```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                     2 ************************************************************************
                                                     3 *
                                                     4 *                      CLCL Unaligned Buffers Test
                                                     5 *
                                                     6 ************************************************************************
                                                     7 *
                                                     8 *  This program tests proper functioning of the CLCL instruction's
                                                     9 *  optimization logic (specifically, the "mem_cmp" function that the
                                                    10 *  CLCL instruction makes use of) to ensure the location of the in-
                                                    11 *  equality is properly reported.
                                                    12 *
                                                    13 *  Depending on the alignment of the two operands being compared, if
                                                    14 *  the length of the compare is large enough that it would cause the
                                                    15 *  comparison to cross a page boundary for both operands and the in-
                                                    16 *  equality occurs at an offset past the distance each operand is
                                                    17 *  from its respective page boundary added together, then the address
                                                    18 *  of the inequality that CLCL returns would be off by the shorter
                                                    19 *  of the two distances.
                                                    20 *
                                                    21 *  For example, if the operand addresses were X'123456' and X'456789'
                                                    22 *  (and the page size was X'800') and the inequality was at (or past)
                                                    23 *  X'123877', then CLCL would incorrectly report the address of the
                                                    24 *  inequality as being at address X'123877' - X'77' = X'123800':
                                                    25 *
                                                    26 *  X'123456' is X'3AA' bytes from the end of its page boundary.
                                                    27 *  X'456789' is X'77'  bytes from the end of its page boundary.
                                                    28 *  The true inequality is at X'123877' (X'123456' + X'77' + X'3AA').
                                                    29 *
                                                    30 *  The optimization logic would perform three separate compares: the
                                                    31 *  first starting at X'123456' for a length of X'77'. The second one
                                                    32 *  at address X'1234CD' (X'123456' + X'77') for a length of X'3AA',
                                                    33 *  and the third and final compare at address X'123877' (X'123456' +
                                                    34 *  X'77' + X'3AA') for a length of at least one byte.
                                                    35 *
                                                    36 *  Due to a bug in the original optimization logic however the length
                                                    37 *  of the first compare would not be added to the calculated offset of
                                                    38 *  where the inequality was located at.  That is to say, the offset of
                                                    39 *  the inequality would be calculated as operand-1 + X'3AA' instead of
                                                    40 *  operand-1 + X'77' + X'3AA'.  The X'77' offset would get erroneously
                                                    41 *  lost, thereby causing the location of the inequality to be reported
                                                    42 *  X'77' bytes BEFORE where the actual inequality was actually located
                                                    43 *  at! (Oops!)
                                                    44 *
                                                    45 *  The bug has since been fixed of course but to ensure such does not
                                                    46 *  occur again, this standalone runtest test performs a series of CLCL
                                                    47 *  comparisons whose parameters are such that they end up tripping the
                                                    48 *  bug as described. Thank you to Dave Kreiss for reporting the bug.
                                                    49 *
                                                    50 ************************************************************************
```

```
LOC  OBJECT CODE  ADDR1  ADDR2  STMT

                                   52 ********************************************************************************
                                   53 *
                                   54 *
                                   55 *                    EXAMPLE RUNTEST TEST CASE
                                   56 *
                                   57 *
                                   58 *            *Testcase CLCL Unaligned Buffers Test
                                   59 *            mainsize    2
                                   60 *            numcpu      1
                                   61 *            sysclear
                                   62 *            archlvl     390
                                   63 *            loadcore    "$(testpath)/CLCL.core"
                                   64 *            runtest     0.1
                                   65 *            *Done
                                   66 *
                                   67 *
                                   68 ********************************************************************************
```

```
 LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                      70          PRINT OFF
                                                    3451          PRINT ON

                                                    3453 ********************************************************************************
                                                    3454 *        SATK prolog stuff...
                                                    3455 ********************************************************************************

                                                    3457          ARCHLVL  ZARCH=NO,MNOTE=NO
                                                    3459+$AL      OPSYN AL
                                                    3460+$ALR     OPSYN ALR
                                                    3461+$B       OPSYN B
                                                    3462+$BAS     OPSYN BAS
                                                    3463+$BASR    OPSYN BASR
                                                    3464+$BC      OPSYN BC
                                                    3465+$BCTR    OPSYN BCTR
                                                    3466+$BE      OPSYN BE
                                                    3467+$BH      OPSYN BH
                                                    3468+$BL      OPSYN BL
                                                    3469+$BM      OPSYN BM
                                                    3470+$BNE     OPSYN BNE
                                                    3471+$BNH     OPSYN BNH
                                                    3472+$BNL     OPSYN BNL
                                                    3473+$BNM     OPSYN BNM
                                                    3474+$BNO     OPSYN BNO
                                                    3475+$BNP     OPSYN BNP
                                                    3476+$BNZ     OPSYN BNZ
                                                    3477+$BO      OPSYN BO
                                                    3478+$BP      OPSYN BP
                                                    3479+$BXLE    OPSYN BXLE
                                                    3480+$BZ      OPSYN BZ
                                                    3481+$CH      OPSYN CH
                                                    3482+$L       OPSYN L
                                                    3483+$LH      OPSYN LH
                                                    3484+$LM      OPSYN LM
                                                    3485+$LPSW    OPSYN LPSW
                                                    3486+$LR      OPSYN LR
                                                    3487+$LTR     OPSYN LTR
                                                    3488+$NR      OPSYN NR
                                                    3489+$SL      OPSYN SL
                                                    3490+$SLR     OPSYN SLR
                                                    3491+$SR      OPSYN SR
                                                    3492+$ST      OPSYN ST
                                                    3493+$STM     OPSYN STM
                                                    3494+$X       OPSYN X
                                                    3495+$AHI     OPSYN AHI
                                                    3496+$B       OPSYN J
                                                    3497+$BC      OPSYN BRC
                                                    3498+$BE      OPSYN JE
                                                    3499+$BH      OPSYN JH
                                                    3500+$BL      OPSYN JL
                                                    3501+$BM      OPSYN JM
                                                    3502+$BNE     OPSYN JNE
```

```
 LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                   3503+$BNH     OPSYN JNH
                                                   3504+$BNL     OPSYN JNL
                                                   3505+$BNM     OPSYN JNM
                                                   3506+$BNO     OPSYN JNO
                                                   3507+$BNP     OPSYN JNP
                                                   3508+$BNZ     OPSYN JNZ
                                                   3509+$BO      OPSYN JO
                                                   3510+$BP      OPSYN JP
                                                   3511+$BXLE    OPSYN JXLE
                                                   3512+$BZ      OPSYN JZ
                                                   3513+$CHI     OPSYN CHI
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  3515 ********************************************************************************
                                                  3516 *        Initiate the CLCL CSECT in the CODE region
                                                  3517 *        with the location counter at 0
                                                  3518 ********************************************************************************

                                                  3520 CLCL     ASALOAD  REGION=CODE
                                00000000  00081031 3521+CLCL     START 0,CODE
00000000  000A0000 00000008                       3523+          PSW   0,0,2,0,X'008'      64-bit Restart ISR Trap New PSW
00000008                        00000008  00000058 3524+          ORG   CLCL+X'058'
00000058  000A0000 00000018                       3526+          PSW   0,0,2,0,X'018'      64-bit External ISR Trap New PSW
00000060  000A0000 00000020                       3527+          PSW   0,0,2,0,X'020'      64-bit Supervisor Call ISR Trap New PSW
00000068  000A0000 00000028                       3528+          PSW   0,0,2,0,X'028'      64-bit Program ISR Trap New PSW
00000070  000A0000 00000030                       3529+          PSW   0,0,2,0,X'030'      64-bit Machine Check Trap New PSW
00000078  000A0000 00000038                       3530+          PSW   0,0,2,0,X'038'      64-bit Input/Output Trap New PSW
00000080                        00000080  00000200 3531+          ORG   CLCL+512

                                                  3533 ********************************************************************************
                                                  3534 *        Create IPL (restart) PSW
                                                  3535 ********************************************************************************

                                                  3537          ASAIPL   IA=BEGIN
                                00000000  00081031 3538+CLCL     CSECT
00000200                        00000200  00000000 3539+          ORG   CLCL
00000000  00080000 00000200                       3540+          PSW   0,0,0,0,BEGIN,24
00000008                        00000008  00000200 3541+          ORG   CLCL+512         Reset CSECT to end of assigned storage area
                                00000000  00081031 3542+CLCL     CSECT



                                                  3544 ********************************************************************************
                                                  3545 *                The actual "CLCL" program itself...
                                                  3546 ********************************************************************************
                                                  3547 *
                                                  3548 *  Architecture Mode:   ESA/390
                                                  3549 *
                                                  3550 *  Addressing Mode:     31-bit
                                                  3551 *
                                                  3552 *  Register Usage:      R12 - R13     Base registers
                                                  3553 *                       R0  - R1      CLCL Operand-1
                                                  3554 *                       R14 - R15     CLCL Operand-2
                                                  3555 *                       R2  - R11     Work registers
                                                  3556 *
                                                  3557 ********************************************************************************


00000200                        00000200          3559          USING  BEGIN,R12       FIRST Base Register
00000200                        00001200          3560          USING  BEGIN+4096,R13  SECOND Base Register

00000200  05C0                                    3562 BEGIN    BALR  R12,0            Initalize FIRST base register
00000202  06C0                                    3563          BCTR  R12,0            Initalize FIRST base register
00000204  06C0                                    3564          BCTR  R12,0            Initalize FIRST base register

00000206  41D0 C800                     00000800  3566          LA    R13,2048(,R12)   Initalize SECOND base register
0000020A  41D0 D800                     00000800  3567          LA    R13,2048(,R13)   Initalize SECOND base register
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    3569 ************************************************************************
                                                    3570 *        Compare DATA1 and DATA2 one BUFFSIZE at a time...
                                                    3571 ************************************************************************

                                                    3573 *                          R4      R5    R6     R7      R8       R9
0000020E  9849 C088                      00000288   3574          LM    R4,R9,=A(BUFFER1,DATA1,BUFFER2,DATA2,BUFFSIZE,DATASIZE)

00000212  1598                                      3576          CLR   R9,R8             DATASIZE greater than BUFFSIZE?
00000214  47B0 C01A                      0000021A   3577          BNL   CHNKLOOP          Yes, get started...
00000218  1889                                      3578          LR    R8,R9             No, only compare however much we have!

                                                    3580 *             Fill buffers with next chunk of data...

0000021A  1804                                      3582 CHNKLOOP LR    R0,R4             R0 --> BUFFER1
0000021C  1825                                      3583          LR    R2,R5             R2 --> DATA1
0000021E  1818                                      3584          LR    R1,R8             R1 <== BUFFSIZE
00000220  1838                                      3585          LR    R3,R8             R3 <== BUFFSIZE
00000222  0E02                                      3586          MVCL  R0,R2             Copy into BUFFER1 <== next DATA1 chunk

00000224  1806                                      3588          LR    R0,R6             R0 --> BUFFER2
00000226  1827                                      3589          LR    R2,R7             R2 --> DATA2
00000228  1818                                      3590          LR    R1,R8             R1 <== BUFFSIZE
0000022A  1838                                      3591          LR    R3,R8             R3 <== BUFFSIZE
0000022C  0E02                                      3592          MVCL  R0,R2             Copy into BUFFER2 <== next DATA2 chunk

                                                    3594 *                  Prepare for CLCL...

0000022E  1804                                      3596          LR    R0,R4             R0  --> BUFFER1
00000230  18E6                                      3597          LR    R14,R6            R14 --> BUFFER2
00000232  1818                                      3598          LR    R1,R8             R1  <== BUFFSIZE
00000234  18F8                                      3599          LR    R15,R8            R15 <== BUFFSIZE

                                                    3601 *                Compare the two buffers...

00000236  0F0E                                      3603 CONTINUE CLCL  R0,R14            Compare BUFFER1 with BUFFER2...
00000238  4780 C056                      00000256   3604          BE    NXTCHUNK          Equal: Buffer compare complete

                                                    3606 *           Inequality found: VERIFY ITS ACCURACY!

0000023C  18A0                                      3608          LR    R10,R0            R10 --> Supposed unequal byte
0000023E  D500 A000 E000      00000000  00000000    3609          CLC   0(1,R10),0(R14)    Valid inequality?
00000244  4780 C078                      00000278   3610          BE    FAILURE           Bogus inequality!  CLCL BUG!  FAIL!

                                                    3612 *           CLCL was correct.  Get past inequality
                                                    3613 *           and finish comparing the buffer data if
                                                    3614 *           there is any data remaining in the buffer
                                                    3615 *           that we haven't compared yet...

00000248  4A00 C0A0                      000002A0   3617          AH    R0,=H'1'          Get past unequal byte
0000024C  4AE0 C0A0                      000002A0   3618          AH    R14,=H'1'         Get past unequal byte
00000250  0610                                      3619          BCTR  R1,0              Get past unequal byte
00000252  46F0 C036                      00000236   3620          BCT   R15,CONTINUE      Go finish buffer if any bytes remain...
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                    3622 *          Go on to next chunk of data  -- if there is one.

00000256  1E58                                      3624 NXTCHUNK ALR   R5,R8            R5 --> Next DATA1 chunk
00000258  1E78                                      3625          ALR   R7,R8            R7 --> Next DATA2 chunk

0000025A  1B98                                      3627          SR    R9,R8            Decrement DATA bytes remaining
0000025C  4780 C06A                       0000026A  3628          BZ    SUCCESS          None: We're done...
00000260  4720 C01A                       0000021A  3629          BP    CHNKLOOP         Lots: Go compare next chunk...
00000264  1089                                      3630          LPR   R8,R9            Some: Make R8 <== positive remaining
00000266  47F0 C01A                       0000021A  3631          B     CHNKLOOP         Go compare final chunk...
```

```
  LOC         OBJECT CODE       ADDR1     ADDR2     STMT

                                                    3633 ********************************************************************************
                                                    3634 *        Normal completion or Abnormal termination PSWs
                                                    3635 ********************************************************************************


                                                    3637 SUCCESS  DWAITEND LOAD=YES          Normal completion
0000026A                                            3639+SUCCESS  DS    0H
0000026A  8200 C070                 00000270        3640+         LPSW  DWAT0008
00000270  000A0000 00000000                         3641+DWAT0008 PSW   0,0,2,0,X'000000'


                                                    3643 FAILURE  DWAIT LOAD=YES,CODE=BAD    Abnormal termination
00000278                                            3644+FAILURE  DS    0H
00000278  8200 C080                 00000280        3645+         LPSW  DWAT0009
00000280  000A0000 00010BAD                         3646+DWAT0009 PSW   0,0,2,0,X'010BAD'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                    3648 ***************************************************************************
                                                    3649 *        Working Storage
                                                    3650 ***************************************************************************
                                                    3651 *
                                                    3652 *        The specific bug that was reported:
                                                    3653 *
                                                    3654 *        4DE   787FE B54   87F46 B54
                                                    3655 *        4DF   787FF B53   87F47 B53
                                                    3656 *
                                                    3657 *        F32   79252 100   8899A 100    (BOGUS!)
                                                    3658 *
                                                    3659 *        FEA   7930A 048   88A52 048
                                                    3660 *        FEB   7930B 047   88A53 047
                                                    3661 *
                                                    3662 ***************************************************************************

00000288                                            3664          LTORG ,                 Literals pool
00000288  00020320 00060000                        3665                =A(BUFFER1,DATA1,BUFFER2,DATA2,BUFFSIZE,DATASIZE)
000002A0  0001                                      3666                =H'1'

                            00002000  00000001    3668 BUFFSIZE EQU   (8*1024)
                            00001032  00000001    3669 DATASIZE EQU   X'1032'

                            00000320  00000001    3671 BUFF1OFF EQU   X'320'
                            00000A68  00000001    3672 BUFF2OFF EQU   X'A68'

000002A2                    000002A2  00020320    3674          ORG   CLCL+(1*(128*1024))+BUFF1OFF
00020320  00000000 00000000                         3675 BUFFER1  DC    (BUFFSIZE/8)XL8'00'

00022320                    00022320  00040A68    3677          ORG   CLCL+(2*(128*1024))+BUFF2OFF
00040A68  00000000 00000000                         3678 BUFFER2  DC    (BUFFSIZE/8)XL8'00'

00042A68                    00042A68  00060000    3680          ORG   CLCL+(3*(128*1024))    X'60000'
00060000  00000000 00000000                         3681 DATA1    DC    (DATASIZE)X'00'        X'60000'

00061032                    00061032  00080000    3683          ORG   CLCL+(4*(128*1024))    X'80000'
00080000  00000000 00000000                         3684 DATA2    DC    (DATASIZE)X'00'        X'80000'

00081032                    00081032  000804DE    3686          ORG   DATA2+X'04DE'
000804DE  FF                                        3687          DC    X'FF'

000804DF                    000804DF  000804DF    3689          ORG   DATA2+X'04DF'
000804DF  FF                                        3690          DC    X'FF'

000804E0                    000804E0  00080FEA    3692          ORG   DATA2+X'0FEA'
00080FEA  FF                                        3693          DC    X'FF'

00080FEB                    00080FEB  00080FEB    3695          ORG   DATA2+X'0FEB'
00080FEB  FF                                        3696          DC    X'FF'

00080FEC                    00080FEC  00081032    3698          ORG   DATA2+DATASIZE
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT
                                                  3700 ********************************************************************************
                                                  3701 *        Register equates
                                                  3702 ********************************************************************************

                                 00000000  00000001 3704 R0       EQU   0
                                 00000001  00000001 3705 R1       EQU   1
                                 00000002  00000001 3706 R2       EQU   2
                                 00000003  00000001 3707 R3       EQU   3
                                 00000004  00000001 3708 R4       EQU   4
                                 00000005  00000001 3709 R5       EQU   5
                                 00000006  00000001 3710 R6       EQU   6
                                 00000007  00000001 3711 R7       EQU   7
                                 00000008  00000001 3712 R8       EQU   8
                                 00000009  00000001 3713 R9       EQU   9
                                 0000000A  00000001 3714 R10      EQU   10
                                 0000000B  00000001 3715 R11      EQU   11
                                 0000000C  00000001 3716 R12      EQU   12
                                 0000000D  00000001 3717 R13      EQU   13
                                 0000000E  00000001 3718 R14      EQU   14
                                 0000000F  00000001 3719 R15      EQU   15


                                                  3721          END
```

```
     SYMBOL         TYPE  VALUE   LENGTH  DEFN  REFERENCES

BEGIN                I    000200       2  3562  3540  3559  3560
BUFF1OFF             U    000320       1  3671  3674
BUFF2OFF             U    000A68       1  3672  3677
BUFFER1              X    020320       8  3675  3574
BUFFER2              X    040A68       8  3678  3574
BUFFSIZE             U    002000       1  3668  3675  3678  3574
CHNKLOOP             I    00021A       2  3582  3577  3629  3631
CLCL                 J    000000  528434  3521  3524  3531  3539  3541  3674  3677  3680  3683
CODE                 2    000000  528434  3521
CONTINUE             I    000236       2  3603  3620
DATA1                X    060000       1  3681  3574
DATA2                X    080000       1  3684  3686  3689  3692  3695  3698  3574
DATASIZE             U    001032       1  3669  3681  3684  3698  3574
DWAT0008             3    000270       8  3641  3640
DWAT0009             3    000280       8  3646  3645
FAILURE              H    000278       2  3644  3610
IMAGE                1    000000  528434     0
NXTCHUNK             I    000256       2  3624  3604
R0                   U    000000       1  3704  3582  3586  3588  3592  3596  3603  3608  3617
R1                   U    000001       1  3705  3584  3590  3598  3619
R10                  U    00000A       1  3714  3608  3609
R11                  U    00000B       1  3715
R12                  U    00000C       1  3716  3559  3562  3563  3564  3566
R13                  U    00000D       1  3717  3560  3566  3567
R14                  U    00000E       1  3718  3597  3603  3609  3618
R15                  U    00000F       1  3719  3599  3620
R2                   U    000002       1  3706  3583  3586  3589  3592
R3                   U    000003       1  3707  3585  3591
R4                   U    000004       1  3708  3574  3582  3596
R5                   U    000005       1  3709  3583  3624
R6                   U    000006       1  3710  3588  3597
R7                   U    000007       1  3711  3589  3625
R8                   U    000008       1  3712  3576  3578  3584  3585  3590  3591  3598  3599  3624  3625  3627  3630
R9                   U    000009       1  3713  3574  3576  3578  3627  3630
SUCCESS              H    00026A       2  3639  3628
=A(BUFFER1,DATA1,BUFFER2,DATA2,BUFFSIZE,DATASIZE)
                     A    000288       4  3665  3574
=H'1'                H    0002A0       2  3666  3617  3618
```

```
 MACRO     DEFN  REFERENCES

ANTR        136
APROB       268
ARCHIND     428   3458
ARCHLVL     569   3457
ASAIPL      695   3537
ASALOAD     775   3520
ASAREA      830
ASAZAREA   1015
CPUWAIT    1098
DSECTS     1424
DWAIT      1627   3638   3643
DWAITEND   1684   3637
ENADEV     1692
ESA390     1792
IOCB       1803
IOCBDS     1979
IOFMT      2013
IOINIT     2351
IOTRFR     2392
ORB        2440
POINTER    2629
PSWFMT     2657
RAWAIT     2791
RAWIO      2887
SIGCPU     3045
SMMGR      3103
SMMGRB     3203
TRAP128    3252
TRAP64     3229   3522   3525
TRAPS      3265
ZARCH      3339
ZEROH      3351
ZEROL      3379
ZEROLH     3407
ZEROLL     3430
```

```
   DESC     SYMBOL    SIZE        POS           ADDR

Entry: 0

Image       IMAGE    528434  00000-81031  00000-81031
  Region    CODE     528434  00000-81031  00000-81031
    CSECT   CLCL     528434  00000-81031  00000-81031
```

STMT FILE NAME

1 c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\CLCL\CLCL.asm
2 C:\Users\Fish\Documents\Visual Studio 2008\Projects\Hercules_Git_Harold\SATK-0\srcasm\satk.mac

** NO ERRORS FOUND **