```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                          2 *Testcase mvsos001:   MVCOS
                                          3 * Created and placed into the public domain
                                          4 * 27 JAN 2021 by Bob Polmanter.

                                00000000  00000001     6 R0       EQU   0          General Purpose Registers
                                00000001  00000001     7 R1       EQU   1
                                00000002  00000001     8 R2       EQU   2
                                00000003  00000001     9 R3       EQU   3
                                00000004  00000001    10 R4       EQU   4
                                00000005  00000001    11 R5       EQU   5
                                00000006  00000001    12 R6       EQU   6
                                00000007  00000001    13 R7       EQU   7
                                00000008  00000001    14 R8       EQU   8
                                00000009  00000001    15 R9       EQU   9
                                0000000A  00000001    16 R10      EQU   10
                                0000000B  00000001    17 R11      EQU   11
                                0000000C  00000001    18 R12      EQU   12
                                0000000D  00000001    19 R13      EQU   13
                                0000000E  00000001    20 R14      EQU   14
                                0000000F  00000001    21 R15      EQU   15
                                00000000  00000001    22 AR0      EQU   0          Access Registers
                                00000001  00000001    23 AR1      EQU   1
                                00000002  00000001    24 AR2      EQU   2
                                00000003  00000001    25 AR3      EQU   3
                                00000004  00000001    26 AR4      EQU   4
                                00000005  00000001    27 AR5      EQU   5
                                00000006  00000001    28 AR6      EQU   6
                                00000007  00000001    29 AR7      EQU   7
                                00000008  00000001    30 AR8      EQU   8
                                00000009  00000001    31 AR9      EQU   9
                                0000000A  00000001    32 AR10     EQU   10
                                0000000B  00000001    33 AR11     EQU   11
                                0000000C  00000001    34 AR12     EQU   12
                                0000000D  00000001    35 AR13     EQU   13
                                0000000E  00000001    36 AR14     EQU   14
                                0000000F  00000001    37 AR15     EQU   15
                                00000000  00000001    38 CR0      EQU   0          Control Registers
                                00000001  00000001    39 CR1      EQU   1
                                00000002  00000001    40 CR2      EQU   2
                                00000003  00000001    41 CR3      EQU   3
                                00000004  00000001    42 CR4      EQU   4
                                00000005  00000001    43 CR5      EQU   5
                                00000006  00000001    44 CR6      EQU   6
                                00000007  00000001    45 CR7      EQU   7
                                00000008  00000001    46 CR8      EQU   8
                                00000009  00000001    47 CR9      EQU   9
                                0000000A  00000001    48 CR10     EQU   10
                                0000000B  00000001    49 CR11     EQU   11
                                0000000C  00000001    50 CR12     EQU   12
                                0000000D  00000001    51 CR13     EQU   13
                                0000000E  00000001    52 CR14     EQU   14
                                0000000F  00000001    53 CR15     EQU   15
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    55 ********************************************************************************
                                                    56 *
                                                    57 *    These tests and this programming were validated on a z114
                                                    58 *    using a z/VM 6.4 virtual machine on 27 January 2021.
                                                    59 *
                                                    60 ********************************************************************************
                                                    61 *
                                                    62 * Tests performed with MVCOS:
                                                    63 *
                                                    64 * 1.  Execute the MVCOS instruction iteratively, each time trying a
                                                    65 *     different combination of machine state, address space control
                                                    66 *     mode, MVCOS operand 1 control modes, MVCOS operand 2 control
                                                    67 *     modes, and key enablement in both operand 1 and then operand 2.
                                                    68 *     These individual tests are nested in a series of loops, so that
                                                    69 *     each state or mode is tested with each other combination of
                                                    70 *     states or modes in an exhaustive way.  A visual description of
                                                    71 *     the nested loops and their related tests is shown below.
                                                    72 *
                                                    73 *     After execution of each MVCOS instruction, the results of the
                                                    74 *     actual data moved are checked to determine if the data fetched
                                                    75 *     indeed came from the specified address space, and if the data
                                                    76 *     stored was indeed placed into the specified address space, as
                                                    77 *     determined by the settings in R0 for operand 2 and operand 1,
                                                    78 *     respectively.
                                                    79 *
                                                    80 * Upon success this is the number of tests performed:
                                                    81 *
                                                    82 *   Successfully completed MVCOS executions:             384
                                                    83 *   Expected protection check events:                  1,152
                                                    84 *   Expected special operation exception events:          92
                                                    85 *   TOTAL TESTS:                                       1,328
                                                    86 *
                                                    87 * MVCOS Results Failure:      Disabled Wait PSW X'BAD1'
                                                    88 * Overall Test Failure:       Disabled Wait PSW X'BAD9'
                                                    89 * Overall Test Success:       Disabled Wait PSW X'0000'
                                                    90 * Unexpected Program Check:   Disabled Wait PSW X'DEAD'
                                                    91 *
                                                    92 * The expected protection checks arise from enabling key controlled
                                                    93 * protection in register 0 as specified by the instruction. Keys
                                                    94 * have been deliberately set to allow some accesses and to fail some
                                                    95 * access attempts. Because either MVCOS operands (or both) could be
                                                    96 * using a failing key, there is more of these failures, whereas
                                                    97 * success requires both operands to have the right key setting.
                                                    98 *
                                                    99 * The expected special operation exceptions arise when the Home
                                                   100 * address space control mode is selected in Register 0 -AND- the
                                                   101 * PSW is in the problem state. This applies to operand 1 OAC only.
                                                   102 * This tests that MVCOS is honoring that specification as documented
                                                   103 * in the Principles.
                                                   104 *
                                                   105 ********************************************************************************
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    107 ************************************************************************
                                                    108 *
                                                    109 *                              TEST METHOD
                                                    110 *
                                                    111 * 1. Setup.
                                                    112 *
                                                    113 * Three address spaces are created: Primary, Secondary, and Home.
                                                    114 * Literals are placed into a page frame belonging to each one of the
                                                    115 * address spaces identifying them by PRI, SEC, and HOM. Those pages
                                                    116 * are the targets of MVCOS operand 1.  A second set of
                                                    117 * literals is placed into another page frame belonging to each
                                                    118 * one of the address spaces identifying them also as FROMPRI, FROMSEC,
                                                    119 * and FROMHOM. These pages are fetched by MVCOS operand 2.
                                                    120 *
                                                    121 * The literals in the target page frames of each address space are
                                                    122 * located at virtual 00010FF0 in each space.
                                                    123 *
                                                    124 * The literals in the 'from' page frames of each address space are
                                                    125 * located at virtual 00012FF8 in each space.
                                                    126 *
                                                    127 * The locations are set to cause MVCOS to move data across page
                                                    128 * boundaries.
                                                    129 *
                                                    130 * The nested loops are entered to set the register 0 MVCOS controls.
                                                    131 *
                                                    132 *
                                                    133 * 2. Executing MVCOS.
                                                    134 *
                                                    135 * A FROM literal is moved to the target page by MVCOS. The address
                                                    136 * space fetched from and the address space target are of course
                                                    137 * determined by the MVCOS controls in Register 0.
                                                    138 *
                                                    139 *
                                                    140 * 3. Validation.
                                                    141 *
                                                    142 * After the MVCOS, the register 0 controls are extracted and used
                                                    143 * to determine programatically which address space literal was
                                                    144 * requested to be moved to which target, and the results are compared
                                                    145 * to determine if those literals are actually where they are supposed
                                                    146 * to be.
                                                    147 *
                                                    148 * After successful validation, the original placement of the literals
                                                    149 * is restored, and the next loop iteration advances to the next test.
                                                    150 *
                                                    151 ************************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    153 ************************************************************************
                                                    154 *
                                                    155 *                    DEBUGGING THIS TEST PROGRAM
                                                    156 *
                                                    157 * If any MVCOS test fails (data from the specified address space is
                                                    158 * not identified as expected), the machine will be halted to preserve
                                                    159 * results and a disabled wait PSW of X'BAD1' will be loaded. Use
                                                    160 * register 0 and the address space control value in byte PSWASC to
                                                    161 * determine what should have been moved to where.  View the literals
                                                    162 * at virtual location X'00010FF0' to determine which space you are
                                                    163 * viewing and what FROM literal was actually  moved to that space.
                                                    164 * Compare that to the R0 controls and PSWASC mode to validate whether
                                                    165 * MVCOS moved correctly.  You can also use the memory map listed
                                                    166 * below to view the real pages by address to inspect the literals in
                                                    167 * each address space.
                                                    168 *
                                                    169 * If an unexpected program check occurs, the PSW will be loaded with
                                                    170 * a disabled wait code X'DEAD'. The machine is halted immediately
                                                    171 * upon occurance and no registers are altered; hence their values
                                                    172 * reflect the state of the failure.
                                                    173 *
                                                    174 *
                                                    175 ************************************************************************
```

```
 LOC  OBJECT CODE  ADDR1  ADDR2  STMT
                                  177 ********************************************************************************
                                  178 *
                                  179 *                    Memory Map - REAL STORAGE
                                  180 *
                                  181 *         Hex
                                  182 * RAddr   Len     Description
                                  183 * -----   ----    ------------------------------------------------------
                                  184 *     0   2000  - Absolute page 0 and 1
                                  185 *  2000   2000  - Program code
                                  186 *  4000   1000  - Segment table, primary space
                                  187 *  5000   1000  - Segment table, secondary space
                                  188 *  6000   1000  - Segment table, home space
                                  189 *  7000    800  - Page tables, primary space
                                  190 *  7800    800  - Page tables, secondary space
                                  191 *  8000    800  - Page tables, home space
                                  192 *  9000   1000  - Primary ASTE, DUCT, DU-AL, ALE blocks
                                  193 *  A000   1000  - Home ASTE block
                                  194 *  B000   5000  - unused; available
                                  195 * 10000  10000  - Pages backing Home virtual space at vaddr 10000
                                  196 * 20000  10000  - Pages backing Primary virtual space at vaddr 10000
                                  197 * 30000  10000  - Pages backing Secondary virtual space at vaddr 10000
                                  198 *
                                  199 *
                                  200 *                    Memory Map - VIRTUAL STORAGE
                                  201 *
                                  202 * VAddr   Len   Raddr Key   Description
                                  203 * -----   ----  ----- ---   ------------------------------------
                                  204 * 00000  10000  00000  00 - Common V=R storage (all address spaces)
                                  205 * 10000  10000  10000  00 - Home space storage
                                  206 * 10000  10000  20000  40 - Primary space storage
                                  207 * 10000  10000  30000  80 - Secondary space storage
                                  208 *
                                  209 *
                                  210 ********************************************************************************
                                  211 *                      IN EACH ADDRESS SPACE:
                                  212 *
                                  213 * VADDR 10FF0 length 32:  Literal identifying the space target
                                  214 *                          (e.g., CL16'PRI-PG1',CL16'PRI-PG2'
                                  215 *
                                  216 * VADDR 12FF8 length 16:  Literal identifying the space source
                                  217 *                          (e.g., CL16'FROMPRI1FROMPRI2'
                                  218 *
                                  219 * After a successful MVCOS, the storage at location 10FF0 would look
                                  220 * like this example with a move from secondary to primary:
                                  221 *
                                  222 * VADDR 10FF0   CL32'PRI-PG1FROMSEC1FROMSEC2       '
                                  223 *
                                  224 * thus showing that the target area is still named PRI,and the data
                                  225 * came from two pages in the secondary space.
                                  226 *
                                  227 *
                                  228 ********************************************************************************
```

```
 LOC  OBJECT CODE  ADDR1  ADDR2  STMT

                                  230 ********************************************************************************
                                  231 *
                                  232 *              VISUAL DESCRIPTION OF NESTED LOOP TESTS
                                  233 *
                                  234 * The sequence of loops nested below allows each combination to be
                                  235 * tested one at a time.  OAC1 and OAC2 are the Operand Access
                                  236 * Control 1 and 2, respectively, in Register 0 that control MVCOS.
                                  237 *
                                  238 * Loop  iterations  Description
                                  239 *   1        2      Supervisor state, then problem state
                                  240 *   2        4      Cycle through each PSW ASC mode P,AR,S,H
                                  241 *   3        2      OAC1 A validity bit off, then on
                                  242 *   4        4      When OAC1 A=1, cycle through each ASC mode in OAC1
                                  243 *   5        3      When OAC1 A=1 & OAC1 is AR, cycle ALETs 0,1,2 oper1
                                  244 *   6        2      OAC2 A validity bit off, then on
                                  245 *   7        4      When OAC2 A=1, cycle through each ASC mode in OAC2
                                  246 *   8        3      When OAC2 A=1 & OAC2 is AR, cycle ALETs 0,1,2 oper2
                                  247 *   9        2      OAC1 K validity bit off, then on
                                  248 *  10        2      OAC2 K validity bit off, then on
                                  249 *
                                  250 *                   Execute MVCOS
                                  251 *
                                  252 *                   Check results; PSW=X'BAD1' if failed
                                  253 *                   Next loop 10
                                  254 *                   Next loop 9
                                  255 *                   Next loop 8
                                  256 *                   Next loop 7
                                  257 *                   Next loop 6
                                  258 *                   Next loop 5
                                  259 *                   Next loop 4
                                  260 *                   Next loop 3
                                  261 *                   Next loop 2
                                  262 *                   Next loop 1
                                  263 *                   Terminate with success, PSW=X'000'
                                  264 *
                                  265 * Note on loop 4 & 7: when either OAC A validity bit is 0, then the
                                  266 * PSW ASC mode is used by MVCOS and the iteration count is 1.
                                  267 *
                                  268 * Note on loop 5 & 8: when either OAC A validity bit is 0 -AND- the
                                  269 * PSW ASC mode is P,S,or H, then the PSW ASC mode is used by MVCOS and
                                  270 * the iteration count is 1.  However, if the PSW ASC mode is AR, then
                                  271 * the iteration count remains 3 for these loops so the ALETs can be
                                  272 * cycled through each test for each operand 1 or operand 2, in turn.
                                  273 *
                                  274 ********************************************************************************
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                    276 ********************************************************************************
                                                    277 *                        Low Core / Prefix Area
                                                    278 ********************************************************************************
                                                    279 *
                             00000000  0000A03F     280 MVCOS001 START 0
                             00000000  00000001     281 STRTLABL EQU   *
00000000                     00000000               282          USING STRTLABL,0


                                                    284 *              Selected z/Arch low core layout
                                                    285 *
00000000                     00000000  00000088     286          ORG   STRTLABL+X'88'      interrupt code area EC mode
00000088  00000000                                  287 SVCINTC  DC    X'00000000'          SVC interrupt code area
0000008C  00000000                                  288 PGMINTC  DC    X'00000000'          Prog check interrupt code area
                                                    289 *
00000090                     00000090  00000140     290          ORG   STRTLABL+X'140'
00000140  00000000 00000000                         291 SVCOPSW  DS    XL16                 SVC old PSW
00000150  00000000 00000000                         292 PGMOPSW  DS    XL16                 Program check old PSW
                                                    293 *
00000160                     00000160  000001A0     294          ORG   STRTLABL+X'1A0'     New PSWs
000001A0  00000000 80000000                         295 RESTART  DC    X'00000000',X'80000000',A(0),A(START)    DAT OFF
000001B0  00000000 00000000                         296 EXTNPSW  DC    XL16'00'
000001C0  04004000 80000000                         297 SVCNPSW  DC    X'04004000',X'80000000',A(0),A(SVCFLIH)  DAT ON, AR MODE
000001D0  04004000 80000000                         298 PGMNPSW  DC    X'04004000',X'80000000',A(0),A(PGMFLIH)  DAT ON, AR MODE


                                                    300 *                     Test Counters
                                                    301 *
000001E0                     000001E0  00000200     302          ORG   STRTLABL+X'200'  Test counters
00000200  0000000C                                  303 MVCOSOK  DC    PL4'0'           # of successful MVCOS   =     384
00000204  0000000C                                  304 PIC04    DC    PL4'0'           # of Pchecks 04         =   1,152
00000208  0000000C                                  305 PIC13    DC    PL4'0'           # of Pchecks 13         =      92
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                   307 ********************************************************************************
                                                   308 *                          Main program
                                                   309 ********************************************************************************
                                                   310 *
0000020C                     0000020C  00002000    311          ORG   STRTLABL+X'2000'
00002000  0DF0                                     312 START    BASR  R15,0
00002002  06F0                                     313          BCTR  R15,0
00002004  06F0                                     314          BCTR  R15,0
00002006                     00002000              315          USING START,R15
                                                   316 *
00002006  1B22                                     317          SR    R2,R2                    STATUS REG SET TO 0
00002008  4130 0001                  00000001      318          LA    R3,1                     R3=1 MEANS SET Z/ARCH MODE
                                                   319 *                                       R3=0 MEANS SET ESA/390 MODE
0000200C  1B44                                     320          SR    R4,R4                    CPU Addr = 0
0000200E  AE24 0012                  00000012      321          SIGP  R2,R4,X'12'              X'12' = SET ARCHITECTURE
                                                   322 *
00002012  4100 0008                  00000008      323          LA    R0,X'08'                 Set KEY=0 fetch prot enabled
00002016  4120 0040                  00000040      324          LA    R2,64                    # of real pages to set
0000201A  1B11                                     325          SR    R1,R1                    Starting addr
                                                   326 *
0000201C  B22B 0001                                327 SET000   SSKE  R0,R1                    Set the key
00002020  A71A 1000                                328          AHI   R1,4096                  Bump to next page
00002024  4620 F01C                  0000201C      329          BCT   R2,SET000
                                                   330 *
00002028  9A0F F530                  00002530      331          LAM   AR0,AR15,AREGS           Clear all ARs
0000202C  EB0F F4B0 002F             000024B0      332          LCTLG CR0,CR15,CREGS           Load all the CRs
00002032  8000 F3D4                  000023D4      333          SSM   =X'04'                   Turn on DAT
                                                   334 *
00002036  5850 F57C                  0000257C      335          L     R5,VADDRTO               Get vaddr in 1st virtual page
0000203A  5860 F580                  00002580      336          L     R6,VADDRFRM              Copy
                                                   337 *
0000203E  D21F 5000 F3E0  00000000   000023E0      338          MVC   0(32,R5),PRIPG1          Set literal identifier in pages
00002044  D20F 6000 F440  00000000   00002440      339          MVC   0(16,R6),FROMPRI         Set literal identifier in pages
                                                   340 *
0000204A  B219 0100                  00000100      341          SAC   SECMODE                  Secondary mode
0000204E  D21F 5000 F400  00000000   00002400      342          MVC   0(32,R5),SECPG1          Set literal identifier in pages
00002054  D20F 6000 F450  00000000   00002450      343          MVC   0(16,R6),FROMSEC         Set literal identifier in pages
                                                   344 *
0000205A  B219 0300                  00000300      345          SAC   HOMEMODE                 Home space mode
0000205E  D21F 5000 F420  00000000   00002420      346          MVC   0(32,R5),HOMPG1          Set literal identifier in pages
00002064  D20F 6000 F460  00000000   00002460      347          MVC   0(16,R6),FROMHOM         Set literal identifier in pages
                                                   348 *
0000206A  B219 0200                  00000200      349          SAC   ARMODE                   Enter AR mode
0000206E  5800 F360                  00002360      350          L     R0,=X'10031003'          Initialize MVCOS controls; on
                                                   351 *                                        first pass below this will be
                                                   352 *                                        set to X'00000000' for 1st test
00002072  920C F599                  00002599      353          MVI   PSWASC,X'0C'             Initialize PSW ASC ctl byte; on
                                                   354 *                                        first pass below this will be
                                                   355 *                                        set to X'00' (PRI) for 1st test
00002076  9201 F598                  00002598      356          MVI   PSWSTATE,X'01'           Initialize PSW state control; on
                                                   357 *                                        first pass below this will be
                                                   358 *                                        set to X'00',SUPRV for 1st test
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    360 ********************************************************************************
0000207A  41E0 0002                     00000002   361          LA    R14,2                    # of PSW state tests
                               0000207E  00000001   362 STATE000 EQU   *
0000207E  9701 F598                     00002598   363          XI    PSWSTATE,X'01'           Flip current PSW state ctl byte
00002082  4310 F598                     00002598   364          IC    R1,PSWSTATE              Get new state ctl byte
00002086  4410 F08E                     0000208E   365          EX    R1,SVCSTATE              Flip to next PSW state
0000208A  47F0 F090                     00002090   366          B     SKIP001                  Continue test prep
                                                    367 *
0000208E  0A00                                     368 SVCSTATE SVC   0                        Executed instruction
                               00002090  00000001   369 SKIP001  EQU   *


                                                    371 ********************************************************************************
00002090  41D0 0004                     00000004   372          LA    R13,4                    # of PSW ASC tests
                               00002094  00000001   373 ASC010   EQU   *
00002094  4310 F599                     00002599   374          IC    R1,PSWASC                Get last mode used
00002098  4110 1004                     00000004   375          LA    R1,X'04'(,R1)            Increment bit 5 to next ASC mode
0000209C  5410 F364                     00002364   376          N     R1,=X'0000000C'          Keep only bits 4 and 5
000020A0  4210 F599                     00002599   377          STC   R1,PSWASC                Set new mode to use


                                                    379 ********************************************************************************
000020A4  41C0 0002                     00000002   380          LA    R12,2                    # From A validity tests
                               000020A8  00000001   381 TOA000   EQU   *
000020A8  1810                                     382          LR    R1,R0                    Copy control bits
000020AA  5410 F368                     00002368   383          N     R1,=A(OAC1A)             Keep only bits we want
000020AE  5710 F368                     00002368   384          X     R1,=A(OAC1A)             Flip these bits
000020B2  5400 F36C                     0000236C   385          N     R0,=A(X'FFFFFFFF'-OAC1A) Force these bits off
000020B6  1601                                     386          OR    R0,R1                    Set ctl based on flip results
000020B8  5400 F370                     00002370   387          N     R0,=A(X'FFFFFFFF'-(ASCHOM*65536)) Force bits off in OAC1
000020BC  41B0 0001                     00000001   388          LA    R11,1                    Assume 1 test if A=0 (using PSW)
000020C0  A700 0001                                389          TMLH  R0,ASCA                  Was A set on or off?
000020C4  4780 F0FE                     000020FE   390          BZ    TOAS200                  A is off, use PSW ASC
                                                    391 *
                                                    392 *                                   A=1: rotate thru OAC1AS modes
000020C8  5600 F374                     00002374   393          O     R0,=A(ASCHOM*65536)      Force on; will wrap to 00 next
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     395 ********************************************************************************
000020CC  41B0 0004                       00000004   396          LA    R11,4                    4 test to rotate thru OAC1 ASCs
                               000020D0   00000001   397 TOAS000  EQU   *
000020D0  A700 0001                                  398          TMLH  R0,ASCA                  Was A set on or off?
000020D4  4780 F0FE                       000020FE   399          BZ    TOAS200                  A is off, use PSW ASC
                                                     400 *                                       A is on, do OAC1 AS changes
000020D8  1810                                       401          LR    R1,R0                    Copy control bits
000020DA  5A10 F378                       00002378   402          A     R1,=X'00400000'          Increment bit to next ASC mode
000020DE  5410 F374                       00002374   403          N     R1,=A(ASCHOM*65536)      Keep only the bits we want
000020E2  5400 F370                       00002370   404          N     R0,=A(X'FFFFFFFF'-(ASCHOM*65536)) Force bits off in OAC1
000020E6  1601                                       405          OR    R0,R1                    Set ctl based on flip results
                                                     406 *
000020E8  41A0 0001                       00000001   407          LA    R10,1                    1 test required if ASC is P,S,H
000020EC  1810                                       408          LR    R1,R0                    Copy control bits
000020EE  5410 F37C                       0000237C   409          N     R1,=A(OAC1A+ASCHOM*65536) Keep only these bits
000020F2  5510 F380                       00002380   410          CL    R1,=A(OAC1A+ASCAR*65536) Using MVCOS control AR mode?
000020F6  4770 F128                       00002128   411          BNE   TOAS290                  No. Use 1 test in R10 for P,S,H
000020FA  47F0 F10A                       0000210A   412          B     TOAS210                  Yes. 3 tests in R10 for AR
                                                     413 *
                               000020FE   00000001   414 TOAS200  EQU   *
000020FE  41A0 0001                       00000001   415          LA    R10,1                    1 test required if PSW is P,S,H
00002102  9504 F599                       00002599   416          CLI   PSWASC,X'04'             Using ASC=AR ?
00002106  4770 F128                       00002128   417          BNE   TOAS290                  No. only 1 test per ASC mode
                                                     418 *
                               0000210A   00000001   419 TOAS210  EQU   *

                                                     421 ********************************************************************************
0000210A  41A0 0003                       00000003   422          LA    R10,3                    3 tests required for ASC=AR
                               0000210E   00000001   423 TOAS220  EQU   *
0000210E  5810 F578                       00002578   424          L     R1,TALET                 Get from ALET
00002112  B24E 0051                                  425          SAR   AR5,R1                   Set in from AR
00002116  4110 1001                       00000001   426          LA    R1,1(,R1)                Bump ALET
0000211A  5910 F384                       00002384   427          C     R1,=F'3'                 Exceeded max of 2?
0000211E  4740 F124                       00002124   428          BL    TOAS230                  No
00002122  1B11                                       429          SR    R1,R1                    Restart back at ALET 0
                                                     430 *
                               00002124   00000001   431 TOAS230  EQU   *
00002124  5010 F578                       00002578   432          ST    R1,TALET                 Save updated ALET
                                                     433 *
                               00002128   00000001   434 TOAS290  EQU   *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                     436 ********************************************************************************
00002128  4190 0002                       00000002   437          LA    R9,2                    # From A validity tests
                               0000212C   00000001   438 FRMA000  EQU   *
0000212C  1810                             00000001   439          LR    R1,R0                   Copy control bits
0000212E  5410 F388                        00002388   440          N     R1,=A(OAC2A)            Keep only bits we want
00002132  5710 F388                        00002388   441          X     R1,=A(OAC2A)            Flip these bits
00002136  5400 F38C                        0000238C   442          N     R0,=A(X'FFFFFFFF'-OAC2A) Force these bits off
0000213A  1601                             00000001   443          OR    R0,R1                   Set ctl based on flip results
0000213C  5400 F390                        00002390   444          N     R0,=A(X'FFFFFFFF'-ASCHOM) Force these bits off in OAC2
00002140  4180 0001                        00000001   445          LA    R8,1                    Assume 1 test if A=0 (using PSW)
00002144  A701 0001                        00000001   446          TMLL  R0,ASCA                 Was A set on or off?
00002148  4780 F182                        00002182   447          BZ    FRMAS200                A is off, use PSW ASC
                                           00000001   448 *
                                           00000001   449 *                                    A=1: rotate thru OAC2AS modes
0000214C  5600 F394                        00002394   450          O     R0,=A(ASCHOM)           Force on; will wrap to 00 next


                                                     452 ********************************************************************************
00002150  4180 0004                        00000004   453          LA    R8,4                    4 test to rotate thru OAC2 ASCs
                               00002154   00000001   454 FRMAS000 EQU   *
00002154  A701 0001                        00000001   455          TMLL  R0,ASCA                 Was A set on or off?
00002158  4780 F182                        00002182   456          BZ    FRMAS200                A is off, use PSW ASC
                                           00000001   457 *                                    A is on, do OAC2 AS changes
0000215C  1810                             00000001   458          LR    R1,R0                   Copy control bits
0000215E  4110 1040                        00000040   459          LA    R1,B'01000000'(,R1)     Increment bit 1 to next ASC mode
00002162  5410 F394                        00002394   460          N     R1,=A(ASCHOM)           Keep only the bits we want
00002166  5400 F390                        00002390   461          N     R0,=A(X'FFFFFFFF'-ASCHOM) Force these bits off in OAC2
0000216A  1601                             00000001   462          OR    R0,R1                   Set ctl based on flip results
                                           00000001   463 *
0000216C  4170 0001                        00000001   464          LA    R7,1                    1 test required if ASC is P,S,H
00002170  1810                             00000001   465          LR    R1,R0                   Copy control bits
00002172  5410 F398                        00002398   466          N     R1,=A(OAC2A+ASCHOM)     Keep only these bits
00002176  5510 F39C                        0000239C   467          CL    R1,=A(OAC2A+ASCAR)      Using MVCOS control AR mode?
0000217A  4770 F1AC                        000021AC   468          BNE   FRMAS290                No. Use 1 test in R7 for P,S,H
0000217E  47F0 F18E                        0000218E   469          B     FRMAS210                Yes. 3 tests in R7 for AR
                                           00000001   470 *
                               00002182    00000001   471 FRMAS200 EQU   *
00002182  4170 0001                        00000001   472          LA    R7,1                    1 test required if PSW is P,S,H
00002186  9504 F599                        00002599   473          CLI   PSWASC,X'04'            Using ASC=AR ?
0000218A  4770 F1AC                        000021AC   474          BNE   FRMAS290                No. only 1 test per ASC mode
                                           00000001   475 *
                               0000218E    00000001   476 FRMAS210 EQU   *
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                    478 ********************************************************************************
0000218E  4170 0003                       00000003  479          LA    R7,3                    3 tests required for ASC=AR
                              00002192    00000001  480 FRMAS220 EQU   *
00002192  5810 F574                       00002574  481          L     R1,FALET                Get from ALET
00002196  B24E 0061                                 482          SAR   AR6,R1                  Set in from AR
0000219A  4110 1001                       00000001  483          LA    R1,1(,R1)               Bump ALET
0000219E  5910 F384                       00002384  484          C     R1,=F'3'                Exceeded max of 2?
000021A2  4740 F1A8                       000021A8  485          BL    FRMAS230                No
000021A6  1B11                                      486          SR    R1,R1                   Restart back at ALET 0
                                                    487 *
                              000021A8    00000001  488 FRMAS230 EQU   *
000021A8  5010 F574                       00002574  489          ST    R1,FALET                Save updated ALET
                                                    490 *
                              000021AC    00000001  491 FRMAS290 EQU   *









                                                    493 ********************************************************************************
000021AC  4140 0002                       00000002  494          LA    R4,2                    # To Key tests
                              000021B0    00000001  495 TKEY000  EQU   *                       To Key
000021B0  1810                                      496          LR    R1,R0                   Copy control bits
000021B2  5410 F3A0                       000023A0  497          N     R1,=A(OAC1KEY+OAC1K)    Keep only bits we want
000021B6  5710 F3A0                       000023A0  498          X     R1,=A(OAC1KEY+OAC1K)    Flip these bits
000021BA  5400 F3A4                       000023A4  499          N     R0,=A(X'FFFFFFFF'-OAC1KEY-OAC1K) Force these bits off
000021BE  1601                                      500          OR    R0,R1                   Set ctl based on flip results


                                                    502 ********************************************************************************
000021C0  4130 0002                       00000002  503          LA    R3,2                    # From Key tests
                              000021C4    00000001  504 FKEY000  EQU   *                       From Key
000021C4  1810                                      505          LR    R1,R0                   Copy control bits
000021C6  5410 F3A8                       000023A8  506          N     R1,=A(OAC2KEY+OAC2K)    Keep only bits we want
000021CA  5710 F3A8                       000023A8  507          X     R1,=A(OAC2KEY+OAC2K)    Flip these bits
000021CE  5400 F3AC                       000023AC  508          N     R0,=A(X'FFFFFFFF'-OAC2KEY-OAC2K) Force these bits off
000021D2  1601                                      509          OR    R0,R1                   Set ctl based on flip results
                                                    510 *
000021D4  1B22                                      511          SR    R2,R2                   Clear for ICM
000021D6  4320 F599                       00002599  512          IC    R2,PSWASC               Get ASC we need to test
000021DA  9101 008B                       0000008B  513          TM    SVCINTC+3,X'01'         Are we in problem state?
000021DE  4780 F1EA                       000021EA  514          BZ    BEGIN000                No.  Do every test
000021E2  BD21 F3D5                       000023D5  515          CLM   R2,1,=X'0C'             Entering HOME ASC mode?
000021E6  4780 F26A                       0000226A  516          BE    NEXTTEST                Y, not permitted in prob state
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     518 ********************************************************************************
                                                     519 *
                                                     520 *     Now do the actual 'MVCOS' and check the results afterwards
                                                     521 *
                                                     522 ********************************************************************************
                                                     523 *
                               000021EA  00000001    524 BEGIN000 EQU   *
000021EA  5812 F588                      00002588    525          L     R1,SACIDX(R2)           Get corresponding SAC bits
000021EE  B219 1000                      00000000    526          SAC   0(R1)                   Put machine in mode we need
                                                     527 *
000021F2  4110 0010                      00000010    528          LA    R1,16                   Length to move
000021F6  C810 5008 6000      00000008  00000000    529          MVCOS 8(R5),0(R6),R1
                               000021FC  00000001    530 MVCOS    EQU   *                       Addr after the instruction
                                                     531 *
000021FC  B219 0200                      00000200    532          SAC   ARMODE                  Resume AR mode
00002200  1B11                                       533          SR    R1,R1                   Clear for IC
00002202  4310 F599                      00002599    534          IC    R1,PSWASC               Get PSW ASC mode bits
00002206  A700 0001                                  535          TMLH  R0,ASCA                 Is AS setting in OAC1 valid?
0000220A  4780 F218                      00002218    536          BZ    CHK010                  No, use the PSW ASC in R1
0000220E  1810                                       537          LR    R1,R0                   Copy current setting
00002210  5410 F3B0                      000023B0    538          N     R1,=X'00C00000'         Keep OAC1 AS bits
00002214  8810 0014                      00000014    539          SRL   R1,20                   Make AS value a 4-byte index
                                                     540 *
                               00002218  00000001    541 CHK010   EQU   *
00002218  4401 F2BA                      000022BA    542          EX    R0,SETAR(R1)            Set AR1 to access the right AS
0000221C  5810 F57C                      0000257C    543          L     R1,VADDRTO              -> literal target area
                                                     544 *
00002220  1B22                                       545          SR    R2,R2                   Clear for IC
00002222  4320 F599                      00002599    546          IC    R2,PSWASC               Get PSW ASC mode bits
00002226  A701 0001                                  547          TMLL  R0,ASCA                 Is AS setting in OAC2 valid?
0000222A  4780 F238                      00002238    548          BZ    CHK020                  No, use the PSW ASC in R2
                                                     549 *
0000222E  1820                                       550          LR    R2,R0                   Copy current setting
00002230  5420 F3B4                      000023B4    551          N     R2,=X'000000C0'         Keep OAC2 AS bits
00002234  8820 0004                      00000004    552          SRL   R2,4                    Make AS value a 4-byte index
                                                     553 *
                               00002238  00000001    554 CHK020   EQU   *
00002238  4402 F2CA                      000022CA    555          EX    R0,GETALET(R2)          Get the MVCOS operand 2 ALET
0000223C  8920 0004                      00000004    556          SLL   R2,4                    Multiply by 16 to make index
00002240  4122 F440                      00002440    557          LA    R2,FROMPRI(R2)          -> space identifier literal
                                                     558 *
00002244  D50F 1008 2000      00000008  00000000    559          CLC   8(16,R1),0(R2)          Check if MVCOS worked
0000224A  4780 F252                      00002252    560          BE    CHK100                  TEST SUCCESS
0000224E  B2B2 F470                      00002470    561          LPSWE BADMVCOS                Stop machine if test failed
                                                     562 *
                               00002252  00000001    563 CHK100   EQU   *
00002252  FA30 0200 F3D6      00000200  000023D6    564          AP    MVCOSOK,=P'1'           Increment # successful tests
00002258  B24F 0021                                  565          EAR   R2,AR1                  Get the ALET we loaded into AR1
0000225C  8920 0005                      00000005    566          SLL   R2,5                    Multiply by 32 to make index
00002260  4122 F3E0                      000023E0    567          LA    R2,PRIPG1(R2)           -> space identifier literal
00002264  D21F 1000 2000      00000000  00000000    568          MVC   0(32,R1),0(R2)          Restore original id literal
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                     570 ********************************************************************************
                                                     571 *                        Loop through all tests
                                                     572 ********************************************************************************
                                0000226A  00000001   573 NEXTTEST EQU   *
0000226A  4630 F1C4                       000021C4   574          BCT   R3,FKEY000               Vary OAC2 K bit
0000226E  4640 F1B0                       000021B0   575          BCT   R4,TKEY000               Vary OAC1 K bit
00002272  4670 F192                       00002192   576          BCT   R7,FRMAS220              Vary from ALETs when OAC2 ASC=AR
00002276  4680 F154                       00002154   577          BCT   R8,FRMAS000              Cycle through OAC2 ASC modes
0000227A  4690 F12C                       0000212C   578          BCT   R9,FRMA000               Vary OAC2 A bit
0000227E  46A0 F10E                       0000210E   579          BCT   R10,TOAS220              Vary To ALETs when OAC1 ASC=AR
00002282  46B0 F0D0                       000020D0   580          BCT   R11,TOAS000              Cycle through OAC1 ASC modes
00002286  46C0 F0A8                       000020A8   581          BCT   R12,TOA000               Vary OAC1 A bit
0000228A  46D0 F094                       00002094   582          BCT   R13,ASC010               Switch to next PSW ASC mode
0000228E  46E0 F07E                       0000207E   583          BCT   R14,STATE000             Switch to next PSW state

                                                     585 ********************************************************************************
                                                     586 *                              END OF TEST
                                                     587 ********************************************************************************
00002292  0A00                                       588          SVC   0                        Back to supervisor state
00002294  F931 0200 F3D0  00000200  000023D0   589          CP    MVCOSOK,=P'384'          Expected count?
0000229A  4770 F2B6                       000022B6   590          BNE   FAILTEST                 No, test failure
0000229E  F932 0204 F3D7  00000204  000023D7   591          CP    PIC04,=P'1152'           Expected count?
000022A4  4770 F2B6                       000022B6   592          BNE   FAILTEST                 No, test failure
000022A8  F931 0208 F3D2  00000208  000023D2   593          CP    PIC13,=P'92'             Expected count?
000022AE  4770 F2B6                       000022B6   594          BNE   FAILTEST                 No, test failure
000022B2  B2B2 F490                       00002490   595          LPSWE TESTGOOD                 Test SUCCESS
000022B6  B2B2 F480                       00002480   596 FAILTEST LPSWE TESTBAD                  Test FAILURE

                                                     598 ********************************************************************************
                                                     599 *        SETAR and GETALET are blocks of EXecuted instructions
                                                     600 ********************************************************************************
                                                     601 *
000022BA  9A11 F3B8                       000023B8   602 SETAR    LAM   AR1,AR1,=F'0'   AS=00   Primary, set ALET=0
000022BE  B24D 0015                                  603          CPYA  AR1,AR5              01   AR, set AR1 to MVCOS Operand 1
000022C2  9A11 F3BC                       000023BC   604          LAM   AR1,AR1,=F'1'        10   Secondary, set ALET=1
000022C6  9A11 F3C0                       000023C0   605          LAM   AR1,AR1,=F'2'        11   Home set ALET=2

000022CA  4120 0000                       00000000   607 GETALET  LA    R2,0              AS=00   Primary, set ALET=0
000022CE  B24F 0026                                  608          EAR   R2,AR6               01   AR, set R2 to MVCOS Operand 2 AR
000022D2  4120 0001                       00000001   609          LA    R2,1                 10   Secondary, set ALET=1
000022D6  4120 0002                       00000002   610          LA    R2,2                 11   Home set ALET=2

                                                     612 ********************************************************************************
                                                     613 * SVC 0: Set supervisor state in PSW, SVC 1: Set problem state in PSW
                                                     614 ********************************************************************************
                          000022DA  00000001   615 SVCFLIH  EQU   *                        SVC Interruption Routine
000022DA  D200 0141 008B  00000141  0000008B   616          MVC   SVCOPSW+1(1),SVCINTC+3   Set state based on SVC num
000022E0  B2B2 0140                       00000140   617          LPSWE SVCOPSW                  Resume execution
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                    619 ************************************************************************
                                                    620 *                    HERE FOR PROGRAM CHECKS
                                                    621 ************************************************************************
                                                    622 *
                               000022E4  00000001   623 PGMFLIH  EQU   *                          Program check interruptions
000022E4  9513 008F                      0000008F   624           CLI   PGMINTC+3,X'13'          Was this a special op exception?
000022E8  4780 F316                      00002316   625           BE    PGM13                    Yes, use microscope
000022EC  9504 008F                      0000008F   626           CLI   PGMINTC+3,X'04'          Was this a protection exception?
000022F0  4770 F35A                      0000235A   627           BNE   PGMSTOP                  No, stop immediate
                                                    628 *
                               000022F4  00000001   629 PGM04    EQU   *                          Examine PIC 4
000022F4  5000 F570                      00002570   630           ST    R0,WORK                  Save current control bits
000022F8  D403 F570 F3C4       00002570  000023C4   631           NC    WORK,=A(OAC1K+OAC2K)     Either key validity bit = 1?
000022FE  4780 F35A                      0000235A   632           BZ    PGMSTOP                  N, PIC 04 from something else
00002302  D503 015C F3C8       0000015C  000023C8   633           CLC   PGMOPSW+12(4),=A(MVCOS)  Was it the MVCOS that failed?
00002308  4770 F35A                      0000235A   634           BNE   PGMSTOP                  Nope, halt the machine
0000230C  FA30 0204 F3D6       00000204  000023D6   635           AP    PIC04,=P'1'              Increment # successful tests
00002312  47F0 F34C                      0000234C   636           B     PGMEXIT                  Exit FLIH
                                                    637 *
                               00002316  00000001   638 PGM13    EQU   *
00002316  9101 008B                      0000008B   639           TM    SVCINTC+3,X'01'          Were we in problem state?
0000231A  4780 F35A                      0000235A   640           BZ    PGMSTOP                  No,error! PIC 13 shouldnt happen
0000231E  D503 015C F3C8       0000015C  000023C8   641           CLC   PGMOPSW+12(4),=A(MVCOS)  Was it the MVCOS that failed?
00002324  4770 F35A                      0000235A   642           BNE   PGMSTOP                  Nope, halt the machine
00002328  5000 F570                      00002570   643           ST    R0,WORK                  Save current control bits
0000232C  D403 F570 F3CC       00002570  000023CC   644           NC    WORK,=A(OAC1A+OAC2A)     Either Access validity bit = 1?
00002332  4780 F35A                      0000235A   645           BZ    PGMSTOP                  N, PIC 13 from something else
00002336  A700 00C0                                 646           TMLH  R0,ASCHOM                Operand 1 ASC is HOME?
0000233A  4710 F346                      00002346   647           BO    PGM13CT                  Yes, PIC13 is ok in prob state
0000233E  A701 00C0                                 648           TMLL  R0,ASCHOM                Operand 2 ASC is HOME?
00002342  4780 F35A                      0000235A   649           BZ    PGMSTOP                  No. Something wrong
                                                    650 *
                               00002346  00000001   651 PGM13CT  EQU   *
00002346  FA30 0208 F3D6       00000208  000023D6   652           AP    PIC13,=P'1'              Increment # successful tests
                                                    653 *
                               0000234C  00000001   654 PGMEXIT  EQU   *                          Exit from FLIH everything OK
0000234C  9101 008B                      0000008B   655           TM    SVCINTC+3,X'01'          Were we in problem state?
00002350  4780 F26A                      0000226A   656           BZ    NEXTTEST                 No, proceed to next test
00002354  0A01                                      657           SVC   1                        Return to problem state
00002356  47F0 F26A                      0000226A   658           B     NEXTTEST                 And return to next test


                                                    660 ************************************************************************
                                                    661 *                      UNEXPECTED PROGRAM CHECK
                                                    662 ************************************************************************
                               0000235A  00000001   663 PGMSTOP  EQU   *                          Halt if something wrong
0000235A  B2B2 F4A0                      000024A0   664           LPSWE HALT                     Here for unexpected prog checks
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    666 ********************************************************************************
                                                    667 *                          WORKING STORAGE
                                                    668 ********************************************************************************
                                                    669 *
00002360                                            670          LTORG
00002360  10031003                                  671                =X'10031003'
00002364  0000000C                                  672                =X'0000000C'
00002368  00010000                                  673                =A(OAC1A)
0000236C  FFFEFFFF                                  674                =A(X'FFFFFFFF'-OAC1A)
00002370  FF3FFFFF                                  675                =A(X'FFFFFFFF'-(ASCHOM*65536))
00002374  00C00000                                  676                =A(ASCHOM*65536)
00002378  00400000                                  677                =X'00400000'
0000237C  00C10000                                  678                =A(OAC1A+ASCHOM*65536)
00002380  00410000                                  679                =A(OAC1A+ASCAR*65536)
00002384  00000003                                  680                =F'3'
00002388  00000001                                  681                =A(OAC2A)
0000238C  FFFFFFFE                                  682                =A(X'FFFFFFFF'-OAC2A)
00002390  FFFFFF3F                                  683                =A(X'FFFFFFFF'-ASCHOM)
00002394  000000C0                                  684                =A(ASCHOM)
00002398  000000C1                                  685                =A(OAC2A+ASCHOM)
0000239C  00000041                                  686                =A(OAC2A+ASCAR)
000023A0  10020000                                  687                =A(OAC1KEY+OAC1K)
000023A4  EFFDFFFF                                  688                =A(X'FFFFFFFF'-OAC1KEY-OAC1K)
000023A8  00001002                                  689                =A(OAC2KEY+OAC2K)
000023AC  FFFFEFFD                                  690                =A(X'FFFFFFFF'-OAC2KEY-OAC2K)
000023B0  00C00000                                  691                =X'00C00000'
000023B4  000000C0                                  692                =X'000000C0'
000023B8  00000000                                  693                =F'0'
000023BC  00000001                                  694                =F'1'
000023C0  00000002                                  695                =F'2'
000023C4  00020002                                  696                =A(OAC1K+OAC2K)
000023C8  000021FC                                  697                =A(MVCOS)
000023CC  00010001                                  698                =A(OAC1A+OAC2A)
000023D0  384C                                      699                =P'384'
000023D2  092C                                      700                =P'92'
000023D4  04                                        701                =X'04'
000023D5  0C                                        702                =X'0C'
000023D6  1C                                        703                =P'1'
000023D7  01152C                                    704                =P'1152'

000023E0                                            706          DC    0D'0'
000023E0  D7D9C960 D7C7F140                         707 PRIPG1   DC    CL16'PRI-PG1'            Eyecatcher
000023F0  D7D9C960 D7C7F240                         708 PRIPG2   DC    CL16'PRI-PG2'            Eyecatcher
00002400  E2C5C360 D7C7F140                         709 SECPG1   DC    CL16'SEC-PG1'            Eyecatcher
00002410  E2C5C360 D7C7F240                         710 SECPG2   DC    CL16'SEC-PG2'            Eyecatcher
00002420  C8D6D460 D7C7F140                         711 HOMPG1   DC    CL16'HOM-PG1'            Eyecatcher
00002430  C8D6D460 D7C7F240                         712 HOMPG2   DC    CL16'HOM-PG2'            Eyecatcher
                                                    713 *
00002440  C6D9D6D4 D7D9C9F1                         714 FROMPRI  DC    CL16'FROMPRI1FROMPRI2'   Eyecatcher
00002450  C6D9D6D4 E2C5C3F1                         715 FROMSEC  DC    CL16'FROMSEC1FROMSEC2'   Eyecatcher
00002460  C6D9D6D4 C8D6D4F1                         716 FROMHOM  DC    CL16'FROMHOM1FROMHOM2'   Eyecatcher
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00002470  04024000 80000000                          718 BADMVCOS DC    X'0402400080000000',XL4'00',X'0000BAD1'  MVCOS failed
00002480  04024000 80000000                          719 TESTBAD  DC    X'0402400080000000',XL4'00',X'0000BAD9'  Test Failure
00002490  04024000 80000000                          720 TESTGOOD DC    X'0402400080000000',XL4'00',X'00000000'  Test Success
000024A0  04024000 80000000                          721 HALT     DC    X'0402400080000000',XL4'00',X'0000DEAD'  Test Crashed!


                                                     723 *        Control registers
                                                     724 *
000024B0                                             725 CREGS    DC    0D'0'
000024B0  00000000 04000000                          726 CREG0    DC    X'00000000',X'04000000' Secondary space control bit 37
000024B8  00000000 00004000                          727 CREG1    DC    X'00000000',A(SEGPRI)   Primary ASCE
000024C0  00000000 00009100                          728 CREG2    DC    X'00000000',A(DUCT)     Dispatchable Unit Ctl Table
000024C8  00000000 C0000001                          729 CREG3    DC    X'00000000',X'C0000001' PKM=C000, Secondary ASN=1
000024D0  00000000 00000000                          730 CREG4    DC    X'00000000',X'00000000' Primary ASN=0
000024D8  00000000 00009000                          731 CREG5    DC    X'00000000',A(PASTEO)   Primary ASTE Origin
000024E0  00000000 00000000                          732 CREG6    DC    X'00000000',X'00000000'
000024E8  00000000 00005000                          733 CREG7    DC    X'00000000',A(SEGSEC)   Secondary ASCE
000024F0  00000000 00000000                          734 CREG8    DC    X'00000000',X'00000000'
000024F8  00000000 00000000                          735 CREG9    DC    X'00000000',X'00000000'
00002500  00000000 00000000                          736 CREG10   DC    X'00000000',X'00000000'
00002508  00000000 00000000                          737 CREG11   DC    X'00000000',X'00000000'
00002510  00000000 00000000                          738 CREG12   DC    X'00000000',X'00000000'
00002518  00000000 00006000                          739 CREG13   DC    X'00000000',A(SEGHOM)   Home ASCE
00002520  00000000 00000000                          740 CREG14   DC    X'00000000',X'00000000'
00002528  00000000 00000000                          741 CREG15   DC    X'00000000',X'00000000'


00002530  00000000 00000000                          743 AREGS    DC    16F'0'                  Init for Access Registers
00002570  00000000                                   744 WORK     DC    F'0'                    Work area
00002574  00000000                                   745 FALET    DC    F'0'                    FROM ALET
00002578  00000000                                   746 TALET    DC    F'0'                    TO ALET
                                                     747 *
0000257C  00010FF0                                   748 VADDRTO  DC    X'00010FF0'             Virtual addr within all 3 addr
                                                     749 *                                        space where the identifying
                                                     750 *                                        space literal is placed
                                                     751 *
00002580  00012FF8                                   752 VADDRFRM DC    X'00012FF8'             Virtual addr within all 3 addr
                                                     753 *                                        space where the 'from'
                                                     754 *                                        identifying literal is placed


00002584                                             756 CONTROL  DC    0F'0'                   R0 MVCOS Control bits
00002584  0000                                       757 OAC1     DC    H'0'                     1st OAC
                          10000000  00000001         758 OAC1KEY  EQU   X'10000000'                1st key
                          00020000  00000001         759 OAC1K    EQU   X'00020000'                1st key validity bit
                          00010000  00000001         760 OAC1A    EQU   X'00010000'                1st ASC validity bit
00002586  0000                                       761 OAC2     DC    H'0'                     2nd OAC
                          00001000  00000001         762 OAC2KEY  EQU   X'00001000'                2nd key
                          00000002  00000001         763 OAC2K    EQU   X'00000002'                2nd key validity bit
                          00000001  00000001         764 OAC2A    EQU   X'00000001'                2nd ASC validity bit
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                   766 *        Bits in OAC1,OAC2 control bytes
                                                   767 *
                             00000100  00000001    768 KEY      EQU   X'0100' .... ...1 .... ....  Key 1 set
                             00000000  00000001    769 ASCPRI   EQU   X'0000' .... .... 00.. ....  Primary space
                             00000040  00000001    770 ASCAR    EQU   X'0040' .... .... 01.. ....  AR
                             00000080  00000001    771 ASCSEC   EQU   X'0080' .... .... 10.. ....  Secondary space
                             000000C0  00000001    772 ASCHOM   EQU   X'00C0' .... .... 11.. ....  Home space
                             00000002  00000001    773 ASCK     EQU   X'0002' .... .... .... ..1.  Specified key valid
                             00000001  00000001    774 ASCA     EQU   X'0001' .... .... .... ...1  Specified AS valid


00002588  00000000                                 776 SACIDX   DC    X'00000000'        Pri   SAC bits to set PSW 16-17=00
0000258C  00000200                                 777          DC    X'00000200'        AR                            01
00002590  00000100                                 778          DC    X'00000100'        Sec                           10
00002594  00000300                                 779          DC    X'00000300'        Home                          11
                                                   780 *
00002598  00                                       781 PSWSTATE DC    X'00'                    Tracks suprv/prob state in PSW
00002599  00                                       782 PSWASC   DC    X'00'                    Tracks ASC Mode setting of PSW
                                                   783 *                                        in bits 4-5 of this byte
                                                   784 *                                        during the MVCOS execution


0000259A                     0000259A  00004000    786          ORG   STRTLABL+X'4000'
00004000  00000000 00007000                        787 SEGPRI   DC    X'00000000',A(PAGPRI)
                                                   788 *
00004008                     00004008  00005000    789          ORG   STRTLABL+X'5000'
00005000  00000000 00007800                        790 SEGSEC   DC    X'00000000',A(PAGSEC)
                                                   791 *
00005008                     00005008  00006000    792          ORG   STRTLABL+X'6000'
00006000  00000000 00008000                        793 SEGHOM   DC    X'00000000',A(PAGHOM)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00006008                     00006008  00007000   795          ORG   STRTLABL+X'7000'
                             00007000  00000001   796 PAGPRI   EQU   *                          Primary Space Page Tables
                                                   797 *
                                                   798 * Addresses 0-FFFF common to all addresses spaces, VIRT=REAL
00007000  00000000 00000000                        799          DC    X'00000000',X'00000000'  R=00000  V=00000
00007008  00000000 00001000                        800          DC    X'00000000',X'00001000'  R=01000  V=01000
00007010  00000000 00002000                        801          DC    X'00000000',X'00002000'  R=02000  V=02000
00007018  00000000 00003000                        802          DC    X'00000000',X'00003000'  R=03000  V=03000
00007020  00000000 00004000                        803          DC    X'00000000',X'00004000'  R=04000  V=04000
00007028  00000000 00005000                        804          DC    X'00000000',X'00005000'  R=05000  V=05000
00007030  00000000 00006000                        805          DC    X'00000000',X'00006000'  R=06000  V=06000
00007038  00000000 00007000                        806          DC    X'00000000',X'00007000'  R=07000  V=07000
00007040  00000000 00008000                        807          DC    X'00000000',X'00008000'  R=08000  V=08000
00007048  00000000 00009000                        808          DC    X'00000000',X'00009000'  R=09000  V=09000
00007050  00000000 0000A000                        809          DC    X'00000000',X'0000A000'  R=0A000  V=0A000
00007058  00000000 0000B000                        810          DC    X'00000000',X'0000B000'  R=0B000  V=0B000
00007060  00000000 0000C000                        811          DC    X'00000000',X'0000C000'  R=0C000  V=0C000
00007068  00000000 0000D000                        812          DC    X'00000000',X'0000D000'  R=0D000  V=0D000
00007070  00000000 0000E000                        813          DC    X'00000000',X'0000E000'  R=0E000  V=0E000
00007078  00000000 0000F000                        814          DC    X'00000000',X'0000F000'  R=0F000  V=0F000



                                                   816 * Begin primary space only storage V-addrs 10000-1FFFF
00007080  00000000 00020000                        817          DC    X'00000000',X'00020000'  R=20000  V=10000
00007088  00000000 00021000                        818          DC    X'00000000',X'00021000'  R=21000  V=11000
00007090  00000000 00022000                        819          DC    X'00000000',X'00022000'  R=22000  V=12000
00007098  00000000 00023000                        820          DC    X'00000000',X'00023000'  R=23000  V=13000
000070A0  00000000 00024000                        821          DC    X'00000000',X'00024000'  R=24000  V=14000
000070A8  00000000 00025000                        822          DC    X'00000000',X'00025000'  R=25000  V=15000
000070B0  00000000 00026000                        823          DC    X'00000000',X'00026000'  R=26000  V=16000
000070B8  00000000 00027000                        824          DC    X'00000000',X'00027000'  R=27000  V=17000
000070C0  00000000 00028000                        825          DC    X'00000000',X'00028000'  R=28000  V=18000
000070C8  00000000 00029000                        826          DC    X'00000000',X'00029000'  R=29000  V=19000
000070D0  00000000 0002A000                        827          DC    X'00000000',X'0002A000'  R=2A000  V=1A000
000070D8  00000000 0002B000                        828          DC    X'00000000',X'0002B000'  R=2B000  V=1B000
000070E0  00000000 0002C000                        829          DC    X'00000000',X'0002C000'  R=2C000  V=1C000
000070E8  00000000 0002D000                        830          DC    X'00000000',X'0002D000'  R=2D000  V=1D000
000070F0  00000000 0002E000                        831          DC    X'00000000',X'0002E000'  R=2E000  V=1E000
000070F8  00000000 0002F000                        832          DC    X'00000000',X'0002F000'  R=2F000  V=1F000
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00007100                     00007100  00007800   834          ORG   STRTLABL+X'7800'
                             00007800  00000001   835 PAGSEC   EQU   *                          Secondary Space Page Tables
                                                   836 *
                                                   837 * Addresses 0-FFFF common to all addresses spaces, VIRT=REAL
00007800  00000000 00000000                        838          DC    X'00000000',X'00000000'  R=00000  V=00000
00007808  00000000 00001000                        839          DC    X'00000000',X'00001000'  R=01000  V=01000
00007810  00000000 00002000                        840          DC    X'00000000',X'00002000'  R=02000  V=02000
00007818  00000000 00003000                        841          DC    X'00000000',X'00003000'  R=03000  V=03000
00007820  00000000 00004000                        842          DC    X'00000000',X'00004000'  R=04000  V=04000
00007828  00000000 00005000                        843          DC    X'00000000',X'00005000'  R=05000  V=05000
00007830  00000000 00006000                        844          DC    X'00000000',X'00006000'  R=06000  V=06000
00007838  00000000 00007000                        845          DC    X'00000000',X'00007000'  R=07000  V=07000
00007840  00000000 00008000                        846          DC    X'00000000',X'00008000'  R=08000  V=08000
00007848  00000000 00009000                        847          DC    X'00000000',X'00009000'  R=09000  V=09000
00007850  00000000 0000A000                        848          DC    X'00000000',X'0000A000'  R=0A000  V=0A000
00007858  00000000 0000B000                        849          DC    X'00000000',X'0000B000'  R=0B000  V=0B000
00007860  00000000 0000C000                        850          DC    X'00000000',X'0000C000'  R=0C000  V=0C000
00007868  00000000 0000D000                        851          DC    X'00000000',X'0000D000'  R=0D000  V=0D000
00007870  00000000 0000E000                        852          DC    X'00000000',X'0000E000'  R=0E000  V=0E000
00007878  00000000 0000F000                        853          DC    X'00000000',X'0000F000'  R=0F000  V=0F000



                                                   855 * Begin secondary space only storage V-addrs 10000-1FFFF
00007880  00000000 00030000                        856          DC    X'00000000',X'00030000'  R=30000  V=10000
00007888  00000000 00031000                        857          DC    X'00000000',X'00031000'  R=31000  V=11000
00007890  00000000 00032000                        858          DC    X'00000000',X'00032000'  R=32000  V=12000
00007898  00000000 00033000                        859          DC    X'00000000',X'00033000'  R=33000  V=13000
000078A0  00000000 00034000                        860          DC    X'00000000',X'00034000'  R=34000  V=14000
000078A8  00000000 00035000                        861          DC    X'00000000',X'00035000'  R=35000  V=15000
000078B0  00000000 00036000                        862          DC    X'00000000',X'00036000'  R=36000  V=16000
000078B8  00000000 00037000                        863          DC    X'00000000',X'00037000'  R=37000  V=17000
000078C0  00000000 00038000                        864          DC    X'00000000',X'00038000'  R=38000  V=18000
000078C8  00000000 00039000                        865          DC    X'00000000',X'00039000'  R=39000  V=19000
000078D0  00000000 0003A000                        866          DC    X'00000000',X'0003A000'  R=3A000  V=1A000
000078D8  00000000 0003B000                        867          DC    X'00000000',X'0003B000'  R=3B000  V=1B000
000078E0  00000000 0003C000                        868          DC    X'00000000',X'0003C000'  R=3C000  V=1C000
000078E8  00000000 0003D000                        869          DC    X'00000000',X'0003D000'  R=3D000  V=1D000
000078F0  00000000 0003E000                        870          DC    X'00000000',X'0003E000'  R=3E000  V=1E000
000078F8  00000000 0003F000                        871          DC    X'00000000',X'0003F000'  R=3F000  V=1F000
```

```
  LOC         OBJECT CODE        ADDR1     ADDR2     STMT

00007900                       00007900  00008000   873          ORG   STRTLABL+X'8000'
                               00008000  00000001   874 PAGHOM   EQU   *                          Home Space Page Tables
                                                    875 *
                                                    876 * Addresses 0-FFFF common to all addresses spaces, VIRT=REAL
00008000  00000000 00000000                         877          DC    X'00000000',X'00000000'  R=00000  V=00000
00008008  00000000 00001000                         878          DC    X'00000000',X'00001000'  R=01000  V=01000
00008010  00000000 00002000                         879          DC    X'00000000',X'00002000'  R=02000  V=02000
00008018  00000000 00003000                         880          DC    X'00000000',X'00003000'  R=03000  V=03000
00008020  00000000 00004000                         881          DC    X'00000000',X'00004000'  R=04000  V=04000
00008028  00000000 00005000                         882          DC    X'00000000',X'00005000'  R=05000  V=05000
00008030  00000000 00006000                         883          DC    X'00000000',X'00006000'  R=06000  V=06000
00008038  00000000 00007000                         884          DC    X'00000000',X'00007000'  R=07000  V=07000
00008040  00000000 00008000                         885          DC    X'00000000',X'00008000'  R=08000  V=08000
00008048  00000000 00009000                         886          DC    X'00000000',X'00009000'  R=09000  V=09000
00008050  00000000 0000A000                         887          DC    X'00000000',X'0000A000'  R=0A000  V=0A000
00008058  00000000 0000B000                         888          DC    X'00000000',X'0000B000'  R=0B000  V=0B000
00008060  00000000 0000C000                         889          DC    X'00000000',X'0000C000'  R=0C000  V=0C000
00008068  00000000 0000D000                         890          DC    X'00000000',X'0000D000'  R=0D000  V=0D000
00008070  00000000 0000E000                         891          DC    X'00000000',X'0000E000'  R=0E000  V=0E000
00008078  00000000 0000F000                         892          DC    X'00000000',X'0000F000'  R=0F000  V=0F000



                                                    894 * Begin home space only storage V-addrs 10000-1FFFF (V=R)
00008080  00000000 00010000                         895          DC    X'00000000',X'00010000'  R=10000  V=10000
00008088  00000000 00011000                         896          DC    X'00000000',X'00011000'  R=11000  V=11000
00008090  00000000 00012000                         897          DC    X'00000000',X'00012000'  R=12000  V=12000
00008098  00000000 00013000                         898          DC    X'00000000',X'00013000'  R=13000  V=13000
000080A0  00000000 00014000                         899          DC    X'00000000',X'00014000'  R=14000  V=14000
000080A8  00000000 00015000                         900          DC    X'00000000',X'00015000'  R=15000  V=15000
000080B0  00000000 00016000                         901          DC    X'00000000',X'00016000'  R=16000  V=16000
000080B8  00000000 00017000                         902          DC    X'00000000',X'00017000'  R=17000  V=17000
000080C0  00000000 00018000                         903          DC    X'00000000',X'00018000'  R=18000  V=18000
000080C8  00000000 00019000                         904          DC    X'00000000',X'00019000'  R=19000  V=19000
000080D0  00000000 0001A000                         905          DC    X'00000000',X'0001A000'  R=1A000  V=1A000
000080D8  00000000 0001B000                         906          DC    X'00000000',X'0001B000'  R=1B000  V=1B000
000080E0  00000000 0001C000                         907          DC    X'00000000',X'0001C000'  R=1C000  V=1C000
000080E8  00000000 0001D000                         908          DC    X'00000000',X'0001D000'  R=1D000  V=1D000
000080F0  00000000 0001E000                         909          DC    X'00000000',X'0001E000'  R=1E000  V=1E000
000080F8  00000000 0001F000                         910          DC    X'00000000',X'0001F000'  R=1F000  V=1F000
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

00008100                     00008100  00009000   912          ORG   STRTLABL+X'9000'
00009000                                          913 PASTEO   DS    0XL64                    Primary ASN Second Table Entry
00009000  00000000                                914          DC    A(0)           +0        ATO
00009004  00000000                                915          DC    A(0)            4        AX,ATL
00009008  00000000 00004000                       916          DC    A(0),A(SEGPRI)  8        Primary ASCE (same as CREG1)
00009010  00000000                                917          DC    A(0)           16        ALD
00009014  00000000                                918          DC    A(0)           20        ASTESN
00009018  00000000                                919          DC    A(0)           24        LTD
0000901C  00000000                                920          DC    A(0)           28        Ctl prog use
00009020  00000000                                921          DC    A(0)           32        Ctl prog use
00009024  00000000                                922          DC    A(0)           36        Ctl prog use
00009028  00000000                                923          DC    A(0)           40        unassigned
0000902C  00000000                                924          DC    A(0)           44        ASTEIN
00009030  00000000                                925          DC    A(0)           48        unassigned
00009034  00000000                                926          DC    A(0)           58        unassigned
00009038  00000000                                927          DC    A(0)           56        unassigned
0000903C  00000000                                928          DC    A(0)           60        unassigned




                                                  930 *
                                                  931 *       Dispatchable Unit Control Table (DUCT)
                                                  932 *
                                                  933 *         This DUCT is used by the primary space programming
                                                  934 *         when in Access Register mode in order to use the DU-AL.
                                                  935 *
00009040                     00009040  00009100   936          ORG   STRTLABL+X'9100'
00009100                                          937 DUCT     DS    0XL64                    Dispatchable Unit Control Tbl
00009100  00000000                                938          DC    A(0)      +0             BASTEO
00009104  00000000                                939          DC    A(0)       4             SSASTEO
00009108  00000000                                940          DC    A(0)       8             unassigned
0000910C  00000000                                941          DC    A(0)      12             SSASTESN
00009110  00009200                                942 DUALD    DC    A(DUAL)   16             DU-AL origin
00009114  00000000                                943          DC    A(0)      20             PSW key masks
00009118  00000000                                944          DC    A(0)      24             unassigned
0000911C  00000000                                945          DC    A(0)      28             unassigned
00009120  00000000                                946          DC    A(0)      32             Return addr high
00009124  00000000                                947          DC    A(0)      36             Return addr low
00009128  00000000                                948          DC    A(0)      40             unassigned
0000912C  00000000                                949          DC    A(0)      44             TRCB
00009130  00000000                                950          DC    A(0)      48             unassigned
00009134  00000000                                951          DC    A(0)      52             unassigned
00009138  00000000                                952          DC    A(0)      56             unassigned
0000913C  00000000                                953          DC    A(0)      60             unassigned
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                     955 *
                                                     956 *        Dispatchable Unit - Access List (DU-AL)
                                                     957 *
                                                     958 *         8 access list entries, only entry 2 is valid (AR ALET = 2)
                                                     959 *
00009140                      00009140  00009200     960          ORG   STRTLABL+X'9200'
                              00009200  00000001     961 DUAL     EQU   *                          DU Access List
00009200  80000000 00000000                          962 ALE0     DC    X'80',15X'00'              ALE 0 invalid
00009210  80000000 00000000                          963 ALE1     DC    X'80',15X'00'              ALE 1 invalid
                                                     964 *
00009220                                             965 ALE2     DS    0XL16                      ALE 2 -> HOME space
00009220  00000000                                   966          DC    A(0)                       I,FO,P,ALESN,ALEAX all 0
00009224  00000000                                   967          DC    A(0)                       unassigned
00009228  0000A000                                   968          DC    A(HASTEO)                  Home space ASTE Origin
0000922C  00000000                                   969          DC    A(0)                       ASTESN seq # set to 0
                                                     970 *
00009230  80000000 00000000                          971 ALE3     DC    X'80',15X'00'              ALE 3 invalid
00009240  80000000 00000000                          972 ALE4     DC    X'80',15X'00'              ALE 4 invalid
00009250  80000000 00000000                          973 ALE5     DC    X'80',15X'00'              ALE 5 invalid
00009260  80000000 00000000                          974 ALE6     DC    X'80',15X'00'              ALE 6 invalid
00009270  80000000 00000000                          975 ALE7     DC    X'80',15X'00'              ALE 7 invalid


                                                     977 *        The HASTE is needed for ALET 2's ALE entry above
                                                     978 *
00009280                      00009280  0000A000     979          ORG   STRTLABL+X'A000'
0000A000                                             980 HASTEO   DS    0XL64                      Home ASN Second Table Entry
0000A000  00000000                                   981          DC    A(0)           +0          ATO
0000A004  00000000                                   982          DC    A(0)            4          AX,ATL
0000A008  00000000 00006000                          983          DC    A(0),A(SEGHOM)  8          Home ASCE (same as CREG13)
0000A010  00000000                                   984          DC    A(0)           16          ALD
0000A014  00000000                                   985          DC    A(0)           20          ASTESN
0000A018  00000000                                   986          DC    A(0)           24          LTD
0000A01C  00000000                                   987          DC    A(0)           28          Ctl prog use
0000A020  00000000                                   988          DC    A(0)           32          Ctl prog use
0000A024  00000000                                   989          DC    A(0)           36          Ctl prog use
0000A028  00000000                                   990          DC    A(0)           40          unassigned
0000A02C  00000000                                   991          DC    A(0)           44          ASTEIN
0000A030  00000000                                   992          DC    A(0)           48          unassigned
0000A034  00000000                                   993          DC    A(0)           58          unassigned
0000A038  00000000                                   994          DC    A(0)           56          unassigned
0000A03C  00000000                                   995          DC    A(0)           60          unassigned


                              00000100  00000001    997 SECMODE  EQU   256       Secondary mode
                              00000300  00000001    998 HOMEMODE EQU   768       Home space mode
                              00000200  00000001    999 ARMODE   EQU   512       Enter AR mode

                                        00002000  1001          END START
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
ALE0	X	00009200	1	962	
ALE1	X	00009210	1	963	
ALE2	X	00009220	16	965	
ALE3	X	00009230	1	971	
ALE4	X	00009240	1	972	
ALE5	X	00009250	1	973	
ALE6	X	00009260	1	974	
ALE7	X	00009270	1	975	
AR0	U	00000000	1	22	331
AR1	U	00000001	1	23	565 602 603 604 605
AR10	U	0000000A	1	32	
AR11	U	0000000B	1	33	
AR12	U	0000000C	1	34	
AR13	U	0000000D	1	35	
AR14	U	0000000E	1	36	
AR15	U	0000000F	1	37	331
AR2	U	00000002	1	24	
AR3	U	00000003	1	25	
AR4	U	00000004	1	26	
AR5	U	00000005	1	27	425 603
AR6	U	00000006	1	28	482 608
AR7	U	00000007	1	29	
AR8	U	00000008	1	30	
AR9	U	00000009	1	31	
AREGS	F	00002530	4	743	331
ARMODE	U	00000200	1	999	349 532
ASC010	U	00002094	1	373	582
ASCA	U	00000001	1	774	389 398 446 455 535 547
ASCAR	U	00000040	1	770	410 467
ASCHOM	U	000000C0	1	772	646 648 387 393 409 444 450 466
ASCK	U	00000002	1	773	
ASCPRI	U	00000000	1	769	
ASCSEC	U	00000080	1	771	
BADMVCOS	X	00002470	8	718	561
BEGIN000	U	000021EA	1	524	514
CHK010	U	00002218	1	541	536
CHK020	U	00002238	1	554	548
CHK100	U	00002252	1	563	560
CONTROL	F	00002584	4	756	
CR0	U	00000000	1	38	332
CR1	U	00000001	1	39	
CR10	U	0000000A	1	48	
CR11	U	0000000B	1	49	
CR12	U	0000000C	1	50	
CR13	U	0000000D	1	51	
CR14	U	0000000E	1	52	
CR15	U	0000000F	1	53	332
CR2	U	00000002	1	40	
CR3	U	00000003	1	41	
CR4	U	00000004	1	42	
CR5	U	00000005	1	43	
CR6	U	00000006	1	44	

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
CR7	U	00000007	1	45	
CR8	U	00000008	1	46	
CR9	U	00000009	1	47	
CREG0	X	000024B0	4	726	
CREG1	X	000024B8	4	727	
CREG10	X	00002500	4	736	
CREG11	X	00002508	4	737	
CREG12	X	00002510	4	738	
CREG13	X	00002518	4	739	
CREG14	X	00002520	4	740	
CREG15	X	00002528	4	741	
CREG2	X	000024C0	4	728	
CREG3	X	000024C8	4	729	
CREG4	X	000024D0	4	730	
CREG5	X	000024D8	4	731	
CREG6	X	000024E0	4	732	
CREG7	X	000024E8	4	733	
CREG8	X	000024F0	4	734	
CREG9	X	000024F8	4	735	
CREGS	D	000024B0	8	725	332
DUAL	U	00009200	1	961	942
DUALD	A	00009110	4	942	
DUCT	X	00009100	64	937	728
EXTNPSW	X	000001B0	16	296	
FAILTEST	I	000022B6	4	596	590 592 594
FALET	F	00002574	4	745	481 489
FKEY000	U	000021C4	1	504	574
FRMA000	U	0000212C	1	438	578
FRMAS000	U	00002154	1	454	577
FRMAS200	U	00002182	1	471	447 456
FRMAS210	U	0000218E	1	476	469
FRMAS220	U	00002192	1	480	576
FRMAS230	U	000021A8	1	488	485
FRMAS290	U	000021AC	1	491	468 474
FROMHOM	C	00002460	16	716	347
FROMPRI	C	00002440	16	714	339 557
FROMSEC	C	00002450	16	715	343
GETALET	I	000022CA	4	607	555
HALT	X	000024A0	8	721	664
HASTEO	X	0000A000	64	980	968
HOMEMODE	U	00000300	1	998	345
HOMPG1	C	00002420	16	711	346
HOMPG2	C	00002430	16	712	
IMAGE	1	00000000	41024	0	
KEY	U	00000100	1	768	
MVCOS	U	000021FC	1	530	633
MVCOS001	J	00000000	41024	280	
MVCOSOK	P	00000200	4	303	564 589
NEXTTEST	U	0000226A	1	573	516 656 658
OAC1	H	00002584	2	757	
OAC1A	U	00010000	1	760	383 385 409 410 644
OAC1K	U	00020000	1	759	497 499 631

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
OAC1KEY	U	10000000	1	758	497 499
OAC2	H	00002586	2	761	
OAC2A	U	00000001	1	764	440 442 466 467 644
OAC2K	U	00000002	1	763	506 508 631
OAC2KEY	U	00001000	1	762	506 508
PAGHOM	U	00008000	1	874	793
PAGPRI	U	00007000	1	796	787
PAGSEC	U	00007800	1	835	790
PASTEO	X	00009000	64	913	731
PGM04	U	000022F4	1	629	
PGM13	U	00002316	1	638	625
PGM13CT	U	00002346	1	651	647
PGMEXIT	U	0000234C	1	654	636
PGMFLIH	U	000022E4	1	623	298
PGMINTC	X	0000008C	4	288	624 626
PGMNPSW	X	000001D0	4	298	
PGMOPSW	X	00000150	16	292	633 641
PGMSTOP	U	0000235A	1	663	627 632 634 640 642 645 649
PIC04	P	00000204	4	304	591 635
PIC13	P	00000208	4	305	593 652
PRIPG1	C	000023E0	16	707	338 567
PRIPG2	C	000023F0	16	708	
PSWASC	X	00002599	1	782	353 374 377 416 473 512 534 546
PSWSTATE	X	00002598	1	781	356 363 364
R0	U	00000000	1	6	323 327 350 382 385 386 387 389 393 398 401 404 405 408 439 442 443 444 446 450 455 458 461 462 465 496 499 500 505 508 509 535 537 542 547 550 555 630 643 646 648
R1	U	00000001	1	7	325 327 328 364 365 374 375 376 377 382 383 384 386 401 402 403 405 408 409 410 424 425 426 427 429 432 439 440 441 443 458 459 460 462 465 466 467 481 482 483 484 486 489 496 497 498 500 505 506 507 509 525 526 528 529 533 534 537 538 539 542 543 559 568
R10	U	0000000A	1	16	407 415 422 579
R11	U	0000000B	1	17	388 396 580
R12	U	0000000C	1	18	380 581
R13	U	0000000D	1	19	372 582
R14	U	0000000E	1	20	361 583
R15	U	0000000F	1	21	312 313 314 315
R2	U	00000002	1	8	317 321 324 329 511 512 515 525 545 546 550 551 552 555 556 557 559 565 566 567 568 607 608 609 610
R3	U	00000003	1	9	318 503 574
R4	U	00000004	1	10	320 321 494 575
R5	U	00000005	1	11	335 338 342 346 529
R6	U	00000006	1	12	336 339 343 347 529
R7	U	00000007	1	13	464 472 479 576
R8	U	00000008	1	14	445 453 577
R9	U	00000009	1	15	437 578
RESTART	X	000001A0	4	295	
SACIDX	X	00002588	4	776	525
SECMODE	U	00000100	1	997	341
SECPG1	C	00002400	16	709	342

```
     SYMBOL          TYPE    VALUE      LENGTH    DEFN  REFERENCES

SECPG2                C     00002410         16    710
SEGHOM                X     00006000          4    793   739   983
SEGPRI                X     00004000          4    787   727   916
SEGSEC                X     00005000          4    790   733
SET000                I     0000201C          4    327   329
SETAR                 I     000022BA          4    602   542
SKIP001               U     00002090          1    369   366
START                 I     00002000          2    312   295   315  1001
STATE000              U     0000207E          1    362   583
STRTLABL              U     00000000          1    281   286   290   294   302   311   786   789   792   795   834   873   912   936
                                                        960   979   282
SVCFLIH               U     000022DA          1    615   297
SVCINTC               X     00000088          4    287   513   616   639   655
SVCNPSW               X     000001C0          4    297
SVCOPSW               X     00000140         16    291   616   617
SVCSTATE              I     0000208E          2    368   365
TALET                 F     00002578          4    746   424   432
TESTBAD               X     00002480          8    719   596
TESTGOOD              X     00002490          8    720   595
TKEY000               U     000021B0          1    495   575
TOA000                U     000020A8          1    381   581
TOAS000               U     000020D0          1    397   580
TOAS200               U     000020FE          1    414   390   399
TOAS210               U     0000210A          1    419   412
TOAS220               U     0000210E          1    423   579
TOAS230               U     00002124          1    431   428
TOAS290               U     00002128          1    434   411   417
VADDRFRM              X     00002580          4    752   336
VADDRTO               X     0000257C          4    748   335   543
WORK                  F     00002570          4    744   630   631   643   644
=A(ASCHOM)            A     00002394          4    684   450   460
=A(ASCHOM*65536)      A     00002374          4    676   393   403
=A(MVCOS)             A     000023C8          4    697   633   641
=A(OAC1A)             A     00002368          4    673   383   384
=A(OAC1A+ASCAR*65536)
                      A     00002380          4    679   410
=A(OAC1A+ASCHOM*65536)
                      A     0000237C          4    678   409
=A(OAC1A+OAC2A)       A     000023CC          4    698   644
=A(OAC1K+OAC2K)       A     000023C4          4    696   631
=A(OAC1KEY+OAC1K)
                      A     000023A0          4    687   497   498
=A(OAC2A)             A     00002388          4    681   440   441
=A(OAC2A+ASCAR)       A     0000239C          4    686   467
=A(OAC2A+ASCHOM)      A     00002398          4    685   466
=A(OAC2KEY+OAC2K)
                      A     000023A8          4    689   506   507
=A(X'FFFFFFFF'-(ASCHOM*65536))
                      A     00002370          4    675   387   404
=A(X'FFFFFFFF'-ASCHOM)
                      A     00002390          4    683   444   461
=A(X'FFFFFFFF'-OAC1A)
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
	A	0000236C	4	674	385
=A(X'FFFFFFFF'-OAC1KEY-OAC1K)					
	A	000023A4	4	688	499
=A(X'FFFFFFFF'-OAC2A)					
	A	0000238C	4	682	442
=A(X'FFFFFFFF'-OAC2KEY-OAC2K)					
	A	000023AC	4	690	508
=F'0'	F	000023B8	4	693	602
=F'1'	F	000023BC	4	694	604
=F'2'	F	000023C0	4	695	605
=F'3'	F	00002384	4	680	427 484
=P'1'	P	000023D6	1	703	564 635 652
=P'1152'	P	000023D7	3	704	591
=P'384'	P	000023D0	2	699	589
=P'92'	P	000023D2	2	700	593
=X'0000000C'	X	00002364	4	672	376
=X'000000C0'	X	000023B4	4	692	551
=X'00400000'	X	00002378	4	677	402
=X'00C00000'	X	000023B0	4	691	538
=X'04'	X	000023D4	1	701	333
=X'0C'	X	000023D5	1	702	515
=X'10031003'	X	00002360	4	671	350

MACRO DEFN REFERENCES

No defined macros

```
   DESC      SYMBOL     SIZE      POS         ADDR

Entry: 2000

Image       IMAGE      41024  0000-A03F  0000-A03F
  Region               41024  0000-A03F  0000-A03F
    CSECT   MVCOS001   41024  0000-A03F  0000-A03F
```

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

** NO ERRORS FOUND **
```