```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     2 ************************************************************************
                                                     3 *
                                                     4 *Testcase IEEE CONVERT TO LOGICAL 32
                                                     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-004-cvttolog.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 2022-03-08
                                                    31 * PADCSECT macro/usage removed by Fish on 2022-03-08
                                                    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-32, RRF-e)
                                                        67 *   CONVERT TO LOGICAL (long BFP to uint-32, RRF-e)
                                                        68 *   CONVERT TO LOGICAL (extended BFP to uint-32, 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-32
                                                        76 * 2) Short BFP to uint-32 with all rounding modes
                                                        77 * 3) Long BFP uint-32
                                                        78 * 3) Long BFP uint-32 with all rounding modes
                                                        79 * 4) Extended BFP to uint-32
                                                        80 * 4) Extended BFP to uint-32 with all rounding modes
                                                        81 *
                                                        82 * Three input test data sets are provided, one each for short, long,
                                                        83 *   and extended precision BFP.  All are converted to uint-32.
                                                        84 *
                                                        85 * Provided test data is 1, 2, 4, 9, QNaN, SNaN, 4294967295.5.
                                                        86 *   The last three values will trigger inexact exceptions when
                                                        87 *   converted to uint-32.  The last value is present only in the long
                                                        88 *   and extended BFP test cases and should overflow a uint-32.
                                                        89 * Provided test data for rounding tests is taken from Table 9-11 on
                                                        90 *   page 9-16 of SA22-7832-10.
                                                        91 *        -1.5, -0.5, +0.5, +1.5, +2.5, +5.5, +9.5.
                                                        92 *   While the table illustrates LOAD FP INTEGER, the same results
                                                        93 *   should be generated when creating a uint-32 or uint-64 from BFP.
                                                        94 * For long BFP and extended BFP rounding mode tests, ond additional
                                                        95 *   test case is included: 4294967294.5.  This case rounds down to a
                                                        96 *   maximum uint-32 and rounds up to overflow; it tests the case where
                                                        97 *   the input is greater than a maximum uint-32 but rounds to a maximum
                                                        98 *   uint-32.  See Table 19-19 on page 19-26 of SA22-7832-10 for details
                                                        99 *   on this boundary condition test.
                                                       100 *
                                                       101 * Also tests the following floating point support instructions
                                                       102 *   LOAD  (Short)
                                                       103 *   LOAD  (Long)
                                                       104 *   LOAD FPC
                                                       105 *   SRNMB (Set BFP Rounding Mode 2-bit)
                                                       106 *   SRNMB (Set BFP Rounding Mode 3-bit)
                                                       107 *   STORE (Short)
                                                       108 *   STORE (Long)
                                                       109 *   STORE FPC
                                                       110 *
                                                       111 *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     112 ************************************************************************
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                      114 *
                                                      115 *  Note: for compatibility with the z/CMS test rig, do not change
                                                      116 *  or use R11, R14, or R15.  Everything else is fair game.
                                                      117 *
                                00000000  0000784B    118 BFPCVTTL START 0
                                00000000  00000001    119 R0       EQU   0
                                00000001  00000001    120 R1       EQU   1
                                00000002  00000001    121 R2       EQU   2
                                00000003  00000001    122 R3       EQU   3
                                00000004  00000001    123 R4       EQU   4
                                00000005  00000001    124 R5       EQU   5
                                00000006  00000001    125 R6       EQU   6
                                00000007  00000001    126 R7       EQU   7
                                00000008  00000001    127 R8       EQU   8
                                00000009  00000001    128 R9       EQU   9
                                0000000A  00000001    129 R10      EQU   10
                                0000000B  00000001    130 R11      EQU   11
                                0000000C  00000001    131 R12      EQU   12
                                0000000D  00000001    132 R13      EQU   13
                                0000000E  00000001    133 R14      EQU   14
                                0000000F  00000001    134 R15      EQU   15
                                                      135 *
                                                      136 * Floating Point Register equates to keep the cross reference clean
                                                      137 *
                                00000000  00000001    138 FPR0     EQU   0
                                00000001  00000001    139 FPR1     EQU   1
                                00000002  00000001    140 FPR2     EQU   2
                                00000003  00000001    141 FPR3     EQU   3
                                00000004  00000001    142 FPR4     EQU   4
                                00000005  00000001    143 FPR5     EQU   5
                                00000006  00000001    144 FPR6     EQU   6
                                00000007  00000001    145 FPR7     EQU   7
                                00000008  00000001    146 FPR8     EQU   8
                                00000009  00000001    147 FPR9     EQU   9
                                0000000A  00000001    148 FPR10    EQU   10
                                0000000B  00000001    149 FPR11    EQU   11
                                0000000C  00000001    150 FPR12    EQU   12
                                0000000D  00000001    151 FPR13    EQU   13
                                0000000E  00000001    152 FPR14    EQU   14
                                0000000F  00000001    153 FPR15    EQU   15
                                                      154 *
00000000                        00000000              155          USING *,R15
00000000                        00007480              156          USING HELPERS,R12
                                                      157 *
                                                      158 * Above works on real iron (R15=0 after sysclear)
                                                      159 * and in z/CMS (R15 points to start of load module)
                                                      160 *


                                                      162 ************************************************************************
                                                      163 *
                                                      164 * Low core definitions, Restart PSW, and Program Check Routine.
                                                      165 *
                                                      166 ************************************************************************
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

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

0000020C  900F F23C                      0000023C  190 PCNOTDTA STM   R0,R15,SAVEREGS  Save registers
00000210  58C0 F27C                      0000027C  191          L     R12,AHELPERS     Get address of helper subroutines
00000214  4DD0 C000                      00007480  192          BAS   R13,PGMCK        Report this unexpected program check
00000218  980F F23C                      0000023C  193          LM    R0,R15,SAVEREGS  Restore registers

0000021C  12EE                                     195          LTR   R14,R14        Return address provided?
0000021E  077E                                     196          BNZR  R14            Yes, return to z/CMS test rig.
00000220  B2B2 F228                      00000228  197          LPSWE PROGPSW        Not data exception, enter disabled wait
00000228  00020000 00000000                        198 PROGPSW  DC    0D'0',X'0002000000000000',XL6'00',X'DEAD' Abnormal end
00000238  B2B2 F2E0                      000002E0  199 FAIL     LPSWE FAILPSW        Not data exception, enter disabled wait
0000023C  00000000 00000000                        200 SAVEREGS DC    16F'0'          Registers save area
0000027C  00007480                                 201 AHELPERS DC    A(HELPERS)      Address of helper subroutines
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    203 ************************************************************************
                                                    204 *
                                                    205 *  Main program.  Enable Advanced Floating Point, process test cases.
                                                    206 *
                                                    207 ************************************************************************


00000280  B600 F2F0                    000002F0    209 START    STCTL R0,R0,CTLR0    Store CR0 to enable AFP
00000284  9604 F2F1                    000002F1    210          OI    CTLR0+1,X'04'  Turn on AFP bit
00000288  B700 F2F0                    000002F0    211          LCTL  R0,R0,CTLR0    Reload updated CR0
                                                    212 *
                                                    213 * Short BFP Input testing
                                                    214 *
0000028C  41A0 F2FC                    000002FC    215          LA    R10,SHORTS     Point to short BFP test inputs
00000290  4DD0 F35C                    0000035C    216          BAS   R13,CLFEBR     Convert values to uint-32 from short BFP
00000294  41A0 F32C                    0000032C    217          LA    R10,RMSHORTS   Point to inputs for rounding mode tests
00000298  4DD0 F3BA                    000003BA    218          BAS   R13,CLFEBRA    Convert using all rounding mode options
                                                    219 *
                                                    220 * Short BFP Input testing
                                                    221 *
0000029C  41A0 F30C                    0000030C    222          LA    R10,LONGS      Point to long BFP test inputs
000002A0  4DD0 F504                    00000504    223          BAS   R13,CLFDBR     Convert values to uint-32 from long BFP
000002A4  41A0 F33C                    0000033C    224          LA    R10,RMLONGS    Point to inputs for rounding mode tests
000002A8  4DD0 F562                    00000562    225          BAS   R13,CLFDBRA    Convert using all rounding mode options
                                                    226 *
                                                    227 * Short BFP Input testing
                                                    228 *
000002AC  41A0 F31C                    0000031C    229          LA    R10,EXTDS      Point to extended BFP test inputs
000002B0  4DD0 F6AC                    000006AC    230          BAS   R13,CLFXBR     Convert values to uint-32 from extended
000002B4  41A0 F34C                    0000034C    231          LA    R10,RMEXTDS    Point to inputs for rounding mode tests
000002B8  4DD0 F70E                    0000070E    232          BAS   R13,CLFXBRA    Convert using all rounding mode options
                                                    233 *
                                                    234 ************************************************************************
                                                    235 *                    Verify test results...
                                                    236 ************************************************************************
                                                    237 *
000002BC  58C0 F27C                    0000027C    238          L     R12,AHELPERS     Get address of helper subroutines
000002C0  4DD0 C0A0                    00007520    239          BAS   R13,VERISUB      Go verify results
000002C4  12EE                                     240          LTR   R14,R14          Was return address provided?
000002C6  077E                                     241          BNZR  R14              Yes, return to z/CMS test rig.
000002C8  B2B2 F2D0                    000002D0    242          LPSWE GOODPSW          Load SUCCESS PSW
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000002D0                                            244          DS    0D             Ensure correct alignment for PSW
000002D0  00020000 00000000                         245 GOODPSW  DC    X'0002000000000000',AD(0)  Normal end - disabled wait
000002E0  00020000 00000000                         246 FAILPSW  DC    X'0002000000000000',XL6'00',X'0BAD' Abnormal end
                                                    247 *
000002F0  00000000                                  248 CTLR0    DS    F
000002F4  00000000                                  249 FPCREGNT DC    X'00000000'  FPCR, trap all IEEE exceptions, zero flags
000002F8  F8000000                                  250 FPCREGTR DC    X'F8000000'  FPCR, trap no IEEE exceptions, zero flags
                                                    251 *
                                                    252 * Input values parameter list, four fullwords:
                                                    253 *      1) Count,
                                                    254 *      2) Address of inputs,
                                                    255 *      3) Address to place results, and
                                                    256 *      4) Address to place DXC/Flags/cc values.
                                                    257 *
000002FC                                            258 SHORTS   DS    0F           Inputs for short BFP testing
000002FC  00000009                                  259          DC    A(SBFPCT/4)
00000300  0000085C                                  260          DC    A(SBFPIN)
00000304  00001000                                  261          DC    A(SINTOUT)
00000308  00001100                                  262          DC    A(SINTFLGS)
                                                    263 *
0000030C                                            264 LONGS    DS    0F           Inputs for long BFP testing
0000030C  00000009                                  265          DC    A(LBFPCT/8)
00000310  000008A8                                  266          DC    A(LBFPIN)
00000314  00002000                                  267          DC    A(LINTOUT)
00000318  00002100                                  268          DC    A(LINTFLGS)
                                                    269 *
0000031C                                            270 EXTDS    DS    0F           Inputs for Extended BFP testing
0000031C  00000009                                  271          DC    A(XBFPCT/16)
00000320  00000940                                  272          DC    A(XBFPIN)
00000324  00003000                                  273          DC    A(XINTOUT)
00000328  00003100                                  274          DC    A(XINTFLGS)
                                                    275 *
0000032C  0000000A                                  276 RMSHORTS DC    A(SBFPRMCT/4)
00000330  00000880                                  277          DC    A(SBFPINRM)  Short BFP rounding mode test inputs
00000334  00001200                                  278          DC    A(SINTRMO)   Short BFP rounding mode test results
00000338  00001600                                  279          DC    A(SINTRMOF)  Short BFP rounding mode test flags
                                                    280 *
0000033C  0000000A                                  281 RMLONGS  DC    A(LBFPRMCT/8)
00000340  000008F0                                  282          DC    A(LBFPINRM)  Long BFP rounding mode test inputs
00000344  00002200                                  283          DC    A(LINTRMO)   Long BFP rounding mode test results
00000348  00002600                                  284          DC    A(LINTRMOF)  Long BFP rounding mode test flags
                                                    285 *
0000034C  0000000A                                  286 RMEXTDS  DC    A(XBFPRMCT/16)
00000350  000009D0                                  287          DC    A(XBFPINRM)  Extended BFP rounding mode test inputs
00000354  00003200                                  288          DC    A(XINTRMO)   Extended BFP rounding mode test results
00000358  00003600                                  289          DC    A(XINTRMOF)  Extended BFP rounding mode test flags
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    291 ************************************************************************
                                                    292 *
                                                    293 * Convert short BFP to uint-32 format.  A pair of results is generated
                                                    294 * for each input: one with all exceptions non-trappable, and the second
                                                    295 * with all exceptions trappable.   The FPCR and condition code is
                                                    296 * stored for each result.
                                                    297 *
                                                    298 ************************************************************************


0000035C  9823 A000                 00000000    300 CLFEBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000360  9878 A008                 00000008    301          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000364  1222                                  302          LTR   R2,R2         Any test cases?
00000366  078D                                  303          BZR   R13           ..No, return to caller
00000368  0DC0                                  304          BASR  R12,0         Set top of loop
                                                    305 *
0000036A  7800 3000                 00000000    306          LE    FPR0,0(,R3)   Get short BFP test value
0000036E  B29D F2F4                 000002F4    307          LFPC  FPCREGNT      Set exceptions non-trappable
00000372  B39C 0010                             308          CLFEBR R1,0,FPR0,0  Cvt float in FPR0 to uint-32 in GPR1
00000376  5010 7000                 00000000    309          ST    R1,0(,R7)     Store int-32 result
0000037A  B29C 8000                 00000000    310          STFPC 0(R8)         Store resulting FPCR flags and DXC
0000037E  B222 0010                             311          IPM   R1            Get condition code and program mask
00000382  8810 001C                 0000001C    312          SRL   R1,28         Isolate CC in low order byte
00000386  4210 8003                 00000003    313          STC   R1,3(,R8)     Save condition code as low byte of FPCR
                                                    314 *
0000038A  B29D F2F8                 000002F8    315          LFPC  FPCREGTR      Set exceptions trappable
0000038E  1711                                  316          XR    R1,R1         Clear any residual result in R1
00000390  0410                                  317          SPM   R1            Clear out any residual nz condition code
00000392  B39C 0010                             318          CLFEBR R1,0,FPR0,0  Cvt float in FPR0 to uint-32 in GPR1
00000396  5010 7004                 00000004    319          ST    R1,4(,R7)     Store short BFP result
0000039A  B29C 8004                 00000004    320          STFPC 4(R8)         Store resulting FPCR flags and DXC
0000039E  B222 0010                             321          IPM   R1            Get condition code and program mask
000003A2  8810 001C                 0000001C    322          SRL   R1,28         Isolate CC in low order byte
000003A6  4210 8007                 00000007    323          STC   R1,7(,R8)     Save condition code as low byte of FPCR
                                                    324 *
000003AA  4130 3004                 00000004    325          LA    R3,4(,R3)     Point to next input values
000003AE  4170 7008                 00000008    326          LA    R7,8(,R7)     Point to next int-32 converted value pair
000003B2  4180 8008                 00000008    327          LA    R8,8(,R8)     Point to next FPCR/CC result area
000003B6  062C                                  328          BCTR  R2,R12        Convert next input value.
000003B8  07FD                                  329          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    331 ************************************************************************
                                                    332 *
                                                    333 * Convert short BFP to integers using each possible rounding mode.
                                                    334 * Ten test results are generated for each input.  A 48-byte test result
                                                    335 * section is used to keep results sets aligned on a quad-double word.
                                                    336 *
                                                    337 * The first four tests use rounding modes specified in the FPCR with
                                                    338 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    339 * the first two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    340 * the last two To get full coverage of that instruction pair.
                                                    341 *
                                                    342 * The next six results use instruction-specified rounding modes.
                                                    343 *
                                                    344 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    345 * prior tests used the default rounding mode.  RNTE is tested
                                                    346 * explicitly as a rounding mode in this section.
                                                    347 *
                                                    348 ************************************************************************


000003BA  9823 A000                 00000000   350 CLFEBRA  LM    R2,R3,0(R10)  Get count and address of test input values
000003BE  9878 A008                 00000008   351          LM    R7,R8,8(R10)  Get address of result area and flag area.
000003C2  1222                                 352          LTR   R2,R2         Any test cases?
000003C4  078D                                 353          BZR   R13           ..No, return to caller
000003C6  0DC0                                 354          BASR  R12,0         Set top of loop
                                               355 *
000003C8  7800 3000                 00000000   356          LE    FPR0,0(,R3)    Get short BFP test value
                                               357 *
                                               358 * Test cases using rounding mode specified in the FPCR
                                               359 *
000003CC  B29D F2F4                 000002F4   360          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003D0  B299 0001                 00000001   361          SRNM  1             SET FPCR to RZ, towards zero.
000003D4  B39C 0410                            362          CLFEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000003D8  5010 7000                 00000000   363          ST    R1,0*4(,R7)   Store uint-32 result
000003DC  B29C 8000                 00000000   364          STFPC 0(R8)         Store resulting FPCR flags and DXC
000003E0  B222 0010                            365          IPM   R1            Get condition code and program mask
000003E4  8810 001C                 0000001C   366          SRL   R1,28         Isolate CC in low order byte
000003E8  4210 8003                 00000003   367          STC   R1,3(,R8)     Save condition code as low byte of FPCR
                                               368 *
000003EC  B29D F2F4                 000002F4   369          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000003F0  B299 0002                 00000002   370          SRNM  2             SET FPCR to RP, to +infinity
000003F4  B39C 0410                            371          CLFEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000003F8  5010 7004                 00000004   372          ST    R1,1*4(,R7)   Store uint-32 result
000003FC  B29C 8004                 00000004   373          STFPC 1*4(R8)       Store resulting FPCR flags and DXC
00000400  B222 0010                            374          IPM   R1            Get condition code and program mask
00000404  8810 001C                 0000001C   375          SRL   R1,28         Isolate CC in low order byte
00000408  4210 8007                 00000007   376          STC   R1,(1*4)+3(,R8)  Save condition code as low byte of FPCR
                                               377 *
0000040C  B29D F2F4                 000002F4   378          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000410  B2B8 0003                 00000003   379          SRNMB 3             SET FPCR to RM, to -infinity
00000414  B39C 0410                            380          CLFEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000418  5010 7008                 00000008   381          ST    R1,2*4(,R7)   Store uint-32 result
0000041C  B29C 8008                 00000008   382          STFPC 2*4(R8)       Store resulting FPCR flags and DXC
00000420  B222 0010                            383          IPM   R1            Get condition code and program mask
00000424  8810 001C                 0000001C   384          SRL   R1,28         Isolate CC in low order byte
00000428  4210 800B                 0000000B   385          STC   R1,(2*4)+3(,R8)  Save condition code as low byte of FPCR
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    386 *
0000042C  B29D F2F4                     000002F4   387          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000430  B2B8 0007                     00000007   388          SRNMB 7             RFS, Prepare for Shorter Precision
00000434  B39C 0410                                389          CLFEBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000438  5010 700C                     0000000C   390          ST    R1,3*4(,R7)   Store uint-32 result
0000043C  B29C 800C                     0000000C   391          STFPC 3*4(R8)       Store resulting FPCR flags and DXC
00000440  B222 0010                                392          IPM   R1            Get condition code and program mask
00000444  8810 001C                     0000001C   393          SRL   R1,28         Isolate CC in low order byte
00000448  4210 800F                     0000000F   394          STC   R1,(3*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    395 *
                                                    396 * Test cases using rounding mode specified in the instruction M3 field
                                                    397 *
0000044C  B29D F2F4                     000002F4   398          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000450  B39C 1010                                399          CLFEBR R1,1,FPR0,B'0000'  RNTA, to nearest, ties away
00000454  5010 7010                     00000010   400          ST    R1,4*4(,R7)   Store uint-32 result
00000458  B29C 8010                     00000010   401          STFPC 4*4(R8)       Store resulting FPCR flags and DXC
0000045C  B222 0010                                402          IPM   R1            Get condition code and program mask
00000460  8810 001C                     0000001C   403          SRL   R1,28         Isolate CC in low order byte
00000464  4210 8013                     00000013   404          STC   R1,(4*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    405 *
00000468  B29D F2F4                     000002F4   406          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000046C  B39C 3010                                407          CLFEBR R1,3,FPR0,B'0000'  RFS, prepare for shorter precision
00000470  5010 7014                     00000014   408          ST    R1,5*4(,R7)   Store uint-32 result
00000474  B29C 8014                     00000014   409          STFPC 5*4(R8)       Store resulting FPCR flags and DXC
00000478  B222 0010                                410          IPM   R1            Get condition code and program mask
0000047C  8810 001C                     0000001C   411          SRL   R1,28         Isolate CC in low order byte
00000480  4210 8017                     00000017   412          STC   R1,(5*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    413 *
00000484  B29D F2F4                     000002F4   414          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000488  B39C 4010                                415          CLFEBR R1,4,FPR0,B'0000'  RNTE, to nearest, ties to even
0000048C  5010 7018                     00000018   416          ST    R1,6*4(,R7)   Store uint-32 result
00000490  B29C 8018                     00000018   417          STFPC 6*4(R8)       Store resulting FPCR flags and DXC
00000494  B222 0010                                418          IPM   R1            Get condition code and program mask
00000498  8810 001C                     0000001C   419          SRL   R1,28         Isolate CC in low order byte
0000049C  4210 801B                     0000001B   420          STC   R1,(6*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    421 *
000004A0  B29D F2F4                     000002F4   422          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004A4  B39C 5010                                423          CLFEBR R1,5,FPR0,B'0000'  RZ, toward zero
000004A8  5010 701C                     0000001C   424          ST    R1,7*4(,R7)   Store uint-32 result
000004AC  B29C 801C                     0000001C   425          STFPC 7*4(R8)       Store resulting FPCR flags and DXC
000004B0  B222 0010                                426          IPM   R1            Get condition code and program mask
000004B4  8810 001C                     0000001C   427          SRL   R1,28         Isolate CC in low order byte
000004B8  4210 801F                     0000001F   428          STC   R1,(7*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    429 *
000004BC  B29D F2F4                     000002F4   430          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004C0  B39C 6010                                431          CLFEBR R1,6,FPR0,B'0000'  RP, to +inf
000004C4  5010 7020                     00000020   432          ST    R1,8*4(,R7)   Store uint-32 result
000004C8  B29C 8020                     00000020   433          STFPC 8*4(R8)       Store resulting FPCR flags and DXC
000004CC  B222 0010                                434          IPM   R1            Get condition code and program mask
000004D0  8810 001C                     0000001C   435          SRL   R1,28         Isolate CC in low order byte
000004D4  4210 8023                     00000023   436          STC   R1,(8*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    437 *
000004D8  B29D F2F4                     000002F4   438          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000004DC  B39C 7010                                439          CLFEBR R1,7,FPR0,B'0000'  RM, to -inf
000004E0  5010 7024                     00000024   440          ST    R1,9*4(,R7)   Store uint-32 result
000004E4  B29C 8024                     00000024   441          STFPC 9*4(R8)       Store resulting FPCR flags and DXC
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000004E8  B222 0010                          442          IPM   R1            Get condition code and program mask
000004EC  8810 001C                 0000001C 443          SRL   R1,28         Isolate CC in low order byte
000004F0  4210 8027                 00000027 444          STC   R1,(9*4)+3(,R8)  Save condition code as low byte of FPCR
                                             445 *
000004F4  4130 3004                 00000004 446          LA    R3,4(,R3)     Point to next input value
000004F8  4170 7030                 00000030 447          LA    R7,12*4(,R7)  Point to next uint-32 result set
000004FC  4180 8030                 00000030 448          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
00000500  062C                               449          BCTR  R2,R12        Convert next input value.
00000502  07FD                               450          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    452 ************************************************************************
                                                    453 *
                                                    454 * Convert long BFP inputs to uint-32.  A pair of results is generated
                                                    455 * for each input: one with all exceptions non-trappable, and the second
                                                    456 * with all exceptions trappable.   The FPCR and condition code is
                                                    457 * stored for each result.
                                                    458 *
                                                    459 ************************************************************************

00000504  9823 A000                 00000000   461 CLFDBR   LM    R2,R3,0(R10)  Get count and address of test input values
00000508  9878 A008                 00000008   462          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000050C  1222                               463          LTR   R2,R2         Any test cases?
0000050E  078D                               464          BZR   R13           ..No, return to caller
00000510  0DC0                               465          BASR  R12,0         Set top of loop
                                             466 *
00000512  6800 3000                 00000000   467          LD    FPR0,0(,R3)   Get long BFP test value
00000516  B29D F2F4                 000002F4   468          LFPC  FPCREGNT      Set exceptions non-trappable
0000051A  B39D 0010                          469          CLFDBR R1,0,FPR0,0  Cvt float in FPR0 to uint-32 in GPR1
0000051E  5010 7000                 00000000   470          ST    R1,0(,R7)     Store long BFP result
00000522  B29C 8000                 00000000   471          STFPC 0(R8)         Store resulting FPCR flags and DXC
00000526  B222 0010                          472          IPM   R1            Get condition code and program mask
0000052A  8810 001C                 0000001C   473          SRL   R1,28         Isolate CC in low order byte
0000052E  4210 8003                 00000003   474          STC   R1,3(,R8)     Save condition code as low byte of FPCR
                                             475 *
00000532  B29D F2F8                 000002F8   476          LFPC  FPCREGTR      Set exceptions trappable
00000536  1711                               477          XR    R1,R1         Clear any residual result in R1
00000538  0410                               478          SPM   R1            Clear out any residual nz condition code
0000053A  B39D 0010                          479          CLFDBR R1,0,FPR0,0  Cvt float in FPR0 to uint-32 in GPR1
0000053E  5010 7004                 00000004   480          ST    R1,4(,R7)     Store int-32 result
00000542  B29C 8004                 00000004   481          STFPC 4(R8)         Store resulting FPCR flags and DXC
00000546  B222 0010                          482          IPM   R1            Get condition code and program mask
0000054A  8810 001C                 0000001C   483          SRL   R1,28         Isolate CC in low order byte
0000054E  4210 8007                 00000007   484          STC   R1,7(,R8)     Save condition code as low byte of FPCR
                                             485 *
00000552  4130 3008                 00000008   486          LA    R3,8(,R3)     Point to next input values
00000556  4170 7008                 00000008   487          LA    R7,8(,R7)     Point to next uint-32 converted value pair
0000055A  4180 8008                 00000008   488          LA    R8,8(,R8)     Point to next FPCR/CC result area
0000055E  062C                               489          BCTR  R2,R12        Convert next input value.
00000560  07FD                               490          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    492 ********************************************************************************
                                                    493 *
                                                    494 * Convert long BFP to integers using each possible rounding mode.
                                                    495 * Ten test results are generated for each input.  A 48-byte test result
                                                    496 * section is used to keep results sets aligned on a quad-double word.
                                                    497 *
                                                    498 * The first four tests use rounding modes specified in the FPCR with
                                                    499 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for
                                                    500 * thefirst two FPCR-controlled tests and SRNMB (3-bit) is used for
                                                    501 * the last two To get full coverage of that instruction pair.
                                                    502 *
                                                    503 * The next six results use instruction-specified rounding modes.
                                                    504 *
                                                    505 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    506 * prior tests used the default rounding mode.  RNTE is tested
                                                    507 * explicitly as a rounding mode in this section.
                                                    508 *
                                                    509 ********************************************************************************


00000562  9823 A000                 00000000   511 CLFDBRA  LM    R2,R3,0(R10)  Get count and address of test input values
00000566  9878 A008                 00000008   512          LM    R7,R8,8(R10)  Get address of result area and flag area.
0000056A  1222                                 513          LTR   R2,R2         Any test cases?
0000056C  078D                                 514          BZR   R13           ..No, return to caller
0000056E  0DC0                                 515          BASR  R12,0         Set top of loop
                                               516 *
00000570  6800 3000                 00000000   517          LD    FPR0,0(,R3)    Get long BFP test value
                                               518 *
                                               519 * Test cases using rounding mode specified in the FPCR
                                               520 *
00000574  B29D F2F4                 000002F4   521          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000578  B299 0001                 00000001   522          SRNM  1             SET FPCR to RZ, towards zero.
0000057C  B39D 0410                            523          CLFDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000580  5010 7000                 00000000   524          ST    R1,0*4(,R7)   Store uint-32 result
00000584  B29C 8000                 00000000   525          STFPC 0(R8)         Store resulting FPCR flags and DXC
00000588  B222 0010                            526          IPM   R1            Get condition code and program mask
0000058C  8810 001C                 0000001C   527          SRL   R1,28         Isolate CC in low order byte
00000590  4210 8003                 00000003   528          STC   R1,3(,R8)     Save condition code as low byte of FPCR
                                               529 *
00000594  B29D F2F4                 000002F4   530          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000598  B299 0002                 00000002   531          SRNM  2             SET FPCR to RP, to +infinity
0000059C  B39D 0410                            532          CLFDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000005A0  5010 7004                 00000004   533          ST    R1,1*4(,R7)   Store uint-32 result
000005A4  B29C 8004                 00000004   534          STFPC 1*4(R8)       Store resulting FPCR flags and DXC
000005A8  B222 0010                            535          IPM   R1            Get condition code and program mask
000005AC  8810 001C                 0000001C   536          SRL   R1,28         Isolate CC in low order byte
000005B0  4210 8007                 00000007   537          STC   R1,(1*4)+3(,R8)  Save condition code as low byte of FPCR
                                               538 *
000005B4  B29D F2F4                 000002F4   539          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000005B8  B2B8 0003                 00000003   540          SRNMB 3             SET FPCR to RM, to -infinity
000005BC  B39D 0410                            541          CLFDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000005C0  5010 7008                 00000008   542          ST    R1,2*4(,R7)   Store uint-32 result
000005C4  B29C 8008                 00000008   543          STFPC 2*4(R8)       Store resulting FPCR flags and DXC
000005C8  B222 0010                            544          IPM   R1            Get condition code and program mask
000005CC  8810 001C                 0000001C   545          SRL   R1,28         Isolate CC in low order byte
000005D0  4210 800B                 0000000B   546          STC   R1,(2*4)+3(,R8)  Save condition code as low byte of FPCR
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    547 *
000005D4  B29D F2F4                     000002F4    548          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000005D8  B2B8 0007                     00000007    549          SRNMB 7             RFS, Prepare for Shorter Precision
000005DC  B39D 0410                                 550          CLFDBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
000005E0  5010 700C                     0000000C    551          ST    R1,3*4(,R7)   Store uint-32 result
000005E4  B29C 800C                     0000000C    552          STFPC 3*4(R8)       Store resulting FPCR flags and DXC
000005E8  B222 0010                                 553          IPM   R1            Get condition code and program mask
000005EC  8810 001C                     0000001C    554          SRL   R1,28         Isolate CC in low order byte
000005F0  4210 800F                     0000000F    555          STC   R1,(3*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    556 *
                                                    557 * Test cases using rounding mode specified in the instruction M3 field
                                                    558 *
000005F4  B29D F2F4                     000002F4    559          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000005F8  B39D 1010                                 560          CLFDBR R1,1,FPR0,B'0000'  RNTA, to nearest, ties away
000005FC  5010 7010                     00000010    561          ST    R1,4*4(,R7)   Store uint-32 result
00000600  B29C 8010                     00000010    562          STFPC 4*4(R8)       Store resulting FPCR flags and DXC
00000604  B222 0010                                 563          IPM   R1            Get condition code and program mask
00000608  8810 001C                     0000001C    564          SRL   R1,28         Isolate CC in low order byte
0000060C  4210 8013                     00000013    565          STC   R1,(4*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    566 *
00000610  B29D F2F4                     000002F4    567          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000614  B39D 3010                                 568          CLFDBR R1,3,FPR0,B'0000'  RFS, prepare for shorter precision
00000618  5010 7014                     00000014    569          ST    R1,5*4(,R7)   Store uint-32 result
0000061C  B29C 8014                     00000014    570          STFPC 5*4(R8)       Store resulting FPCR flags and DXC
00000620  B222 0010                                 571          IPM   R1            Get condition code and program mask
00000624  8810 001C                     0000001C    572          SRL   R1,28         Isolate CC in low order byte
00000628  4210 8017                     00000017    573          STC   R1,(5*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    574 *
0000062C  B29D F2F4                     000002F4    575          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000630  B39D 4010                                 576          CLFDBR R1,4,FPR0,B'0000'  RNTE, to nearest, ties to even
00000634  5010 7018                     00000018    577          ST    R1,6*4(,R7)   Store uint-32 result
00000638  B29C 8018                     00000018    578          STFPC 6*4(R8)       Store resulting FPCR flags and DXC
0000063C  B222 0010                                 579          IPM   R1            Get condition code and program mask
00000640  8810 001C                     0000001C    580          SRL   R1,28         Isolate CC in low order byte
00000644  4210 801B                     0000001B    581          STC   R1,(6*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    582 *
00000648  B29D F2F4                     000002F4    583          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
0000064C  B39D 5010                                 584          CLFDBR R1,5,FPR0,B'0000'  RZ, toward zero
00000650  5010 701C                     0000001C    585          ST    R1,7*4(,R7)   Store uint-32 result
00000654  B29C 801C                     0000001C    586          STFPC 7*4(R8)       Store resulting FPCR flags and DXC
00000658  B222 0010                                 587          IPM   R1            Get condition code and program mask
0000065C  8810 001C                     0000001C    588          SRL   R1,28         Isolate CC in low order byte
00000660  4210 801F                     0000001F    589          STC   R1,(7*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    590 *
00000664  B29D F2F4                     000002F4    591          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000668  B39D 6010                                 592          CLFDBR R1,6,FPR0,B'0000'  RP, to +inf
0000066C  5010 7020                     00000020    593          ST    R1,8*4(,R7)   Store uint-32 result
00000670  B29C 8020                     00000020    594          STFPC 8*4(R8)       Store resulting FPCR flags and DXC
00000674  B222 0010                                 595          IPM   R1            Get condition code and program mask
00000678  8810 001C                     0000001C    596          SRL   R1,28         Isolate CC in low order byte
0000067C  4210 8023                     00000023    597          STC   R1,(8*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    598 *
00000680  B29D F2F4                     000002F4    599          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000684  B39D 7010                                 600          CLFDBR R1,7,FPR0,B'0000'  RM, to -inf
00000688  5010 7024                     00000024    601          ST    R1,9*4(,R7)   Store uint-32 result
0000068C  B29C 8024                     00000024    602          STFPC 9*4(R8)       Store resulting FPCR flags and DXC
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000690  B222 0010                          603          IPM   R1            Get condition code and program mask
00000694  8810 001C                0000001C  604          SRL   R1,28         Isolate CC in low order byte
00000698  4210 8027                00000027  605          STC   R1,(9*4)+3(,R8)  Save condition code as low byte of FPCR
                                             606 *
0000069C  4130 3008                00000008  607          LA    R3,8(,R3)     Point to next input values
000006A0  4170 7030                00000030  608          LA    R7,12*4(,R7)  Point to next long BFP converted values
000006A4  4180 8030                00000030  609          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
000006A8  062C                             610          BCTR  R2,R12        Convert next input value.
000006AA  07FD                             611          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    613 ************************************************************************
                                                    614 *
                                                    615 * Convert extended BFP to uint-32.  A pair of results is generated
                                                    616 * for each input: one with all exceptions non-trappable, and the
                                                    617 * second with all exceptions trappable.   The FPCR and condition code
                                                    618 * are stored for each result.
                                                    619 *
                                                    620 ************************************************************************


000006AC  9823 A000                 00000000   622 CLFXBR   LM    R2,R3,0(R10)  Get count and address of test input values
000006B0  9878 A008                 00000008   623          LM    R7,R8,8(R10)  Get address of result area and flag area.
000006B4  1222                                 624          LTR   R2,R2         Any test cases?
000006B6  078D                                 625          BZR   R13           ..No, return to caller
000006B8  0DC0                                 626          BASR  R12,0         Set top of loop
                                               627 *
000006BA  6800 3000                 00000000   628          LD    FPR0,0(,R3)   Get extended BFP test value part 1
000006BE  6820 3008                 00000008   629          LD    FPR2,8(,R3)   Get extended BFP test value part 1
000006C2  B29D F2F4                 000002F4   630          LFPC  FPCREGNT      Set exceptions non-trappable
000006C6  B39E 0010                            631          CLFXBR R1,0,FPR0,0  Cvt float in FPR0-FPR2 to uint-32 in GPR1
000006CA  5010 7000                 00000000   632          ST    R1,0(,R7)     Store uint-32 result
000006CE  B29C 8000                 00000000   633          STFPC 0(R8)         Store resulting FPCR flags and DXC
000006D2  B222 0010                            634          IPM   R1            Get condition code and program mask
000006D6  8810 001C                 0000001C   635          SRL   R1,28         Isolate CC in low order byte
000006DA  4210 8003                 00000003   636          STC   R1,3(,R8)     Save condition code as low byte of FPCR
                                               637 *
000006DE  B29D F2F8                 000002F8   638          LFPC  FPCREGTR      Set exceptions trappable
000006E2  1711                                 639          XR    R1,R1         Clear any residual result in R1
000006E4  0410                                 640          SPM   R1            Clear out any residual nz condition code
000006E6  B39E 0010                            641          CLFXBR R1,0,FPR0,0  Cvt float in FPR0-FPR2 to uint-32 in GPR1
000006EA  5010 7004                 00000004   642          ST    R1,4(,R7)     Store uint-32 result
000006EE  B29C 8004                 00000004   643          STFPC 4(R8)         Store resulting FPCR flags and DXC
000006F2  B222 0010                            644          IPM   R1            Get condition code and program mask
000006F6  8810 001C                 0000001C   645          SRL   R1,28         Isolate CC in low order byte
000006FA  4210 8007                 00000007   646          STC   R1,7(,R8)     Save condition code as low byte of FPCR
                                               647 *
000006FE  4130 3010                 00000010   648          LA    R3,16(,R3)    Point to next extended BFP input value
00000702  4170 7008                 00000008   649          LA    R7,8(,R7)     Point to next uint-32 converted value pair
00000706  4180 8008                 00000008   650          LA    R8,8(,R8)     Point to next FPCR/CC result area
0000070A  062C                                 651          BCTR  R2,R12        Convert next input value.
0000070C  07FD                                 652          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    654 ********************************************************************************
                                                    655 *
                                                    656 * Convert extended BFP to integers using each possible rounding mode.
                                                    657 * Ten test results are generated for each input.  A 48-byte test result
                                                    658 * section is used to keep results sets aligned on a quad-double word.
                                                    659 *
                                                    660 * The first four tests use rounding modes specified in the FPCR with
                                                    661 * the IEEE Inexact exception supressed.  SRNM (2-bit) is used  for the
                                                    662 * first two FPCR-controlled tests and SRNMB (3-bit) is used for the
                                                    663 * last two To get full coverage of that instruction pair.
                                                    664 *
                                                    665 * The next six results use instruction-specified rounding modes.
                                                    666 *
                                                    667 * The default rounding mode (0 for RNTE) is not tested in this section;
                                                    668 * prior tests used the default rounding mode.  RNTE is tested
                                                    669 * explicitly as a rounding mode in this section.
                                                    670 *
                                                    671 ********************************************************************************


0000070E  9823 A000                 00000000    673 CLFXBRA  LM    R2,R3,0(R10)  Get count and address of test input values
00000712  9878 A008                 00000008    674          LM    R7,R8,8(R10)  Get address of result area and flag area.
00000716  1222                                  675          LTR   R2,R2         Any test cases?
00000718  078D                                  676          BZR   R13           ..No, return to caller
0000071A  0DC0                                  677          BASR  R12,0         Set top of loop
                                                    678 *
0000071C  6800 3000                 00000000    679          LD    FPR0,0(,R3)   Get extended BFP test value part 1
00000720  6820 3008                 00000008    680          LD    FPR2,8(,R3)   Get extended BFP test value part 2
                                                    681 *
                                                    682 * Test cases using rounding mode specified in the FPCR
                                                    683 *
00000724  B29D F2F4                 000002F4    684          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000728  B299 0001                 00000001    685          SRNM  1             Set FPCR to RZ, towards zero.
0000072C  B39E 0410                             686          CLFXBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000730  5010 7000                 00000000    687          ST    R1,0*4(,R7)   Store uint-32 result
00000734  B29C 8000                 00000000    688          STFPC 0(R8)         Store resulting FPCR flags and DXC
00000738  B222 0010                             689          IPM   R1            Get condition code and program mask
0000073C  8810 001C                 0000001C    690          SRL   R1,28         Isolate CC in low order byte
00000740  4210 8003                 00000003    691          STC   R1,3(,R8)     Save condition code as low byte of FPCR
                                                    692 *
00000744  B29D F2F4                 000002F4    693          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000748  B299 0002                 00000002    694          SRNM  2             SET FPCR to RP, to +infinity
0000074C  B39E 0410                             695          CLFXBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000750  5010 7004                 00000004    696          ST    R1,1*4(,R7)   Store uint-32 result
00000754  B29C 8004                 00000004    697          STFPC 1*4(R8)       Store resulting FPCR flags and DXC
00000758  B222 0010                             698          IPM   R1            Get condition code and program mask
0000075C  8810 001C                 0000001C    699          SRL   R1,28         Isolate CC in low order byte
00000760  4210 8007                 00000007    700          STC   R1,(1*4)+3(,R8)  Save condition code as low byte of FPCR
                                                    701 *
00000764  B29D F2F4                 000002F4    702          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000768  B2B8 0003                 00000003    703          SRNMB 3             SET FPCR to RM, to -infinity
0000076C  B39E 0410                             704          CLFXBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000770  5010 7008                 00000008    705          ST    R1,2*4(,R7)   Store uint-32 result
00000774  B29C 8008                 00000008    706          STFPC 2*4(R8)       Store resulting FPCR flags and DXC
00000778  B222 0010                             707          IPM   R1            Get condition code and program mask
0000077C  8810 001C                 0000001C    708          SRL   R1,28         Isolate CC in low order byte
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00000780  4210 800B                     0000000B   709          STC   R1,(2*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   710 *
00000784  B29D F2F4                     000002F4   711          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000788  B2B8 0007                     00000007   712          SRNMB 7             RFS, Prepare for Shorter Precision
0000078C  B39E 0410                                713          CLFXBR R1,0,FPR0,B'0100'  FPCR ctl'd rounding, inexact masked
00000790  5010 700C                     0000000C   714          ST    R1,3*4(,R7)   Store uint-32 result
00000794  B29C 800C                     0000000C   715          STFPC 3*4(R8)       Store resulting FPCR flags and DXC
00000798  B222 0010                                716          IPM   R1            Get condition code and program mask
0000079C  8810 001C                     0000001C   717          SRL   R1,28         Isolate CC in low order byte
000007A0  4210 800F                     0000000F   718          STC   R1,(3*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   719 *
                                                   720 * Test cases using rounding mode specified in the instruction M3 field
                                                   721 *
000007A4  B29D F2F4                     000002F4   722          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007A8  B39E 1010                                723          CLFXBR R1,1,FPR0,B'0000'  RNTA, to nearest, ties away
000007AC  5010 7010                     00000010   724          ST    R1,4*4(,R7)   Store uint-32 result
000007B0  B29C 8010                     00000010   725          STFPC 4*4(R8)       Store resulting FPCR flags and DXC
000007B4  B222 0010                                726          IPM   R1            Get condition code and program mask
000007B8  8810 001C                     0000001C   727          SRL   R1,28         Isolate CC in low order byte
000007BC  4210 8013                     00000013   728          STC   R1,(4*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   729 *
000007C0  B29D F2F4                     000002F4   730          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007C4  B39E 3010                                731          CLFXBR R1,3,FPR0,B'0000'  RFS, prepare for shorter precision
000007C8  5010 7014                     00000014   732          ST    R1,5*4(,R7)   Store uint-32 result
000007CC  B29C 8014                     00000014   733          STFPC 5*4(R8)       Store resulting FPCR flags and DXC
000007D0  B222 0010                                734          IPM   R1            Get condition code and program mask
000007D4  8810 001C                     0000001C   735          SRL   R1,28         Isolate CC in low order byte
000007D8  4210 8017                     00000017   736          STC   R1,(5*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   737 *
000007DC  B29D F2F4                     000002F4   738          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007E0  B39E 4010                                739          CLFXBR R1,4,FPR0,B'0000'  RNTE, to nearest, ties to even
000007E4  5010 7018                     00000018   740          ST    R1,6*4(,R7)   Store uint-32 result
000007E8  B29C 8018                     00000018   741          STFPC 6*4(R8)       Store resulting FPCR flags and DXC
000007EC  B222 0010                                742          IPM   R1            Get condition code and program mask
000007F0  8810 001C                     0000001C   743          SRL   R1,28         Isolate CC in low order byte
000007F4  4210 801B                     0000001B   744          STC   R1,(6*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   745 *
000007F8  B29D F2F4                     000002F4   746          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
000007FC  B39E 5010                                747          CLFXBR R1,5,FPR0,B'0000'  RZ, toward zero
00000800  5010 701C                     0000001C   748          ST    R1,7*4(,R7)   Store uint-32 result
00000804  B29C 801C                     0000001C   749          STFPC 7*4(R8)       Store resulting FPCR flags and DXC
00000808  B222 0010                                750          IPM   R1            Get condition code and program mask
0000080C  8810 001C                     0000001C   751          SRL   R1,28         Isolate CC in low order byte
00000810  4210 801F                     0000001F   752          STC   R1,(7*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   753 *
00000814  B29D F2F4                     000002F4   754          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000818  B39E 6010                                755          CLFXBR R1,6,FPR0,B'0000'  RP, to +inf
0000081C  5010 7020                     00000020   756          ST    R1,8*4(,R7)   Store uint-32 result
00000820  B29C 8020                     00000020   757          STFPC 8*4(R8)       Store resulting FPCR flags and DXC
00000824  B222 0010                                758          IPM   R1            Get condition code and program mask
00000828  8810 001C                     0000001C   759          SRL   R1,28         Isolate CC in low order byte
0000082C  4210 8023                     00000023   760          STC   R1,(8*4)+3(,R8)  Save condition code as low byte of FPCR
                                                   761 *
00000830  B29D F2F4                     000002F4   762          LFPC  FPCREGNT      Set exceptions non-trappable, clear flags
00000834  B39E 7010                                763          CLFXBR R1,7,FPR0,B'0000'  RM, to -inf
00000838  5010 7024                     00000024   764          ST    R1,9*4(,R7)   Store uint-32 result
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

0000083C  B29C 8024                    00000024   765          STFPC 9*4(R8)       Store resulting FPCR flags and DXC
00000840  B222 0010                               766          IPM   R1            Get condition code and program mask
00000844  8810 001C                    0000001C   767          SRL   R1,28         Isolate CC in low order byte
00000848  4210 8027                    00000027   768          STC   R1,(9*4)+3(,R8)  Save condition code as low byte of FPCR
                                                  769 *
0000084C  4130 3010                    00000010   770          LA    R3,16(,R3)    Point to next input value
00000850  4170 7030                    00000030   771          LA    R7,12*4(,R7)  Point to next long BFP converted values
00000854  4180 8030                    00000030   772          LA    R8,12*4(,R8)  Point to next FPCR/CC result area
00000858  062C                                    773          BCTR  R2,R12        Convert next input value.
0000085A  07FD                                    774          BR    R13           All converted; return.
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    776 ********************************************************************************
                                                    777 *
                                                    778 * Floating point inputs for Convert From Fixed testing.  The same test
                                                    779 * values in the appropriate input format are used for short, long,
                                                    780 * and extended format tests.  The last four values should generate
                                                    781 * exceptions.
                                                    782 *
                                                    783 ********************************************************************************


                                                    785 *
                                                    786 * Inputs for short BFP testing of trap and non-trap results
                                                    787 *
0000085C                                            788 SBFPIN   DS    0F             Inputs for short BFP testing
0000085C  3F800000                                  789          DC    X'3F800000'     +1.0
00000860  40000000                                  790          DC    X'40000000'     +2.0
00000864  40800000                                  791          DC    X'40800000'     +4.0
00000868  7F810000                                  792          DC    X'7F810000'     SNaN
0000086C  7FC10000                                  793          DC    X'7FC10000'     QNaN
00000870  4F800000                                  794          DC    X'4F800000'     Max uint-32 + 1 rounded to short BFP
                                                    795 *                               ...4 294 967 296
00000874  4F7FFFFF                                  796          DC    X'4F7FFFFF'     Max uint-32 value representable
                                                    797 *                               ...in short bfp (4 294 967 040)
00000878  3F400000                                  798          DC    X'3F400000'     +0.75
0000087C  3E800000                                  799          DC    X'3E800000'     +0.25
                          00000024  00000001        800 SBFPCT   EQU   *-SBFPIN     Count of short BFP in list * 4
                                                    801 *
                                                    802 * Inputs for short BFP exhaustive rounding mode testing.
                                                    803 *
00000880                                            804 SBFPINRM DS    0F
00000880  BFC00000                                  805          DC    X'BFC00000'     -1.5
00000884  BF000000                                  806          DC    X'BF000000'     -0.5
00000888  3F000000                                  807          DC    X'3F000000'     +0.5
0000088C  3FC00000                                  808          DC    X'3FC00000'     +1.5
00000890  40200000                                  809          DC    X'40200000'     +2.5
00000894  40B00000                                  810          DC    X'40B00000'     +5.5
00000898  41180000                                  811          DC    X'41180000'     +9.5
0000089C  4F7FFFFF                                  812          DC    X'4F7FFFFF'     largest uint-32 value representable
                                                    813 *                               ...in short bfp (4 294 967 040)
000008A0  3F400000                                  814          DC    X'3F400000'     +0.75
000008A4  3E800000                                  815          DC    X'3E800000'     +0.25
                          00000028  00000001        816 SBFPRMCT EQU   *-SBFPINRM   Count of short BFP for rounding tests * 4
                                                    817 *
                                                    818 * Inputs for long BFP testing of trap and non-trap results
                                                    819 *
000008A8                                            820 LBFPIN   DS    0F                 Inputs for long BFP testing
000008A8  3FF00000 00000000                         821          DC    X'3FF0000000000000'    +1.0
000008B0  40000000 00000000                         822          DC    X'4000000000000000'    +2.0
000008B8  40100000 00000000                         823          DC    X'4010000000000000'    +4.0
000008C0  7FF01000 00000000                         824          DC    X'7FF0100000000000'    SNaN
000008C8  7FF81000 00000000                         825          DC    X'7FF8100000000000'    QNaN
000008D0  41EFFFFF FFF00000                         826          DC    X'41EFFFFFFFF00000'    max uint-32 + 0.5
                                                    827 *                                     ...4 294 967 295.5
                                                    828 *                                     Rounds up on RNTE, overflows
000008D8  41EFFFFF FFECCCCD                         829          DC    X'41EFFFFFFECCCCD'     max uint-32 + 0.4
                                                    830 *                                     ...4 294 967 295.4, exceeds
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    831 *                                       uint-32, but rounds down to fit
000008E0  3FE80000 00000000                         832          DC    X'3FE8000000000000'    +0.75
000008E8  3FD00000 00000000                         833          DC    X'3FD0000000000000'    +0.25
                             00000048  00000001     834 LBFPCT   EQU   *-LBFPIN      Count of long BFP in list * 8
                                                    835 *
                                                    836 * Inputs for long BFP exhaustive rounding mode testing.
                                                    837 *
000008F0                                            838 LBFPINRM DS    0F
000008F0  BFF80000 00000000                         839          DC    X'BFF8000000000000'    -1.5
000008F8  BFE00000 00000000                         840          DC    X'BFE0000000000000'    -0.5
00000900  3FE00000 00000000                         841          DC    X'3FE0000000000000'    +0.5
00000908  3FF80000 00000000                         842          DC    X'3FF8000000000000'    +1.5
00000910  40040000 00000000                         843          DC    X'4004000000000000'    +2.5
00000918  40160000 00000000                         844          DC    X'4016000000000000'    +5.5
00000920  40230000 00000000                         845          DC    X'4023000000000000'    +9.5
00000928  41EFFFFF FFF00000                         846          DC    X'41EFFFFFFFF00000'    max uint-32 + 0.5
                                                    847 *                                       ...4 294 967 295.5
00000930  3FE80000 00000000                         848          DC    X'3FE8000000000000'    +0.75
00000938  3FD00000 00000000                         849          DC    X'3FD0000000000000'    +0.25
                             00000050  00000001     850 LBFPRMCT EQU   *-LBFPINRM   Count of long BFP for rounding tests * 8
                                                    851 *
                                                    852 * Inputs for short BFP testing of trap and non-trap results
                                                    853 *
00000940                                            854 XBFPIN   DS    0D                  Inputs for long BFP testing
00000940  3FFF0000 00000000                         855          DC    X'3FFF0000000000000000000000000000'    +1.0
00000950  40000000 00000000                         856          DC    X'4000000000000000000000000000000'    +2.0
00000960  40010000 00000000                         857          DC    X'4001000000000000000000000000000'    +4.0
00000970  7FFF0100 00000000                         858          DC    X'7FFF0100000000000000000000000000'    SNaN
00000980  7FFF8100 00000000                         859          DC    X'7FFF8100000000000000000000000000'    QNaN
00000990  401EFFFF FFFF0000                         860          DC    X'401EFFFFFFFF0000000000000000000'    max uint-32+0.5
                                                    861 *                                      ...4 294 967 295.5
                                                    862 *                                     Rounds up on RNTE, overflows
000009A0  401EFFFF FFFECCCC                         863          DC    X'401EFFFFFFFECCCCCCCCCCCCCCCCCD'    max uint-32+0.4
                                                    864 *                                      ...4 294 967 295.4, exceeds
                                                    865 *                                      uint-32, but rounds down to fit
000009B0  3FFE8000 00000000                         866          DC    X'3FFE8000000000000000000000000000'         0.75
000009C0  3FFD0000 00000000                         867          DC    X'3FFD0000000000000000000000000000'         0.25
                             00000090  00000001     868 XBFPCT   EQU   *-XBFPIN     Count of extended BFP in list * 16
                                                    869 *
                                                    870 * Inputs for extended BFP exhaustive rounding mode testing.
                                                    871 *
000009D0                                            872 XBFPINRM DS    0D
000009D0  BFFF8000 00000000                         873          DC    X'BFFF8000000000000000000000000000'   -1.5
000009E0  BFFE0000 00000000                         874          DC    X'BFFE0000000000000000000000000000'   -0.5
000009F0  3FFE0000 00000000                         875          DC    X'3FFE0000000000000000000000000000'   +0.5
00000A00  3FFF8000 00000000                         876          DC    X'3FFF8000000000000000000000000000'   +1.5
00000A10  40004000 00000000                         877          DC    X'4000400000000000000000000000000'   +2.5
00000A20  40016000 00000000                         878          DC    X'4001600000000000000000000000000'   +5.5
00000A30  40023000 00000000                         879          DC    X'4002300000000000000000000000000'   +9.5
00000A40  401EFFFF FFFF0000                         880          DC    X'401EFFFFFFFF0000000000000000000'   max uint-32 + 0.5
                                                    881 *                                                 (+4 294 967 295.5)
                                                    882 *    Above is always inexact, and may overflow based on rounding mode
00000A50  3FFE8000 00000000                         883          DC    X'3FFE8000000000000000000000000000'         0.75
00000A60  3FFD0000 00000000                         884          DC    X'3FFD0000000000000000000000000000'         0.25
                             000000A0  00000001     885 XBFPRMCT EQU   *-XBFPINRM   Count of extended BFP rounding tests * 16
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                          887 ************************************************************************
                                          888 *                    ACTUAL results saved here
                                          889 ************************************************************************
                                          890 *
                                          891 *                Locations for ACTUAL results
                                          892 *
                                          893 *
                            00001000  00000000    894 SINTOUT  EQU   BFPCVTTL+X'1000'    uint-32 values from short BFP
                                          895 *                                        ..9 pairs used, room for 16
                            00001100  00000000    896 SINTFLGS EQU   BFPCVTTL+X'1100'    FPCR flags and DXC from short BFP
                                          897 *                                        ..9 pairs used, room for 16
                            00001200  00000000    898 SINTRMO  EQU   BFPCVTTL+X'1200'    Short rounding mode test results
                                          899 *                                        ..10 sets used, room for 20
                            00001600  00000000    900 SINTRMOF EQU   BFPCVTTL+X'1600'    Short rounding mode FPCR contents
                                          901 *                                        ..10 sets used, room for 20
                                          902 *
                            00002000  00000000    903 LINTOUT  EQU   BFPCVTTL+X'2000'    uint-32 values from long BFP
                                          904 *                                        ..9 pairs used, room for 16
                            00002100  00000000    905 LINTFLGS EQU   BFPCVTTL+X'2100'    FPCR flags and DXC from long BFP
                                          906 *                                        ..9 pairs used, room for 16
                            00002200  00000000    907 LINTRMO  EQU   BFPCVTTL+X'2200'    Long rounding mode test results
                                          908 *                                        ..10 sets used, room for 20
                            00002600  00000000    909 LINTRMOF EQU   BFPCVTTL+X'2600'    Long rounding mode FPCR contents
                                          910 *                                        ..10 sets used, room for 20
                                          911 *
                            00003000  00000000    912 XINTOUT  EQU   BFPCVTTL+X'3000'    uint-32 values from extended BFP
                                          913 *                                        ..9 pairs used, room for 16
                            00003100  00000000    914 XINTFLGS EQU   BFPCVTTL+X'3100'    FPCR flags and DXC from extended BFP
                                          915 *                                        ..9 pairs used, room for 16
                            00003200  00000000    916 XINTRMO  EQU   BFPCVTTL+X'3200'    Extended rounding mode test results
                                          917 *                                        ..10 sets used, room for 20
                            00003600  00000000    918 XINTRMOF EQU   BFPCVTTL+X'3600'    Extended rounding mode FPCR contents
                                          919 *                                        ..10 sets used, room for 20
                                          920 *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    922 ********************************************************************************
                                                    923 *                       EXPECTED results
                                                    924 ********************************************************************************
                                                    925 *
00000A70                      00000A70  00004000    926          ORG   BFPCVTTL+X'4000'   (past end of actual results)
                                                    927 *
                              00004000  00000001    928 SINTOUT_GOOD EQU *
00004000  C3D3C6C5 C2D94099                         929  DC CL48'CLFEBR result pairs 1-2'
00004030  00000001 00000001                         930  DC XL16'00000001000000010000000200000002'
00004040  C3D3C6C5 C2D94099                         931  DC CL48'CLFEBR result pairs 3-4'
00004070  00000004 00000004                         932  DC XL16'00000004000000040000000000000000'
00004080  C3D3C6C5 C2D94099                         933  DC CL48'CLFEBR result pairs 5-6'
000040B0  00000000 00000000                         934  DC XL16'00000000000000000FFFFFFF00000000'
000040C0  C3D3C6C5 C2D94099                         935  DC CL48'CLFEBR result pairs 7-8'
000040F0  FFFFFF00 FFFFFF00                         936  DC XL16'FFFFFF00FFFFFF0000000000100000001'
00004100  C3D3C6C5 C2D94099                         937  DC CL48'CLFEBR result pair 9'
00004130  00000000 00000000                         938  DC XL16'00000000000000000000000000000000'
                              00000005  00000001    939 SINTOUT_NUM EQU (*-SINTOUT_GOOD)/64
                                                    940 *
                                                    941 *
                              00004140  00000001    942 SINTFLGS_GOOD EQU *
00004140  C3D3C6C5 C2D940C6                         943  DC CL48'CLFEBR FPC pairs 1-2'
00004170  00000002 F8000002                         944  DC XL16'00000002F800000200000002F8000002'
00004180  C3D3C6C5 C2D940C6                         945  DC CL48'CLFEBR FPC pairs 3-4'
000041B0  00000002 F8000002                         946  DC XL16'00000002F800000200880003F8008000'
000041C0  C3D3C6C5 C2D940C6                         947  DC CL48'CLFEBR FPC pairs 5-6'
000041F0  00880003 F8008000                         948  DC XL16'00880003F800800000880003F8008000'
00004200  C3D3C6C5 C2D940C6                         949  DC CL48'CLFEBR FPC pairs 7-8'
00004230  00000002 F8000002                         950  DC XL16'00000002F800000200080002F8000C02'
00004240  C3D3C6C5 C2D940C6                         951  DC CL48'CLFEBR FPC pair 9'
00004270  00080002 F8000802                         952  DC XL16'00080002F8000802000000000000000'
                              00000005  00000001    953 SINTFLGS_NUM EQU (*-SINTFLGS_GOOD)/64
                                                    954 *
                                                    955 *
                              00004280  00000001    956 SINTRMO_GOOD EQU *
00004280  C3D3C6C5 C2D94060                         957  DC CL48'CLFEBR -1.5 FPC modes 1-3, 7'
000042B0  00000000 00000000                         958  DC XL16'00000000000000000000000000000000'
000042C0  C3D3C6C5 C2D94060                         959  DC CL48'CLFEBR -1.5 M3 modes 1, 3-5'
000042F0  00000000 00000000                         960  DC XL16'00000000000000000000000000000000'
00004300  C3D3C6C5 C2D94060                         961  DC CL48'CLFEBR -1.5 M3 modes 6, 7'
00004330  00000000 00000000                         962  DC XL16'00000000000000000000000000000000'
00004340  C3D3C6C5 C2D94060                         963  DC CL48'CLFEBR -0.5 FPC modes 1-3, 7'
00004370  00000000 00000000                         964  DC XL16'00000000000000000000000000000000'
00004380  C3D3C6C5 C2D94060                         965  DC CL48'CLFEBR -0.5 M3 modes 1, 3-5'
000043B0  00000000 00000000                         966  DC XL16'00000000000000000000000000000000'
000043C0  C3D3C6C5 C2D94060                         967  DC CL48'CLFEBR -0.5 M3 modes 6, 7'
000043F0  00000000 00000000                         968  DC XL16'00000000000000000000000000000000'
00004400  C3D3C6C5 C2D940F0                         969  DC CL48'CLFEBR 0.5 FPC modes 1-3, 7'
00004430  00000000 00000001                         970  DC XL16'00000000000000010000000000000001'
00004440  C3D3C6C5 C2D940F0                         971  DC CL48'CLFEBR 0.5 M3 modes 1, 3-5'
00004470  00000001 00000001                         972  DC XL16'00000001000000010000000000000000'
00004480  C3D3C6C5 C2D940F0                         973  DC CL48'CLFEBR 0.5 M3 modes 6, 7'
000044B0  00000001 00000000                         974  DC XL16'00000001000000000000000000000000'
000044C0  C3D3C6C5 C2D940F1                         975  DC CL48'CLFEBR 1.5 FPC modes 1-3, 7'
000044F0  00000001 00000002                         976  DC XL16'00000001000000020000000100000001'
00004500  C3D3C6C5 C2D940F1                         977  DC CL48'CLFEBR 1.5 M3 modes 1, 3-5'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00004530  00000002 00000001                          978  DC XL16'00000002000000010000000200000001'
00004540  C3D3C6C5 C2D940F1                          979  DC CL48'CLFEBR 1.5 M3 modes 6, 7'
00004570  00000002 00000001                          980  DC XL16'00000002000000010000000000000000'
00004580  C3D3C6C5 C2D940F2                          981  DC CL48'CLFEBR 2.5 FPC modes 1-3, 7'
000045B0  00000002 00000003                          982  DC XL16'00000002000000030000000200000003'
000045C0  C3D3C6C5 C2D940F2                          983  DC CL48'CLFEBR 2.5 M3 modes 1, 3-5'
000045F0  00000003 00000003                          984  DC XL16'00000003000000030000000200000002'
00004600  C3D3C6C5 C2D940F2                          985  DC CL48'CLFEBR 2.5 M3 modes 6, 7'
00004630  00000003 00000002                          986  DC XL16'00000003000000020000000000000000'
00004640  C3D3C6C5 C2D940F5                          987  DC CL48'CLFEBR 5.5 FPC modes 1-3, 7'
00004670  00000005 00000006                          988  DC XL16'00000005000000060000000500000005'
00004680  C3D3C6C5 C2D940F5                          989  DC CL48'CLFEBR 5.5 M3 modes 1, 3-5'
000046B0  00000006 00000005                          990  DC XL16'00000006000000050000000600000005'
000046C0  C3D3C6C5 C2D940F5                          991  DC CL48'CLFEBR 5.5 M3 modes 6, 7'
000046F0  00000006 00000005                          992  DC XL16'00000006000000050000000000000000'
00004700  C3D3C6C5 C2D940F9                          993  DC CL48'CLFEBR 9.5 FPC modes 1-3, 7'
00004730  00000009 0000000A                          994  DC XL16'00000009000000A0000000900000009'
00004740  C3D3C6C5 C2D940F9                          995  DC CL48'CLFEBR 9.5 M3 modes 1, 3-5'
00004770  0000000A 00000009                          996  DC XL16'0000000A00000009000000A00000009'
00004780  C3D3C6C5 C2D940F9                          997  DC CL48'CLFEBR 9.5 M3 modes 6, 7'
000047B0  0000000A 00000009                          998  DC XL16'0000000A00000009000000000000000'
000047C0  C3D3C6C5 C2D94094                          999  DC CL48'CLFEBR max FPC modes 1-3, 7'
000047F0  FFFFFF00 FFFFFF00                         1000  DC XL16'FFFFFF00FFFFFF00FFFFFF00FFFFFF00'
00004800  C3D3C6C5 C2D94094                         1001  DC CL48'CLFEBR max M3 modes 1, 3-5'
00004830  FFFFFF00 FFFFFF00                         1002  DC XL16'FFFFFF00FFFFFF00FFFFFF00FFFFFF00'
00004840  C3D3C6C5 C2D94094                         1003  DC CL48'CLFEBR max M3 modes 6, 7'
00004870  FFFFFF00 FFFFFF00                         1004  DC XL16'FFFFFF00FFFFFF00000000000000000'
00004880  C3D3C6C5 C2D940F0                         1005  DC CL48'CLFEBR 0.75 FPC modes 1-3, 7'
000048B0  00000000 00000001                         1006  DC XL16'00000000000000010000000000000001'
000048C0  C3D3C6C5 C2D940F0                         1007  DC CL48'CLFEBR 0.75 M3 modes 1, 3-5'
000048F0  00000001 00000001                         1008  DC XL16'00000001000000010000000100000000'
00004900  C3D3C6C5 C2D940F0                         1009  DC CL48'CLFEBR 0.75 M3 modes 6, 7'
00004930  00000001 00000000                         1010  DC XL16'00000001000000000000000000000000'
00004940  C3D3C6C5 C2D940F0                         1011  DC CL48'CLFEBR 0.25 FPC modes 1-3, 7'
00004970  00000000 00000001                         1012  DC XL16'00000000000000010000000000000001'
00004980  C3D3C6C5 C2D940F0                         1013  DC CL48'CLFEBR 0.25 M3 modes 1, 3-5'
000049B0  00000000 00000001                         1014  DC XL16'00000000000000010000000000000000'
000049C0  C3D3C6C5 C2D940F0                         1015  DC CL48'CLFEBR 0.25 M3 modes 6, 7'
000049F0  00000001 00000000                         1016  DC XL16'00000001000000000000000000000000'
                            0000001E  00000001      1017 SINTRMO_NUM EQU (*-SINTRMO_GOOD)/64
                                                    1018 *
                                                    1019 *
                            00004A00  00000001      1020 SINTRMOF_GOOD EQU *
00004A00  C3D3C6C5 C2D94060                         1021  DC CL48'CLFEBR -1.5 FPC modes 1-3, 7 FPCR'
00004A30  00800003 00800003                         1022  DC XL16'00800003008000030080000300800003'
00004A40  C3D3C6C5 C2D94060                         1023  DC CL48'CLFEBR -1.5 M3 modes 1, 3-5 FPCR'
00004A70  00880003 00880003                         1024  DC XL16'00880003008800030088000300880003'
00004A80  C3D3C6C5 C2D94060                         1025  DC CL48'CLFEBR -1.5 M3 modes 6, 7 FPCR'
00004AB0  00880003 00880003                         1026  DC XL16'00880003008800030000000000000000'
00004AC0  C3D3C6C5 C2D94060                         1027  DC CL48'CLFEBR -0.5 FPC modes 1-3, 7 FPCR'
00004AF0  00000001 00000001                         1028  DC XL16'00000001000000010080000300800003'
00004B00  C3D3C6C5 C2D94060                         1029  DC CL48'CLFEBR -0.5 M3 modes 1, 3-5 FPCR'
00004B30  00880003 00880003                         1030  DC XL16'00880003008800030008000100080001'
00004B40  C3D3C6C5 C2D94060                         1031  DC CL48'CLFEBR -0.5 M3 modes 6, 7 FPCR'
00004B70  00080001 00880003                         1032  DC XL16'00080001008800030000000000000000'
00004B80  C3D3C6C5 C2D9404E                         1033  DC CL48'CLFEBR +0.5 FPC modes 1-3, 7 FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00004BB0  00000002 00000002                         1034  DC XL16'00000002000000020000000200000002'
00004BC0  C3D3C6C5 C2D9404E                         1035  DC CL48'CLFEBR +0.5 M3 modes 1, 3-5 FPCR'
00004BF0  00080002 00080002                         1036  DC XL16'00080002000800020008000200080002'
00004C00  C3D3C6C5 C2D9404E                         1037  DC CL48'CLFEBR +0.5 M3 modes 6, 7 FPCR'
00004C30  00080002 00080002                         1038  DC XL16'00080002000800020000000000000000'
00004C40  C3D3C6C5 C2D9404E                         1039  DC CL48'CLFEBR +1.5 FPC modes 1-3, 7 FPCR'
00004C70  00000002 00000002                         1040  DC XL16'00000002000000020000000200000002'
00004C80  C3D3C6C5 C2D9404E                         1041  DC CL48'CLFEBR +1.5 M3 modes 1, 3-5 FPCR'
00004CB0  00080002 00080002                         1042  DC XL16'00080002000800020008000200080002'
00004CC0  C3D3C6C5 C2D9404E                         1043  DC CL48'CLFEBR +1.5 M3 modes 6, 7 FPCR'
00004CF0  00080002 00080002                         1044  DC XL16'00080002000800020000000000000000'
00004D00  C3D3C6C5 C2D9404E                         1045  DC CL48'CLFEBR +2.5 FPC modes 1-3, 7 FPCR'
00004D30  00000002 00000002                         1046  DC XL16'00000002000000020000000200000002'
00004D40  C3D3C6C5 C2D9404E                         1047  DC CL48'CLFEBR +2.5 M3 modes 1, 3-5 FPCR'
00004D70  00080002 00080002                         1048  DC XL16'00080002000800020008000200080002'
00004D80  C3D3C6C5 C2D9404E                         1049  DC CL48'CLFEBR +2.5 M3 modes 6, 7 FPCR'
00004DB0  00080002 00080002                         1050  DC XL16'00080002000800020000000000000000'
00004DC0  C3D3C6C5 C2D9404E                         1051  DC CL48'CLFEBR +5.5 FPC modes 1-3, 7 FPCR'
00004DF0  00000002 00000002                         1052  DC XL16'00000002000000020000000200000002'
00004E00  C3D3C6C5 C2D9404E                         1053  DC CL48'CLFEBR +5.5 M3 modes 1, 3-5 FPCR'
00004E30  00080002 00080002                         1054  DC XL16'00080002000800020008000200080002'
00004E40  C3D3C6C5 C2D9404E                         1055  DC CL48'CLFEBR +5.5 M3 modes 6, 7 FPCR'
00004E70  00080002 00080002                         1056  DC XL16'00080002000800020000000000000000'
00004E80  C3D3C6C5 C2D9404E                         1057  DC CL48'CLFEBR +9.5 FPC modes 1-3, 7 FPCR'
00004EB0  00000002 00000002                         1058  DC XL16'00000002000000020000000200000002'
00004EC0  C3D3C6C5 C2D9404E                         1059  DC CL48'CLFEBR +9.5 M3 modes 1, 3-5 FPCR'
00004EF0  00080002 00080002                         1060  DC XL16'00080002000800020008000200080002'
00004F00  C3D3C6C5 C2D9404E                         1061  DC CL48'CLFEBR +9.5 M3 modes 6, 7 FPCR'
00004F30  00080002 00080002                         1062  DC XL16'00080002000800020000000000000000'
00004F40  C3D3C6C5 C2D94094                         1063  DC CL48'CLFEBR max FPC modes 1-3, 7 FPCR'
00004F70  00000002 00000002                         1064  DC XL16'00000002000000020000000200000002'
00004F80  C3D3C6C5 C2D94094                         1065  DC CL48'CLFEBR max M3 modes 1, 3-5 FPCR'
00004FB0  00000002 00000002                         1066  DC XL16'00000002000000020000000200000002'
00004FC0  C3D3C6C5 C2D94094                         1067  DC CL48'CLFEBR max M3 modes 5-7'
00004FF0  00000002 00000002                         1068  DC XL16'00000002000000020000000000000000'
00005000  C3D3C6C5 C2D9404E                         1069  DC CL48'CLFEBR +0.75 FPC modes 1-3, 7 FPCR'
00005030  00000002 00000002                         1070  DC XL16'00000002000000020000000200000002'
00005040  C3D3C6C5 C2D9404E                         1071  DC CL48'CLFEBR +0.75 M3 modes 1, 3-5 FPCR'
00005070  00080002 00080002                         1072  DC XL16'00080002000800020008000200080002'
00005080  C3D3C6C5 C2D9404E                         1073  DC CL48'CLFEBR +0.75 M3 modes 6, 7 FPCR'
000050B0  00080002 00080002                         1074  DC XL16'00080002000800020000000000000000'
000050C0  C3D3C6C5 C2D9404E                         1075  DC CL48'CLFEBR +0.25 FPC modes 1-3, 7 FPCR'
000050F0  00000002 00000002                         1076  DC XL16'00000002000000020000000200000002'
00005100  C3D3C6C5 C2D9404E                         1077  DC CL48'CLFEBR +0.25 M3 modes 1, 3-5 FPCR'
00005130  00080002 00080002                         1078  DC XL16'00080002000800020008000200080002'
00005140  C3D3C6C5 C2D9404E                         1079  DC CL48'CLFEBR +0.25 M3 modes 6, 7 FPCR'
00005170  00080002 00080002                         1080  DC XL16'00080002000800020000000000000000'
                            0000001E  00000001      1081 SINTRMOF_NUM EQU (*-SINTRMOF_GOOD)/64
                                                    1082 *
                                                    1083 *
                            00005180  00000001      1084 LINTOUT_GOOD EQU *
00005180  C3D3C6C4 C2D94099                         1085  DC CL48'CLFDBR result pairs 1-2'
000051B0  00000001 00000001                         1086  DC XL16'00000001000000010000000200000002'
000051C0  C3D3C6C4 C2D94099                         1087  DC CL48'CLFDBR result pairs 3-4'
000051F0  00000004 00000004                         1088  DC XL16'00000004000000040000000000000000'
00005200  C3D3C6C4 C2D94099                         1089  DC CL48'CLFDBR result pairs 5-6'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00005230  00000000 00000000                        1090  DC XL16'0000000000000000FFFFFFFF00000000'
00005240  C3D3C6C4 C2D94099                        1091  DC CL48'CLFDBR result pairs 7-8'
00005270  FFFFFFFF FFFFFFFF                        1092  DC XL16'FFFFFFFFFFFFFFFF0000000100000001'
00005280  C3D3C6C4 C2D94099                        1093  DC CL48'CLFDBR result pair 9'
000052B0  00000000 00000000                        1094  DC XL16'00000000000000000000000000000000'
                             00000005  00000001   1095 LINTOUT_NUM EQU (*-LINTOUT_GOOD)/64
                                                  1096 *
                                                  1097 *
                             000052C0  00000001   1098 LINTFLGS_GOOD EQU *
000052C0  C3D3C6C4 C2D940C6                        1099  DC CL48'CLFDBR FPC pairs 1-2'
000052F0  00000002 F8000002                        1100  DC XL16'00000002F800000200000002F8000002'
00005300  C3D3C6C4 C2D940C6                        1101  DC CL48'CLFDBR FPC pairs 3-4'
00005330  00000002 F8000002                        1102  DC XL16'00000002F800000200880003F8008000'
00005340  C3D3C6C4 C2D940C6                        1103  DC CL48'CLFDBR FPC pairs 5-6'
00005370  00880003 F8008000                        1104  DC XL16'00880003F800800000880003F8008000'
00005380  C3D3C6C4 C2D940C6                        1105  DC CL48'CLFDBR FPC pairs 7-8'
000053B0  00080002 F8000802                        1106  DC XL16'00080002F800080200080002F8000C02'
000053C0  C3D3C6C4 C2D940C6                        1107  DC CL48'CLFDBR FPC pair 9'
000053F0  00080002 F8000802                        1108  DC XL16'00080002F8000802000000000000000'
                             00000005  00000001   1109 LINTFLGS_NUM EQU (*-LINTFLGS_GOOD)/64
                                                  1110 *
                                                  1111 *
                             00005400  00000001   1112 LINTRMO_GOOD EQU *
00005400  C3D3C6C4 C2D94060                        1113  DC CL48'CLFDBR -1.5 FPC modes 1-3, 7'
00005430  00000000 00000000                        1114  DC XL16'00000000000000000000000000000000'
00005440  C3D3C6C4 C2D94060                        1115  DC CL48'CLFDBR -1.5 M3 modes 1, 3-5'
00005470  00000000 00000000                        1116  DC XL16'00000000000000000000000000000000'
00005480  C3D3C6C4 C2D94060                        1117  DC CL48'CLFDBR -1.5 M3 modes 6, 7'
000054B0  00000000 00000000                        1118  DC XL16'00000000000000000000000000000000'
000054C0  C3D3C6C4 C2D94060                        1119  DC CL48'CLFDBR -0.5 FPC modes 1-3, 7'
000054F0  00000000 00000000                        1120  DC XL16'00000000000000000000000000000000'
00005500  C3D3C6C4 C2D94060                        1121  DC CL48'CLFDBR -0.5 M3 modes 1, 3-5'
00005530  00000000 00000000                        1122  DC XL16'00000000000000000000000000000000'
00005540  C3D3C6C4 C2D94060                        1123  DC CL48'CLFDBR -0.5 M3 modes 6, 7'
00005570  00000000 00000000                        1124  DC XL16'00000000000000000000000000000000'
00005580  C3D3C6C4 C2D940F0                        1125  DC CL48'CLFDBR 0.5 FPC modes 1-3, 7'
000055B0  00000000 00000001                        1126  DC XL16'00000000000000010000000000000001'
000055C0  C3D3C6C4 C2D940F0                        1127  DC CL48'CLFDBR 0.5 M3 modes 1, 3-5'
000055F0  00000001 00000001                        1128  DC XL16'00000001000000010000000000000000'
00005600  C3D3C6C4 C2D940F0                        1129  DC CL48'CLFDBR 0.5 M3 modes 6, 7'
00005630  00000001 00000000                        1130  DC XL16'00000001000000000000000000000000'
00005640  C3D3C6C4 C2D940F1                        1131  DC CL48'CLFDBR 1.5 FPC modes 1-3, 7'
00005670  00000001 00000002                        1132  DC XL16'00000001000000020000000100000001'
00005680  C3D3C6C4 C2D940F1                        1133  DC CL48'CLFDBR 1.5 M3 modes 1, 3-5'
000056B0  00000002 00000001                        1134  DC XL16'00000002000000010000000200000001'
000056C0  C3D3C6C4 C2D940F1                        1135  DC CL48'CLFDBR 1.5 M3 modes 6, 7'
000056F0  00000002 00000001                        1136  DC XL16'00000002000000010000000000000000'
00005700  C3D3C6C4 C2D940F2                        1137  DC CL48'CLFDBR 2.5 FPC modes 1-3, 7'
00005730  00000002 00000003                        1138  DC XL16'00000002000000030000000200000003'
00005740  C3D3C6C4 C2D940F2                        1139  DC CL48'CLFDBR 2.5 M3 modes 1, 3-5'
00005770  00000003 00000003                        1140  DC XL16'00000003000000030000000200000002'
00005780  C3D3C6C4 C2D940F2                        1141  DC CL48'CLFDBR 2.5 M3 modes 6, 7'
000057B0  00000003 00000002                        1142  DC XL16'00000003000000020000000000000000'
000057C0  C3D3C6C4 C2D940F5                        1143  DC CL48'CLFDBR 5.5 FPC modes 1-3, 7'
000057F0  00000005 00000006                        1144  DC XL16'00000005000000060000000500000005'
00005800  C3D3C6C4 C2D940F5                        1145  DC CL48'CLFDBR 5.5 M3 modes 1, 3-5'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00005830  00000006 00000005                         1146  DC XL16'00000006000000050000000600000005'
00005840  C3D3C6C4 C2D940F5                         1147  DC CL48'CLFDBR 5.5 M3 modes 6, 7'
00005870  00000006 00000005                         1148  DC XL16'00000006000000050000000000000000'
00005880  C3D3C6C4 C2D940F9                         1149  DC CL48'CLFDBR 9.5 FPC modes 1-3, 7'
000058B0  00000009 0000000A                         1150  DC XL16'0000000900000000A0000000900000009'
000058C0  C3D3C6C4 C2D940F9                         1151  DC CL48'CLFDBR 9.5 M3 modes 1, 3-5'
000058F0  0000000A 00000009                         1152  DC XL16'0000000A000000090000000A00000009'
00005900  C3D3C6C4 C2D940F9                         1153  DC CL48'CLFDBR 9.5 M3 modes 6, 7'
00005930  0000000A 00000009                         1154  DC XL16'0000000A0000000900000000000000000'
00005940  C3D3C6C4 C2D94094                         1155  DC CL48'CLFDBR max FPC modes 1-3, 7'
00005970  FFFFFFFF FFFFFFFF                         1156  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00005980  C3D3C6C4 C2D94094                         1157  DC CL48'CLFDBR max M3 modes 1, 3-5'
000059B0  FFFFFFFF FFFFFFFF                         1158  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
000059C0  C3D3C6C4 C2D94094                         1159  DC CL48'CLFDBR max M3 modes 6, 7'
000059F0  FFFFFFFF FFFFFFFF                         1160  DC XL16'FFFFFFFFFFFFFFFF0000000000000000'
00005A00  C3D3C6C4 C2D940F0                         1161  DC CL48'CLFDBR 0.75 FPC modes 1-3, 7'
00005A30  00000000 00000001                         1162  DC XL16'00000000000000010000000000000001'
00005A40  C3D3C6C4 C2D940F0                         1163  DC CL48'CLFDBR 0.75 M3 modes 1, 3-5'
00005A70  00000001 00000001                         1164  DC XL16'00000001000000010000000100000000'
00005A80  C3D3C6C4 C2D940F0                         1165  DC CL48'CLFDBR 0.75 M3 modes 6, 7'
00005AB0  00000001 00000000                         1166  DC XL16'00000001000000000000000000000000'
00005AC0  C3D3C6C4 C2D940F0                         1167  DC CL48'CLFDBR 0.25 FPC modes 1-3, 7'
00005AF0  00000000 00000001                         1168  DC XL16'00000000000000010000000000000001'
00005B00  C3D3C6C4 C2D940F0                         1169  DC CL48'CLFDBR 0.25 M3 modes 1, 3-5'
00005B30  00000000 00000001                         1170  DC XL16'00000000000000010000000000000000'
00005B40  C3D3C6C4 C2D940F0                         1171  DC CL48'CLFDBR 0.25 M3 modes 6, 7'
00005B70  00000001 00000000                         1172  DC XL16'00000001000000000000000000000000'
                            0000001E  00000001      1173 LINTRMO_NUM EQU (*-LINTRMO_GOOD)/64
                                                    1174 *
                                                    1175 *
                            00005B80  00000001      1176 LINTRMOF_GOOD EQU *
00005B80  C3D3C6C4 C2D94060                         1177  DC CL48'CLFDBR -1.5 FPC modes 1-3, 7 FPCR'
00005BB0  00800003 00800003                         1178  DC XL16'00800003008000030080000300800003'
00005BC0  C3D3C6C4 C2D94060                         1179  DC CL48'CLFDBR -1.5 M3 modes 1, 3-5 FPCR'
00005BF0  00880003 00880003                         1180  DC XL16'00880003008800030088000300880003'
00005C00  C3D3C6C4 C2D94060                         1181  DC CL48'CLFDBR -1.5 M3 modes 6, 7 FPCR'
00005C30  00880003 00880003                         1182  DC XL16'00880003008800030000000000000000'
00005C40  C3D3C6C4 C2D94060                         1183  DC CL48'CLFDBR -0.5 FPC modes 1-3, 7 FPCR'
00005C70  00000001 00000001                         1184  DC XL16'00000001000000010080000300800003'
00005C80  C3D3C6C4 C2D94060                         1185  DC CL48'CLFDBR -0.5 M3 modes 1, 3-5 FPCR'
00005CB0  00880003 00880003                         1186  DC XL16'00880003008800030008000100080001'
00005CC0  C3D3C6C4 C2D94060                         1187  DC CL48'CLFDBR -0.5 M3 modes 6, 7 FPCR'
00005CF0  00080001 00880003                         1188  DC XL16'00080001008800030000000000000000'
00005D00  C3D3C6C4 C2D9404E                         1189  DC CL48'CLFDBR +0.5 FPC modes 1-3, 7 FPCR'
00005D30  00000002 00000002                         1190  DC XL16'00000002000000020000000200000002'
00005D40  C3D3C6C4 C2D9404E                         1191  DC CL48'CLFDBR +0.5 M3 modes 1, 3-5 FPCR'
00005D70  00080002 00080002                         1192  DC XL16'00080002000800020008000200080002'
00005D80  C3D3C6C4 C2D9404E                         1193  DC CL48'CLFDBR +0.5 M3 modes 6, 7 FPCR'
00005DB0  00080002 00080002                         1194  DC XL16'00080002000800020000000000000000'
00005DC0  C3D3C6C4 C2D9404E                         1195  DC CL48'CLFDBR +1.5 FPC modes 1-3, 7 FPCR'
00005DF0  00000002 00000002                         1196  DC XL16'00000002000000020000000200000002'
00005E00  C3D3C6C4 C2D9404E                         1197  DC CL48'CLFDBR +1.5 M3 modes 1, 3-5 FPCR'
00005E30  00080002 00080002                         1198  DC XL16'00080002000800020008000200080002'
00005E40  C3D3C6C4 C2D9404E                         1199  DC CL48'CLFDBR +1.5 M3 modes 6, 7 FPCR'
00005E70  00080002 00080002                         1200  DC XL16'00080002000800020000000000000000'
00005E80  C3D3C6C4 C2D9404E                         1201  DC CL48'CLFDBR +2.5 FPC modes 1-3, 7 FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

00005EB0  00000002 00000002                         1202  DC XL16'00000002000000020000000200000002'
00005EC0  C3D3C6C4 C2D9404E                         1203  DC CL48'CLFDBR +2.5 M3 modes 1, 3-5 FPCR'
00005EF0  00080002 00080002                         1204  DC XL16'00080002000800020008000200080002'
00005F00  C3D3C6C4 C2D9404E                         1205  DC CL48'CLFDBR +2.5 M3 modes 6, 7 FPCR'
00005F30  00080002 00080002                         1206  DC XL16'00080002000800020000000000000000'
00005F40  C3D3C6C4 C2D9404E                         1207  DC CL48'CLFDBR +5.5 FPC modes 1-3, 7 FPCR'
00005F70  00000002 00000002                         1208  DC XL16'00000002000000020000000200000002'
00005F80  C3D3C6C4 C2D9404E                         1209  DC CL48'CLFDBR +5.5 M3 modes 1, 3-5 FPCR'
00005FB0  00080002 00080002                         1210  DC XL16'00080002000800020008000200080002'
00005FC0  C3D3C6C4 C2D9404E                         1211  DC CL48'CLFDBR +5.5 M3 modes 6, 7 FPCR'
00005FF0  00080002 00080002                         1212  DC XL16'00080002000800020000000000000000'
00006000  C3D3C6C4 C2D9404E                         1213  DC CL48'CLFDBR +9.5 FPC modes 1-3, 7 FPCR'
00006030  00000002 00000002                         1214  DC XL16'00000002000000020000000200000002'
00006040  C3D3C6C4 C2D9404E                         1215  DC CL48'CLFDBR +9.5 M3 modes 1, 3-5 FPCR'
00006070  00080002 00080002                         1216  DC XL16'00080002000800020008000200080002'
00006080  C3D3C6C4 C2D9404E                         1217  DC CL48'CLFDBR +9.5 M3 modes 6, 7 FPCR'
000060B0  00080002 00080002                         1218  DC XL16'00080002000800020000000000000000'
000060C0  C3D3C6C4 C2D94094                         1219  DC CL48'CLFDBR max FPC modes 1-3, 7 FPCR'
000060F0  00000002 00800003                         1220  DC XL16'00000002008000030000000200000002'
00006100  C3D3C6C4 C2D94094                         1221  DC CL48'CLFDBR max M3 modes 1, 3-5 FPCR'
00006130  00880003 00080002                         1222  DC XL16'00880003000800020088000300080002'
00006140  C3D3C6C4 C2D94094                         1223  DC CL48'CLFDBR max M3 modes 6, 7 FPCR'
00006170  00880003 00080002                         1224  DC XL16'00880003000800020000000000000000'
00006180  C3D3C6C4 C2D9404E                         1225  DC CL48'CLFDBR +0.75 FPC modes 1-3, 7 FPCR'
000061B0  00000002 00000002                         1226  DC XL16'00000002000000020000000200000002'
000061C0  C3D3C6C4 C2D9404E                         1227  DC CL48'CLFDBR +0.75 M3 modes 1, 3-5 FPCR'
000061F0  00080002 00080002                         1228  DC XL16'00080002000800020008000200080002'
00006200  C3D3C6C4 C2D9404E                         1229  DC CL48'CLFDBR +0.75 M3 modes 6, 7 FPCR'
00006230  00080002 00080002                         1230  DC XL16'00080002000800020000000000000000'
00006240  C3D3C6C4 C2D9404E                         1231  DC CL48'CLFDBR +0.25 FPC modes 1-3, 7 FPCR'
00006270  00000002 00000002                         1232  DC XL16'00000002000000020000000200000002'
00006280  C3D3C6C4 C2D9404E                         1233  DC CL48'CLFDBR +0.25 M3 modes 1, 3-5 FPCR'
000062B0  00080002 00080002                         1234  DC XL16'00080002000800020008000200080002'
000062C0  C3D3C6C4 C2D9404E                         1235  DC CL48'CLFDBR +0.25 M3 modes 6, 7 FPCR'
000062F0  00080002 00080002                         1236  DC XL16'00080002000800020000000000000000'
                            0000001E  00000001      1237 LINTRMOF_NUM EQU (*-LINTRMOF_GOOD)/64
                                                    1238 *
                                                    1239 *
                            00006300  00000001      1240 XINTOUT_GOOD EQU *
00006300  C3D3C6E7 C2D94099                         1241  DC CL48'CLFXBR result pairs 1-2'
00006330  00000001 00000001                         1242  DC XL16'00000001000000010000000200000002'
00006340  C3D3C6E7 C2D94099                         1243  DC CL48'CLFXBR result pairs 3-4'
00006370  00000004 00000004                         1244  DC XL16'00000004000000040000000000000000'
00006380  C3D3C6E7 C2D94099                         1245  DC CL48'CLFXBR result pairs 5-6'
000063B0  00000000 00000000                         1246  DC XL16'00000000000000000FFFFFFF00000000'
000063C0  C3D3C6E7 C2D94099                         1247  DC CL48'CLFXBR result pairs 7-8'
000063F0  FFFFFFFF FFFFFFFF                         1248  DC XL16'FFFFFFFFFFFFFFFF0000000100000001'
00006400  C3D3C6E7 C2D94099                         1249  DC CL48'CLFXBR result pair 9'
00006430  00000000 00000000                         1250  DC XL16'00000000000000000000000000000000'
                            00000005  00000001      1251 XINTOUT_NUM EQU (*-XINTOUT_GOOD)/64
                                                    1252 *
                                                    1253 *
                            00006440  00000001      1254 XINTFLGS_GOOD EQU *
00006440  C3D3C6E7 C2D940C6                         1255  DC CL48'CLFXBR FPC pairs 1-2'
00006470  00000002 F8000002                         1256  DC XL16'00000002F800000200000002F8000002'
00006480  C3D3C6E7 C2D940C6                         1257  DC CL48'CLFXBR FPC pairs 3-4'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

000064B0  00000002 F8000002                       1258  DC XL16'00000002F800000200880003F8008000'
000064C0  C3D3C6E7 C2D940C6                       1259  DC CL48'CLFXBR FPC pairs 5-6'
000064F0  00880003 F8008000                       1260  DC XL16'00880003F800800000880003F8008000'
00006500  C3D3C6E7 C2D940C6                       1261  DC CL48'CLFXBR FPC pairs 7-8'
00006530  00080002 F8000802                       1262  DC XL16'00080002F800080200080002F8000C02'
00006540  C3D3C6E7 C2D940C6                       1263  DC CL48'CLFXBR FPC pair 9'
00006570  00080002 F8000802                       1264  DC XL16'00080002F8000802000000000000000'
                             00000005  00000001  1265 XINTFLGS_NUM EQU (*-XINTFLGS_GOOD)/64
                                                 1266 *
                                                 1267 *
                             00006580  00000001  1268 XINTRMO_GOOD EQU *
00006580  C3D3C6E7 C2D94060                       1269  DC CL48'CLFXBR -1.5 FPC modes 1-3, 7'
000065B0  00000000 00000000                       1270  DC XL16'00000000000000000000000000000000'
000065C0  C3D3C6E7 C2D94060                       1271  DC CL48'CLFXBR -1.5 M3 modes 1, 3-5'
000065F0  00000000 00000000                       1272  DC XL16'00000000000000000000000000000000'
00006600  C3D3C6E7 C2D94060                       1273  DC CL48'CLFXBR -1.5 M3 modes 6, 7'
00006630  00000000 00000000                       1274  DC XL16'00000000000000000000000000000000'
00006640  C3D3C6E7 C2D94060                       1275  DC CL48'CLFXBR -0.5 FPC modes 1-3, 7'
00006670  00000000 00000000                       1276  DC XL16'00000000000000000000000000000000'
00006680  C3D3C6E7 C2D94060                       1277  DC CL48'CLFXBR -0.5 M3 modes 1, 3-5'
000066B0  00000000 00000000                       1278  DC XL16'00000000000000000000000000000000'
000066C0  C3D3C6E7 C2D94060                       1279  DC CL48'CLFXBR -0.5 M3 modes 6, 7'
000066F0  00000000 00000000                       1280  DC XL16'00000000000000000000000000000000'
00006700  C3D3C6E7 C2D940F0                       1281  DC CL48'CLFXBR 0.5 FPC modes 1-3, 7'
00006730  00000000 00000001                       1282  DC XL16'00000000000000010000000000000001'
00006740  C3D3C6E7 C2D940F0                       1283  DC CL48'CLFXBR 0.5 M3 modes 1, 3-5'
00006770  00000001 00000001                       1284  DC XL16'00000001000000010000000000000000'
00006780  C3D3C6E7 C2D940F0                       1285  DC CL48'CLFXBR 0.5 M3 modes 6, 7'
000067B0  00000001 00000000                       1286  DC XL16'00000001000000000000000000000000'
000067C0  C3D3C6E7 C2D940F1                       1287  DC CL48'CLFXBR 1.5 FPC modes 1-3, 7'
000067F0  00000001 00000002                       1288  DC XL16'00000001000000020000000100000001'
00006800  C3D3C6E7 C2D940F1                       1289  DC CL48'CLFXBR 1.5 M3 modes 1, 3-5'
00006830  00000002 00000001                       1290  DC XL16'00000002000000010000000200000001'
00006840  C3D3C6E7 C2D940F1                       1291  DC CL48'CLFXBR 1.5 M3 modes 6, 7'
00006870  00000002 00000001                       1292  DC XL16'00000002000000010000000000000000'
00006880  C3D3C6E7 C2D940F2                       1293  DC CL48'CLFXBR 2.5 FPC modes 1-3, 7'
000068B0  00000002 00000003                       1294  DC XL16'00000002000000030000000200000003'
000068C0  C3D3C6E7 C2D940F2                       1295  DC CL48'CLFXBR 2.5 M3 modes 1, 3-5'
000068F0  00000003 00000003                       1296  DC XL16'00000003000000030000000200000002'
00006900  C3D3C6E7 C2D940F2                       1297  DC CL48'CLFXBR 2.5 M3 modes 6, 7'
00006930  00000003 00000002                       1298  DC XL16'00000003000000020000000000000000'
00006940  C3D3C6E7 C2D940F5                       1299  DC CL48'CLFXBR 5.5 FPC modes 1-3, 7'
00006970  00000005 00000006                       1300  DC XL16'00000005000000060000000500000005'
00006980  C3D3C6E7 C2D940F5                       1301  DC CL48'CLFXBR 5.5 M3 modes 1, 3-5'
000069B0  00000006 00000005                       1302  DC XL16'00000006000000050000000600000005'
000069C0  C3D3C6E7 C2D940F5                       1303  DC CL48'CLFXBR 5.5 M3 modes 6, 7'
000069F0  00000006 00000005                       1304  DC XL16'00000006000000050000000000000000'
00006A00  C3D3C6E7 C2D940F9                       1305  DC CL48'CLFXBR 9.5 FPC modes 1-3, 7'
00006A30  00000009 0000000A                       1306  DC XL16'00000009000000A0000000900000009'
00006A40  C3D3C6E7 C2D940F9                       1307  DC CL48'CLFXBR 9.5 M3 modes 1, 3-5'
00006A70  0000000A 00000009                       1308  DC XL16'0000000A00000009000000A00000009'
00006A80  C3D3C6E7 C2D940F9                       1309  DC CL48'CLFXBR 9.5 M3 modes 6, 7'
00006AB0  0000000A 00000009                       1310  DC XL16'0000000A00000009000000000000000'
00006AC0  C3D3C6E7 C2D94094                       1311  DC CL48'CLFXBR max FPC modes 1-3, 7'
00006AF0  FFFFFFFF FFFFFFFF                       1312  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00006B00  C3D3C6E7 C2D94094                       1313  DC CL48'CLFXBR max M3 modes 1, 3-5'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00006B30  FFFFFFFF FFFFFFFF                        1314  DC XL16'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00006B40  C3D3C6E7 C2D94094                        1315  DC CL48'CLFXBR max M3 modes 6, 7'
00006B70  FFFFFFFF FFFFFFFF                        1316  DC XL16'FFFFFFFFFFFFFFFF0000000000000000'
00006B80  C3D3C6E7 C2D940F0                        1317  DC CL48'CLFXBR 0.75 FPC modes 1-3, 7'
00006BB0  00000000 00000001                        1318  DC XL16'00000000000000001000000000000001'
00006BC0  C3D3C6E7 C2D940F0                        1319  DC CL48'CLFXBR 0.75 M3 modes 1, 3-5'
00006BF0  00000001 00000001                        1320  DC XL16'00000001000000010000000100000000'
00006C00  C3D3C6E7 C2D940F0                        1321  DC CL48'CLFXBR 0.75 M3 modes 6, 7'
00006C30  00000001 00000000                        1322  DC XL16'00000001000000000000000000000000'
00006C40  C3D3C6E7 C2D940F0                        1323  DC CL48'CLFXBR 0.25 FPC modes 1-3, 7'
00006C70  00000000 00000001                        1324  DC XL16'00000000000000001000000000000001'
00006C80  C3D3C6E7 C2D940F0                        1325  DC CL48'CLFXBR 0.25 M3 modes 1, 3-5'
00006CB0  00000000 00000001                        1326  DC XL16'00000000000000001000000000000000'
00006CC0  C3D3C6E7 C2D940F0                        1327  DC CL48'CLFXBR 0.25 M3 modes 6, 7'
00006CF0  00000001 00000000                        1328  DC XL16'00000001000000000000000000000000'
                            0000001E  00000001     1329 XINTRMO_NUM EQU (*-XINTRMO_GOOD)/64
                                                   1330 *
                                                   1331 *
                            00006D00  00000001     1332 XINTRMOF_GOOD EQU *
00006D00  C3D3C6E7 C2D94060                        1333  DC CL48'CLFXBR -1.5 FPC modes 1-3, 7 FPCR'
00006D30  00800003 00800003                        1334  DC XL16'00800003008000030080000300800003'
00006D40  C3D3C6E7 C2D94060                        1335  DC CL48'CLFXBR -1.5 M3 modes 1, 3-5 FPCR'
00006D70  00880003 00880003                        1336  DC XL16'00880003008800030088000300880003'
00006D80  C3D3C6E7 C2D94060                        1337  DC CL48'CLFXBR -1.5 M3 modes 6, 7 FPCR'
00006DB0  00880003 00880003                        1338  DC XL16'00880003008800030000000000000000'
00006DC0  C3D3C6E7 C2D94060                        1339  DC CL48'CLFXBR -0.5 FPC modes 1-3, 7 FPCR'
00006DF0  00000001 00000001                        1340  DC XL16'00000001000000010080000300800003'
00006E00  C3D3C6E7 C2D94060                        1341  DC CL48'CLFXBR -0.5 M3 modes 1, 3-5 FPCR'
00006E30  00880003 00880003                        1342  DC XL16'00880003008800030008000100080001'
00006E40  C3D3C6E7 C2D94060                        1343  DC CL48'CLFXBR -0.5 M3 modes 6, 7 FPCR'
00006E70  00080001 00880003                        1344  DC XL16'00080001008800030000000000000000'
00006E80  C3D3C6E7 C2D9404E                        1345  DC CL48'CLFXBR +0.5 FPC modes 1-3, 7 FPCR'
00006EB0  00000002 00000002                        1346  DC XL16'00000002000000020000000200000002'
00006EC0  C3D3C6E7 C2D9404E                        1347  DC CL48'CLFXBR +0.5 M3 modes 1, 3-5 FPCR'
00006EF0  00080002 00080002                        1348  DC XL16'00080002000800020008000200080002'
00006F00  C3D3C6E7 C2D9404E                        1349  DC CL48'CLFXBR +0.5 M3 modes 6, 7 FPCR'
00006F30  00080002 00080002                        1350  DC XL16'00080002000800020000000000000000'
00006F40  C3D3C6E7 C2D9404E                        1351  DC CL48'CLFXBR +1.5 FPC modes 1-3, 7 FPCR'
00006F70  00000002 00000002                        1352  DC XL16'00000002000000020000000200000002'
00006F80  C3D3C6E7 C2D9404E                        1353  DC CL48'CLFXBR +1.5 M3 modes 1, 3-5 FPCR'
00006FB0  00080002 00080002                        1354  DC XL16'00080002000800020008000200080002'
00006FC0  C3D3C6E7 C2D9404E                        1355  DC CL48'CLFXBR +1.5 M3 modes 6, 7 FPCR'
00006FF0  00080002 00080002                        1356  DC XL16'00080002000800020000000000000000'
00007000  C3D3C6E7 C2D9404E                        1357  DC CL48'CLFXBR +2.5 FPC modes 1-3, 7 FPCR'
00007030  00000002 00000002                        1358  DC XL16'00000002000000020000000200000002'
00007040  C3D3C6E7 C2D9404E                        1359  DC CL48'CLFXBR +2.5 M3 modes 1, 3-5 FPCR'
00007070  00080002 00080002                        1360  DC XL16'00080002000800020008000200080002'
00007080  C3D3C6E7 C2D9404E                        1361  DC CL48'CLFXBR +2.5 M3 modes 5-7'
000070B0  00080002 00080002                        1362  DC XL16'00080002000800020000000000000000'
000070C0  C3D3C6E7 C2D9404E                        1363  DC CL48'CLFXBR +5.5 FPC modes 1-3, 7 FPCR'
000070F0  00000002 00000002                        1364  DC XL16'00000002000000020000000200000002'
00007100  C3D3C6E7 C2D9404E                        1365  DC CL48'CLFXBR +5.5 M3 modes 1, 3-5 FPCR'
00007130  00080002 00080002                        1366  DC XL16'00080002000800020008000200080002'
00007140  C3D3C6E7 C2D9404E                        1367  DC CL48'CLFXBR +5.5 M3 modes 6, 7 FPCR'
00007170  00080002 00080002                        1368  DC XL16'00080002000800020000000000000000'
00007180  C3D3C6E7 C2D9404E                        1369  DC CL48'CLFXBR +9.5 FPC modes 1-3, 7 FPCR'
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

000071B0  00000002 00000002                         1370  DC XL16'00000002000000020000000200000002'
000071C0  C3D3C6E7 C2D9404E                         1371  DC CL48'CLFXBR +9.5 M3 modes 1, 3-5 FPCR'
000071F0  00080002 00080002                         1372  DC XL16'00080002000800020008000200080002'
00007200  C3D3C6E7 C2D9404E                         1373  DC CL48'CLFXBR +9.5 M3 modes 6, 7 FPCR'
00007230  00080002 00080002                         1374  DC XL16'00080002000800020000000000000000'
00007240  C3D3C6E7 C2D94094                         1375  DC CL48'CLFXBR max FPC modes 1-3, 7 FPCR'
00007270  00000002 00800003                         1376  DC XL16'00000002008000030000000200000002'
00007280  C3D3C6E7 C2D94094                         1377  DC CL48'CLFXBR max M3 modes 1, 3-5 FPCR'
000072B0  00880003 00080002                         1378  DC XL16'00880003000800020088000300080002'
000072C0  C3D3C6E7 C2D94094                         1379  DC CL48'CLFXBR max M3 modes 6, 7 FPCR'
000072F0  00880003 00080002                         1380  DC XL16'00880003000800020000000000000000'
00007300  C3D3C6E7 C2D9404E                         1381  DC CL48'CLFXBR +0.75 FPC modes 1-3, 7 FPCR'
00007330  00000002 00000002                         1382  DC XL16'00000002000000020000000200000002'
00007340  C3D3C6E7 C2D9404E                         1383  DC CL48'CLFXBR +0.75 M3 modes 1, 3-5 FPCR'
00007370  00080002 00080002                         1384  DC XL16'00080002000800020008000200080002'
00007380  C3D3C6E7 C2D9404E                         1385  DC CL48'CLFXBR +0.75 M3 modes 6, 7 FPCR'
000073B0  00080002 00080002                         1386  DC XL16'00080002000800020000000000000000'
000073C0  C3D3C6E7 C2D9404E                         1387  DC CL48'CLFXBR +0.25 FPC modes 1-3, 7 FPCR'
000073F0  00000002 00000002                         1388  DC XL16'00000002000000020000000200000002'
00007400  C3D3C6E7 C2D9404E                         1389  DC CL48'CLFXBR +0.25 M3 modes 1, 3-5 FPCR'
00007430  00080002 00080002                         1390  DC XL16'00080002000800020008000200080002'
00007440  C3D3C6E7 C2D9404E                         1391  DC CL48'CLFXBR +0.25 M3 modes 6, 7 FPCR'
00007470  00080002 00080002                         1392  DC XL16'00080002000800020000000000000000'
                            0000001E  00000001  1393 XINTRMOF_NUM EQU (*-XINTRMOF_GOOD)/64
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00007480                                          1395 HELPERS  DS    0H       (R12 base of helper subroutines)

                                                  1397 ********************************************************************
                                                  1398 *                 REPORT UNEXPECTED PROGRAM CHECK
                                                  1399 ********************************************************************

00007480                                          1401 PGMCK    DS    0H
00007480  F342 C072 F08E      000074F2  0000008E  1402          UNPK  PROGCODE(L'PROGCODE+1),PCINTCD(L'PCINTCD+1)
00007486  926B C076                     000074F6  1403          MVI   PGMCOMMA,C','
0000748A  DC03 C072 C178      000074F2  000075F8  1404          TR    PROGCODE,HEXTRTAB

00007490  F384 C07C F150      000074FC  00000150  1406          UNPK  PGMPSW+(0*9)(9),PCOLDPSW+(0*4)(5)
00007496  9240 C084                     00007504  1407          MVI   PGMPSW+(0*9)+8,C' '
0000749A  DC07 C07C C178      000074FC  000075F8  1408          TR    PGMPSW+(0*9)(8),HEXTRTAB

000074A0  F384 C085 F154      00007505  00000154  1410          UNPK  PGMPSW+(1*9)(9),PCOLDPSW+(1*4)(5)
000074A6  9240 C08D                     0000750D  1411          MVI   PGMPSW+(1*9)+8,C' '
000074AA  DC07 C085 C178      00007505  000075F8  1412          TR    PGMPSW+(1*9)(8),HEXTRTAB

000074B0  F384 C08E F158      0000750E  00000158  1414          UNPK  PGMPSW+(2*9)(9),PCOLDPSW+(2*4)(5)
000074B6  9240 C096                     00007516  1415          MVI   PGMPSW+(2*9)+8,C' '
000074BA  DC07 C08E C178      0000750E  000075F8  1416          TR    PGMPSW+(2*9)(8),HEXTRTAB

000074C0  F384 C097 F15C      00007517  0000015C  1418          UNPK  PGMPSW+(3*9)(9),PCOLDPSW+(3*4)(5)
000074C6  9240 C09F                     0000751F  1419          MVI   PGMPSW+(3*9)+8,C' '
000074CA  DC07 C097 C178      00007517  000075F8  1420          TR    PGMPSW+(3*9)(8),HEXTRTAB

000074D0  4100 0042                     00000042  1422          LA    R0,L'PROGMSG     R0 <== length of message
000074D4  4110 C05E                     000074DE  1423          LA    R1,PROGMSG       R1 --> the message text itself
000074D8  4520 C27A                     000076FA  1424          BAL   R2,MSG           Go display this message
                                                  1425
000074DC  07FD                                    1426          BR    R13              Return to caller




000074DE                                          1428 PROGMSG  DS   0CL66
000074DE  D7D9D6C7 D9C1D440                       1429          DC    CL20'PROGRAM CHECK! CODE '
000074F2  88888888                                1430 PROGCODE DC    CL4'hhhh'
000074F6  6B                                      1431 PGMCOMMA DC    CL1','
000074F7  40D7E2E6 40                             1432          DC    CL5' PSW '
000074FC  88888888 88888888                       1433 PGMPSW   DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                    1435 ************************************************************************
                                                    1436 *                      VERIFICATION ROUTINE
                                                    1437 ************************************************************************

00007520                                            1439 VERISUB  DS    0H
                                                    1440 *
                                                    1441 **       Loop through the VERIFY TABLE...
                                                    1442 *

00007520  4110 C32C                    000077AC     1444          LA    R1,VERIFTAB      R1 --> Verify table
00007524  4120 000C                    0000000C     1445          LA    R2,VERIFLEN      R2 <== Number of entries
00007528  0D30                                      1446          BASR  R3,0             Set top of loop

0000752A  9846 1000                    00000000     1448          LM    R4,R6,0(R1)      Load verify table values
0000752E  4D70 C0C2                    00007542     1449          BAS   R7,VERIFY        Verify results
00007532  4110 100C                    0000000C     1450          LA    R1,12(,R1)       Next verify table entry
00007536  0623                                      1451          BCTR  R2,R3            Loop through verify table

00007538  9500 C278                    000076F8     1453          CLI   FAILFLAG,X'00'   Did all tests verify okay?
0000753C  078D                                      1454          BER   R13              Yes, return to caller
0000753E  47F0 F238                    00000238     1455          B     FAIL             No, load FAILURE disabled wait PSW


                                                    1457 *
                                                    1458 **       Loop through the ACTUAL / EXPECTED results...
                                                    1459 *

00007542  0D80                                      1461 VERIFY   BASR  R8,0            Set top of loop

00007544  D50F 4000 5030      00000000  00000030    1463          CLC   0(16,R4),48(R5)  Actual results == Expected results?
0000754A  4770 C0DA                     0000755A    1464          BNE   VERIFAIL        No, show failure
0000754E  4140 4010                     00000010    1465 VERINEXT LA    R4,16(,R4)       Next actual result
00007552  4150 5040                     00000040    1466          LA    R5,64(,R5)       Next expected result
00007556  0668                                      1467          BCTR  R6,R8            Loop through results

00007558  07F7                                      1469          BR    R7               Return to caller
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                          1471 ************************************************************************
                                          1472 *                     Report the failure...
                                          1473 ************************************************************************

0000755A  9005 C250                 000076D0  1475 VERIFAIL STM   R0,R5,SAVER0R5   Save registers
0000755E  92FF C278                 000076F8  1476          MVI   FAILFLAG,X'FF'   Remember verification failure
                                          1477 *
                                          1478 **       First, show them the description...
                                          1479 *
00007562  D22F C1E0 5000      00007660  00000000  1480          MVC   FAILDESC,0(R5)   Save results/test description
00007568  4100 0044                 00000044  1481          LA    R0,L'FAILMSG1    R0 <== length of message
0000756C  4110 C1CC                 0000764C  1482          LA    R1,FAILMSG1      R1 --> the message text itself
00007570  4520 C27A                 000076FA  1483          BAL   R2,MSG           Go display this message
                                          1484 *
                                          1485 **       Save address of actual and expected results
                                          1486 *
00007574  5040 C24C                 000076CC  1487          ST    R4,AACTUAL       Save A(actual results)
00007578  4150 5030                 00000030  1488          LA    R5,48(,R5)       R5 ==> expected results
0000757C  5050 C248                 000076C8  1489          ST    R5,AEXPECT       Save A(expected results)
                                          1490 *
                                          1491 **       Format and show them the EXPECTED ("Want") results...
                                          1492 *
00007580  D205 C210 C3C0      00007690  00007840  1493          MVC   WANTGOT,=CL6'Want: '
00007586  F384 C216 C248      00007696  000076C8  1494          UNPK  FAILADR(L'FAILADR+1),AEXPECT(L'AEXPECT+1)
0000758C  9240 C21E                 0000769E  1495          MVI   BLANKEQ,C' '
00007590  DC07 C216 C178      00007696  000075F8  1496          TR    FAILADR,HEXTRTAB

00007596  F384 C221 5000      000076A1  00000000  1498          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R5)
0000759C  9240 C229                 000076A9  1499          MVI   FAILVALS+(0*9)+8,C' '
000075A0  DC07 C221 C178      000076A1  000075F8  1500          TR    FAILVALS+(0*9)(8),HEXTRTAB

000075A6  F384 C22A 5004      000076AA  00000004  1502          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R5)
000075AC  9240 C232                 000076B2  1503          MVI   FAILVALS+(1*9)+8,C' '
000075B0  DC07 C22A C178      000076AA  000075F8  1504          TR    FAILVALS+(1*9)(8),HEXTRTAB

000075B6  F384 C233 5008      000076B3  00000008  1506          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R5)
000075BC  9240 C23B                 000076BB  1507          MVI   FAILVALS+(2*9)+8,C' '
000075C0  DC07 C233 C178      000076B3  000075F8  1508          TR    FAILVALS+(2*9)(8),HEXTRTAB

000075C6  F384 C23C 500C      000076BC  0000000C  1510          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R5)
000075CC  9240 C244                 000076C4  1511          MVI   FAILVALS+(3*9)+8,C' '
000075D0  DC07 C23C C178      000076BC  000075F8  1512          TR    FAILVALS+(3*9)(8),HEXTRTAB

000075D6  4100 0035                 00000035  1514          LA    R0,L'FAILMSG2    R0 <== length of message
000075DA  4110 C210                 00007690  1515          LA    R1,FAILMSG2      R1 --> the message text itself
000075DE  4520 C27A                 000076FA  1516          BAL   R2,MSG           Go display this message
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                   1518 *
                                                   1519 **       Format and show them the ACTUAL ("Got") results...
                                                   1520 *
000075E2  D205 C210 C3C6      00007690  00007846  1521          MVC   WANTGOT,=CL6'Got:  '
000075E8  F384 C216 C24C      00007696  000076CC  1522          UNPK  FAILADR(L'FAILADR+1),AACTUAL(L'AACTUAL+1)
000075EE  9240 C21E                     0000769E  1523          MVI   BLANKEQ,C' '
000075F2  DC07 C216 C178      00007696  000075F8  1524          TR    FAILADR,HEXTRTAB

000075F8  F384 C221 4000      000076A1  00000000  1526          UNPK  FAILVALS+(0*9)(9),(0*4)(5,R4)
000075FE  9240 C229                     000076A9  1527          MVI   FAILVALS+(0*9)+8,C' '
00007602  DC07 C221 C178      000076A1  000075F8  1528          TR    FAILVALS+(0*9)(8),HEXTRTAB

00007608  F384 C22A 4004      000076AA  00000004  1530          UNPK  FAILVALS+(1*9)(9),(1*4)(5,R4)
0000760E  9240 C232                     000076B2  1531          MVI   FAILVALS+(1*9)+8,C' '
00007612  DC07 C22A C178      000076AA  000075F8  1532          TR    FAILVALS+(1*9)(8),HEXTRTAB

00007618  F384 C233 4008      000076B3  00000008  1534          UNPK  FAILVALS+(2*9)(9),(2*4)(5,R4)
0000761E  9240 C23B                     000076BB  1535          MVI   FAILVALS+(2*9)+8,C' '
00007622  DC07 C233 C178      000076B3  000075F8  1536          TR    FAILVALS+(2*9)(8),HEXTRTAB

00007628  F384 C23C 400C      000076BC  0000000C  1538          UNPK  FAILVALS+(3*9)(9),(3*4)(5,R4)
0000762E  9240 C244                     000076C4  1539          MVI   FAILVALS+(3*9)+8,C' '
00007632  DC07 C23C C178      000076BC  000075F8  1540          TR    FAILVALS+(3*9)(8),HEXTRTAB

00007638  4100 0035                     00000035  1542          LA    R0,L'FAILMSG2    R0 <== length of message
0000763C  4110 C210                     00007690  1543          LA    R1,FAILMSG2      R1 --> the message text itself
00007640  4520 C27A                     000076FA  1544          BAL   R2,MSG           Go display this message

00007644  9805 C250                     000076D0  1546          LM    R0,R5,SAVER0R5   Restore registers
00007648  47F0 C0CE                     0000754E  1547          B     VERINEXT         Continue with verification...



0000764C                                          1549 FAILMSG1 DS    0CL68
0000764C  C3D6D4D7 C1D9C9E2                       1550          DC    CL20'COMPARISON FAILURE! '
00007660  4D8485A2 83998997                       1551 FAILDESC DC    CL48'(description)'


00007690                                          1553 FAILMSG2 DS    0CL53
00007690  40404040 4040                           1554 WANTGOT  DC    CL6' '           'Want: ' -or- 'Got:  '
00007696  C1C1C1C1 C1C1C1C1                       1555 FAILADR  DC    CL8'AAAAAAAA'
0000769E  407E40                                  1556 BLANKEQ  DC    CL3' = '
000076A1  88888888 88888888                       1557 FAILVALS DC    CL36'hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh '

000076C8  00000000                                1559 AEXPECT  DC    F'0'             ==> Expected ("Want") results
000076CC  00000000                                1560 AACTUAL  DC    F'0'             ==> Actual ("Got") results
000076D0  00000000 00000000                       1561 SAVER0R5 DC    6F'0'            Registers R0 - R5 save area
000076E8  F0F1F2F3 F4F5F6F7                       1562 CHARHEX  DC    CL16'0123456789ABCDEF'
                            000075F8  00000010    1563 HEXTRTAB EQU   CHARHEX-X'F0'    Hexadecimal translation table
000076F8  00                                      1564 FAILFLAG DC    X'00'            FF = Fail, 00 = Success
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          1566 ********************************************************************************
                                          1567 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                          1568 ********************************************************************************

000076FA  4900 C3BC                 0000783C  1570 MSG      CH    R0,=H'0'                Do we even HAVE a message?
000076FE  07D2                              1571           BNHR  R2                      No, ignore

00007700  9002 C2B0                 00007730  1573           STM   R0,R2,MSGSAVE           Save registers

00007704  4900 C3BE                 0000783E  1575           CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00007708  47D0 C290                 00007710  1576           BNH   MSGOK                   Yes, continue
0000770C  4100 005F                 0000005F  1577           LA    R0,L'MSGMSG             No, set to maximum

00007710  1820                              1579 MSGOK    LR    R2,R0                   Copy length to work register
00007712  0620                              1580           BCTR  R2,0                    Minus-1 for execute
00007714  4420 C2BC                 0000773C  1581           EX    R2,MSGMVC               Copy message to O/P buffer

00007718  4120 200A                 0000000A  1583           LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
0000771C  4110 C2C2                 00007742  1584           LA    R1,MSGCMD               Point to true command

00007720  83120008                          1586           DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
00007724  4780 C2AA                 0000772A  1587           BZ    MSGRET                  Return if successful
00007728  0000                              1588           DC    H'0'                    CRASH for debugging purposes

0000772A  9802 C2B0                 00007730  1590 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
0000772E  07F2                              1591           BR    R2                      Return to caller
```

```
00007730  00000000 00000000                 1593 MSGSAVE  DC    3F'0'                   Registers save area
0000773C  D200 C2CB 1000      0000774B  00000000  1594 MSGMVC   MVC   MSGMSG(0),0(R1)       Executed instruction

00007742  D4E2C7D5 D6C8405C                 1596 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
0000774B  40404040 40404040                 1597 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    1599 ********************************************************************************
                                                    1600 *                              VERIFY TABLE
                                                    1601 ********************************************************************************
                                                    1602 *
                                                    1603 *        A(actual results), A(expected results), A(#of results)
                                                    1604 *
                                                    1605 ********************************************************************************

000077AC                                            1607 VERIFTAB DC    0F'0'
000077AC  00001000                                  1608          DC    A(SINTOUT)
000077B0  00004000                                  1609          DC    A(SINTOUT_GOOD)
000077B4  00000005                                  1610          DC    A(SINTOUT_NUM)
                                                    1611 *
000077B8  00001100                                  1612          DC    A(SINTFLGS)
000077BC  00004140                                  1613          DC    A(SINTFLGS_GOOD)
000077C0  00000005                                  1614          DC    A(SINTFLGS_NUM)
                                                    1615 *
000077C4  00001200                                  1616          DC    A(SINTRMO)
000077C8  00004280                                  1617          DC    A(SINTRMO_GOOD)
000077CC  0000001E                                  1618          DC    A(SINTRMO_NUM)
                                                    1619 *
000077D0  00001600                                  1620          DC    A(SINTRMOF)
000077D4  00004A00                                  1621          DC    A(SINTRMOF_GOOD)
000077D8  0000001E                                  1622          DC    A(SINTRMOF_NUM)
                                                    1623 *
000077DC  00002000                                  1624          DC    A(LINTOUT)
000077E0  00005180                                  1625          DC    A(LINTOUT_GOOD)
000077E4  00000005                                  1626          DC    A(LINTOUT_NUM)
                                                    1627 *
000077E8  00002100                                  1628          DC    A(LINTFLGS)
000077EC  000052C0                                  1629          DC    A(LINTFLGS_GOOD)
000077F0  00000005                                  1630          DC    A(LINTFLGS_NUM)
                                                    1631 *
000077F4  00002200                                  1632          DC    A(LINTRMO)
000077F8  00005400                                  1633          DC    A(LINTRMO_GOOD)
000077FC  0000001E                                  1634          DC    A(LINTRMO_NUM)
                                                    1635 *
00007800  00002600                                  1636          DC    A(LINTRMOF)
00007804  00005B80                                  1637          DC    A(LINTRMOF_GOOD)
00007808  0000001E                                  1638          DC    A(LINTRMOF_NUM)
                                                    1639 *
0000780C  00003000                                  1640          DC    A(XINTOUT)
00007810  00006300                                  1641          DC    A(XINTOUT_GOOD)
00007814  00000005                                  1642          DC    A(XINTOUT_NUM)
                                                    1643 *
00007818  00003100                                  1644          DC    A(XINTFLGS)
0000781C  00006440                                  1645          DC    A(XINTFLGS_GOOD)
00007820  00000005                                  1646          DC    A(XINTFLGS_NUM)
                                                    1647 *
00007824  00003200                                  1648          DC    A(XINTRMO)
00007828  00006580                                  1649          DC    A(XINTRMO_GOOD)
0000782C  0000001E                                  1650          DC    A(XINTRMO_NUM)
                                                    1651 *
00007830  00003600                                  1652          DC    A(XINTRMOF)
00007834  00006D00                                  1653          DC    A(XINTRMOF_GOOD)
00007838  0000001E                                  1654          DC    A(XINTRMOF_NUM)
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                    1655 *
                              0000000C  00000001  1656 VERIFLEN EQU   (*-VERIFTAB)/12    #of entries in verify table
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

0000783C                                          1658          END
0000783C  0000                                    1659                =H'0'
0000783E  005F                                    1660                =AL2(L'MSGMSG)
00007840  E68195A3 7A40                           1661                =CL6'Want: '
00007846  C796A37A 4040                           1662                =CL6'Got:  '
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AACTUAL	F	0076CC	4	1560	1487 1522
AEXPECT	F	0076C8	4	1559	1489 1494
AHELPERS	A	00027C	4	201	191 238
BFPCVTTL	J	000000	30796	118	168 171 173 176 184 894 896 898 900 903 905 907 909 912 914 916 918 926
BLANKEQ	C	00769E	3	1556	1495 1523
CHARHEX	C	0076E8	16	1562	1563
CLFDBR	I	000504	4	461	223
CLFDBRA	I	000562	4	511	225
CLFEBR	I	00035C	4	300	216
CLFEBRA	I	0003BA	4	350	218
CLFXBR	I	0006AC	4	622	230
CLFXBRA	I	00070E	4	673	232
CTLR0	F	0002F0	4	248	209 210 211
EXTDS	F	00031C	4	270	229
FAIL	I	000238	4	199	1455
FAILADR	C	007696	8	1555	1494 1496 1522 1524
FAILDESC	C	007660	48	1551	1480
FAILFLAG	X	0076F8	1	1564	1453 1476
FAILMSG1	C	00764C	68	1549	1481 1482
FAILMSG2	C	007690	53	1553	1514 1515 1542 1543
FAILPSW	X	0002E0	8	246	199
FAILVALS	C	0076A1	36	1557	1498 1499 1500 1502 1503 1504 1506 1507 1508 1510 1511 1512 1526 1527 1528 1530 1531 1532 1534 1535 1536 1538 1539 1540
FPCREGNT	X	0002F4	4	249	307 360 369 378 387 398 406 414 422 430 438 468 521 530 539 548 559 567 575 583 591 599 630 684 693 702 711 722 730 738 746 754 762
FPCREGTR	X	0002F8	4	250	315 476 638
FPR0	U	000000	1	138	306 308 318 356 362 371 380 389 399 407 415 423 431 439 467 469 479 517 523 532 541 550 560 568 576 584 592 600 628 631 641 679 686 695 704 713 723 731 739 747 755 763
FPR1	U	000001	1	139	
FPR10	U	00000A	1	148	
FPR11	U	00000B	1	149	
FPR12	U	00000C	1	150	
FPR13	U	00000D	1	151	
FPR14	U	00000E	1	152	
FPR15	U	00000F	1	153	
FPR2	U	000002	1	140	629 680
FPR3	U	000003	1	141	
FPR4	U	000004	1	142	
FPR5	U	000005	1	143	
FPR6	U	000006	1	144	
FPR7	U	000007	1	145	
FPR8	U	000008	1	146	
FPR9	U	000009	1	147	
GOODPSW	X	0002D0	8	245	242
HELPERS	H	007480	2	1395	156 201
HEXTRTAB	U	0075F8	16	1563	1404 1408 1412 1416 1420 1496 1500 1504 1508 1512 1524 1528 1532 1536 1540
IMAGE	1	000000	30796	0	
LBFPCT	U	000048	1	834	265
LBFPIN	F	0008A8	4	820	834 266
LBFPINRM	F	0008F0	4	838	850 282
LBFPRMCT	U	000050	1	850	281
LINTFLGS	U	002100	0	905	268 1628

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
LINTFLGS_GOOD	U	0052C0	1	1098	1109 1629
LINTFLGS_NUM	U	000005	1	1109	1630
LINTOUT	U	002000	0	903	267 1624
LINTOUT_GOOD	U	005180	1	1084	1095 1625
LINTOUT_NUM	U	000005	1	1095	1626
LINTRMO	U	002200	0	907	283 1632
LINTRMOF	U	002600	0	909	284 1636
LINTRMOF_GOOD	U	005B80	1	1176	1237 1637
LINTRMOF_NUM	U	00001E	1	1237	1638
LINTRMO_GOOD	U	005400	1	1112	1173 1633
LINTRMO_NUM	U	00001E	1	1173	1634
LONGS	F	00030C	4	264	222
MSG	I	0076FA	4	1570	1424 1483 1516 1544
MSGCMD	C	007742	9	1596	1583 1584
MSGMSG	C	00774B	95	1597	1577 1594 1575
MSGMVC	I	00773C	6	1594	1581
MSGOK	I	007710	2	1579	1576
MSGRET	I	00772A	4	1590	1587
MSGSAVE	F	007730	4	1593	1573 1590
PCINTCD	H	00008E	2	169	186 1402
PCNOTDTA	I	00020C	4	190	187
PCOLDPSW	U	000150	0	171	188 1406 1410 1414 1418
PGMCK	H	007480	2	1401	192
PGMCOMMA	C	0074F6	1	1431	1403
PGMPSW	C	0074FC	36	1433	1406 1407 1408 1410 1411 1412 1414 1415 1416 1418 1419 1420
PROGCHK	H	000200	2	185	177
PROGCODE	C	0074F2	4	1430	1402 1404
PROGMSG	C	0074DE	66	1428	1422 1423
PROGPSW	D	000228	8	198	197
R0	U	000000	1	119	190 193 209 211 1422 1475 1481 1514 1542 1546 1570 1573 1575 1577 1579 1590
R1	U	000001	1	120	308 309 311 312 313 316 317 318 319 321 322 323 362 363 365 366 367 371 372 374 375 376 380 381 383 384 385 389 390 392 393 394 399 400 402 403 404 407 408 410 411 412 415 416 418 419 420 423 424 426 427 428 431 432 434 435 436 439 440 442 443 444 469 470 472 473 474 477 478 479 480 482 483 484 523 524 526 527 528 532 533 535 536 537 541 542 544 545 546 550 551 553 554 555 560 561 563 564 565 568 569 571 572 573 576 577 579 580 581 584 585 587 588 589 592 593 595 596 597 600 601 603 604 605 631 632 634 635 636 639 640 641 642 644 645 646 686 687 689 690 691 695 696 698 699 700 704 705 707 708 709 713 714 716 717 718 723 724 726 727 728 731 732 734 735 736 739 740 742 743 744 747 748 750 751 752 755 756 758 759 760 763 764 766 767 768 1423 1444 1448 1450 1482 1515 1543 1584 1594
R10	U	00000A	1	129	215 217 222 224 229 231 300 301 350 351 461 462 511 512 622 623 673 674
R11	U	00000B	1	130	
R12	U	00000C	1	131	156 191 238 304 328 354 449 465 489 515 610 626 651 677 773
R13	U	00000D	1	132	192 216 218 223 225 230 232 239 303 329 353 450 464 490 514 611 625 652 676 774 1426 1454
R14	U	00000E	1	133	195 196 240 241
R15	U	00000F	1	134	155 190 193
R2	U	000002	1	121	300 302 328 350 352 449 461 463 489 511 513 610 622 624 651 673 675 773 1424 1445 1451 1483 1516 1544 1571 1573 1579 1580

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
					1581 1583 1590 1591
R3	U	000003	1	122	300 306 325 350 356 446 461 467 486 511 517 607 622 628
					629 648 673 679 680 770 1446 1451
R4	U	000004	1	123	1448 1463 1465 1487 1526 1530 1534 1538
R5	U	000005	1	124	1463 1466 1475 1480 1488 1489 1498 1502 1506 1510 1546
R6	U	000006	1	125	1448 1467
R7	U	000007	1	126	301 309 319 326 351 363 372 381 390 400 408 416 424 432
					440 447 462 470 480 487 512 524 533 542 551 561 569 577
					585 593 601 608 623 632 642 649 674 687 696 705 714 724
					732 740 748 756 764 771 1449 1469
R8	U	000008	1	127	301 310 313 320 323 327 351 364 367 373 376 382 385 391
					394 401 404 409 412 417 420 425 428 433 436 441 444 448
					462 471 474 481 484 488 512 525 528 534 537 543 546 552
					555 562 565 570 573 578 581 586 589 594 597 602 605 609
					623 633 636 643 646 650 674 688 691 697 700 706 709 715
					718 725 728 733 736 741 744 749 752 757 760 765 768 772
					1461 1467
R9	U	000009	1	128	
RMEXTDS	A	00034C	4	286	231
RMLONGS	A	00033C	4	281	224
RMSHORTS	A	00032C	4	276	217
SAVER0R5	F	0076D0	4	1561	1475 1546
SAVEREGS	F	00023C	4	200	190 193
SBFPCT	U	000024	1	800	259
SBFPIN	F	00085C	4	788	800 260
SBFPINRM	F	000880	4	804	816 277
SBFPRMCT	U	000028	1	816	276
SHORTS	F	0002FC	4	258	215
SINTFLGS	U	001100	0	896	262 1612
SINTFLGS_GOOD	U	004140	1	942	953 1613
SINTFLGS_NUM	U	000005	1	953	1614
SINTOUT	U	001000	0	894	261 1608
SINTOUT_GOOD	U	004000	1	928	939 1609
SINTOUT_NUM	U	000005	1	939	1610
SINTRMO	U	001200	0	898	278 1616
SINTRMOF	U	001600	0	900	279 1620
SINTRMOF_GOOD	U	004A00	1	1020	1081 1621
SINTRMOF_NUM	U	00001E	1	1081	1622
SINTRMO_GOOD	U	004280	1	956	1017 1617
SINTRMO_NUM	U	00001E	1	1017	1618
START	I	000280	4	209	174
VERIFAIL	I	00755A	4	1475	1464
VERIFLEN	U	00000C	1	1656	1445
VERIFTAB	F	0077AC	4	1607	1656 1444
VERIFY	I	007542	2	1461	1449
VERINEXT	I	00754E	4	1465	1547
VERISUB	H	007520	2	1439	239
WANTGOT	C	007690	6	1554	1493 1521
XBFPCT	U	000090	1	868	271
XBFPIN	D	000940	8	854	868 272
XBFPINRM	D	0009D0	8	872	885 287
XBFPRMCT	U	0000A0	1	885	286
XINTFLGS	U	003100	0	914	274 1644
XINTFLGS_GOOD	U	006440	1	1254	1265 1645
XINTFLGS_NUM	U	000005	1	1265	1646
XINTOUT	U	003000	0	912	273 1640

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

XINTOUT_GOOD         U    006300       1  1240  1251  1641
XINTOUT_NUM          U    000005       1  1251  1642
XINTRMO              U    003200       0   916   288  1648
XINTRMOF             U    003600       0   918   289  1652
XINTRMOF_GOOD        U    006D00       1  1332  1393  1653
XINTRMOF_NUM         U    00001E       1  1393  1654
XINTRMO_GOOD         U    006580       1  1268  1329  1649
XINTRMO_NUM          U    00001E       1  1329  1650
=AL2(L'MSGMSG)       R    00783E       2  1660  1575
=CL6'Got:  '         C    007846       6  1662  1521
=CL6'Want: '         C    007840       6  1661  1493
=H'0'                H    00783C       2  1659  1570
```

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 0

Image       IMAGE      30796  0000-784B  0000-784B
  Region               30796  0000-784B  0000-784B
    CSECT   BFPCVTTL   30796  0000-784B  0000-784B
```

```
   STMT                                                    FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\bfp-004-cvttolog\bfp-004-cvttolog.asm

** NO ERRORS FOUND **
```