```
 LOC       OBJECT CODE      ADDR1    ADDR2    STMT

                                                2 ************************************************************************
                                                3 *
                                                4 *        CLCE  instruction tests
                                                5 *
                                                6 *        NOTE: This is a copy of the CLCL-et-al Test
                                                7 *              modified to only test the CLCLE instruction.
                                                8 *              Specifically, instuction
                                                9 *
                                               10 *                    CLCL  R10,R12
                                               11 *
                                               12 *                    was changed to
                                               13 *
                                               14 *                    CLCLE R10,R12,0
                                               15 *                    BC    B'0001',*-4          not finished?
                                               16 *
                                               17 *
                                               18 *        James Wekel August 2022
                                               19 ************************************************************************
                                               20 ************************************************************************
                                               21 *
                                               22 *  This program tests proper functioning of the CLCLE instructions.
                                               23 *
                                               24 *  PLEASE NOTE that the tests are very SIMPLE TESTS designed to catch
                                               25 *  obvious coding errors.  None of the tests are thorough.  They are
                                               26 *  NOT designed to test all aspects of any of the instructions.
                                               27 *
                                               28 ************************************************************************
                                               29 *
                                               30 *  Example Hercules Testcase:
                                               31 *
                                               32 *
                                               33 *        *Testcase CLCE-03-basic (Test CLCLE instructions)
                                               34 *
                                               35 *        archlvl     390
                                               36 *        mainsize    3
                                               37 *        numcpu      1
                                               38 *        sysclear
                                               39 *
                                               40 *        loadcore    "CLCLE-03-basic.core" 0x0
                                               41 *
                                               42 *        runtest     1
                                               43 *        *Done
                                               44 *
                                               45 ************************************************************************
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                       47          PRINT OFF
                                     3428          PRINT ON

                                     3430 ********************************************************************************
                                     3431 *        SATK prolog stuff...
                                     3432 ********************************************************************************

                                     3434          ARCHLVL  ZARCH=NO,MNOTE=NO
                                     3436+$AL      OPSYN AL
                                     3437+$ALR     OPSYN ALR
                                     3438+$B       OPSYN B
                                     3439+$BAS     OPSYN BAS
                                     3440+$BASR    OPSYN BASR
                                     3441+$BC      OPSYN BC
                                     3442+$BCTR    OPSYN BCTR
                                     3443+$BE      OPSYN BE
                                     3444+$BH      OPSYN BH
                                     3445+$BL      OPSYN BL
                                     3446+$BM      OPSYN BM
                                     3447+$BNE     OPSYN BNE
                                     3448+$BNH     OPSYN BNH
                                     3449+$BNL     OPSYN BNL
                                     3450+$BNM     OPSYN BNM
                                     3451+$BNO     OPSYN BNO
                                     3452+$BNP     OPSYN BNP
                                     3453+$BNZ     OPSYN BNZ
                                     3454+$BO      OPSYN BO
                                     3455+$BP      OPSYN BP
                                     3456+$BXLE    OPSYN BXLE
                                     3457+$BZ      OPSYN BZ
                                     3458+$CH      OPSYN CH
                                     3459+$L       OPSYN L
                                     3460+$LH      OPSYN LH
                                     3461+$LM      OPSYN LM
                                     3462+$LPSW    OPSYN LPSW
                                     3463+$LR      OPSYN LR
                                     3464+$LTR     OPSYN LTR
                                     3465+$NR      OPSYN NR
                                     3466+$SL      OPSYN SL
                                     3467+$SLR     OPSYN SLR
                                     3468+$SR      OPSYN SR
                                     3469+$ST      OPSYN ST
                                     3470+$STM     OPSYN STM
                                     3471+$X       OPSYN X
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                             3473 ********************************************************************
                                             3474 *        Initiate the CLCLE03 CSECT in the CODE region
                                             3475 *        with the location counter at 0
                                             3476 ********************************************************************

                                             3478 CLCLE03  ASALOAD  REGION=CODE
                           000000  003000    3479+CLCLE03  START 0,CODE
000000  000A0000 00000008                    3481+         PSW   0,0,2,0,X'008'      64-bit Restart ISR Trap New PSW
000008                     000008  000058    3482+         ORG   CLCLE03+X'058'
000058  000A0000 00000018                    3484+         PSW   0,0,2,0,X'018'      64-bit External ISR Trap New PSW
000060  000A0000 00000020                    3485+         PSW   0,0,2,0,X'020'      64-bit Supervisor Call ISR Trap New PSW
000068  000A0000 00000028                    3486+         PSW   0,0,2,0,X'028'      64-bit Program ISR Trap New PSW
000070  000A0000 00000030                    3487+         PSW   0,0,2,0,X'030'      64-bit Machine Check Trap New PSW
000078  000A0000 00000038                    3488+         PSW   0,0,2,0,X'038'      64-bit Input/Output Trap New PSW
000080                     000080  000200    3489+         ORG   CLCLE03+512

                                             3491 ********************************************************************
                                             3492 *        Create IPL (restart) PSW
                                             3493 ********************************************************************

                                             3495          ASAIPL   IA=BEGIN
                           000000  003000    3496+CLCLE03  CSECT
000200                     000200  000000    3497+         ORG   CLCLE03
000000  00080000 00000200                    3498+         PSW   0,0,0,0,BEGIN,24
000008                     000008  000200    3499+         ORG   CLCLE03+512      Reset CSECT to end of assigned storage area
                           000000  003000    3500+CLCLE03  CSECT
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                      3502 ********************************************************************
                                      3503 *               The actual "CLCLE03" program itself...
                                      3504 ********************************************************************
                                      3505 *
                                      3506 *  Architecture Mode: 390
                                      3507 *  Addressing Mode:   31-bit
                                      3508 *  Register Usage:
                                      3509 *
                                      3510 *        R0       (work)
                                      3511 *        R1
                                      3512 *        R2       First base register
                                      3513 *        R3
                                      3514 *        R4
                                      3515 *        R5-R7    (work)
                                      3516 *        R8
                                      3517 *        R9       Second base register
                                      3518 *        R10-R13  (work)
                                      3519 *        R14      Subroutine call
                                      3520 *        R15      Secondary Subroutine call or work
                                      3521 *
                                      3522 ********************************************************************

000200                        000000  3524          USING  ASA,R0          Low core addressability
000200                        000200  3525          USING  BEGIN,R2        FIRST Base Register
000200                        001200  3526          USING  BEGIN+4096,R9   SECOND Base Register

000200  0520                         3528 BEGIN     BALR   R2,0            Initalize FIRST base register
000202  0620                         3529           BCTR   R2,0            Initalize FIRST base register
000204  0620                         3530           BCTR   R2,0            Initalize FIRST base register

000206  4190 2800             000800  3532          LA     R9,2048(,R2)    Initalize SECOND base register
00020A  4190 9800             000800  3533          LA     R9,2048(,R9)    Initalize SECOND base register

                                      3535 *
                                      3536 **       Run the tests...
                                      3537 *
00020E  45E0 202A             00022A  3538          BAL    R14,TEST01      Test CLCLE  instruction
                                      3539 *
000212  45E0 2134             000334  3540          BAL    R14,TEST91      Test CLCLE page fault handling
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3542 ********************************************************************
                                     3543 *         Test for normal or unexpected test completion...
                                     3544 ********************************************************************

000216  9591 9FFE                   0021FE  3546          CLI   TESTNUM,X'91'    Did we end on expected test?
00021A  4770 2298                   000498  3547          BNE   FAILTEST         No?! Then FAIL the test!

00021E  9510 9FFF                   0021FF  3549          CLI   SUBTEST,X'10'    Did we end on expected SUB-test?
000222  4770 2298                   000498  3550          BNE   FAILTEST         No?! Then FAIL the test!

000226  47F0 228A                   00048A  3552          B     EOJ              Yes, then normal completion!
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                      3554 ********************************************************************************
                                      3555 *        TEST01                    Test CLCLE instruction
                                      3556 ********************************************************************************

00022A  9201 9FFE                     0021FE  3558 TEST01   MVI   TESTNUM,X'01'
                                      3559 *
                                      3560 **       Initialize test parameters...
                                      3561 *
00022E  9856 2364                     000564  3562          LM     R5,R6,CLCL4     CLCL4 test Op1 address and length
000232  1E56                           000000  3563          ALR    R5,R6           Point past last byte
000234  0650                           000000  3564          BCTR   R5,0            Backup to last byte
000236  92FF 5000                     000000  3565          MVI    0(R5),X'FF'     Force unequal compare (op1 high)
                                      3566 *
00023A  9856 2384                     000584  3567          LM     R5,R6,CLCLOP1   (same thing for CLCLOP1 test)
00023E  1E56                           000000  3568          ALR    R5,R6                    "
000240  0650                           000000  3569          BCTR   R5,0                     "
000242  92FF 5000                     000000  3570          MVI    0(R5),X'FF'              "
                                      3571 *
000246  9856 237C                     00057C  3572          LM     R5,R6,CLCL8+8   CLCL8 test ===> OP2 <===
00024A  1E56                           000000  3573          ALR    R5,R6
00024C  0650                           000000  3574          BCTR   R5,0
00024E  92FF 5000                     000000  3575          MVI    0(R5),X'FF'     ===> OPERAND-2 high (OP1 LOW) <===
                                      3576 *
                                      3577 **       Neither cross (one byte)
                                      3578 *
000252  9201 9FFF                     0021FF  3579          MVI   SUBTEST,X'01'
000256  98AD 2304                     000504  3580          LM    R10,R13,CLCL1
00025A  A9AC 0000                     000000  3581          CLCLE R10,R12,0
00025E  4710 205A                     00025A  3582          BC    B'0001',*-4          not finished?
000262  4770 2298                     000498  3583          BNE   FAILTEST
000266  4150 23A4                     0005A4  3584          LA    R5,ECLCL1
00026A  45F0 227A                     00047A  3585          BAL   R15,ENDCLCL
                                      3586 *
                                      3587 **       Neither cross (two bytes)
                                      3588 *
00026E  9202 9FFF                     0021FF  3589          MVI   SUBTEST,X'02'
000272  98AD 2314                     000514  3590          LM    R10,R13,CLCL2
000276  A9AC 0000                     000000  3591          CLCLE R10,R12,0
00027A  4710 2076                     000276  3592          BC    B'0001',*-4          not finished?
00027E  4770 2298                     000498  3593          BNE   FAILTEST
000282  4150 23B4                     0005B4  3594          LA    R5,ECLCL2
000286  45F0 227A                     00047A  3595          BAL   R15,ENDCLCL
                                      3596 *
                                      3597 **       Neither cross (four bytes)
                                      3598 **       (inequality on last byte of op1)
                                      3599 *
00028A  9204 9FFF                     0021FF  3600          MVI   SUBTEST,X'04'
00028E  98AD 2364                     000564  3601          LM    R10,R13,CLCL4
000292  A9AC 0000                     000000  3602          CLCLE R10,R12,0
000296  4710 2092                     000292  3603          BC    B'0001',*-4          not finished?
00029A  47D0 2298                     000498  3604          BNH   FAILTEST                 (see INIT; CLCL4:   op1 > op2)
00029E  4150 2404                     000604  3605          LA    R5,ECLCL4
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT
0002A2  45F0 227A                   00047A  3606          BAL   R15,ENDCLCL
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT
                                             3608 *
                                             3609 **       Neither cross (eight bytes)
                                             3610 **       (inequality on last byte of op2)
                                             3611 *
0002A6  9208 9FFF                    0021FF  3612          MVI   SUBTEST,X'08'
0002AA  98AD 2374                    000574  3613          LM    R10,R13,CLCL8
0002AE  A9AC 0000                    000000  3614          CLCLE R10,R12,0
0002B2  4710 20AE                    0002AE  3615          BC    B'0001',*-4          not finished?
0002B6  47B0 2298                    000498  3616          BNL   FAILTEST                 (see INIT; CLCL8:   op1 < op2)
0002BA  4150 2414                    000614  3617          LA    R5,ECLCL8
0002BE  45F0 227A                    00047A  3618          BAL   R15,ENDCLCL
                                             3619 *
                                             3620 **       Neither cross (1K bytes)
                                             3621 *
0002C2  9200 9FFF                    0021FF  3622          MVI   SUBTEST,X'00'
0002C6  98AD 2334                    000534  3623          LM    R10,R13,CLCL1K
0002CA  A9AC 0000                    000000  3624          CLCLE R10,R12,0
0002CE  4710 20CA                    0002CA  3625          BC    B'0001',*-4          not finished?
0002D2  4770 2298                    000498  3626          BNE   FAILTEST
0002D6  4150 23D4                    0005D4  3627          LA    R5,ECLCL1K
0002DA  45F0 227A                    00047A  3628          BAL   R15,ENDCLCL
                                             3629 *
                                             3630 **       Both cross
                                             3631 *
0002DE  9222 9FFF                    0021FF  3632          MVI   SUBTEST,X'22'
0002E2  98AD 2344                    000544  3633          LM    R10,R13,CLCLBOTH
0002E6  A9AC 0000                    000000  3634          CLCLE R10,R12,0
0002EA  4710 20E6                    0002E6  3635          BC    B'0001',*-4          not finished?
0002EE  4770 2298                    000498  3636          BNE   FAILTEST
0002F2  4150 23E4                    0005E4  3637          LA    R5,ECLCLBTH
0002F6  45F0 227A                    00047A  3638          BAL   R15,ENDCLCL
                                             3639 *
                                             3640 **       Only op1 crosses
                                             3641 **       (inequality on last byte of op1)
                                             3642 *
0002FA  9210 9FFF                    0021FF  3643          MVI   SUBTEST,X'10'
0002FE  98AD 2384                    000584  3644          LM    R10,R13,CLCLOP1
000302  A9AC 0000                    000000  3645          CLCLE R10,R12,0
000306  4710 2102                    000302  3646          BC    B'0001',*-4          not finished?
00030A  47D0 2298                    000498  3647          BNH   FAILTEST                 (see INIT; CLCLOP1: op1 > op2)
00030E  4150 2424                    000624  3648          LA    R5,ECLCLOP1
000312  45F0 227A                    00047A  3649          BAL   R15,ENDCLCL
                                             3650 *
                                             3651 **       Only op2 crosses
                                             3652 *
000316  9220 9FFF                    0021FF  3653          MVI   SUBTEST,X'20'
00031A  98AD 2354                    000554  3654          LM    R10,R13,CLCLOP2
00031E  A9AC 0000                    000000  3655          CLCLE R10,R12,0
000322  4710 211E                    00031E  3656          BC    B'0001',*-4          not finished?
000326  4770 2298                    000498  3657          BNE   FAILTEST
00032A  4150 23F4                    0005F4  3658          LA    R5,ECLCLOP2
00032E  45F0 227A                    00047A  3659          BAL   R15,ENDCLCL
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                             3660 *
000332  07FE                                 3661          BR    R14
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT
                                      3663 ********************************************************************************
                                      3664 *        TEST91                    Test CLCLE page fault handling
                                      3665 ********************************************************************************

000334  9291 9FFE                    0021FE  3667 TEST91   MVI   TESTNUM,X'91'
000338  9200 9FFF                    0021FF  3668          MVI   SUBTEST,X'00'
                                      3669 *
                                      3670 **       First, make sure we start clean!
                                      3671 *
00033C  98AD 2394                    000594  3672          LM    R10,R13,CLCLPF     Retrieve CLCLE PF test parameters
000340  0EAC                          000594  3673          MVCL  R10,R12            (forces full comparison)
                                      3674 *
                                      3675 **       Initialize Dynamic Address Translation tables...
                                      3676 *
000342  58A0 22A8                    0004A8  3677          L     R10,=A(SEGTABLS)   Segment Tables Origin
000346  41B0 0020                    000020  3678          LA    R11,NUMPGTBS       Number of Segment Table Entries
00034A  58C0 22AC                    0004AC  3679          L     R12,=A(PAGETABS)   Page Tables Origin
00034E  1F00                          000000  3680          SLR   R0,R0              First Page Frame Address
000350  4160 0004                    000004  3681          LA    R6,4               Size of one table entry
000354  5870 22B0                    0004B0  3682          L     R7,=A(PAGE)        Size of one Page Frame

000358  50C0 A000                    000000  3684 SEGLOOP  ST    R12,0(,R10)        Seg Table Entry <= Page Table Origin
00035C  960F A003                    000003  3685          OI    3(R10),X'0F'       Seg Table Entry <= Page Table Length
000360  1EA6                          000000  3686          ALR   R10,R6             Bump to next Segment Table Entry

000362  41D0 0010                    000010  3688          LA    R13,16             Page Table Entries per Page Table
000366  5000 C000                    000000  3689 PAGELOOP ST    R0,0(,R12)         Page Table Entry = Page Frame Address
00036A  1E07                          000000  3690          ALR   R0,R7              Increment to next Page Frame Address
00036C  1EC6                          000000  3691          ALR   R12,R6             Bump to next Page Table Entry
00036E  46D0 2166                    000366  3692          BCT   R13,PAGELOOP       Loop until Page table is complete

000372  46B0 2158                    000358  3694          BCT   R11,SEGLOOP        Loop until all
                                      3695 *                                           Segment Table Entries built
                                      3696 *
                                      3697 **       Update desired page table entry to cause page fault
                                      3698 *
000376  98AD 2394                    000594  3699          LM    R10,R13,CLCLPF     Retrieve CLCLE PF test parameters
00037A  185A                          000594  3700          LR    R5,R10             R5 --> Operand-1
00037C  5E50 22B4                    0004B4  3701          AL    R5,=A(PFPGBYTS)    R5 --> Operand-1 Page Fault address
000380  1865                          000000  3702          LR    R6,R5              R6 --> Address where PF should occur
000382  8850 000C                    00000C  3703          SRL   R5,12              R5 = Page Frame number
000386  8950 0002                    000002  3704          SLL   R5,2               R5 = Page Table Entry number

00038A  9204 9FFF                    0021FF  3706          MVI   SUBTEST,X'04'
00038E  5E50 22AC                    0004AC  3707          AL    R5,=A(PAGETABS)    R5 --> Page Table Entry
000392  9604 5002                    000002  3708          OI    2(R5),X'04'        Mark this page invalid
```

```
 LOC       OBJECT CODE       ADDR1   ADDR2   STMT

                                             3710 *
                                             3711 **       Install program check routine to catch the page fault
                                             3712 *
000396  9202 9FFF                   0021FF   3713          MVI   SUBTEST,X'02'
00039A  D207 21D8 0068       0003D8  000068   3714          MVC   SVPGMNEW,PGMNPSW   Save original Program New PSW
0003A0  4100 21E8                    0003E8   3715          LA    R0,MYPGMNEW        Point to temporary Pgm New routine
0003A4  5000 006C                    00006C   3716          ST    R0,PGMNPSW+4       Point Program New PSW to our routine
0003A8  9208 0069                    000069   3717          MVI   PGMNPSW+1,X'08'    Make it a non-disabled-wait PSW!
                                             3718 *
                                             3719 **       Run the test: should cause a page fault
                                             3720 *
0003AC  920F 9FFF                   0021FF   3721          MVI   SUBTEST,X'0F'
0003B0  B700 22BC                    0004BC   3722          LCTL  R0,R0,CRLREG0      Switch to DAT mode
0003B4  B711 22C0                    0004C0   3723          LCTL  R1,R1,CTLREG1      Switch to DAT mode
0003B8  8200 21E0                    0003E0   3724          LPSW  DATONPSW           Switch to DAT mode
0003BC  4700 21BC                    0003BC   3725 BEGDATON NOP   *                  (pad)
0003C0  4700 21C0                    0003C0   3726          NOP   *                  (pad)
0003C4  B20D 0000                    000000   3727          PTLB  ,                  Purge Translation Lookaside Buffer
0003C8  A9AC 0000                    000000   3728 PFINSADR CLCLE R10,R12,0          Page Fault should occur on this instr
0003CC  4710 21C8                    0003C8   3729          BC    B'0001',*-4        not finished?
0003D0                                       3730          CNOP  0,8                      (align to doubleword)
0003D0  00000000 00000000                    3731 LOGICERR DC    D'0'                 We should never reach here!
0003D8  00000000 00000000                    3732 SVPGMNEW DC    D'0'                 Original Program New PSW
0003E0  04080000 000003BC                    3733 DATONPSW DC    XL4'04080000',A(BEGDATON)  Enable DAT PSW
                                             3734 *
                                             3735 **       Temporary Program New routine:
                                             3736 **       Restore original Program New PSW
                                             3737 *
0003E8  D207 0068 21D8       000068  0003D8  3738 MYPGMNEW MVC   PGMNPSW,SVPGMNEW   Restore original Program New PSW
                                             3739 *
                                             3740 **       Verify Program Check occurred on expected instruction
                                             3741 *
0003EE  9268 9FFF                   0021FF   3742          MVI   SUBTEST,X'68'
0003F2  D503 22B8 002C       0004B8  00002C   3743          CLC   =A(PFINSADR),PGMOPSW+4   Program Check where expected?
0003F8  4770 2298                    000498   3744          BNE   FAILTEST                 No?! Something is VERY WRONG!
                                             3745 *
                                             3746 **       Verify Program Check was indeed a page fault
                                             3747 *
0003FC  9211 9FFF                   0021FF   3748          MVI   SUBTEST,X'11'
000400  9511 008F                    00008F   3749          CLI   PGMICODE+1,X'11'   Verify it's a Page Fault interrupt
000404  4770 2298                    000498   3750          BNE   FAILTEST           If not then something is VERY WRONG!
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT
                                             3752 *
                                             3753 **       Verify Page Fault occurred on expected Page
                                             3754 *
000408  9205 9FFF                  0021FF  3755          MVI   SUBTEST,X'05'
00040C  5800 0090                  000090  3756          L     R0,PGMTRX          Get where Page Fault occurred
000410  8800 000C                  00000C  3757          SRL   R0,12
000414  8900 000C                  00000C  3758          SLL   R0,12

000418  8860 000C                  00000C  3760          SRL   R6,12              Where Page Fault is expected
00041C  8960 000C                  00000C  3761          SLL   R6,12

000420  1506                              3763          CLR   R0,R6              Page Fault occur on expected Page?
000422  4770 2298                  000498  3764          BNE   FAILTEST           No? Then something is very wrong!
                                             3765 *
                                             3766 **       Verify CLCLE instruction registers were updated as expected
                                             3767 *
000426  9206 9FFF                  0021FF  3768          MVI   SUBTEST,X'06'
00042A  55A0 2394                  000594  3769          CL    R10,CLCLPF         (op1 greater than starting value?)
00042E  47D0 2298                  000498  3770          BNH   FAILTEST
000432  55C0 239C                  00059C  3771          CL    R12,CLCLPF+4+4     (op2 greater than starting value?)
000436  47D0 2298                  000498  3772          BNH   FAILTEST

00043A  9207 9FFF                  0021FF  3774          MVI   SUBTEST,X'07'
00043E  15BD                              3775          CLR   R11,R13            (same remaining lengths?)
000440  4770 2298                  000498  3776          BNE   FAILTEST
000444  55B0 2398                  000598  3777          CL    R11,CLCLPF+4       (op1 len less than starting value?)
000448  47B0 2298                  000498  3778          BNL   FAILTEST
00044C  55D0 23A0                  0005A0  3779          CL    R13,CLCLPF+4+4+4   (op2 len less than starting value?)
000450  47B0 2298                  000498  3780          BNL   FAILTEST

000454  9208 9FFF                  0021FF  3782          MVI   SUBTEST,X'08'
000458  55A0 2434                  000634  3783          CL    R10,ECLCLPF        (stop before end?)
00045C  47B0 2298                  000498  3784          BNL   FAILTEST

000460  9209 9FFF                  0021FF  3786          MVI   SUBTEST,X'09'
000464  15A6                              3787          CLR   R10,R6             (stop at or before expected page?)
000466  4720 2298                  000498  3788          BH    FAILTEST

00046A  9210 9FFF                  0021FF  3790          MVI   SUBTEST,X'10'
00046E  187A                              3791          LR    R7,R10             (op1 stopped address)
000470  1E7B                              3792          ALR   R7,R11             (add remaining length)
000472  1576                              3793          CLR   R7,R6              (would remainder reach PF page?)
000474  47D0 2298                  000498  3794          BNH   FAILTEST

000478  07FE                              3796          BR    R14                Success!
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3798 ********************************************************************
                                     3799 *        Verify CLCLE ending register values
                                     3800 * R10-R12 = actual ending values, R5 --> expected ending values
                                     3801 ********************************************************************

00047A  90AD 2444                    000644  3803 ENDCLCL  STM    R10,R13,CLCLEND     Save actual ending register values
00047E  D50F 5000 2444       000000  000644  3804          CLC    0(4*4,R5),CLCLEND   Do they have the expected values?
000484  4770 2298                    000498  3805          BNE    FAILTEST            If not then the test has failed
000488  07FF                          3806          BR     R15                 Otherwise return to caller
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                             3809 ************************************************************************
                                             3810 *        Normal completion or Abnormal termination PSWs
                                             3811 ************************************************************************




                                             3813 EOJ      DWAITEND LOAD=YES          Normal completion
00048A                                       3815+EOJ      DS    0H
00048A  8200 2290                    000490  3816+          LPSW  DWAT0008
000490  000A0000 00000000                    3817+DWAT0008 PSW   0,0,2,0,X'000000'



                                             3819 FAILTEST DWAIT LOAD=YES,CODE=BAD    Abnormal termination
000498                                       3820+FAILTEST DS    0H
000498  8200 22A0                    0004A0  3821+          LPSW  DWAT0009
0004A0  000A0000 00010BAD                    3822+DWAT0009 PSW   0,0,2,0,X'010BAD'
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3824 ********************************************************************************
                                     3825 *        Working Storage
                                     3826 ********************************************************************************


0004A8                               3828          LTORG ,                  Literals pool
0004A8  00003000                     3829                =A(SEGTABLS)
0004AC  00003080                     3830                =A(PAGETABS)
0004B0  00001000                     3831                =A(PAGE)
0004B4  00005000                     3832                =A(PFPGBYTS)
0004B8  000003C8                     3833                =A(PFINSADR)

                         000400  000001  3835 K        EQU   1024             One KB
                         001000  000001  3836 PAGE     EQU   (4*K)            Size of one page
                         010000  000001  3837 K64      EQU   (64*K)           64 KB
                         100000  000001  3838 MB       EQU   (K*K)             1 MB

                         0021FE  000001  3840 TESTADDR EQU   (2*PAGE+X'200'-2)  Where test/subtest numbers will go

                         200000  000001  3842 MAINSIZE EQU   (2*MB)                    Minimum required storage size
                         000020  000001  3843 NUMPGTBS EQU   ((MAINSIZE+K64-1)/K64)   Number of Page Tables needed
                         000002  000001  3844 NUMSEGTB EQU   ((NUMPGTBS*4)/(16*4))    Number of Segment Tables
                         003000  000001  3845 SEGTABLS EQU   (3*PAGE)                 Segment Tables Origin
                         003080  000001  3846 PAGETABS EQU   (SEGTABLS+(NUMPGTBS*4))  Page Tables Origin
0004BC  00B00060                     3847 CRLREG0  DC    0A(0),XL4'00B00060'      Control Register 0
0004C0  00003002                     3848 CTLREG1  DC    A(SEGTABLS+NUMSEGTB)     Control Register 1
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3851 ************************************************************************
                                     3852 *        CLCLE Test Parameters:   A(operand-1),A(operand-2)
                                     3853 ************************************************************************

0004C4  00010000 00110000            3855 CLC1     DC    A(1*K64),A(MB+(1*K64))                        both equal
0004CC  00010000 00110000            3856 CLC2     DC    A(1*K64),A(MB+(1*K64))                        both equal
0004D4  0000FFF4 0010FFDE            3857 CLCBOTH  DC    A(1*K64-12),A(MB+(1*K64)-34)                  both equal
0004DC  00010000 0010FFDE            3858 CLCOP2   DC    A(1*K64),A(MB+(1*K64)-34)                     both equal

0004E4  00020000 00120000            3860 CLC4     DC    A(2*K64),A(MB+(2*K64))                          op1 HIGH
0004EC  00030000 00130000            3861 CLC8     DC    A(3*K64),A(MB+(3*K64))                          op1 LOW!
0004F4  00040000 00140000            3862 CLC256   DC    A(4*K64),A(MB+(4*K64))                          op1 HIGH
0004FC  0004FFF4 00150000            3863 CLCOP1   DC    A(5*K64-12),A(MB+(5*K64))                       op1 HIGH

                         000000  003000  3865 CLCLE03 CSECT ,
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3867 ************************************************************************
                                     3868 *        CLCLE Test Parameters
                                     3869 ************************************************************************
000504  00060000 00000001                  3871 CLCL1    DC    A(6*K64),A(1),A(MB+(6*K64)),A(1)              both equal

000514  00060000 00000002                  3873 CLCL2    DC    A(6*K64),A(2),A(MB+(6*K64)),A(2)              both equal

000524  00060000 00000100                  3875 CLCL256  DC    A(6*K64),A(256),A(MB+(6*K64)),A(256)          both equal

000534  00060000 00000400                  3877 CLCL1K   DC    A(6*K64),A(K),A(MB+(6*K64)),A(K)              both equal

000544  0005FFF4 00010000                  3879 CLCLBOTH DC    A(6*K64-12),A(K64),A(MB+(6*K64)-34),A(K64)    both equal

000554  00060000 00001000                  3881 CLCLOP2  DC    A(6*K64),A(PAGE),A(MB+(6*K64)-34),A(K64)      both equal

000564  00070000 00000004                  3883 CLCL4    DC    A(7*K64),A(4),A(MB+(7*K64)),A(4)                op1 HIGH

000574  00080000 00000008                  3885 CLCL8    DC    A(8*K64),A(8),A(MB+(8*K64)),A(8)                op1 LOW!

000584  0008FFF4 00010000                  3887 CLCLOP1  DC    A(9*K64-12),A(K64),A(MB+(9*K64)),A(PAGE)        op1 HIGH

000594  000A0000 00010000                  3889 CLCLPF   DC    A(10*K64),A(K64),A(MB+(10*K64)),A(K64)        page fault
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3891 ************************************************************************
                                     3892 *        CLCLE Expected Ending Register Values
                                     3893 ************************************************************************

0005A4  00060001 00000000            3895 ECLCL1   DC    A(6*K64+1),A(0),A(MB+(6*K64)+1),A(0)         both equal

0005B4  00060002 00000000            3897 ECLCL2   DC    A(6*K64+2),A(0),A(MB+(6*K64)+2),A(0)         both equal

0005C4  00060100 00000000            3899 ECLCL256 DC    A(6*K64+256),A(0),A(MB+(6*K64)+256),A(0)     both equal

0005D4  00060400 00000000            3901 ECLCL1K  DC    A(6*K64+K),A(0),A(MB+(6*K64)+K),A(0)         both equal

0005E4  0006FFF4 00000000            3903 ECLCLBTH DC    A(6*K64-12+K64),A(0),A(MB+(6*K64)-34+K64),A(0) bth equl

0005F4  00061000 00000000            3905 ECLCLOP2 DC    A(6*K64+PAGE),A(0),A(MB+(6*K64)-34+K64),A(0) both equal

000604  00070003 00000001            3907 ECLCL4   DC    A(7*K64+4-1),A(1),A(MB+(7*K64)+4-1),A(1)       op1 HIGH

000614  00080007 00000001            3909 ECLCL8   DC    A(8*K64+8-1),A(1),A(MB+(8*K64)+8-1),A(1)       op1 LOW!

000624  0009FFF3 00000001            3911 ECLCLOP1 DC    A(9*K64-12+K64-1),A(1),A(MB+(9*K64)+PAGE),A(0) op1 HIGH

000634  000B0000 00000000            3913 ECLCLPF  DC    A(10*K64+K64),A(0),A(MB+(10*K64)+K64),A(0)   page fault

000644  00000000 00000000            3915 CLCLEND  DC    4F'0'          (actual ending register values)
                           000005  000001  3916 PFPAGE   EQU   5              (page the Page Fault should occur on)
                           005000  000001  3917 PFPGBYTS EQU   (PFPAGE*PAGE)  (number of bytes into operand-1)
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3919 ********************************************************************
                                     3920 *        Fixed storage locations
                                     3921 ********************************************************************



000654                    000654  0021FE  3923          ORG   CLCLE03+TESTADDR      (s/b @ X'21FE', X'21FF')

0021FE  00                          3925 TESTNUM  DC    X'00'      Test number of active test
0021FF  00                          3926 SUBTEST  DC    X'00'      Active test sub-test number



002200                    002200  003000  3928          ORG   CLCLE03+SEGTABLS      (s/b @ X'3000')

003000  00                          3930 DATTABS  DC    X'00'      Segment and Page Tables will go here...
```

```
 LOC       OBJECT CODE      ADDR1   ADDR2   STMT

                                     3932 ********************************************************************
                                     3933 *        (other DSECTS needed by SATK)
                                     3934 ********************************************************************

                                     3936          DSECTS PRINT=OFF,NAME=(ASA,SCHIB)









                                     4149          PRINT ON
                                     4150 ********************************************************************
                                     4151 *        Register equates
                                     4152 ********************************************************************


                          000000  000001  4154 R0       EQU   0
                          000001  000001  4155 R1       EQU   1
                          000002  000001  4156 R2       EQU   2
                          000003  000001  4157 R3       EQU   3
                          000004  000001  4158 R4       EQU   4
                          000005  000001  4159 R5       EQU   5
                          000006  000001  4160 R6       EQU   6
                          000007  000001  4161 R7       EQU   7
                          000008  000001  4162 R8       EQU   8
                          000009  000001  4163 R9       EQU   9
                          00000A  000001  4164 R10      EQU   10
                          00000B  000001  4165 R11      EQU   11
                          00000C  000001  4166 R12      EQU   12
                          00000D  000001  4167 R13      EQU   13
                          00000E  000001  4168 R14      EQU   14
                          00000F  000001  4169 R15      EQU   15


                                     4171          END
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
ASA	4	000000	512	3940	3524
ASBEGIN	U	000000	1	3941	3946 3988 4024 4033 4051 4058 4064 4068 4072 4078 4095
ASEND	U	000200	1	4094	4095
ASLENGTH	U	000200	1	4095	
BCEXTCOD	H	00001A	2	3958	
BCIOCOD	H	00003A	2	3966	
BCMCKCOD	H	000032	2	3964	
BCPGMCOD	H	00002A	2	3962	
BCSVCCOD	H	000022	2	3960	
BEGDATON	I	0003BC	4	3725	3733
BEGIN	I	000200	2	3528	3498 3525 3526
CAW	F	000048	4	3970	
CAWADDR	R	000049	3	3973	
CAWKEY	X	000048	1	3971	
CAWSUSP	U	000008	1	3972	
CHANID	F	0000A8	4	4025	
CLC1	A	0004C4	4	3855	
CLC2	A	0004CC	4	3856	
CLC256	A	0004F4	4	3862	
CLC4	A	0004E4	4	3860	
CLC8	A	0004EC	4	3861	
CLCBOTH	A	0004D4	4	3857	
CLCL1	A	000504	4	3871	3580
CLCL1K	A	000534	4	3877	3623
CLCL2	A	000514	4	3873	3590
CLCL256	A	000524	4	3875	
CLCL4	A	000564	4	3883	3562 3601
CLCL8	A	000574	4	3885	3572 3613
CLCLBOTH	A	000544	4	3879	3633
CLCLE03	J	000000	12289	3479	3482 3489 3497 3499 3923 3928
CLCLEND	F	000644	4	3915	3803 3804
CLCLOP1	A	000584	4	3887	3567 3644
CLCLOP2	A	000554	4	3881	3654
CLCLPF	A	000594	4	3889	3672 3699 3769 3771 3777 3779
CLCOP1	A	0004FC	4	3863	
CLCOP2	A	0004DC	4	3858	
CODE	2	000000	12289	3479	
CPUID	U	00031B	1	4097	
CRLREG0	A	0004BC	4	3847	3722
CSW	F	000040	8	3969	
CTLREG1	A	0004C0	4	3848	3723
DATONPSW	X	0003E0	4	3733	3724
DATTABS	X	003000	1	3930	
DWAT0008	3	000490	8	3817	3816
DWAT0009	3	0004A0	8	3822	3821
ECLCL1	A	0005A4	4	3895	3584
ECLCL1K	A	0005D4	4	3901	3627
ECLCL2	A	0005B4	4	3897	3594
ECLCL256	A	0005C4	4	3899	
ECLCL4	A	000604	4	3907	3605
ECLCL8	A	000614	4	3909	3617
ECLCLBTH	A	0005E4	4	3903	3637

```
     SYMBOL         TYPE  VALUE    LENGTH  DEFN  REFERENCES

ECLCLOP1             A    000624        4  3911  3648
ECLCLOP2             A    0005F4        4  3905  3658
ECLCLPF              A    000634        4  3913  3783
ENDCLCL              I    00047A        4  3803  3585  3595  3606  3618  3628  3638  3649  3659
EOJ                  H    00048A        2  3815  3552
EXTCPUAD             H    000084        2  3990
EXTICODE             H    000086        2  3991
EXTIPARM             F    000080        4  3989
EXTNPSW              F    000058        8  3979
EXTOPSW              F    000018        8  3951  3957
FAILTEST             H    000498        2  3820  3547  3550  3583  3593  3604  3616  3626  3636  3647  3657  3744  3750  3764  3770
                                               3772  3776  3778  3780  3784  3788  3794  3805
IMAGE                1    000000    12289     0
IOELADDR             F    0000AC        4  4026
IOICODE              H    0000BA        2  4031
IOIID                F    0000C0        4  4036
IOIPARM              F    0000BC        4  4035
IONPSW               F    000078        8  3983
IOOPSW               F    000038        8  3955  3965
IOSSID               F    0000B8        4  4034
IPLCCW1              F    000008        8  3943
IPLCCW2              F    000010        8  3944
IPLPSW               F    000000        8  3942
K                    U    000400        1  3835  3836  3837  3838  3877  3901
K64                  U    010000        1  3837  3843  3855  3856  3857  3858  3860  3861  3862  3863  3871  3873  3875  3877  3879
                                               3881  3883  3885  3887  3889  3895  3897  3899  3901  3903  3905  3907  3909  3911
                                               3913
LCHANLOG             F    0000B0        4  4027
LOGICERR             D    0003D0        8  3731
MAINSIZE             U    200000        1  3842  3843
MB                   U    100000        1  3838  3842  3855  3856  3857  3858  3860  3861  3862  3863  3871  3873  3875  3877  3879
                                               3881  3883  3885  3887  3889  3895  3897  3899  3901  3903  3905  3907  3909  3911
                                               3913
MCKLOG               F    000100        4  4059
MCKNPSW              F    000070        8  3982
MCKOPSW              F    000030        8  3954  3963
MEASUREB             X    0000B9        1  4030
MKARCHMD             X    0000A3        1  4018
MKARS                F    000120        4  4057
MKCLKCMP             F    0000E0        8  4043
MKCPUTIM             F    0000D8        8  4042
MKCRS                F    0001C0        4  4062
MKDMGCOD             F    0000F4        4  4046
MKFAILA              F    0000F8        4  4048
MKFPRS               D    000160        8  4060
MKICODE              F    0000E8        4  4044
MKLOGOUT             F    000100        4  4050
MKMODEL              F    0000FC        4  4049
MKXSAA               F    0000D4        4  4041
MONCLS               H    000094        2  4006
MONCODE              F    00009C        4  4013
MONNUMBR             X    000095        1  4008
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
MPGACCID	X	0000A2	1	4016	
MYPGMNEW	I	0003E8	6	3738	3715
NKGRS	F	000180	4	4061	
NUMPGTBS	U	000020	1	3843	3844 3846 3678
NUMSEGTB	U	000002	1	3844	3848
PAGE	U	001000	1	3836	3840 3845 3917 3682 3881 3887 3905 3911
PAGELOOP	I	000366	4	3689	3692
PAGETABS	U	003080	1	3846	3679
PCFETO	A	0000C4	4	4037	
PERACCID	X	0000A1	1	4015	
PERADDR	F	000098	4	4012	
PERCODE	X	000096	1	4009	
PERCODMK	U	0000F0	1	4010	
PFINSADR	I	0003C8	4	3728	3743
PFPAGE	U	000005	1	3916	3917
PFPGBYTS	U	005000	1	3917	3701
PGMACCID	X	0000A0	1	4014	
PGMDXC	F	000090	4	4004	
PGMICODE	H	00008E	2	4003	3749
PGMIID	F	00008C	4	3999	
PGMIILC	X	00008D	1	4001	
PGMIILCM	U	00000C	1	4002	
PGMNPSW	F	000068	8	3981	3714 3716 3717 3738
PGMOPSW	F	000028	8	3953	3961 3743
PGMTRX	F	000090	4	4005	3756
PMCW1_0	X	000004	1	4103	
PMCW1_8	X	000005	1	4106	
PMCWB	U	000004	1	4138	
PMCWCHP0	X	000010	1	4127	
PMCWCHP1	X	000011	1	4128	
PMCWCHP2	X	000012	1	4129	
PMCWCHP3	X	000013	1	4130	
PMCWCHP4	X	000014	1	4131	
PMCWCHP5	X	000015	1	4132	
PMCWCHP6	X	000016	1	4133	
PMCWCHP7	X	000017	1	4134	
PMCWDNUM	H	000006	2	4118	
PMCWE	U	000080	1	4107	
PMCWEXC	X	00001B	1	4137	
PMCWIP	F	000000	4	4102	
PMCWISCM	U	000038	1	4104	
PMCWLM	U	000060	1	4108	
PMCWLMG	U	000020	1	4109	
PMCWLML	U	000040	1	4110	
PMCWLPM	X	000008	1	4120	
PMCWLPUM	X	00000A	1	4122	
PMCWM	U	000004	1	4114	
PMCWMBI	H	00000C	2	4124	
PMCWMM	U	000018	1	4111	
PMCWMMC	U	000008	1	4113	
PMCWMME	U	000010	1	4112	
PMCWPAM	X	00000F	1	4126	

```
     SYMBOL         TYPE  VALUE    LENGTH  DEFN  REFERENCES

PMCWPIM              X    00000B        1  4123
PMCWPNOM             X    000009        1  4121
PMCWPOM              X    00000E        1  4125
PMCWRES1             X    000018        4  4135
PMCWRES2             X    000018        3  4136
PMCWS                U    000001        1  4140
PMCWT                U    000002        1  4115
PMCWV                U    000001        1  4116
PMCWX                U    000002        1  4139
R0                   U    000000        1  4154  3524  3680  3689  3690  3715  3716  3722  3756  3757  3758  3763
R1                   U    000001        1  4155  3723
R10                  U    00000A        1  4164  3580  3581  3590  3591  3601  3602  3613  3614  3623  3624  3633  3634  3644  3645
                                                  3654  3655  3672  3673  3677  3684  3685  3686  3699  3700  3728  3769  3783  3787
                                                  3791  3803
R11                  U    00000B        1  4165  3678  3694  3775  3777  3792
R12                  U    00000C        1  4166  3581  3591  3602  3614  3624  3634  3645  3655  3673  3679  3684  3689  3691  3728
                                                  3771
R13                  U    00000D        1  4167  3580  3590  3601  3613  3623  3633  3644  3654  3672  3688  3692  3699  3775  3779
                                                  3803
R14                  U    00000E        1  4168  3538  3540  3661  3796
R15                  U    00000F        1  4169  3585  3595  3606  3618  3628  3638  3649  3659  3806
R2                   U    000002        1  4156  3525  3528  3529  3530  3532
R3                   U    000003        1  4157
R4                   U    000004        1  4158
R5                   U    000005        1  4159  3562  3563  3564  3565  3567  3568  3569  3570  3572  3573  3574  3575  3584  3594
                                                  3605  3617  3627  3637  3648  3658  3700  3701  3702  3703  3704  3707  3708  3804
R6                   U    000006        1  4160  3562  3563  3567  3568  3572  3573  3681  3686  3691  3702  3760  3761  3763  3787
                                                  3793
R7                   U    000007        1  4161  3682  3690  3791  3792  3793
R8                   U    000008        1  4162
R9                   U    000009        1  4163  3526  3532  3533
RSTNPSW              F    000000        8  3947
RSTOPSW              F    000008        8  3948
SCANOUT              X    000080        1  3985  3986
SCANOUTL             U    000000        1  3986
SCHIB                4    000000       52  4099  4146
SCHIBL               U    000034        1  4146
SCHMBA               A    000028        8  4144
SCHMDA1              X    000030        4  4145
SCHMDA3              X    000028       12  4143
SCHPMCW              X    000000       28  4101
SCHSCSW              X    00001C       12  4142
SEGLOOP              I    000358        4  3684  3694
SEGTABLS             U    003000        1  3845  3846  3928  3677  3848
SSARCHMD             X    0000A3        1  4017
SSARS                F    000120        4  4073
SSCLKCMP             F    0000E0        8  4067
SSCPUTIM             F    0000D8        8  4066
SSCRS                F    0001C0        4  4076
SSFPRS               D    000160        8  4074
SSGRS                F    000180        4  4075
SSMODEL              F    00010C        4  4071
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
SSPREFIX	F	000108	4	4070	
SSPSW	F	000100	8	4069	
SSXSAA	A	0000D4	4	4065	
STFLDATA	F	0000C8	4	4038	
SUBTEST	X	0021FF	1	3926	3549 3579 3589 3600 3612 3622 3632 3643 3653 3668 3706 3713 3721 3742 3748 3755 3768 3774 3782 3786 3790
SVCICODE	H	00008A	2	3997	
SVCIID	F	000088	4	3993	
SVCIILC	X	000089	1	3995	
SVCIILCM	U	00000C	1	3996	
SVCNPSW	F	000060	8	3980	
SVCOPSW	F	000020	8	3952	3959
SVPGMNEW	D	0003D8	8	3732	3714 3738
TEST01	I	00022A	4	3558	3538
TEST91	I	000334	4	3667	3540
TESTADDR	U	0021FE	1	3840	3923
TESTNUM	X	0021FE	1	3925	3546 3558 3667
TIMER	F	000050	4	3976	
TTDES	F	000054	4	3977	
UA0	F	000010	8	3949	
UA1	F	00004C	4	3974	
UA2	F	0000A4	4	4019	
UA3	F	0000B4	4	4028	
UA4	X	0000B8	1	4029	
UA5	X	0000CC	8	4039	
UA6	X	0000EC	8	4045	
UA7	F	000118	8	4056	
UA8	X	000180	32	4085	
ZBRKADDR	A	000110	8	4055	
ZEMONCNT	F	00010C	4	4054	
ZEMONCTR	A	000100	8	4052	
ZEMONSIZ	F	000108	4	4053	
ZEXTNPSW	X	0001B0	16	4088	
ZEXTOPSW	X	000130	16	4080	
ZIONPSW	X	0001F0	16	4092	
ZIOOPSW	X	000170	16	4084	
ZMCKNPSW	X	0001E0	16	4091	
ZMCKOPSW	X	000160	16	4083	
ZMKFAILA	F	0000F8	8	4047	
ZMONCODE	F	0000B0	8	4022	
ZPGMNPSW	X	0001D0	16	4090	
ZPGMOPSW	X	000150	16	4082	
ZPGMTRX	F	0000A8	8	4021	
ZRSTNPSW	X	0001A0	16	4087	
ZRSTOPSW	X	000120	16	4079	
ZSASDISP	U	0011C0	1	4093	
ZSVCNPSW	X	0001C0	16	4089	
ZSVCOPSW	X	000140	16	4081	
=A(PAGE)	A	0004B0	4	3831	3682
=A(PAGETABS)	A	0004AC	4	3830	3679 3707
=A(PFINSADR)	A	0004B8	4	3833	3743
=A(PFPGBYTS)	A	0004B4	4	3832	3701

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
=A(SEGTABLS)	A	0004A8	4	3829	3677

```
 MACRO     DEFN  REFERENCES

ANTR        113
APROB       245
ARCHIND     405   3435
ARCHLVL     546   3434
ASAIPL      672   3495
ASALOAD     752   3478
ASAREA      807   3939
ASAZAREA    992
CPUWAIT    1075
DSECTS     1401   3936
DWAIT      1604   3814   3819
DWAITEND   1661   3813
ENADEV     1669
ESA390     1769
IOCB       1780
IOCBDS     1956
IOFMT      1990   4098
IOINIT     2328
IOTRFR     2369
ORB        2417
POINTER    2606
PSWFMT     2634
RAWAIT     2768
RAWIO      2864
SIGCPU     3022
SMMGR      3080
SMMGRB     3180
TRAP128    3229
TRAP64     3206   3480   3483
TRAPS      3242
ZARCH      3316
ZEROH      3328
ZEROL      3356
ZEROLH     3384
ZEROLL     3407
```

```
   DESC     SYMBOL   SIZE     POS        ADDR

Entry: 0

Image      IMAGE    12289  0000-3000  0000-3000
  Region   CODE     12289  0000-3000  0000-3000
    CSECT  CLCLE03  12289  0000-3000  0000-3000
```

```
   STMT                          FILE NAME

1     /devstor/dev/satk/samples/tests/./CLCLE-03-basic.asm
2     /home/tn529/dev/satk/srcasm/satk.mac


** NO ERRORS FOUND **
```