```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                     2 ***********************************************************************
                                                     3 *
                                                     4 *  Testcase cmpxchg16 as used by CDSG, STPQ and LPQ instructions
                                                     5 *
                                                     6 ***********************************************************************
                                                     7 *
                                                     8 *  CDSG is one of the instructions that the POP refers to as having
                                                     9 *  Interlocked-Update References, also known as being 'atomic'.  This
                                                    10 *  instruction can be implemented using compiler intrinsics for certain
                                                    11 *  host architectures.  On X86-64 / AMD64 processors, the "cmpxchg16b"
                                                    12 *  instruction can be used when available.
                                                    13 *
                                                    14 *  This means that whilst one CPU performs a CDSG instruction, its
                                                    15 *  memory references are interlocked against those by other CPU's.
                                                    16 *  This test attempts to verify this, with an approach, similar to the
                                                    17 *  one in the CBUC test.
                                                    18 *
                                                    19 *  The STPQ and LPQ instructions also use "cmpxchg" and are tested
                                                    20 *  in here as well.
                                                    21 *
                                                    22 ***********************************************************************


                                                    24 ***********************************************************************
                                                    25 *
                                                    26 *                        Example test scripts
                                                    27 *
                                                    28 *                             (CDSG.tst)
                                                    29 *
                                                    30 * *Testcase for cmpxchg16 as used by CDSG, STPQ and LPQ instructions
                                                    31 * mainsize    1
                                                    32 * numcpu      2
                                                    33 * sysclear
                                                    34 * archlvl     z/Arch
                                                    35 * loadcore    "$(testpath)/CDSG.core"
                                                    36 * runtest     1
                                                    37 * v 900.B0
                                                    38 * *Done
                                                    39 *
                                                    40 ***********************************************************************
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    42 ***********************************************************************
                                                    43 *
                                                    44 *                         PROGRAMMING NOTE
                                                    45 *
                                                    46 *  During initialisation we test the functionality of the Store Pair
                                                    47 *  to Quadword (STPQ) and load Pair from Quadword (LPQ) instructions.
                                                    48 *  We merely do this as these two intructions also rely on the
                                                    49 *  Hercules macro cmpchxg16 (as of the most recent updates), as does
                                                    50 *  the CDSG implementation.  This cmpxchg16 macro may be assisted,
                                                    51 *  depending on the Hercules host architecture.
                                                    52 *
                                                    53 *  After initialisation, a second CPU is started, at which point in
                                                    54 *  time both CPU's perform a loop lasting LOOPMAX iterations.  The
                                                    55 *  first of the loop centers around the CDSG instruction.  The second
                                                    56 *  loop performs 2 CSG instructions, the first one of which overlaps
                                                    57 *  with the 2nd half destination of CDSG.  These overlapping causes
                                                    58 *  CC=1 when it occurs.  As a result, the non-overlapping destinations
                                                    59 *  gets incremented exactly LOOPMAX times, the overlapping part twice
                                                    60 *  that.  This is what is being checked for success by the test.  By
                                                    61 *  inspecting the CDSGCNTR and CSG_CNTR, one can see how many attempts
                                                    62 *  were needed. These are always higher that LOOPMAX. The exact number
                                                    63 *  varies on every test run.
                                                    64 *
                                                    65 *  The DESTination area is 24 bytes long, and is initialised as
                                                    66 *  follows:
                                                    67 *
                                                    68 *     '1A1B2A2B3A3B00004A4B5A5B6A6B00007A7B8A8B9A9B0000'X
                                                    69 *
                                                    70 *  The first loop works against DEST1+DEST2 (the leftmost 16 bytes),
                                                    71 *  the second loop against DEST2+DEST3 (the rightmost 16 bytes).
                                                    72 *  Both loops implement an atomic "increment" with the value:
                                                    73 *
                                                    74 *     '00000000000000010000000000000001'X
                                                    75 *
                                                    76 *  Thus also for the second loop:
                                                    77 *
                                                    78 *                     '00000000000000010000000000000001'X
                                                    79 *
                                                    80 *  The process ends when both loops have incremented LOOPMAX times,
                                                    81 *  and the doubleword in the middle will have been incremented
                                                    82 *  exactly twice that amount -- that is, if the CDSG operation is
                                                    83 *  really atomic.  And that is what will decide a successfull CDSG
                                                    84 *  instruction or not.
                                                    85 *
                                                    86 ***********************************************************************
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    88          PRINT OFF
                                                  3469          PRINT ON

                                                  3471 ***********************************************************************
                                                  3472 *        SATK prolog stuff...
                                                  3473 ***********************************************************************

                                                  3475          ARCHLVL  MNOTE=NO
                                                  3477+$AL      OPSYN AL
                                                  3478+$ALR     OPSYN ALR
                                                  3479+$B       OPSYN B
                                                  3480+$BAS     OPSYN BAS
                                                  3481+$BASR    OPSYN BASR
                                                  3482+$BC      OPSYN BC
                                                  3483+$BCTR    OPSYN BCTR
                                                  3484+$BE      OPSYN BE
                                                  3485+$BH      OPSYN BH
                                                  3486+$BL      OPSYN BL
                                                  3487+$BM      OPSYN BM
                                                  3488+$BNE     OPSYN BNE
                                                  3489+$BNH     OPSYN BNH
                                                  3490+$BNL     OPSYN BNL
                                                  3491+$BNM     OPSYN BNM
                                                  3492+$BNO     OPSYN BNO
                                                  3493+$BNP     OPSYN BNP
                                                  3494+$BNZ     OPSYN BNZ
                                                  3495+$BO      OPSYN BO
                                                  3496+$BP      OPSYN BP
                                                  3497+$BXLE    OPSYN BXLE
                                                  3498+$BZ      OPSYN BZ
                                                  3499+$CH      OPSYN CH
                                                  3500+$L       OPSYN L
                                                  3501+$LH      OPSYN LH
                                                  3502+$LM      OPSYN LM
                                                  3503+$LPSW    OPSYN LPSW
                                                  3504+$LR      OPSYN LR
                                                  3505+$LTR     OPSYN LTR
                                                  3506+$NR      OPSYN NR
                                                  3507+$SL      OPSYN SL
                                                  3508+$SLR     OPSYN SLR
                                                  3509+$SR      OPSYN SR
                                                  3510+$ST      OPSYN ST
                                                  3511+$STM     OPSYN STM
                                                  3512+$X       OPSYN X
                                                  3513+$AHI     OPSYN AHI
                                                  3514+$B       OPSYN J
                                                  3515+$BC      OPSYN BRC
                                                  3516+$BE      OPSYN JE
                                                  3517+$BH      OPSYN JH
                                                  3518+$BL      OPSYN JL
                                                  3519+$BM      OPSYN JM
                                                  3520+$BNE     OPSYN JNE
                                                  3521+$BNH     OPSYN JNH
                                                  3522+$BNL     OPSYN JNL
                                                  3523+$BNM     OPSYN JNM
                                                  3524+$BNO     OPSYN JNO
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3525+$BNP     OPSYN JNP
                                                  3526+$BNZ     OPSYN JNZ
                                                  3527+$BO      OPSYN JO
                                                  3528+$BP      OPSYN JP
                                                  3529+$BXLE    OPSYN JXLE
                                                  3530+$BZ      OPSYN JZ
                                                  3531+$CHI     OPSYN CHI
                                                  3532+$AHI     OPSYN AGHI
                                                  3533+$AL      OPSYN ALG
                                                  3534+$ALR     OPSYN ALGR
                                                  3535+$BCTR    OPSYN BCTGR
                                                  3536+$BXLE    OPSYN JXLEG
                                                  3537+$CH      OPSYN CGH
                                                  3538+$CHI     OPSYN CGHI
                                                  3539+$L       OPSYN LG
                                                  3540+$LH      OPSYN LGH
                                                  3541+$LM      OPSYN LMG
                                                  3542+$LPSW    OPSYN LPSWE
                                                  3543+$LR      OPSYN LGR
                                                  3544+$LTR     OPSYN LTGR
                                                  3545+$NR      OPSYN NGR
                                                  3546+$SL      OPSYN SLG
                                                  3547+$SLR     OPSYN SLGR
                                                  3548+$SR      OPSYN SGR
                                                  3549+$ST      OPSYN STG
                                                  3550+$STM     OPSYN STMG
                                                  3551+$X       OPSYN XG
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3553 **********************************************************************
                                                  3554 *        Initiate the CDSG CSECT in the CODE region
                                                  3555 *        with the location counter at 0
                                                  3556 **********************************************************************

                                                  3558 CDSGTEST ASALOAD  REGION=CODE
                                00000000  000409CB  3559+CDSGTEST START 0,CODE
00000000  00020000 00000000                       3561+         PSW   0,0,2,0,X'008'       64-bit Restart ISR Trap New PSW
00000010                        00000010  00000058  3562+         ORG   CDSGTEST+X'058'
00000058  00020000 00000000                       3564+         PSW   0,0,2,0,X'018'       64-bit External ISR Trap New PSW
00000068  00020000 00000000                       3565+         PSW   0,0,2,0,X'020'       64-bit Supervisor Call ISR Trap New PSW
00000078  00020000 00000000                       3566+         PSW   0,0,2,0,X'028'       64-bit Program ISR Trap New PSW
00000088  00020000 00000000                       3567+         PSW   0,0,2,0,X'030'       64-bit Machine Check Trap New PSW
00000098  00020000 00000000                       3568+         PSW   0,0,2,0,X'038'       64-bit Input/Output Trap New PSW
000000A8                        000000A8  000001A0  3569+         ORG   CDSGTEST+X'1A0'
000001A0  00020000 00000000                       3571+         PSWZ  0,0,2,0,X'120'      Restart ISR Trap New PSW
000001B0  00020000 00000000                       3572+         PSWZ  0,0,2,0,X'130'      External ISR Trap New PSW
000001C0  00020000 00000000                       3573+         PSWZ  0,0,2,0,X'140'      Supervisor Call ISR Trap New PSW
000001D0  00020000 00000000                       3574+         PSWZ  0,0,2,0,X'150'      Program ISR Trap New PSW
000001E0  00020000 00000000                       3575+         PSWZ  0,0,2,0,X'160'      Machine Check Trap New PSW
000001F0  00020000 00000000                       3576+         PSWZ  0,0,2,0,X'170'      Input/Output Trap New PSW


                                                  3578 **********************************************************************
                                                  3579 *        Define the z/Arch RESTART PSW
                                                  3580 **********************************************************************

                                00000200  00000001  3582 PREVORG  EQU   *
00000200                        00000200  000001A0  3583          ORG   CDSGTEST+X'1A0'
                                                  3584 *        PSWZ  <sys>,<key>,<mwp>,<prog>,<addr>[,amode]
000001A0  00000001 80000000                       3585          PSWZ  0,0,0,0,X'200',64
000001B0                        000001B0  00000200  3586          ORG   PREVORG


                                                  3588 **********************************************************************
                                                  3589 *        Create IPL (restart) PSW
                                                  3590 **********************************************************************

                                                  3592          ASAIPL   IA=BEGIN
                                00000000  000409CB  3593+CDSGTEST CSECT
00000200                        00000200  00000000  3594+         ORG   CDSGTEST
00000000  00080000 00000200                       3595+         PSWE390 0,0,0,0,BEGIN,24
00000008                        00000008  00000200  3596+         ORG   CDSGTEST+512     Reset CSECT to end of assigned storage area
                                00000000  000409CB  3597+CDSGTEST CSECT
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3599 ********************************************************************
                                                  3600 *              The actual CDSG program itself...
                                                  3601 ********************************************************************

00000200                     00000970             3603          USING CDSG,R0                     No base registers needed

00000200  1F00                                    3605 BEGIN    SLR   R0,R0                       Start clean
00000202  4110 0001                     00000001  3606          LA    R1,1                        Request z/Arch mode
00000206  1F22                                    3607          SLR   R2,R2                       Start clean
00000208  1F33                                    3608          SLR   R3,R3                       Start clean
0000020A  AE02 0012                     00000012  3609          SIGP  R0,R2,X'12'                 Request z/Arch mode

0000020E  1F11                                    3611          SLR   R1,R1                       Start clean
00000210  4120 0000                     00000000  3612          LA    R2,0                        Get our CPU number
00000214  4140 0224                     00000224  3613          LA    R4,BEGIN2                   Our restart entry point
00000218  4040 01AE                     000001AE  3614          STH   R4,X'1AE'                   Update restart PSW
0000021C  AE02 0006                     00000006  3615          SIGP  R0,R2,X'06'                 Restart our CPU
00000220  47F0 03E8                     000003E8  3616          B     SIG1FAIL                    WTF?! How did we get here?!

00000224                                          3618 BEGIN2   DS    0H
00000224  E340 0920 008F                00000920  3619          LPQ   R4,INIT1                    Load INIT1+INIT2 using LPQ
0000022A  E340 0920 0020                00000920  3620          CG    R4,INIT1                    Did LPQ high DW work ...
00000230  4770 0408                     00000408  3621          BNE   LPQFAIL1                    ... or not ?
00000234  E350 0928 0020                00000928  3622          CG    R5,INIT2                    Did LPQ low DW work ...
0000023A  4770 0420                     00000420  3623          BNE   LPQFAIL2                    ... or not ?
0000023E  E340 0900 008E                00000900  3624          STPQ  R4,DEST1                    Store DEST1+DEST2 for CDSG use
                                                  3625 *                                          R4+R5 for CDSGLOOP to use
00000244  E340 0900 0020                00000900  3626          CG    R4,DEST1                    Did STPQ high DW work ...
0000024A  4770 0438                     00000438  3627          BNE   STPQFAL1                    ... or not ?
0000024E  E350 0908 0020                00000908  3628          CG    R5,DEST2                    Did STPQ low DW work ...
00000254  4770 0450                     00000450  3629          BNE   STPQFAL2                    ... or not ?
00000258  41E0 09D0                     000009D0  3630          LA    R14,TRACE                   Initialize CDSG trace pointer

0000025C  B904 0065                               3632          LGR   R6,R5                       R6+R7 for CSG_LOOP to use
00000260  E370 0930 0004                00000930  3633          LG    R7,INIT3                    Load INIT3 to initialize ...
00000266  E370 0910 0024                00000910  3634          STG   R7,DEST3                    DEST3 so CSG_CPU can use it

0000026C  4120 0001                     00000001  3636          LA    R2,1                        Second CPU number
00000270  4180 02EA                     000002EA  3637          LA    R8,CSG_CPU                  Point to its entry point
00000274  4080 01AE                     000001AE  3638          STH   R8,X'1AE'                   Update restart PSW
00000278  AE02 0006                     00000006  3639          SIGP  R0,R2,X'06'                 Restart second CPU
0000027C  4770 03F8                     000003F8  3640          BNZ   SIG2FAIL                    WTF?! (SIGP failed!)
                                                  3641 *        B     CDSG_CPU                    Otherwise get started
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3643 **********************************************************************
                                                  3644 *  The first loop incremenets the rightmost halfwords of DEST1 and
                                                  3645 *  DEST2 with a single atomic CDSG instruction.  But the 2nd CPU will
                                                  3646 *  attempt to do the same thing against DEST2 with a CSG instruction,
                                                  3647 *  thus causing collisions now and then.  We keep track of the total
                                                  3648 *  CDSG's, which is limited to CNTRMAX.  The loop is expected to end
                                                  3649 *  before that, when exactly LOOPMAX successful increments (i.e. SWAP
                                                  3650 *  operations) have taken place.
                                                  3651 **********************************************************************

00000280  B982 00CC                               3653 CDSG_CPU XGR   R12,R12                  Initialise CDSG counter

00000284                                          3655 CDSGLOOP DS    0H
00000284  9201 E003                     00000003  3656          MVI   3(R14),X'1'              Trace CC=1 from previous CDSG

00000288                                          3658 CDSG_CC0 DS    0H
00000288  9500 09C6                     000009C6  3659          CLI   STOPFLAG,X'00'           Are we being asked to stop?
0000028C  4770 0364                     00000364  3660          BNE   GOODEOJ                  Yes, then do so.

00000290  41CC 0001                     00000001  3662          LA    R12,1(R12)               Increment the CDSG counter
00000294  50C0 099C                     0000099C  3663          ST    R12,CDSGCNTR             Update CDSG counter
00000298  59C0 09B0                     000009B0  3664          C     R12,CNTRMAX              CDSG counter overrun
0000029C  4720 0468                     00000468  3665          BH    CDSGCNTO                 Yes, that should never happen

000002A0  4184 0001                     00000001  3667          LA    R8,1(R4)                 Increment DEST1
000002A4  4195 0001                     00000001  3668          LA    R9,1(R5)                 Increment DEST1

000002A8  41EE 0010                     00000010  3670          LA    R14,16(R14)              Point to the next TRACE entry
000002AC  409E 0000                     00000000  3671          STH   R9,0(R14)                Trace the CDSG DEST2 update
000002B0  408E 0004                     00000004  3672          STH   R8,4(R14)                Trace the CDSG DEST1 update

000002B4  EB48 0900 003E                00000900  3674          CDSG  R4,R8,DEST1              CDSG to attempt doing it
000002BA  4770 0284                     00000284  3675          BNE   CDSGLOOP                 The CSG_CPU came in between

000002BE  BD83 09B4                     000009B4  3677          CLM   R8,B'0011',LOOPMAX       End value reached ?
000002C2  47B0 02D2                     000002D2  3678          BNL   CDSGEND                  Yes, CDSG incrementing ended

000002C6  B904 0048                               3680          LGR   R4,R8                    Copy the incremented DEST1
000002CA  B904 0059                               3681          LGR   R5,R9                    Copy the incremented DEST2
000002CE  47F0 0288                     00000288  3682          B     CDSG_CC0                 Go try the next CDSG

000002D2                                          3684 CDSGEND  DS    0H
000002D2  D501 0916 09B4      00000916  000009B4  3685          CLC   DEST3+6(2),LOOPMAX       Is also CSG_LOOP ended yet ?
000002D8  4740 02D2                     000002D2  3686          BL    CDSGEND                  Spin-loop style waiting for it
                                                  3687 *
                                                  3688 **       OK, both loops are finished!
                                                  3689 *
000002DC  D501 090E 09B6      0000090E  000009B6  3690          CLC   DEST2+6(2),LOOPMAX2      DEST2 must be =2*LOOPMAX
000002E2  4780 0364                     00000364  3691          BE    GOODEOJ                  Yes, then we have success!
000002E6  47F0 03D8                     000003D8  3692          B     FAILEOJ                  No, the test failed!!
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3694 ***********************************************************************
                                                  3695 *  The second loop incremenets the rightmost halfwords of DEST2 and
                                                  3696 *  DEST3, both of which use the atomic CSG instruction.  But the one
                                                  3697 *  against DEST2 will collide now and then with the CDSG atomic
                                                  3698 *  increments of DEST1+DEST2.  We keep track of the total number of
                                                  3699 *  CSG's against DEST2, which is limited to CNTRMAX.  The loop is
                                                  3700 *  expected to end before that, when DEST2 (and DEST3 also) have been
                                                  3701 *  able to increment (i.e. SWAP) exactly LOOPMAX times.
                                                  3702 ***********************************************************************

000002EA  B982 00DD                               3704 CSG_CPU  XGR   R13,R13                  Initialise CSG counter
000002EE  E360 0908 0004            00000908  3705          LG    R6,DEST2                 Initialise R6 for CSG
000002F4  E370 0910 0004            00000910  3706          LG    R7,DEST3                 Initialise R7 for CSG
000002FA  41F0 09D8                 000009D8  3707          LA    R15,TRACE+8              Initialize CSG trace pointer

000002FE                                      3709 CSG_LOOP DS    0H
000002FE  9201 F003                 00000003  3710          MVI   3(R15),X'1'              Trace CC=1 from previous CSG

00000302                                      3712 CSG_CC0  DS    0H
00000302  9500 09C6                 000009C6  3713          CLI   STOPFLAG,X'00'           Are we being asked to stop?
00000306  4770 03C8                 000003C8  3714          BNE   ENDNOTOK                 Yes, then do so.

0000030A  41DD 0001                 00000001  3716          LA    R13,1(R13)               Increment the CSG counter
0000030E  50D0 09AC                 000009AC  3717          ST    R13,CSG_CNTR             Update CSG counter
00000312  59D0 09B0                 000009B0  3718          C     R13,CNTRMAX              CSG counter overrun
00000316  4720 0480                 00000480  3719          BH    CSGCNTO                  Yes, that should never happen

0000031A  41A6 0001                 00000001  3721          LA    R10,1(R6)                Increment DEST2
0000031E  41FF 0010                 00000010  3722          LA    R15,16(R15)              Point to the next TRACE entry
00000322  40AF 0000                 00000000  3723          STH   R10,0(R15)               Trace the CSG DEST2 update
00000326  40BF 0004                 00000004  3724          STH   R11,4(R15)               Trace the previous DEST3 update

0000032A  EB6A 0908 0030            00000908  3726          CSG   R6,R10,DEST2             CSG to attempt doing it
00000330  4770 02FE                 000002FE  3727          BNE   CSG_LOOP                 The CDSG_CPU came in between

00000334  41B7 0001                 00000001  3729          LA    R11,1(R7)                Increments DEST3

00000338                                      3731 CSGLOOP2 DS    0H
00000338  EB7B 0910 0030            00000910  3732          CSG   R7,R11,DEST3             CSG to attempt doing it
0000033E  4770 0338                 00000338  3733          BNE   CSGLOOP2                 CDSGEND read came in between

00000342  BDB3 09B4                 000009B4  3735          CLM   R11,B'0011',LOOPMAX      End value reached ?
00000346  47B0 0356                 00000356  3736          BNL   CSG_END                  Yes, CSG incrementing ended

0000034A  B904 006A                           3738          LGR   R6,R10                   Copy the incremented DEST2
0000034E  B904 007B                           3739          LGR   R7,R11                   Copy the incremented DEST3
00000352  47F0 0302                 00000302  3740          B     CSG_CC0                  Go try the next CSG

00000356                                      3742 CSG_END  DS    0H
00000356  D501 0906 09B4  00000906  000009B4  3743          CLC   DEST1+6(2),LOOPMAX       Is also CDSGLOOP ended yet ?
0000035C  4740 0356                 00000356  3744          BL    CSG_END                  Spin-loop style waiting for it
00000360  47F0 03BA                 000003BA  3745          B     ENDOK                    OK, both loops are finished
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3747 ***********************************************************************
                                                  3748 *                        GOOD End Of Job
                                                  3749 ***********************************************************************


00000364                                          3751 GOODEOJ  DS       0H
00000364  D21F 0940 0498      00000940  00000498  3752          MVC      STATUS,=CL32'Success! CDSG, STPQ and LPQ: OK!'

0000036A  58C0 099C                     0000099C  3754          L        R12,CDSGCNTR            Load CSDG counter
0000036E  4EC0 09B8                     000009B8  3755          CVD      R12,PACKED              Convert CDSG counter to packed
00000372  D205 0974 09C0      00000974  000009C0  3756          MVC      CDSGCMPR,EDIT           Copy EDIT mask in CDSG counter
00000378  DE05 0974 09BD      00000974  000009BD  3757          ED       CDSGCMPR,PACKED+5       CDSG counter in zoned format

0000037E  48C0 0906                     00000906  3759          LH       R12,DEST1+6             Load CDSG SWAPs counter
00000382  4EC0 09B8                     000009B8  3760          CVD      R12,PACKED              Convert it to packed
00000386  D205 097A 09C0      0000097A  000009C0  3761          MVC      CDSGSWAP,EDIT           Copy EDIT mask in CDSG SWAP ctr
0000038C  DE05 097A 09BD      0000097A  000009BD  3762          ED       CDSGSWAP,PACKED+5       CDSG SWAPS counter in zoned

00000392  58D0 09AC                     000009AC  3764          L        R13,CSG_CNTR            Load CSG counter
00000396  4ED0 09B8                     000009B8  3765          CVD      R13,PACKED              Convert CSG  counter to packed
0000039A  D205 0984 09C0      00000984  000009C0  3766          MVC      CSG_CMPR,EDIT           Copy EDIT mask in CSG  counter
000003A0  DE05 0984 09BD      00000984  000009BD  3767          ED       CSG_CMPR,PACKED+5       CSG  counter in zoned format

000003A6  48D0 0916                     00000916  3769          LH       R13,DEST3+6             Load CSG SWAPs counter
000003AA  4ED0 09B8                     000009B8  3770          CVD      R13,PACKED              Convert it to packed
000003AE  D205 098A 09C0      0000098A  000009C0  3771          MVC      CSG_SWAP,EDIT           Copy EDIT mask in CSG SWAP cntr
000003B4  DE05 098A 09BD      0000098A  000009BD  3772          ED       CSG_SWAP,PACKED+5       CSG SWAPS counter in zoned

                                                  3774 *        B        ENDOK                   Load successful completion PSW
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3776 ***********************************************************************
                                                  3777 *                            PSWs
                                                  3778 ***********************************************************************


000003BA                                          3780 ENDOK    DS       0H
                                                  3781          DWAITEND LOAD=YES                Normal completion
000003BA  8200 03C0                     000003C0  3783+         LPSW  DWAT0009
000003C0  000A0000 00000000                       3784+DWAT0009 PSWE390 0,0,2,0,X'000000'



000003C8                                          3786 ENDNOTOK DS       0H
                                                  3787          DWAIT    LOAD=YES,CODE=BAD     Abnormal termination
000003C8  8200 03D0                     000003D0  3788+         LPSW  DWAT0010
000003D0  000A0000 00010BAD                       3789+DWAT0010 PSWE390 0,0,2,0,X'010BAD'



000003D8  92FF 09C6                     000009C6  3791 FAILEOJ  MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
                                                  3792          DWAIT    LOAD=YES,CODE=BAD     Abnormal termination
000003DC  8200 03E0                     000003E0  3793+         LPSW  DWAT0011
000003E0  000A0000 00010BAD                       3794+DWAT0011 PSWE390 0,0,2,0,X'010BAD'



000003E8  92FF 09C6                     000009C6  3796 SIG1FAIL MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
                                                  3797          DWAIT    LOAD=YES,CODE=111     First SIGP failed
000003EC  8200 03F0                     000003F0  3798+         LPSW  DWAT0012
000003F0  000A0000 00010111                       3799+DWAT0012 PSWE390 0,0,2,0,X'010111'



000003F8  92FF 09C6                     000009C6  3801 SIG2FAIL MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
                                                  3802          DWAIT    LOAD=YES,CODE=222     Second SIGP failed
000003FC  8200 0400                     00000400  3803+         LPSW  DWAT0013
00000400  000A0000 00010222                       3804+DWAT0013 PSWE390 0,0,2,0,X'010222'
```

```
  LOC         OBJECT CODE        ADDR1     ADDR2    STMT

00000408  92FF 09C6                       000009C6  3806 LPQFAIL1 MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
0000040C  D21F 0940 04B8      00000940  000004B8  3807          MVC      STATUS,=CL32'Failure! LPQ  Hi  does NOT match'
                                                  3808          DWAIT    LOAD=YES,CODE=333     LPQ High part failed
00000412  8200 0418                       00000418  3809+         LPSW  DWAT0014
00000418  000A0000 00010333                       3810+DWAT0014 PSWE390 0,0,2,0,X'010333'



00000420  92FF 09C6                       000009C6  3812 LPQFAIL2 MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
00000424  D21F 0940 04D8      00000940  000004D8  3813          MVC      STATUS,=CL32'Failure! LPQ  Lo  does NOT match'
                                                  3814          DWAIT    LOAD=YES,CODE=444     LPQ Low part failed
0000042A  8200 0430                       00000430  3815+         LPSW  DWAT0015
00000430  000A0000 00010444                       3816+DWAT0015 PSWE390 0,0,2,0,X'010444'



00000438  92FF 09C6                       000009C6  3818 STPQFAL1 MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
0000043C  D21F 0940 04F8      00000940  000004F8  3819          MVC      STATUS,=CL32'Failure! STPQ Hi  does NOT match'
                                                  3820          DWAIT    LOAD=YES,CODE=555     STPQ High part failes
00000442  8200 0448                       00000448  3821+         LPSW  DWAT0016
00000448  000A0000 00010555                       3822+DWAT0016 PSWE390 0,0,2,0,X'010555'



00000450  92FF 09C6                       000009C6  3824 STPQFAL2 MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
00000454  D21F 0940 0518      00000940  00000518  3825          MVC      STATUS,=CL32'Failure! STPQ Lo  does NOT match'
                                                  3826          DWAIT    LOAD=YES,CODE=666     STPQ Low part failed
0000045A  8200 0460                       00000460  3827+         LPSW  DWAT0017
00000460  000A0000 00010666                       3828+DWAT0017 PSWE390 0,0,2,0,X'010666'



00000468  92FF 09C6                       000009C6  3830 CDSGCNTO MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
0000046C  D21F 0940 0538      00000940  00000538  3831          MVC      STATUS,=CL32'Failure! CDSG    Counter Overrun'
                                                  3832          DWAIT    LOAD=YES,CODE=777     CDSG Counter Overrun
00000472  8200 0478                       00000478  3833+         LPSW  DWAT0018
00000478  000A0000 00010777                       3834+DWAT0018 PSWE390 0,0,2,0,X'010777'



00000480  92FF 09C6                       000009C6  3836 CSGCNTO  MVI      STOPFLAG,X'FF'        Tell the other CPU to stop
00000484  D21F 0940 0558      00000940  00000558  3837          MVC      STATUS,=CL32'Failure! CBG     Counter Overrun'
                                                  3838          DWAIT    LOAD=YES,CODE=888     CSG Counter Overrun
0000048A  8200 0490                       00000490  3839+         LPSW  DWAT0019
00000490  000A0000 00010888                       3840+DWAT0019 PSWE390 0,0,2,0,X'010888'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3842 ********************************************************************
                                                  3843 *                      Working Storage
                                                  3844 ********************************************************************

00000498                                          3846          LTORG ,                         Literals pool
00000498  E2A48383 85A2A25A                       3847                =CL32'Success! CDSG, STPQ and LPQ: OK!'
000004B8  C6818993 A499855A                       3848                =CL32'Failure! LPQ  Hi  does NOT match'
000004D8  C6818993 A499855A                       3849                =CL32'Failure! LPQ  Lo  does NOT match'
000004F8  C6818993 A499855A                       3850                =CL32'Failure! STPQ Hi  does NOT match'
00000518  C6818993 A499855A                       3851                =CL32'Failure! STPQ Lo  does NOT match'
00000538  C6818993 A499855A                       3852                =CL32'Failure! CDSG    Counter Overrun'
00000558  C6818993 A499855A                       3853                =CL32'Failure! CBG     Counter Overrun'

00000578                     00000578  00000900  3855          ORG   CDSGTEST+X'900'
00000900                                          3856          CNOP  0,16                      MUST be quadword ALIGNED!
00000900  00000000 00000000                       3857 DEST1    DC    XL8'0000000000000000'     DEST1+DEST2 updated using CDSG
00000908  00000000 00000000                       3858 DEST2    DC    XL8'0000000000000000'     DEST2 updated using CSG
00000910  00000000 00000000                       3859 DEST3    DC    XL8'0000000000000000'     DEST3 updated whenever CSG
                                                  3860 *                                        successfully updates DEST
00000918  07000700 07000700                       3861          CNOP  0,16                      MUST be quadword ALIGNED!
00000920  1A1B2A2B 3A3B0000                       3862 INIT1    DC    XL8'1A1B2A2B3A3B0000'     Initial value for DEST1
00000928  4A4B5A5B 6A6B0000                       3863 INIT2    DC    XL8'4A4B5A5B6A6B0000'     Initial value for DEST2
00000930  7A7B8A8B 9A9B0000                       3864 INIT3    DC    XL8'7A7B8A8B9A9B0000'     Initial value for DEST3

00000938  07000700 07000700                       3866          CNOP  0,16                      So that the output looks better
00000940  C6818993 A499855A                       3867 STATUS   DC    CL32'Failure!'            Overall status message
00000960  C995A2A3 994B40C3                       3868 COUNTERS DC    CL16'Instr. CMP  SWAP'    Title column
00000970  C3C4E2C7                                3869 CDSG     DC    CL4'CDSG'                 CDSG Instruction
00000974  40404040 4040                           3870 CDSGCMPR DC    CL6' '                    CDSG instructions attempted
0000097A  40404040 4040                           3871 CDSGSWAP DC    CL6' '                    CDSG successful swaps done
00000980  C3E2C740                                3872 CSG      DC    CL4'CSG '                 CSG  Instruction
00000984  40404040 4040                           3873 CSG_CMPR DC    CL6' '                    CSG  DEST2 instructions done
0000098A  40404040 4040                           3874 CSG_SWAP DC    CL6' '                    CSG  DEST2 successful swaps

00000990                                          3876          CNOP  0,16                      So that the output looks better
00000990  C3C4E2C7 C3D5E3D9                       3877 CDSGCNTL DC    CL8'CDSGCNTR'             CDSG counter label
00000998  40404040                                3878          DC    CL4' '
0000099C  00000000                                3879 CDSGCNTR DC    F'0'                      CDSG instructions attempted
000009A0  C3E2C76D C3D5E3D9                       3880 CSG_CNTL DC    CL8'CSG_CNTR'             CSG  counter label
000009A8  40404040                                3881          DC    CL4' '
000009AC  00000000                                3882 CSG_CNTR DC    F'0'                      CSG  instructions attempted

000009B0  0000EA60                                3884 CNTRMAX  DC    F'60000'                  Counter Overrun Maximum
000009B4  3A98                                    3885 LOOPMAX  DC    H'15000'                  Maximum number of loops
000009B6  7530                                    3886 LOOPMAX2 DC    H'30000'                  Maximum number of loops * 2

000009B8  00000000 0000000C                       3888 PACKED   DC    PL8'0'                    Packed decimal work area
000009C0  40202020 2020                           3889 EDIT     DC    XL6'402020202020'         EDIT mask with leading blank
000009C6  00                                      3890 STOPFLAG DC    X'00'                     Set to non-zero to stop test

000009C8  07000700 07000700                       3892          CNOP  0,16                      Beautify trace table looks
000009D0  00000000 00000000                       3893 TRACE    DC    65535F'0'                 Trace area for debugging
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT



                              00000000  00000001  3896 R0        EQU   0
                              00000001  00000001  3897 R1        EQU   1
                              00000002  00000001  3898 R2        EQU   2
                              00000003  00000001  3899 R3        EQU   3
                              00000004  00000001  3900 R4        EQU   4
                              00000005  00000001  3901 R5        EQU   5
                              00000006  00000001  3902 R6        EQU   6
                              00000007  00000001  3903 R7        EQU   7
                              00000008  00000001  3904 R8        EQU   8
                              00000009  00000001  3905 R9        EQU   9
                              0000000A  00000001  3906 R10       EQU   10
                              0000000B  00000001  3907 R11       EQU   11
                              0000000C  00000001  3908 R12       EQU   12
                              0000000D  00000001  3909 R13       EQU   13
                              0000000E  00000001  3910 R14       EQU   14
                              0000000F  00000001  3911 R15       EQU   15



                                                  3913           END
```

```
     SYMBOL        TYPE  VALUE   LENGTH  DEFN  REFERENCES

BEGIN               I    000200       2  3605  3595
BEGIN2              H    000224       2  3618  3613
CDSG                C    000970       4  3869  3603
CDSGCMPR            C    000974       6  3870  3756  3757
CDSGCNTL            C    000990       8  3877
CDSGCNTO            I    000468       4  3830  3665
CDSGCNTR            F    00099C       4  3879  3663  3754
CDSGEND             H    0002D2       2  3684  3678  3686
CDSGLOOP            H    000284       2  3655  3675
CDSGSWAP            C    00097A       6  3871  3761  3762
CDSGTEST            J    000000  264652  3559  3562  3569  3583  3594  3596  3855
CDSG_CC0            H    000288       2  3658  3682
CDSG_CPU            I    000280       4  3653
CNTRMAX             F    0009B0       4  3884  3664  3718
CODE                2    000000  264652  3559
COUNTERS            C    000960      16  3868
CSG                 C    000980       4  3872
CSGCNTO             I    000480       4  3836  3719
CSGLOOP2            H    000338       2  3731  3733
CSG_CC0             H    000302       2  3712  3740
CSG_CMPR            C    000984       6  3873  3766  3767
CSG_CNTL            C    0009A0       8  3880
CSG_CNTR            F    0009AC       4  3882  3717  3764
CSG_CPU             I    0002EA       4  3704  3637
CSG_END             H    000356       2  3742  3736  3744
CSG_LOOP            H    0002FE       2  3709  3727
CSG_SWAP            C    00098A       6  3874  3771  3772
DEST1               X    000900       8  3857  3624  3626  3674  3743  3759
DEST2               X    000908       8  3858  3628  3690  3705  3726
DEST3               X    000910       8  3859  3634  3685  3706  3732  3769
DWAT0009            3    0003C0       8  3784  3783
DWAT0010            3    0003D0       8  3789  3788
DWAT0011            3    0003E0       8  3794  3793
DWAT0012            3    0003F0       8  3799  3798
DWAT0013            3    000400       8  3804  3803
DWAT0014            3    000418       8  3810  3809
DWAT0015            3    000430       8  3816  3815
DWAT0016            3    000448       8  3822  3821
DWAT0017            3    000460       8  3828  3827
DWAT0018            3    000478       8  3834  3833
DWAT0019            3    000490       8  3840  3839
EDIT                X    0009C0       6  3889  3756  3761  3766  3771
ENDNOTOK            H    0003C8       2  3786  3714
ENDOK               H    0003BA       2  3780  3745
FAILEOJ             I    0003D8       4  3791  3692
GOODEOJ             H    000364       2  3751  3660  3691
IMAGE               1    000000  264652     0
INIT1               X    000920       8  3862  3619  3620
INIT2               X    000928       8  3863  3622
INIT3               X    000930       8  3864  3633
LOOPMAX             H    0009B4       2  3885  3677  3685  3735  3743
LOOPMAX2            H    0009B6       2  3886  3690
LPQFAIL1            I    000408       4  3806  3621
LPQFAIL2            I    000420       4  3812  3623
PACKED              P    0009B8       8  3888  3755  3757  3760  3762  3765  3767  3770  3772
PREVORG             U    000200       1  3582  3586
```

```
     SYMBOL        TYPE  VALUE   LENGTH  DEFN  REFERENCES

R0                  U    000000       1  3896  3603  3605  3609  3615  3639
R1                  U    000001       1  3897  3606  3611
R10                 U    00000A       1  3906  3721  3723  3726  3738
R11                 U    00000B       1  3907  3724  3729  3732  3735  3739
R12                 U    00000C       1  3908  3653  3662  3663  3664  3754  3755  3759  3760
R13                 U    00000D       1  3909  3704  3716  3717  3718  3764  3765  3769  3770
R14                 U    00000E       1  3910  3630  3656  3670  3671  3672
R15                 U    00000F       1  3911  3707  3710  3722  3723  3724
R2                  U    000002       1  3898  3607  3609  3612  3615  3636  3639
R3                  U    000003       1  3899  3608
R4                  U    000004       1  3900  3613  3614  3619  3620  3624  3626  3667  3674  3680
R5                  U    000005       1  3901  3622  3628  3632  3668  3681
R6                  U    000006       1  3902  3632  3705  3721  3726  3738
R7                  U    000007       1  3903  3633  3634  3706  3729  3732  3739
R8                  U    000008       1  3904  3637  3638  3667  3672  3674  3677  3680
R9                  U    000009       1  3905  3668  3671  3681
SIG1FAIL            I    0003E8       4  3796  3616
SIG2FAIL            I    0003F8       4  3801  3640
STATUS              C    000940      32  3867  3752  3807  3813  3819  3825  3831  3837
STOPFLAG            X    0009C6       1  3890  3659  3713  3791  3796  3801  3806  3812  3818  3824  3830  3836
STPQFAL1            I    000438       4  3818  3627
STPQFAL2            I    000450       4  3824  3629
TRACE               F    0009D0       4  3893  3630  3707
=CL32'Failure! CBG     Counter Overrun'
                    C    000558      32  3853  3837
=CL32'Failure! CDSG    Counter Overrun'
                    C    000538      32  3852  3831
=CL32'Failure! LPQ  Hi  does NOT match'
                    C    0004B8      32  3848  3807
=CL32'Failure! LPQ  Lo  does NOT match'
                    C    0004D8      32  3849  3813
=CL32'Failure! STPQ Hi  does NOT match'
                    C    0004F8      32  3850  3819
=CL32'Failure! STPQ Lo  does NOT match'
                    C    000518      32  3851  3825
=CL32'Success! CDSG, STPQ and LPQ: OK!'
                    C    000498      32  3847  3752
```

```
 MACRO    DEFN  REFERENCES

ANTR       154
APROB      286
ARCHIND    446   3476
ARCHLVL    587   3475
ASAIPL     713   3592
ASALOAD    793   3558
ASAREA     848
ASAZAREA  1033
CPUWAIT   1116
DSECTS    1442
DWAIT     1645   3782   3787   3792   3797   3802   3808   3814   3820   3826   3832   3838
DWAITEND  1702   3781
ENADEV    1710
ESA390    1810
IOCB      1821
IOCBDS    1997
IOFMT     2031
IOINIT    2369
IOTRFR    2410
ORB       2458
POINTER   2647
PSWFMT    2675
RAWAIT    2809
RAWIO     2905
SIGCPU    3063
SMMGR     3121
SMMGRB    3221
TRAP128   3270   3570
TRAP64    3247   3560   3563
TRAPS     3283
ZARCH     3357
ZEROH     3369
ZEROL     3397
ZEROLH    3425
ZEROLL    3448
```

```
   DESC     SYMBOL    SIZE       POS          ADDR

Entry: 0

Image      IMAGE     264652  00000-409CB  00000-409CB
  Region   CODE      264652  00000-409CB  00000-409CB
    CSECT  CDSGTEST  264652  00000-409CB  00000-409CB
   STMT                                                 FILE NAME

1     c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\CDSG\CDSG.asm
2     C:\Users\Fish\Documents\Visual Studio 2008\Projects\Hercules\_Git\_Harold\SATK-0\srcasm\satk.mac


** NO ERRORS FOUND **
```