```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                        2 *
                                                        3 ********************************************************************************
                                                        4 *
                                                        5 *Testcase str-001-clst  --  Test cases for variations on the CLST
                                                        6 *                           (Compare Logical String) instruction.
                                                        7 *
                                                        8 ********************************************************************************
                                                        9 *
                                                       10 *                          str-001-clst.asm
                                                       11 *
                                                       12 * Created and placed into the public domain 2018-12-27 by Bob Polmanter
                                                       13 * Remove runtest *Compare dependency on 2022-03-08 by Fish
                                                       14 *
                                                       15 * The CLST instruction is tested against the definition in the
                                                       16 * z/Architecture Principles of Operation, SA22-7832.
                                                       17 *
                                                       18 * Test data is assembled into this program, and some test data is
                                                       19 * generated by this program. The program itself verifies the resulting
                                                       20 * status of registers and condition codes via simple CLC comparison.
                                                       21 *
                                                       22 *
                                                       23 *
                                                       24 * Tests performed with CLST (Compare Logical String):
                                                       25 *
                                                       26 * 1.  Ensure that a non-zero bit in R0 bits 32-55 gives PIC06
                                                       27 * 2.  Simple equality test; no operands cross page boundary
                                                       28 * 3.  Operand 1 first byte is the termination character
                                                       29 * 4.  Operand 2 first byte is the termination character
                                                       30 * 5.  Operand 1 string "less than" operand 2 string
                                                       31 * 6.  Operand 1 string "greater than" operand 2 string
                                                       32 * 7.  Operand 1 string "shorter than" operand 2 string
                                                       33 * 8.  Operand 1 string "longer than" operand 2 string
                                                       34 * 9.  Operand 1 (only) crosses a page boundary
                                                       35 * 10. Operand 2 (only) crosses a page boundary
                                                       36 * 11. Both operands cross, operand 1 closer to boundary
                                                       37 * 12. Both operands cross, operand 2 closer to boundary
                                                       38 * 13. Both operands cross, ops equidistant, large multipage compare.
                                                       39 *
                                                       40 *
                                                       41 * NOTE - the nature of the string instructions is such that this test
                                                       42 *        case will only validate properly for the string instruction
                                                       43 *        improvement modifications committed in December 2018.  The
                                                       44 *        computation of the CPU determined number of bytes is an
                                                       45 *        unpredictable number on real hardware (at least above the
                                                       46 *        minimum value) and the method used in Hercules prior to
                                                       47 *        instruction improvements calculated it differently than the
                                                       48 *        improved method.  As a result, the operand registers will
                                                       49 *        likely contain different values when compared by the test
                                                       50 *        due to the different CPU number of bytes determined.
                                                       51 *        None of the methods are wrong, and failing results in the
                                                       52 *        test are not necessarily wrong.  But this program and the
                                                       53 *        resulting test comparisons were written for the method used
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                      54 *        by the improved string instructions (CLST, MVST, SRST).
                                                      55 *
                                                      56 *
                                                      57 ********************************************************************************
                                                      58 *
                                                      59 *
                              00000000  000008BF      60 CLST001  START 0
                              00000000  00000001      61 STRTLABL EQU   *
                              00000000  00000001      62 R0       EQU   0
                              00000001  00000001      63 R1       EQU   1
                              00000002  00000001      64 R2       EQU   2
                              00000003  00000001      65 R3       EQU   3
                              00000004  00000001      66 R4       EQU   4
                              00000005  00000001      67 R5       EQU   5
                              00000006  00000001      68 R6       EQU   6
                              00000007  00000001      69 R7       EQU   7
                              00000008  00000001      70 R8       EQU   8
                              00000009  00000001      71 R9       EQU   9
                              0000000A  00000001      72 R10      EQU   10
                              0000000B  00000001      73 R11      EQU   11
                              0000000C  00000001      74 R12      EQU   12
                              0000000D  00000001      75 R13      EQU   13
                              0000000E  00000001      76 R14      EQU   14
                              0000000F  00000001      77 R15      EQU   15
                                                      78 *
                                                      79 *
00000000                      00000000                80          USING *,R15
                                                      81 *
                                                      82 * Selected z/Arch low core layout
                                                      83 *
00000000                      00000000  0000008C      84          ORG   STRTLABL+X'8C'      Program check interrution code
0000008C  00000000                                    85 PGMINTC  DS    F
                                                      86 *
                              00000150  00000001      87 PGMOPSW  EQU   STRTLABL+X'150'     z/Arch Program check old PSW
                                                      88 *
00000090                      00000090  000001A0      89          ORG   STRTLABL+X'1A0'     z/Arch Restart PSW
000001A0  00000001 80000000                           90          DC    X'0000000180000000',A(0,START)
                                                      91 *
000001B0                      000001B0  000001D0      92          ORG   STRTLABL+X'1D0'     z/Arch Program check new PSW
000001D0  00000001 80000000                           93 PGMNPSW  DC    X'0000000180000000',A(0,PROGCHK)
                                                      94 *
                                                      95 * Program check routine.  We are looking for a single specification
                                                      96 * exeception.  Any other program check is not expected to occur and
                                                      97 * results in a hard wait.
                                                      98 *
000001E0                      000001E0  00000200      99          ORG   STRTLABL+X'200'
00000200                                             100 PROGCHK  DS    0H                Program check occured...
00000200  9500 F21C                     0000021C     101          CLI   DIDTHIS,X'00'    First/only time here?
00000204  4770 F218                     00000218     102          BNE   FAIL              No?! Then something is wrong!
00000208  9506 F08F                     0000008F     103          CLI   PGMINTC+3,X'06'   Specification Exception?
0000020C  4770 F218                     00000218     104          BNE   FAIL              No?! Then something is wrong!
00000210  92FF F21C                     0000021C     105          MVI   DIDTHIS,X'FF'    Remember we did this once already
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2    STMT

00000214  47F0 F22E                       0000022E   106           B     CONTINUE         Continue, as this is expected (once!)
00000218  B2B2 F3E8                       000003E8   107 FAIL      LPSWE FAILPSW          Unexpected PIC, disabled wait
0000021C  00                                         108 DIDTHIS   DC    X'00'            X'FF' == we already did this
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT
                                                      110 ********************************************************************************
                                                      111 *
                                                      112 *  Main program.
                                                      113 *
0000021E                                              114 START    DS    0H
                                                      115 *
                                                      116 **********
                                                      117 * TEST 1 * Ensure any non-zero bits in R0 bits 32-55 gives PIC 06
                                                      118 **********
                                                      119 *
0000021E  4100 0400                   00000400        120          LA    R0,X'400'              Set invalid termination char
00000222  4160 F700                   00000700        121          LA    R6,SHORT1              -> short string
00000226  4170 F710                   00000710        122          LA    R7,SHORT2              -> another short string
0000022A  B25D 0067                                   123          CLST  R6,R7                  Attempt a CLST, should get PIC 6
                                                      124 *
                          0000022E    00000001        125 CONTINUE EQU   *
0000022E  95FF F21C                   0000021C        126          CLI   DIDTHIS,X'FF'          Did PIC 06 happen?
00000232  4770 F218                   00000218        127          BNE   FAIL                   No?! Then something is wrong!
00000236  D207 F1D0 F3E8  000001D0    000003E8        128          MVC   PGMNPSW,FAILPSW       Going forward, all other
                                                      129 *                                      program checks should halt.
                                                      130 *
                                                      131 **********
                                                      132 * TEST 2 * Compare short equal strings; no page boundary crossings.
                                                      133 **********
                                                      134 *
0000023C  4160 F700                   00000700        135          LA    R6,SHORT1              -> string 1
00000240  4170 F710                   00000710        136          LA    R7,SHORT2              -> string 2
00000244  4D90 F3BA                   000003BA        137          BAS   R9,COMPARE             Compare the string
00000248  9068 F800                   00000800        138          STM   R6,R8,RESULT2          Save test result regs
                                                      139 *
                                                      140 **********
                                                      141 * TEST 3 * Compare a short string; operand 1 is the termination
                                                      142 ********** character in the first byte.
                                                      143 *
0000024C  4160 F750                   00000750        144          LA    R6,TERM                -> string 1
00000250  4170 F710                   00000710        145          LA    R7,SHORT2              -> string 2
00000254  4D90 F3BA                   000003BA        146          BAS   R9,COMPARE             Compare the string
00000258  9068 F810                   00000810        147          STM   R6,R8,RESULT3          Save test result regs
                                                      148 *
                                                      149 **********
                                                      150 * TEST 4 * Compare a short string; operand 2 is the termination
                                                      151 ********** character in the first byte.
                                                      152 *
0000025C  4160 F700                   00000700        153          LA    R6,SHORT1              -> string 1
00000260  4170 F750                   00000750        154          LA    R7,TERM                -> string 2
00000264  4D90 F3BA                   000003BA        155          BAS   R9,COMPARE             Compare the string
00000268  9068 F820                   00000820        156          STM   R6,R8,RESULT4          Save test result regs
                                                      157 *
                                                      158 **********
                                                      159 * TEST 5 * Compare a short string; operand 1 string is "lesser"
                                                      160 ********** than the operand 2 string.
                                                      161 *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
0000026C  4160 F720                    00000720   162          LA    R6,LESSER               -> string 1
00000270  4170 F730                    00000730   163          LA    R7,GREATER              -> string 2
00000274  4D90 F3BA                    000003BA   164          BAS   R9,COMPARE              Compare the string
00000278  9068 F830                    00000830   165          STM   R6,R8,RESULT5           Save test result regs
                                                  166 *
                                                  167 **********
                                                  168 * TEST 6 * Compare a short string; operand 1 string is "greater"
                                                  169 ********** than the operand 2 string.
                                                  170 *
0000027C  4160 F730                    00000730   171          LA    R6,GREATER              -> string 1
00000280  4170 F720                    00000720   172          LA    R7,LESSER               -> string 2
00000284  4D90 F3BA                    000003BA   173          BAS   R9,COMPARE              Compare the string
00000288  9068 F840                    00000840   174          STM   R6,R8,RESULT6           Save test result regs
                                                  175 *
                                                  176 **********
                                                  177 * TEST 7 * Compare a short string; operand 1 string is "shorter"
                                                  178 ********** than the operand 2 string.
                                                  179 *
0000028C  4160 F700                    00000700   180          LA    R6,SHORT1               -> string 1
00000290  4170 F740                    00000740   181          LA    R7,LONGER               -> string 2
00000294  4D90 F3BA                    000003BA   182          BAS   R9,COMPARE              Compare the string
00000298  9068 F850                    00000850   183          STM   R6,R8,RESULT7           Save test result regs
                                                  184 *
                                                  185 **********
                                                  186 * TEST 8 * Compare a short string; operand 1 string is "longer"
                                                  187 ********** than the operand 2 string.
                                                  188 *
0000029C  4160 F740                    00000740   189          LA    R6,LONGER               -> string 1
000002A0  4170 F710                    00000710   190          LA    R7,SHORT2               -> string 2
000002A4  4D90 F3BA                    000003BA   191          BAS   R9,COMPARE              Compare the string
000002A8  9068 F860                    00000860   192          STM   R6,R8,RESULT8           Save test result regs
                                                  193 *
                                                  194 **********
                                                  195 * PREP   * Prepare a multi-page frame area for more lengthy compares.
                                                  196 **********
                                                  197 *
000002AC  9825 F760                    00000760   198          LM    R2,R5,AREA              -> large area and length
000002B0  0E24                                    199          MVCL  R2,R4                   Pad it full of X'AA'
                                                  200 *
                                                  201 **********
                                                  202 * TEST 9 * Compare a string; operand 1 string crosses a
                                                  203 ********** page boundary.
                                                  204 *
000002B2  9847 F770                    00000770   205          LM    R4,R7,TEST9             Get lengths and string ptrs
000002B6  925B 4000                    00000000   206          MVI   0(R4),C'$'              Set a termination char
000002BA  925B 5000                    00000000   207          MVI   0(R5),C'$'              Set a termination char
000002BE  4D90 F3BA                    000003BA   208          BAS   R9,COMPARE              Compare the string
000002C2  9068 F870                    00000870   209          STM   R6,R8,RESULT9           Save test result regs
000002C6  92AA 4000                    00000000   210          MVI   0(R4),X'AA'             Reset the termination char
000002CA  92AA 5000                    00000000   211          MVI   0(R5),X'AA'             Reset the termination char
                                                  212 *
                                                  213 ***********
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    214 * TEST 10 * Compare a string; operand 2 string crosses a
                                                    215 ********** page boundary.
                                                    216 *
000002CE  9847 F780                      00000780   217          LM    R4,R7,TEST10           Get lengths and string ptrs
000002D2  925B 4000                      00000000   218          MVI   0(R4),C'$'             Set a termination char
000002D6  925B 5000                      00000000   219          MVI   0(R5),C'$'             Set a termination char
000002DA  4D90 F3BA                      000003BA   220          BAS   R9,COMPARE             Compare the string
000002DE  9068 F880                      00000880   221          STM   R6,R8,RESULT10         Save test result regs
000002E2  92AA 4000                      00000000   222          MVI   0(R4),X'AA'            Reset the termination char
000002E6  92AA 5000                      00000000   223          MVI   0(R5),X'AA'            Reset the termination char
                                                    224 *
                                                    225 **********
                                                    226 * TEST 11 * Compare a string; both operands cross page boundaries;
                                                    227 ********** operand 1 is closer to the boundary
                                                    228 *
000002EA  9847 F790                      00000790   229          LM    R4,R7,TEST11           Get lengths and string ptrs
000002EE  925B 4000                      00000000   230          MVI   0(R4),C'$'             Set a termination char
000002F2  925B 5000                      00000000   231          MVI   0(R5),C'$'             Set a termination char
000002F6  4D90 F3BA                      000003BA   232          BAS   R9,COMPARE             Compare the string
000002FA  9068 F890                      00000890   233          STM   R6,R8,RESULT11         Save test result regs
000002FE  92AA 4000                      00000000   234          MVI   0(R4),X'AA'            Reset the termination char
00000302  92AA 5000                      00000000   235          MVI   0(R5),X'AA'            Reset the termination char
                                                    236 *
                                                    237 **********
                                                    238 * TEST 12 * Compare a string; both operands cross page boundaries;
                                                    239 ********** operand 2 is closer to the boundary
                                                    240 *
00000306  9847 F7A0                      000007A0   241          LM    R4,R7,TEST12           Get lengths and string ptrs
0000030A  925B 4000                      00000000   242          MVI   0(R4),C'$'             Set a termination char
0000030E  925B 5000                      00000000   243          MVI   0(R5),C'$'             Set a termination char
00000312  4D90 F3BA                      000003BA   244          BAS   R9,COMPARE             Compare the string
00000316  9068 F8A0                      000008A0   245          STM   R6,R8,RESULT12         Save test result regs
0000031A  92AA 4000                      00000000   246          MVI   0(R4),X'AA'            Reset the termination char
0000031E  92AA 5000                      00000000   247          MVI   0(R5),X'AA'            Reset the termination char
                                                    248 *
                                                    249 **********
                                                    250 * TEST 13 * Compare a string; both operands cross page boundaries;
                                                    251 ********** both operands equidistant from boundary; large compare.
                                                    252 *
00000322  9847 F7B0                      000007B0   253          LM    R4,R7,TEST13           Get lengths and string ptrs
00000326  925B 4000                      00000000   254          MVI   0(R4),C'$'             Set a termination char
0000032A  925B 5000                      00000000   255          MVI   0(R5),C'$'             Set a termination char
0000032E  4D90 F3BA                      000003BA   256          BAS   R9,COMPARE             Compare the string
00000332  9068 F8B0                      000008B0   257          STM   R6,R8,RESULT13         Save test result regs
00000336  92AA 4000                      00000000   258          MVI   0(R4),X'AA'            Reset the termination char
0000033A  92AA 5000                      00000000   259          MVI   0(R5),X'AA'            Reset the termination char
                                                    260 *
                                                    261 **       Verify results...
                                                    262 *
0000033E  D50B F3F8 F800      000003F8   00000800   263          CLC   GRESLT2,RESULT2        Expected results?
00000344  4770 F218                      00000218   264          BNE   FAIL                   No?! Then something is wrong!
00000348  D50B F404 F810      00000404   00000810   265          CLC   GRESLT3,RESULT3        Expected results?
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

0000034E  4770 F218                    00000218   266          BNE   FAIL                    No?! Then something is wrong!
00000352  D50B F410 F820      00000410  00000820   267          CLC   GRESLT4,RESULT4         Expected results?
00000358  4770 F218                    00000218   268          BNE   FAIL                    No?! Then something is wrong!
0000035C  D50B F41C F830      0000041C  00000830   269          CLC   GRESLT5,RESULT5         Expected results?
00000362  4770 F218                    00000218   270          BNE   FAIL                    No?! Then something is wrong!
00000366  D50B F428 F840      00000428  00000840   271          CLC   GRESLT6,RESULT6         Expected results?
0000036C  4770 F218                    00000218   272          BNE   FAIL                    No?! Then something is wrong!
00000370  D50B F434 F850      00000434  00000850   273          CLC   GRESLT7,RESULT7         Expected results?
00000376  4770 F218                    00000218   274          BNE   FAIL                    No?! Then something is wrong!
0000037A  D50B F440 F860      00000440  00000860   275          CLC   GRESLT8,RESULT8         Expected results?
00000380  4770 F218                    00000218   276          BNE   FAIL                    No?! Then something is wrong!
00000384  D50B F44C F870      0000044C  00000870   277          CLC   GRESLT9,RESULT9         Expected results?
0000038A  4770 F218                    00000218   278          BNE   FAIL                    No?! Then something is wrong!
0000038E  D50B F458 F880      00000458  00000880   279          CLC   GRESLT10,RESULT10       Expected results?
00000394  4770 F218                    00000218   280          BNE   FAIL                    No?! Then something is wrong!
00000398  D50B F464 F890      00000464  00000890   281          CLC   GRESLT11,RESULT11       Expected results?
0000039E  4770 F218                    00000218   282          BNE   FAIL                    No?! Then something is wrong!
000003A2  D50B F470 F8A0      00000470  000008A0   283          CLC   GRESLT12,RESULT12       Expected results?
000003A8  4770 F218                    00000218   284          BNE   FAIL                    No?! Then something is wrong!
000003AC  D50B F47C F8B0      0000047C  000008B0   285          CLC   GRESLT13,RESULT13       Expected results?
000003B2  4770 F218                    00000218   286          BNE   FAIL                    No?! Then something is wrong!
                                                  287 *
000003B6  B2B2 F3D8                    000003D8   288          LPSWE GOODPSW                 load SUCCESS disabled wait PSW
                                                  289 *
                                                  290 *-- CLST routine used by the tests
                                                  291 *
                               000003BA  00000001 292 COMPARE  EQU   *
000003BA  4100 005B                    0000005B   293          LA    R0,C'$'                 Load termination character
000003BE  1B88                                    294          SR    R8,R8                   Init CLST counter
                                                  295 *
                               000003C0  00000001 296 INVOKE   EQU   *
000003C0  B25D 0067                               297          CLST  R6,R7                   Compare the strings
000003C4  4180 8001                    00000001   298          LA    R8,1(,R8)               Count executions of CLST
000003C8  4710 F3C0                    000003C0   299          BC    1,INVOKE                Restart the compare
000003CC  B222 0080                               300          IPM   R8                      Put final CC in high R8
000003D0  07F9                                    301          BR    R9                      Return
                                                  302 *
                                                  303 *
000003D8                                          304          DS    0D              Ensure correct alignment for psw
000003D8  00020000 00000000                       305 GOODPSW  DC    X'0002000000000000',A(0,0) Normal end - disabled wait
000003E8  00020000 00000000                       306 FAILPSW  DC    X'0002000000000000',XL4'00',X'0000DEAD' Abnormal end
                                                  307 *
000003F8  00000700 00000710                       308 GRESLT2  DC    XL12'000007000000071000000001'
00000404  00000750 00000710                       309 GRESLT3  DC    XL12'000007500000071010000001'
00000410  00000700 00000750                       310 GRESLT4  DC    XL12'000007000000075020000001'
0000041C  00000727 00000737                       311 GRESLT5  DC    XL12'000007270000073710000001'
00000428  00000737 00000727                       312 GRESLT6  DC    XL12'000007370000072720000001'
00000434  0000070C 0000074C                       313 GRESLT7  DC    XL12'0000070C0000074C10000001'
00000440  0000074C 0000071C                       314 GRESLT8  DC    XL12'0000074C0000071C20000001'
0000044C  00003000 00004300                       315 GRESLT9  DC    XL12'000030000000430000000002'
00000458  00002A00 00005000                       316 GRESLT10 DC    XL12'00002A000000500000000002'
00000464  00003080 00005000                       317 GRESLT11 DC    XL12'000030800000500000000002'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
00000470  00002F40 00004F80                          318 GRESLT12 DC    XL12'00002F4000004F8000000001'
0000047C  00006000 0000C000                          319 GRESLT13 DC    XL12'00006000000C00000000005'
                                                     320 *
                                                     321 *                                    core
                                                     322 *                                   offset
00000488                      00000488  00000700     323          ORG   STRTLABL+X'700'       7xx
00000700  E2C8D6D9 E340E2E3                          324 SHORT1   DC    CL16'SHORT STRING$   '   00
00000710  E2C8D6D9 E340E2E3                          325 SHORT2   DC    CL16'SHORT STRING$   '   10
00000720  E2E3D9C9 D5C7404C                          326 LESSER   DC    CL16'STRING < LOW   $'   20
00000730  E2E3D9C9 D5C7406E                          327 GREATER  DC    CL16'STRING > HIGH  $'   30
00000740  E2C8D6D9 E340E2E3                          328 LONGER   DC    CL16'SHORT STRING XL$'   40
00000750  5B                                        329 TERM     DC    C'$'                     50
00000751  FFFFFFFF FFFFFFFF                          330 FFS      DC    15X'FF'                  51
                                                     331 *
00000760  00002000                                   332 AREA     DC    X'00002000'              -> start of multi-page area
00000764  00010000                                   333 AREALEN  DC    A(4096*16)               Size of multi=page area
00000768  00000000                                   334 ZERO     DC    A(0)
0000076C  AA000000                                   335 PAD      DC    X'AA000000'              MVCL pad char
                                                     336 *
                                                     337 *-- Storage addresses for Tests 9-13.  Four addresses are
                                                     338 *-- provided:  where to place the termination character in
                                                     339 *-- strings 1 and 2, and where string 1 and 2 start.
                                                     340 *
                                                     341 *
00000770                                             342 TEST9    DS    0F                       Op 1 (only) crosses page
00000770  00003200                                   343          DC    X'00003200'   len=x400   -> where to place term chr op 1
00000774  00004500                                   344          DC    X'00004500'   len=x400   -> where to place term chr op 2
00000778  00002E00                                   345          DC    X'00002E00'              -> start of string (operand 1)
0000077C  00004100                                   346          DC    X'00004100'              -> start of string (operand 2)
                                                     347 *
00000780                                             348 TEST10   DS    0F                       Op 2 (only) crosses page
00000780  00002B00                                   349          DC    X'00002B00'   len=x800   -> where to place term chr op 1
00000784  00005100                                   350          DC    X'00005100'   len=x800   -> where to place term chr op 2
00000788  00002300                                   351          DC    X'00002300'              -> start of string (operand 1)
0000078C  00004900                                   352          DC    X'00004900'              -> start of string (operand 2)
                                                     353 *
00000790                                             354 TEST11   DS    0F                       Both cross; Op1 closer to bound
00000790  00003090                                   355          DC    X'00003090'   len=x110   -> where to place term chr op 1
00000794  00005010                                   356          DC    X'00005010'   len=x110   -> where to place term chr op 2
00000798  00002F80                                   357          DC    X'00002F80'              -> start of string (operand 1)
0000079C  00004F00                                   358          DC    X'00004F00'              -> start of string (operand 2)
                                                     359 *
000007A0                                             360 TEST12   DS    0F                       Both cross; Op2 closer to bound
000007A0  00003030                                   361          DC    X'00003030'   len=x0F0   -> where to place term chr op 1
000007A4  00005070                                   362          DC    X'00005070'   len=x0F0   -> where to place term chr op 2
000007A8  00002F40                                   363          DC    X'00002F40'              -> start of string (operand 1)
000007AC  00004F80                                   364          DC    X'00004F80'              -> start of string (operand 2)
                                                     365 *
000007B0                                             366 TEST13   DS    0F                       Both cross; ops equidistant
000007B0  00006080                                   367          DC    X'00006080'   len=x3480  -> where to place term chr op 1
000007B4  0000C080                                   368          DC    X'0000C080'   len=x3480  -> where to place term chr op 2
000007B8  00002C00                                   369          DC    X'00002C00'              -> start of string (operand 1)
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

000007BC  00008C00                                    370          DC    X'00008C00'             -> start of string (operand 2)
                                                      371 *
                                                      372 *
                                                      373 *  Locations for results
                                                      374 *
                                                      375 * Result fields are kept on 16-byte boundaries to more easily
                                                      376 * track their assembled offsets.
                                                      377 *
                                                      378 *                                  offset
000007C0                          000007C0  00000800  379          ORG   STRTLABL+X'800'   8xx
00000800  00000000 00000000                           380 RESULT2  DS    4F                 00   Register results test 2
00000810  00000000 00000000                           381 RESULT3  DS    4F                 10   Register results test 3
00000820  00000000 00000000                           382 RESULT4  DS    4F                 20   Register results test 4
00000830  00000000 00000000                           383 RESULT5  DS    4F                 30   Register results test 5
00000840  00000000 00000000                           384 RESULT6  DS    4F                 40   Register results test 6
00000850  00000000 00000000                           385 RESULT7  DS    4F                 50   Register results test 7
00000860  00000000 00000000                           386 RESULT8  DS    4F                 60   Register results test 8
00000870  00000000 00000000                           387 RESULT9  DS    4F                 70   Register results test 9
00000880  00000000 00000000                           388 RESULT10 DS    4F                 80   Register results test 10
00000890  00000000 00000000                           389 RESULT11 DS    4F                 90   Register results test 11
000008A0  00000000 00000000                           390 RESULT12 DS    4F                 A0   Register results test 12
000008B0  00000000 00000000                           391 RESULT13 DS    4F                 B0   Register results test 13
                                                      392 *
                                                      393          END
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AREA	X	000760	4	332	198
AREALEN	A	000764	4	333	
CLST001	J	000000	2240	60	
COMPARE	U	0003BA	1	292	137 146 155 164 173 182 191 208 220 232 244 256
CONTINUE	U	00022E	1	125	106
DIDTHIS	X	00021C	1	108	101 105 126
FAIL	I	000218	4	107	102 104 127 264 266 268 270 272 274 276 278 280 282 284 286
FAILPSW	X	0003E8	8	306	107 128
FFS	X	000751	1	330	
GOODPSW	X	0003D8	8	305	288
GREATER	C	000730	16	327	163 171
GRESLT10	X	000458	12	316	279
GRESLT11	X	000464	12	317	281
GRESLT12	X	000470	12	318	283
GRESLT13	X	00047C	12	319	285
GRESLT2	X	0003F8	12	308	263
GRESLT3	X	000404	12	309	265
GRESLT4	X	000410	12	310	267
GRESLT5	X	00041C	12	311	269
GRESLT6	X	000428	12	312	271
GRESLT7	X	000434	12	313	273
GRESLT8	X	000440	12	314	275
GRESLT9	X	00044C	12	315	277
IMAGE	1	000000	2240	0	
INVOKE	U	0003C0	1	296	299
LESSER	C	000720	16	326	162 172
LONGER	C	000740	16	328	181 189
PAD	X	00076C	4	335	
PGMINTC	F	00008C	4	85	103
PGMNPSW	X	0001D0	8	93	128
PGMOPSW	U	000150	1	87	
PROGCHK	H	000200	2	100	93
R0	U	000000	1	62	120 293
R1	U	000001	1	63	
R10	U	00000A	1	72	
R11	U	00000B	1	73	
R12	U	00000C	1	74	
R13	U	00000D	1	75	
R14	U	00000E	1	76	
R15	U	00000F	1	77	80
R2	U	000002	1	64	198 199
R3	U	000003	1	65	
R4	U	000004	1	66	199 205 206 210 217 218 222 229 230 234 241 242 246 253 254 258
R5	U	000005	1	67	198 207 211 219 223 231 235 243 247 255 259
R6	U	000006	1	68	121 123 135 138 144 147 153 156 162 165 171 174 180 183 189 192 209 221 233 245 257 297
R7	U	000007	1	69	122 123 136 145 154 163 172 181 190 205 217 229 241 253 297
R8	U	000008	1	70	138 147 156 165 174 183 192 209 221 233 245 257 294 298 300
R9	U	000009	1	71	137 146 155 164 173 182 191 208 220 232 244 256 301
RESULT10	F	000880	4	388	221 279
RESULT11	F	000890	4	389	233 281
RESULT12	F	0008A0	4	390	245 283

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
RESULT13	F	0008B0	4	391	257 285
RESULT2	F	000800	4	380	138 263
RESULT3	F	000810	4	381	147 265
RESULT4	F	000820	4	382	156 267
RESULT5	F	000830	4	383	165 269
RESULT6	F	000840	4	384	174 271
RESULT7	F	000850	4	385	183 273
RESULT8	F	000860	4	386	192 275
RESULT9	F	000870	4	387	209 277
SHORT1	C	000700	16	324	121 135 153 180
SHORT2	C	000710	16	325	122 136 145 190
START	H	00021E	2	114	90
STRTLABL	U	000000	1	61	84 87 89 92 99 323 379
TERM	C	000750	1	329	144 154
TEST10	F	000780	4	348	217
TEST11	F	000790	4	354	229
TEST12	F	0007A0	4	360	241
TEST13	F	0007B0	4	366	253
TEST9	F	000770	4	342	205
ZERO	A	000768	4	334	

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL   SIZE    POS       ADDR

Entry: 0

Image       IMAGE    2240  000-8BF  000-8BF
  Region             2240  000-8BF  000-8BF
    CSECT   CLST001  2240  000-8BF  000-8BF
```

```
   STMT                                                FILE NAME
1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\str-001-clst\str-001-clst.asm

** NO ERRORS FOUND **
```