FPR3                 U    000003       1   150
FPR4                 U    000004       1   151
FPR5                 U    000005       1   152
FPR6                 U    000006       1   153
FPR7                 U    000007       1   154
FPR8                 U    000008       1   155
FPR9                 U    000009       1   156
GOODPSW              X    0002D0       8   254   251
HELPERS              H    0092C0       2  1546   165   210
HEXTRTAB             U    009438      16  1714  1555  1559  1563  1567  1571  1647  1651  1655  1659  1663  1675  1679  1683  1687
                                                 1691
IMAGE                1    000000   38540     0
LBFPCT               U    000048       1   855   274
LBFPIN               F    0008F8       4   843   855   275
LBFPINRM             F    000940       4   858   878   293
LBFPRMCT             U    000048       1   878   292
LINTFLGS             U    002200       0   940   277  1779
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
LINTFLGS_GOOD	U	0069C0	1	1191	1202 1780
LINTFLGS_NUM	U	000005	1	1202	1781
LINTOUT	U	002000	0	938	276 1775
LINTOUT_GOOD	U	006780	1	1169	1188 1776
LINTOUT_NUM	U	000009	1	1188	1777
LINTRMO	U	002300	0	942	294 1783
LINTRMOF	U	002800	0	944	295 1787
LINTRMOF_GOOD	U	007500	1	1289	1338 1788
LINTRMOF_NUM	U	000018	1	1338	1789
LINTRMO_GOOD	U	006B00	1	1205	1286 1784
LINTRMO_NUM	U	000028	1	1286	1785
LONGS	F	00030C	4	273	231
MSG	I	00953A	4	1721	1575 1634 1667 1695
MSGCMD	C	009582	9	1747	1734 1735
MSGMSG	C	00958B	95	1748	1728 1745 1726
MSGMVC	I	00957C	6	1745	1732
MSGOK	I	009550	2	1730	1727
MSGRET	I	00956A	4	1741	1738
MSGSAVE	F	009570	4	1744	1724 1741
PCINTCD	H	00008E	2	178	195 1553
PCNOTDTA	I	00020C	4	199	196
PCOLDPSW	U	000150	0	180	197 1557 1561 1565 1569
PGMCK	H	0092C0	2	1552	201
PGMCOMMA	C	009336	1	1582	1554
PGMPSW	C	00933C	36	1584	1557 1558 1559 1561 1562 1563 1565 1566 1567 1569 1570 1571
PROGCHK	H	000200	2	194	186
PROGCODE	C	009332	4	1581	1553 1555
PROGMSG	C	00931E	66	1579	1573 1574
PROGPSW	D	000228	8	207	206
R0	U	000000	1	128	199 202 218 220 692 699 708 717 726 736 744 752 760 768 776 1573 1626 1632 1665 1693 1697 1721 1724 1726 1728 1730 1741
R1	U	000001	1	129	321 322 324 325 326 329 330 331 332 334 335 336 375 376 378 379 380 384 385 387 388 389 393 394 396 397 398 402 403 405 406 407 412 413 415 416 417 420 421 423 424 425 428 429 431 432 433 436 437 439 440 441 444 445 447 448 449 452 453 455 456 457 482 483 485 486 487 490 491 492 493 495 496 497 536 537 539 540 541 545 546 548 549 550 554 555 557 558 559 563 564 566 567 568 573 574 576 577 578 581 582 584 585 586 589 590 592 593 594 597 598 600 601 602 605 606 608 609 610 613 614 616 617 618 644 645 647 648 649 652 653 654 655 657 658 659 699 700 702 703 704 708 709 711 712 713 717 718 720 721 722 726 727 729 730 731 736 737 739 740 741 744 745 747 748 749 752 753 755 756 757 760 761 763 764 765 768 769 771 772 773 776 777 779 780 781 1574 1595 1599 1601 1633 1666 1694 1735 1745
R10	U	00000A	1	138	224 226 231 233 238 240 313 314 363 364 474 475 524 525 635 636 686 687
R11	U	00000B	1	139	
R12	U	00000C	1	140	165 200 247 317 341 367 462 478 502 528 623 639 664 690 786
R13	U	00000D	1	141	201 225 227 232 234 239 241 248 316 342 366 463 477 503 527 624 638 665 689 787 1577 1605
R14	U	00000E	1	142	204 205 249 250
R15	U	00000F	1	143	164 199 202
R2	U	000002	1	130	313 315 341 363 365 462 474 476 502 524 526 623 635 637 642 664 686 688 693 786 1575 1596 1602 1634 1667 1695 1722 1724

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
					1730 1731 1732 1734 1741 1742
R3	U	000003	1	131	313 319 338 363 369 459 474 480 499 524 530 620 635 641
					642 661 686 692 693 783 1597 1602
R4	U	000004	1	132	1599 1614 1616 1638 1677 1681 1685 1689
R5	U	000005	1	133	1614 1617 1626 1631 1639 1640 1649 1653 1657 1661 1697
R6	U	000006	1	134	1599 1618
R7	U	000007	1	135	314 322 332 339 364 376 385 394 403 413 421 429 437 445
					453 460 475 483 493 500 525 537 546 555 564 574 582 590
					598 606 614 621 636 645 655 662 687 700 709 718 727 737
					745 753 761 769 777 784 1600 1620
R8	U	000008	1	136	314 323 326 333 336 340 364 377 380 386 389 395 398 404
					407 414 417 422 425 430 433 438 441 446 449 454 457 461
					475 484 487 494 497 501 525 538 541 547 550 556 559 565
					568 575 578 583 586 591 594 599 602 607 610 615 618 622
					636 646 649 656 659 663 687 701 704 710 713 719 722 728
					731 738 741 746 749 754 757 762 765 770 773 778 781 785
					1612 1618
R9	U	000009	1	137	
RMEXTDS	F	00034C	4	297	240
RMLONGS	F	00033C	4	291	233
RMSHORTS	F	00032C	4	285	226
SAVER0R5	F	009510	4	1712	1626 1697
SAVEREGS	F	00023C	4	209	199 202
SBFPCT	U	000024	1	817	268
SBFPIN	F	0008AC	4	804	817 269
SBFPINRM	F	0008D0	4	820	840 287
SBFPRMCT	U	000028	1	840	286
SHORTS	F	0002FC	4	267	224
SINTFLGS	U	001200	0	931	271 1763
SINTFLGS_GOOD	U	005240	1	987	998 1764
SINTFLGS_NUM	U	000005	1	998	1765
SINTOUT	U	001000	0	929	270 1759
SINTOUT_GOOD	U	005000	1	965	984 1760
SINTOUT_NUM	U	000009	1	984	1761
SINTRMO	U	001300	0	933	288 1767
SINTRMOF	U	001800	0	935	289 1771
SINTRMOF_GOOD	U	006000	1	1105	1166 1772
SINTRMOF_NUM	U	00001E	1	1166	1773
SINTRMO_GOOD	U	005380	1	1001	1102 1768
SINTRMO_NUM	U	000032	1	1102	1769
START	I	000280	4	218	183
VERIFAIL	I	00939A	4	1626	1615
VERIFLEN	U	00000C	1	1807	1596
VERIFTAB	F	0095EC	4	1758	1807 1595
VERIFY	I	009382	2	1612	1600
VERINEXT	I	00938E	4	1616	1698
VERISUB	H	009360	2	1590	248
WANTGOT	C	0094D0	6	1705	1644 1672
XBFPCT	U	0000A0	1	896	280
XBFPIN	D	000988	8	881	896 281
XBFPINRM	D	000A28	8	899	920 299
XBFPRMCT	U	0000A0	1	920	298
XINTFLGS	U	003200	0	949	283 1795
XINTFLGS_GOOD	U	007D80	1	1365	1376 1796
XINTFLGS_NUM	U	000005	1	1376	1797
XINTOUT	U	003000	0	947	282 1791

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

XINTOUT_GOOD         U    007B00       1  1341  1362  1792
XINTOUT_NUM          U    00000A       1  1362  1793
XINTRMO              U    003300       0   951   300  1799
XINTRMOF             U    003800       0   953   301  1803
XINTRMOF_GOOD        U    008B40       1  1483  1544  1804
XINTRMOF_NUM         U    00001E       1  1544  1805
XINTRMO_GOOD         U    007EC0       1  1379  1480  1800
XINTRMO_NUM          U    000032       1  1480  1801
=AL2(L'MSGMSG)       R    00967E       2  1811  1726
=CL6'Got:  '         C    009686       6  1813  1672
=CL6'Want: '         C    009680       6  1812  1644
=H'0'                H    00967C       2  1810  1721
```

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 0

Image       IMAGE      38540  0000-968B  0000-968B
  Region               38540  0000-968B  0000-968B
    CSECT   BFPCVTTL   38540  0000-968B  0000-968B
```

```
   STMT                                                    FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-005-cvttolog64\bfp-005-cvttolog64.asm

** NO ERRORS FOUND **
```