```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                     2 ************************************************************************
                                                     3 *
                                                     4 *              TRTRE instruction tests
                                                     5 *
                                                     6 *        NOTE: This test is based the CLCL-et-al Test
                                                     7 *              modified to only test the Performance
                                                     8 *              of the TRTRE instruction.
                                                     9 *
                                                    10 *              The MSG routine is from the Hercules Binary
                                                    11 *              Floating Point Validation Package by Stephen R. Orso
                                                    12
                                                    13 *                        ********************
                                                    14 *                        **   IMPORTANT!   **
                                                    15 *                        ********************
                                                    16 *
                                                    17 *              This test uses the Hercules Diagnose X'008' interface
                                                    18 *              to display messages and thus your .tst runtest script
                                                    19 *              MUST contain a "DIAG8CMD ENABLE" statement within it!
                                                    20 *
                                                    21 *        James Wekel October 2022
                                                    22 ************************************************************************


                                                    24 ************************************************************************
                                                    25 *
                                                    26 *            TRTRE Performance instruction tests
                                                    27 *
                                                    28 ************************************************************************
                                                    29 *
                                                    30 *  This program ONLY tests the performance of the TRTRE
                                                    31 *  instructions.
                                                    32 *        Tests:
                                                    33 *              All tests are ' TRTRE  R2,R4,12 '
                                                    34 *              where the FC table is 128K in length,
                                                    35 *              FC is 2 bytes and an argument length of 2 bytes.
                                                    36 *
                                                    37 *              M3=12 requires page crossover tests for both FC and
                                                    38 *              the argument and has the worst performance compared to
                                                    39 *              M3=0 with the FC table and operand contained within
                                                    40 *              a page. The test should provide a lower bound on
                                                    41 *              performance improvement.
                                                    42 *
                                                    43 *              1. TRTRE of 512 bytes
                                                    44 *              2. TRTRE of 512 bytes that crosses a page boundary,
                                                    45 *                 which results in a CC=3, and a branch back
                                                    46 *                 to complete the TRTRE instruction.
                                                    47 *              3. TRTRE of 2048 bytes
                                                    48 *              4. TRTRE of 2048 bytes that crosses a page boundary,
                                                    49 *                 which results in a CC=3, and a branch back
                                                    50 *                 to complete the TRTRE instruction
                                                    51 *
                                                    52 ************************************************************************


                                                    54 ************************************************************************
                                                    55 *
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    56 *  Example Hercules Testcase:
                                                    57 *
                                                    58 *
                                                    59 *        *Testcase TRTRE-02-performance (Test TRTRE instructions)
                                                    60 *        mainsize   16
                                                    61 *        numcpu     1
                                                    62 *        sysclear
                                                    63 *        archlvl    z/Arch
                                                    64 *
                                                    65 *        loadcore   "$(testpath)/TRTRE-02-performance.core" 0x0
                                                    66 *
                                                    67 *        diag8cmd   enable  # (needed for messages to Hercules console)
                                                    68 *        #r         408=ff  # (enable timing tests)
                                                    69 *        runtest    200     # (test duration, depends on host)
                                                    70 *        diag8cmd   disable # (reset back to default)
                                                    71 *
                                                    72 *        *Done
                                                    73 *
                                                    74 *
                                                    75 ************************************************************************
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT
                                                    77 ********************************************************************
                                                    78 *                Low Core Definitions
                                                    79 ********************************************************************
                                                    80 *
                               00000000  000C3C15   81 TRTRE2TST START 0
00000000                       00000000             82          USING TRTRE2TST,R0            Low core addressability



00000000                       00000000  000001A0   84          ORG   TRTRE2TST+X'1A0'        z/Architecure RESTART PSW
000001A0  00000001 80000000                       85          DC    X'0000000180000000'
000001A8  00000000 00000200                       86          DC    AD(BEGIN)

000001B0                       000001B0  000001D0   88          ORG   TRTRE2TST+X'1D0'        z/Architecure PROGRAM CHECK PSW
000001D0  00020001 80000000                       89          DC    X'0002000180000000'
000001D8  00000000 0000DEAD                       90          DC    AD(X'DEAD')



000001E0                       000001E0  00000200   92          ORG   TRTRE2TST+X'200'        Start of actual test program...
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                      94 ************************************************************************
                                                      95 *                The actual "TRTRE2TST" program itself...
                                                      96 ************************************************************************
                                                      97 *
                                                      98 *  Architecture Mode: 370
                                                      99 *  Register Usage:
                                                     100 *
                                                     101 *   R0       (work)
                                                     102 *   R1       (work)
                                                     103 *   R2       (work) or MSG subroutine call
                                                     104 *   R3       (work)
                                                     105 *   R4       (work)
                                                     106 *   R5       TRTRETEST Base (of current test)
                                                     107 *   R5-R7    (work)
                                                     108 *   R8       (work)
                                                     109 *   R9       Second base register
                                                     110 *   R10-R12  (work)
                                                     111 *   R13      First base register
                                                     112 *   R14      Subroutine call
                                                     113 *   R15      Secondary Subroutine call or work
                                                     114 *
                                                     115 ************************************************************************

00000200                          00000200           117          USING  BEGIN,R13        FIRST Base Register
00000200                          00001200           118          USING  BEGIN+4096,R9    SECOND Base Register

00000200  05D0                                       120 BEGIN    BALR   R13,0             Initalize FIRST base register
00000202  06D0                                       121          BCTR   R13,0             Initalize FIRST base register
00000204  06D0                                       122          BCTR   R13,0             Initalize FIRST base register

00000206  4190 D800                      00000800    124          LA     R9,2048(,R13)     Initalize SECOND base register
0000020A  4190 9800                      00000800    125          LA     R9,2048(,R9)      Initalize SECOND base register

                                                     127 *
                                                     128 **       Run the performance tests...
                                                     129 *
0000020E  45E0 D328                      00000528    130          BAL    R14,TEST91        Time TRTRE   instruction  (speed test)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    132 ********************************************************************
                                                    133 *         Test for normal or unexpected test completion...
                                                    134 ********************************************************************

00000212  95FF D208                       00000408  136          CLI   TIMEOPT,X'FF'    Was this a timing run?
00000216  4770 DD80                       00000F80  137          BNE   EOJ              No, timing run; just go end normally

0000021A  95FC D200                       00000400  139          CLI   TESTNUM,X'FC'    Did we end on expected test?
0000021E  4770 DD98                       00000F98  140          BNE   FAILTEST         No?! Then FAIL the test!

00000222  9599 D201                       00000401  142          CLI   SUBTEST,X'99'    Did we end on expected SUB-test?
00000226  4770 DD98                       00000F98  143          BNE   FAILTEST         No?! Then FAIL the test!

0000022A  47F0 DD80                       00000F80  145          B     EOJ              Yes, then normal completion!




                                                    147 ********************************************************************
                                                    148 *        Fixed test storage locations ...
                                                    149 ********************************************************************



0000022E                        0000022E  00000400  151          ORG   BEGIN+X'200'
                                                    152
00000400                                            153 TESTADDR DS    0D                Where test/subtest numbers will go
00000400  99                                        154 TESTNUM  DC    X'99'      Test number of active test
00000401  99                                        155 SUBTEST  DC    X'99'      Active test sub-test number


00000408                                            157          DS    0D
00000408  00                                        158 TIMEOPT  DC    X'00'      Set to non-zero to run timing tests


00000410                                            160          DS    0D
00000410  00000000 00000000                         161 SAVE1T4  DC    4F'0'
00000420  00000000                                  162 SAVER2   DC    F'0'
00000424  00000000                                  163 SAVER5   DC    F'0'


00000428                        00000428  00000528  165          ORG   *+X'100'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    167 ********************************************************************************
                                                    168 *        TEST91                  Time TRTRE instruction  (speed test)
                                                    169 ********************************************************************************

00000528  91FF D208                         00000408   171 TEST91   TM    TIMEOPT,X'FF'    Is timing tests option enabled?
0000052C  078E                                         172          BZR   R14              No, skip timing tests

0000052E  4150 DE40                         00001040   174          LA    R5,TRTREPERF         Point R5 --> testing control table
00000532                          00000000             175          USING TRTRETEST,R5         What each table entry looks like
                                                    176 *
                                  00000532  00000001   177 TST91LOP EQU   *
00000532  5050 D224                         00000424   178          ST    R5,SAVER5          save current pref table base
                                                    179 *
00000536  4360 5000                         00000000   180          IC    R6,TNUM            Set test number
0000053A  4260 D200                         00000400   181          STC   R6,TESTNUM
                                                    182 *
                                                    183 **       Initialize operand data  (move data to testing address)
                                                    184 *
0000053E  58A0 5018                         00000018   185          L     R10,OP1WHERE         Where to move operand-1 data to
00000542  58B0 5008                         00000008   186          L     R11,OP1LEN           operand-1 length
00000546  50B0 501C                         0000001C   187          ST    R11,OP1WLEN            and save for later
0000054A  5860 5004                         00000004   188          L     R6,OP1DATA           Where op1 data is right now
0000054E  5870 5008                         00000008   189          L     R7,OP1LEN            How much of it there is
00000552  0EA6                                         190          MVCL  R10,R6
                                                    191 *
00000554  58A0 5014                         00000014   192          L     R10,OP2WHERE         Where to move operand-2 data to
00000558  58B0 5010                         00000010   193          L     R11,OP2LEN           How much of it there is
0000055C  5860 500C                         0000000C   194          L     R6,OP2DATA           Where op2 data is right now
00000560  5870 5010                         00000010   195          L     R7,OP2LEN            How much of it there is
00000564  0EA6                                         196          MVCL  R10,R6


00000566  9814 5014                         00000014   198          LM    R1,R4,OPSWHERE     get TRTRE input; set OP addr to end
0000056A  1A23                                         199          AR    R2,R3              add OP length
0000056C  0620                                         200          BCTR  R2,0               M3=12 so op addr -2
0000056E  0620                                         201          BCTR  R2,0
00000570  9014 DB90                         00000D90   202          STM   R1,R4,OPSPERF      save for preformance test
                                                    203 *
                                                    204 **       Next, time the overhead...
                                                    205 *
00000574  5870 DDB4                         00000FB4   206          L     R7,NUMLOOPS
00000578  B205 DDB8                         00000FB8   207          STCK  BEGCLOCK
0000057C  9014 D210                         00000410   208          STM   R1,R4,SAVE1T4
00000580  0560                                         209          BALR  R6,0


00000582  9814 DB90                         00000D90   211          LM    R1,R4,OPSPERF     get TRTRE operands
00000586  4710 D382                         00000582   212          BC    B'0001',*-4            not finished
0000058A  9814 DB90                         00000D90   213          LM    R1,R4,OPSPERF
0000058E  4710 D392                         00000592   214          BC    B'0001',*+4
                                                    215 *        .........ETC.........
                                                    216          PRINT OFF
                                                    411          PRINT ON
00000892  9814 DB90                         00000D90   412          LM    R1,R4,OPSPERF
00000896  4710 D69A                         0000089A   413          BC    B'0001',*+4
0000089A  9814 DB90                         00000D90   414          LM    R1,R4,OPSPERF
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

0000089E  4710 D6A2                     000008A2   415          BC    B'0001',*+4
                                                   416 *
000008A2  0676                                     417          BCTR  R7,R6
000008A4  B205 DDC0                     00000FC0   418          STCK  ENDCLOCK
000008A8  45F0 DC30                     00000E30   419          BAL   R15,CALCDUR
000008AC  D207 DDD0 DDC8      00000FD0  00000FC8   420          MVC   OVERHEAD,DURATION
                                                   421 *
                                                   422 **       Now do the actual timing run...
                                                   423 *
000008B2  5870 DDB4                     00000FB4   424          L     R7,NUMLOOPS
000008B6  B205 DDB8                     00000FB8   425          STCK  BEGCLOCK
000008BA  0560                                     426          BALR  R6,0
                                                   427 *
000008BC  9814 DB90                     00000D90   428          LM    R1,R4,OPSPERF           Load TRTRE operands
000008C0  B9BD C024                                429          TRTRE  R2,R4,12                 do TRTRE
000008C4  4710 D6C0                     000008C0   430          BC    B'0001',*-4                not finished?
000008C8  9814 DB90                     00000D90   431          LM    R1,R4,OPSPERF           Load TRTRE operands
000008CC  B9BD C024                                432          TRTRE  R2,R4,12                 do TRTRE
000008D0  4710 D6CC                     000008CC   433          BC    B'0001',*-4                not finished?
                                                   434 *        .........ETC.........
                                                   435          PRINT OFF
                                                   734          PRINT ON
00000D4C  9814 DB90                     00000D90   735          LM    R1,R4,OPSPERF
00000D50  B9BD C024                                736          TRTRE  R2,R4,12
00000D54  4710 DB50                     00000D50   737          BC    B'0001',*-4
00000D58  9814 DB90                     00000D90   738          LM    R1,R4,OPSPERF
00000D5C  B9BD C024                                739          TRTRE  R2,R4,12
00000D60  4710 DB5C                     00000D5C   740          BC    B'0001',*-4


00000D64  0676                                     742          BCTR  R7,R6
00000D66  B205 DDC0                     00000FC0   743          STCK  ENDCLOCK



00000D6A  9814 D210                     00000410   745          LM    R1,R4,SAVE1T4
00000D6E  D204 DE11 DDA8      00001011  00000FA8   746          MVC   PRTLINE+33(5),=CL5'TRTRE'
00000D74  45F0 DBB0                     00000DB0   747          BAL   R15,RPTSPEED
                                                   748 *
                                                   749 *  more performance tests?
                                                   750 *
00000D78  5850 D224                     00000424   751          L     R5,SAVER5           restore perf table base
00000D7C  4150 5034                     00000034   752          LA    R5,TRTRENEXT         Go on to next table entry
00000D80  D503 DD9C 5000      00000F9C  00000000   753          CLC   =F'0',0(R5)         End of table?
00000D86  4770 D332                     00000532   754          BNE   TST91LOP            No, loop...
00000D8A  07FE                                     755          BR    R14                 Return to caller or FAILTEST


00000D90  00000000 00000000                        757 OPSPERF  DS    4D                  Performance test R1-R4
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    759 ************************************************************************
                                                    760 *        RPTSPEED                  Report instruction speed
                                                    761 ************************************************************************

00000DB0  50F0 DC18                    00000E18    763 RPTSPEED ST    R15,RPTSAVE        Save return address
00000DB4  5050 DC1C                    00000E1C    764          ST    R5,RPTSVR5         Save R5
                                                    765 *
00000DB8  45F0 DC30                    00000E30    766          BAL   R15,CALCDUR        Calculate duration
                                                    767 *
00000DBC  4150 DDD0                    00000FD0    768          LA    R5,OVERHEAD        Subtract overhead
00000DC0  4160 DDC8                    00000FC8    769          LA    R6,DURATION        From raw timing
00000DC4  4170 DDC8                    00000FC8    770          LA    R7,DURATION        Yielding true instruction timing
00000DC8  45F0 DC84                    00000E84    771          BAL   R15,SUBDWORD       Do it
                                                    772 *
00000DCC  98AB DDC8                    00000FC8    773          LM    R10,R11,DURATION   Convert to...
00000DD0  8CA0 000C                    0000000C    774          SRDL  R10,12             ... microseconds
                                                    775 *
00000DD4  4EA0 DDD8                    00000FD8    776          CVD   R10,TICKSAAA       convert HIGH part to decimal
00000DD8  4EB0 DDE0                    00000FE0    777          CVD   R11,TICKSBBB       convert LOW  part to decimal
                                                    778 *
00000DDC  F877 DDE8 DDD8   00000FE8    00000FD8    779          ZAP   TICKSTOT,TICKSAAA           Calculate...
00000DE2  FC75 DDE8 DDAD   00000FE8    00000FAD    780          MP    TICKSTOT,=P'4294967296'     ...decimal...
00000DE8  FA77 DDE8 DDE0   00000FE8    00000FE0    781          AP    TICKSTOT,TICKSBBB           ...microseconds
                                                    782 *
00000DEE  D20B DE1B DE34   0000101B    00001034    783          MVC   PRTLINE+43(L'EDIT),EDIT         (edit into...
00000DF4  DE0B DE1B DDEB   0000101B    00000FEB    784          ED    PRTLINE+43(L'EDIT),TICKSTOT+3    ...print line)
```

```
                                                    786 *
                                                    787 *        Use Hercules Diagnose for Message to console
                                                    788 *
00000DFA  9002 DC20                    00000E20    789          STM   R0,R2,RPTDWSAV       save regs used by MSG
00000DFE  4100 0044                    00000044    790          LA    R0,PRTLNG          message length
00000E02  4110 DDF0                    00000FF0    791          LA    R1,PRTLINE         messagfe address
00000E06  4520 DCB8                    00000EB8    792          BAL   R2,MSG             call Hercules console MSG display
00000E0A  9802 DC20                    00000E20    793          LM    R0,R2,RPTDWSAV     restore regs

00000E0E  5850 DC1C                    00000E1C    795          L     R5,RPTSVR5         Restore R5
00000E12  58F0 DC18                    00000E18    796          L     R15,RPTSAVE        Restore return address
00000E16  07FF                                     797          BR    R15                Return to caller

00000E18  00000000                                 799 RPTSAVE  DC    F'0'               R15 save area
00000E1C  00000000                                 800 RPTSVR5  DC    F'0'               R5 save area

00000E20  00000000 00000000                        802 RPTDWSAV DC    2D'0'              R0-R2 save area for MSG call
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    804 ********************************************************************************
                                                    805 *        CALCDUR                  Calculate DURATION
                                                    806 ********************************************************************************

00000E30  50F0 DC74                    00000E74    808 CALCDUR  ST    R15,CALCRET        Save return address
00000E34  9057 DC78                    00000E78    809          STM   R5,R7,CALCWORK     Save work registers
                                                    810 *
00000E38  9867 DDB8                    00000FB8    811          LM    R6,R7,BEGCLOCK     Remove CPU number from clock value
00000E3C  8C60 0006                    00000006    812          SRDL  R6,6                       "
00000E40  8D60 0006                    00000006    813          SLDL  R6,6                       "
00000E44  9067 DDB8                    00000FB8    814          STM   R6,R7,BEGCLOCK             "
                                                    815 *
00000E48  9867 DDC0                    00000FC0    816          LM    R6,R7,ENDCLOCK     Remove CPU number from clock value
00000E4C  8C60 0006                    00000006    817          SRDL  R6,6                       "
00000E50  8D60 0006                    00000006    818          SLDL  R6,6                       "
00000E54  9067 DDC0                    00000FC0    819          STM   R6,R7,ENDCLOCK             "
                                                    820 *
00000E58  4150 DDB8                    00000FB8    821          LA    R5,BEGCLOCK        Starting time
00000E5C  4160 DDC0                    00000FC0    822          LA    R6,ENDCLOCK        Ending time
00000E60  4170 DDC8                    00000FC8    823          LA    R7,DURATION        Difference
00000E64  45F0 DC84                    00000E84    824          BAL   R15,SUBDWORD       Calculate duration
                                                    825 *
00000E68  9857 DC78                    00000E78    826          LM    R5,R7,CALCWORK     Restore work registers
00000E6C  58F0 DC74                    00000E74    827          L     R15,CALCRET        Restore return address
00000E70  07FF                                     828          BR    R15                Return to caller

00000E74  00000000                                 830 CALCRET  DC    F'0'               R15 save area
00000E78  00000000 00000000                        831 CALCWORK DC    3F'0'              R5-R7 save area



                                                    833 ********************************************************************************
                                                    834 *        SUBDWORD                 Subtract two doublewords
                                                    835 *        R5 --> subtrahend, R6 --> minuend, R7 --> result
                                                    836 ********************************************************************************

00000E84  9014 DCA8                    00000EA8    838 SUBDWORD STM   R1,R4,SUBDWSAV   Save registers
                                                    839 *
00000E88  9812 5000                    00000000    840          LM    R1,R2,0(R5)        Subtrahend  (value to subtract)
00000E8C  9834 6000                    00000000    841          LM    R3,R4,0(R6)        Minuend     (what to subtract FROM)
00000E90  1F42                                     842          SLR   R4,R2              Subtract LOW part
00000E92  47B0 DC9A                    00000E9A    843          BNM   *+4+4                (branch if no borrow)
00000E96  5F30 DDA0                    00000FA0    844          SL    R3,=F'1'           (otherwise do borrow)
00000E9A  1F31                                     845          SLR   R3,R1              Subtract HIGH part
00000E9C  9034 7000                    00000000    846          STM   R3,R4,0(R7)        Store results
                                                    847 *
00000EA0  9814 DCA8                    00000EA8    848          LM    R1,R4,SUBDWSAV     Restore registers
00000EA4  07FF                                     849          BR    R15                Return to caller

00000EA8  00000000 00000000                        851 SUBDWSAV DC    2D'0'              R1-R4 save area
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    853 ********************************************************************
                                                    854 *        Issue HERCULES MESSAGE pointed to by R1, length in R0
                                                    855 *              R2 = return address
                                                    856 ********************************************************************

00000EB8  4900 DDA4                  00000FA4   858 MSG      CH    R0,=H'0'                Do we even HAVE a message?
00000EBC  07D2                                  859          BNHR  R2                      No, ignore

00000EBE  9002 DCF0                  00000EF0   861          STM   R0,R2,MSGSAVE           Save registers

00000EC2  4900 DDA6                  00000FA6   863          CH    R0,=AL2(L'MSGMSG)       Message length within limits?
00000EC6  47D0 DCCE                  00000ECE   864          BNH   MSGOK                   Yes, continue
00000ECA  4100 005F                  0000005F   865          LA    R0,L'MSGMSG             No, set to maximum

00000ECE  1820                                  867 MSGOK    LR    R2,R0                   Copy length to work register
00000ED0  0620                                  868          BCTR  R2,0                    Minus-1 for execute
00000ED2  4420 DCFC                  00000EFC   869          EX    R2,MSGMVC               Copy message to O/P buffer

00000ED6  4120 200A                  0000000A   871          LA    R2,1+L'MSGCMD(,R2)      Calculate true command length
00000EDA  4110 DD02                  00000F02   872          LA    R1,MSGCMD               Point to true command

00000EDE  83120008                              874          DC    X'83',X'12',X'0008'     Issue Hercules Diagnose X'008'
00000EE2  4780 DCE8                  00000EE8   875          BZ    MSGRET                  Return if successful
00000EE6  0000                                  876          DC    H'0'                    CRASH for debugging purposes

00000EE8  9802 DCF0                  00000EF0   878 MSGRET   LM    R0,R2,MSGSAVE           Restore registers
00000EEC  07F2                                  879          BR    R2                      Return to caller



00000EF0  00000000 00000000                     881 MSGSAVE  DC    3F'0'                   Registers save area
00000EFC  D200 DD0B 1000       00000F0B  00000000   882 MSGMVC   MVC   MSGMSG(0),0(R1)       Executed instruction


00000F02  D4E2C7D5 D6C8405C                     884 MSGCMD   DC    C'MSGNOH * '            *** HERCULES MESSAGE COMMAND ***
00000F0B  40404040 40404040                     885 MSGMSG   DC    CL95' '                 The message text to be displayed
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    887 ************************************************************************
                                                    888 *        Normal completion or Abnormal termination PSWs
                                                    889 ************************************************************************


00000F70  00020001 80000000                         891 EOJPSW   DC    0D'0',X'0002000180000000',AD(0)

00000F80  B2B2 DD70                  00000F70       893 EOJ      LPSWE EOJPSW               Normal completion


00000F88  00020001 80000000                         895 FAILPSW  DC    0D'0',X'0002000180000000',AD(X'BAD')

00000F98  B2B2 DD88                  00000F88       897 FAILTEST LPSWE FAILPSW              Abnormal termination
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                    899 ********************************************************************************
                                                    900 *        Working Storage
                                                    901 ********************************************************************************



00000F9C                                            903          LTORG ,                 Literals pool
00000F9C  00000000                                  904                =F'0'
00000FA0  00000001                                  905                =F'1'
00000FA4  0000                                      906                =H'0'
00000FA6  005F                                      907                =AL2(L'MSGMSG)
00000FA8  E3D9E3D9 C5                               908                =CL5'TRTRE'
00000FAD  04294967 296C                             909                =P'4294967296'

                          00000400  00000001        911 K        EQU   1024            One KB
                          00001000  00000001        912 PAGE     EQU   (4*K)           Size of one page
                          00010000  00000001        913 K64      EQU   (64*K)          64 KB
                          00100000  00000001        914 MB       EQU   (K*K)            1 MB

00000FB4  00002710                                  916 NUMLOOPS DC    F'10000'         10,000 * 100 = 1,000,000

00000FB8  BBBBBBBB BBBBBBBB                         918 BEGCLOCK DC    0D'0',8X'BB'     Begin
00000FC0  EEEEEEEE EEEEEEEE                         919 ENDCLOCK DC    0D'0',8X'EE'     End
00000FC8  DDDDDDDD DDDDDDDD                         920 DURATION DC    0D'0',8X'DD'     Diff
00000FD0  FFFFFFFF FFFFFFFF                         921 OVERHEAD DC    0D'0',8X'FF'     Overhead

00000FD8  00000000 0000000C                         923 TICKSAAA DC    PL8'0'           Clock ticks high part
00000FE0  00000000 0000000C                         924 TICKSBBB DC    PL8'0'           Clock ticks low part
00000FE8  00000000 0000000C                         925 TICKSTOT DC    PL8'0'           Total clock ticks

00000FF0  40404040 40404040                         927 PRTLINE  DC    C'         1,000,000 iterations of XXXXX'
00001016  40A39696 9240F9F9                         928          DC    C' took 999,999,999 microseconds'
                          00000044  00000001        929 PRTLNG   EQU   *-PRTLINE
00001034  40202020 6B202020                         930 EDIT     DC    X'402020206B2020206B202120'
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                                     932 ********************************************************************************
                                                     933 *        TRTRETEST DSECT
                                                     934 ********************************************************************************



                                                     936 TRTRETEST DSECT ,
00000000  00                                         937 TNUM     DC    X'00'          TRTRE table Number
00000001  00                                         938          DC    X'00'
00000002  00                                         939          DC    X'00'
00000003  00                                         940 M3       DC    X'00'          M3 byte stored into TRTRE instruction



00000004  00000000                                   942 OP1DATA  DC    A(0)           Pointer to Operand-1 data
00000008  00000000                                   943 OP1LEN   DC    F'0'           How much data is there - 1
0000000C  00000000                                   944 OP2DATA  DC    A(0)           Pointer to FC table data
00000010  00000000                                   945 OP2LEN   DC    F'0'           How much data is there - FC Table


                          00000014  00000001         947 OPSWHERE EQU   *
00000014  00000000                                   948 OP2WHERE DC    A(0)           Where FC Table  data should be placed
00000018  00000000                                   949 OP1WHERE DC    A(0)           Where Operand-1 data should be placed
0000001C  00000000                                   950 OP1WLEN  DC    F'0'           How much data is there - 1
00000020  00000000                                   951          DC    A(0)           pollute - found FC


00000024  00000000                                   953 FAILMASK DC    A(0)           Failure Branch on Condition mask


                                                     955 *                           Ending register values
00000028  00000000                                   956 ENDREGS  DC    A(0)              Operand 1 address
0000002C  00000000                                   957          DC    A(0)              Operand 1 length
00000030  00000000                                   958          DC    A(0)              Function Code


                        00000034  00000001         960 TRTRENEXT EQU   *             Start of next table entry...



                        AABBCCDD  00000001         962 REG2PATT EQU   X'AABBCCDD'    Polluted Register pattern
                        000000DD  00000001         963 REG2LOW  EQU         X'DD'    (last byte above)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                               00000000  000C3C15   965 TRTRE2TST CSECT ,
                                                    966 ********************************************************************
                                                    967 *        TRTRE Performace Test data...
                                                    968 ********************************************************************
00001040                                            969 TRTREPERF DC    0A(0)      start of table




                                                    971 ********************************************************************
                                                    972 *        Check performance tests are valid.
                                                    973 *        tests with   M3: A=1,F=1,L=0, reserved=0    (12)
                                                    974 *                     FC Table : SIZE: 131,072 (2 BYTE ARGUMENT)
                                                    975 *                                Function Code is 2 bytes
                                                    976 *
                                                    977 *              Note: Op1 length must be a multiple of 2
                                                    978 ********************************************************************


00001040                                            980 F12T8    DS    0F
00001040  F8                                        981          DC    X'F8'                        Test Num
00001041  0000                                      982          DC    X'00',X'00'
00001043  C0                                        983          DC    X'C0'                        M3: A=1,F=1,L=0,---=0
00001044  00001418 00000200                        984          DC    A(TRTOP1F1),A(512)           Source - Op 1 & length
0000104C  000A3A16 00020000                        985          DC    A(TRTOPCF1),A(2*K64)         Source - FC Table & length
                                                    986 *                                         Target -
00001054  00710000 00910000                        987          DC    A(7*MB+(1*K64)),A(9*MB+(1*K64)),A(0)  FC, Op1, Op1L
00001060  AABBCCDD                                  988          DC    A(REG2PATT)
00001064  0000000B                                  989          DC    A(11) CC1
00001068  00910000 00000002                        990          DC    A(9*MB+(1*K64)),A(2),XL4'F1'



00001074                                            992 F12T8A   DS    0F
00001074  F9                                        993          DC    X'F9'                        Test Num
00001075  0000                                      994          DC    X'00',X'00'
00001077  C0                                        995          DC    X'C0'                        M3: A=1,F=1,L=0,---=0
00001078  00001418 00000200                        996          DC    A(TRTOP1F1),A(512)           Source - Op 1 & length
00001080  000A3A16 00020000                        997          DC    A(TRTOPCF1),A(2*K64)         Source - FC Table & length
                                                    998 *                                         Target - FC, Op1, Op1L
00001088  0072FF81 0092FF81                        999          DC    A(7*MB+(3*K64)-127),A(9*MB+(3*K64)-127),A(0)
00001094  AABBCCDD                                 1000          DC    A(REG2PATT)
00001098  0000000A                                 1001          DC    A(10) CC1 or CC3
0000109C  0092FF81 00000002                       1002          DC    A(9*MB+(3*K64)-127),A(2),XL4'F1'



000010A8                                           1004 F12T11  DS    0F
000010A8  FB                                       1005          DC    X'FB'                        Test Num
000010A9  0000                                     1006          DC    X'00',X'00'
000010AB  C0                                       1007          DC    X'C0'                        M3: A=1,F=1,L=0,---=0
000010AC  00002618 00000800                       1008          DC    A(TRTO1LF0),A(2048)          Source - Op 1 & length
000010B4  00083818 00020000                       1009          DC    A(TRTOPCF0),A(2*K64)         Source - FC Table & length
                                                   1010 *                                         Target -
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2     STMT

000010BC  00760000 00960000                        1011          DC    A(7*MB+(6*K64)),A(9*MB+(6*K64)),A(0) FC, Op1, Op1L
000010C8  AABBCCDD                                 1012          DC    A(REG2PATT)
000010CC  0000000B                                 1013          DC    A(11) CC1
000010D0  00960000 00000002                        1014          DC    A(9*MB+(6*K64)),A(2),XL4'F0'



000010DC                                           1016 F12T11A  DS    0F
000010DC  FC                                       1017          DC    X'FC'                        Test Num
000010DD  0000                                     1018          DC    X'00',X'00'
000010DF  C0                                       1019          DC    X'C0'                        M3: A=1,F=1,L=0,--=0
000010E0  00002618 00000800                        1020          DC    A(TRTO1LF0),A(2048)          Source - Op 1 & length
000010E8  00083818 00020000                        1021          DC    A(TRTOPCF0),A(2*K64)         Source - FC Table & length
                                                   1022 *                                           Target - FC, Op1, Op1L
000010F0  0078FE1F 0098FE1F                        1023          DC    A(7*MB+(9*K64)-481),A(9*MB+(9*K64)-481),A(0)
000010FC  AABBCCDD                                 1024          DC    A(REG2PATT)
00001100  0000000A                                 1025          DC    A(10) CC1 or CC3
00001104  0098FE1F 00000002                        1026          DC    A(9*MB+(9*K64)-481),A(2),XL4'F0'



00001110  00000000                                 1028          DC    A(0)        end of table
00001114  00000000                                 1029          DC    A(0)        end of table
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                   1031 ************************************************************************
                                                   1032 *        TRTRE op1 scan data...
                                                   1033 ************************************************************************

00001118  78125634 78125634                        1035 TRTOP10  DC    64XL4'78125634'    (CC0)

00001218  78125634 78125634                        1037 TRTOP111 DC    59XL4'78125634',X'00110000',04XL4'78125634'    (CC1)

00001318  00F00000 78125634                        1039 TRTOP1F0 DC    X'00F00000',63XL4'78125634'    (CC1)

00001418  00F10000 78125634                        1041 TRTOP1F1 DC    X'00F10000',127XL4'78125634'   (CC1)

00001618  98765432 98765432                        1043 TRTO1L0  DC    512XL4'98765432'    (CC0)

00001E18  98765432 98765432                        1045 TRTO1L11 DC    256XL4'98765432',X'00110000',255XL4'98765432'    (CC1)

00002618  00F00000 98765432                        1047 TRTO1LF0 DC    XL4'00F00000',511XL4'98765432'    (CC1)
```

```
  LOC        OBJECT CODE       ADDR1     ADDR2    STMT

                                                  1049 ********************************************************************
                                                  1050 *        Function Code (FC) Tables (GR1)
                                                  1051 ********************************************************************

00002E18  00000000 00000000                       1053 TRTOP20  DC    256X'00'            no stop
00002F18                      00002F18  00022F18  1054          ORG   *+2*K64

00022F18  00000000 00000000                       1056 TRTOP211 DC    17X'00',X'11',238X'00'       stop on X'11'

00023018  00000000 00000000                       1058 TRTOP2F0 DC    240X'00',X'F0',15X'00'       stop on X'F0'

00023118  00000000 00000000                       1060 TRTOP411 DC    34X'00',X'0011',476X'00'     stop on X'11'

00023318  00000000 00000000                       1062 TRTOP4F0 DC    480X'00',X'00F0',30X'00'     stop on X'F0'

00023518  00000000 00000000                       1064 TRTOP811 DC    17X'00',X'11',238X'00'       stop on X'11'
00023618                      00023618  00043618  1065          ORG   *+2*K64

00043618  00000000 00000000                       1067 TRTOP8F0 DC    240X'00',X'F0',15X'00'       stop on X'F0'
00043718                      00043718  00063718  1068          ORG   *+2*K64

00063718  00000000 00000000                       1070 TRTOP8F1 DC    240X'00',X'00',X'F1',14X'00' stop on X'F1'
00063818                      00063818  00083818  1071          ORG   *+2*K64

00083818  00000000 00000000                       1073 TRTOPCF0 DC    480X'00',X'00F0',28X'00'     stop on X'F0'
00083A16                      00083A16  000A3A16  1074          ORG   *+2*K64

                                                  1076 *                                           stop on X'F1'
000A3A16  00000000 00000000                       1077 TRTOPCF1 DC    480X'00',X'0000',X'00F1',28X'00'
000A3C16                      000A3C16  000C3C16  1078          ORG   *+2*K64
```

```
 LOC        OBJECT CODE       ADDR1     ADDR2     STMT

                                         1080 ************************************************************************
                                         1081 *        Register equates
                                         1082 ************************************************************************


                             00000000  00000001  1084 R0       EQU   0
                             00000001  00000001  1085 R1       EQU   1
                             00000002  00000001  1086 R2       EQU   2
                             00000003  00000001  1087 R3       EQU   3
                             00000004  00000001  1088 R4       EQU   4
                             00000005  00000001  1089 R5       EQU   5
                             00000006  00000001  1090 R6       EQU   6
                             00000007  00000001  1091 R7       EQU   7
                             00000008  00000001  1092 R8       EQU   8
                             00000009  00000001  1093 R9       EQU   9
                             0000000A  00000001  1094 R10      EQU   10
                             0000000B  00000001  1095 R11      EQU   11
                             0000000C  00000001  1096 R12      EQU   12
                             0000000D  00000001  1097 R13      EQU   13
                             0000000E  00000001  1098 R14      EQU   14
                             0000000F  00000001  1099 R15      EQU   15


                                         1101          END
```

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
BEGCLOCK	D	00000FB8	8	918	207 425 811 814 821
BEGIN	I	00000200	2	120	151 86 117 118
CALCDUR	I	00000E30	4	808	419 766
CALCRET	F	00000E74	4	830	808 827
CALCWORK	F	00000E78	4	831	809 826
DURATION	D	00000FC8	8	920	420 769 770 773 823
EDIT	X	00001034	12	930	783 784
ENDCLOCK	D	00000FC0	8	919	418 743 816 819 822
ENDREGS	A	00000028	4	956	
EOJ	I	00000F80	4	893	137 145
EOJPSW	D	00000F70	8	891	893
F12T11	F	000010A8	4	1004	
F12T11A	F	000010DC	4	1016	
F12T8	F	00001040	4	980	
F12T8A	F	00001074	4	992	
FAILMASK	A	00000024	4	953	
FAILPSW	D	00000F88	8	895	897
FAILTEST	I	00000F98	4	897	140 143
IMAGE	1	00000000	801814	0	
K	U	00000400	1	911	912 913 914
K64	U	00010000	1	913	1054 1065 1068 1071 1074 1078 985 987 990 997 999 1002 1009 1011 1014 1021 1023 1026
M3	X	00000003	1	940	
MB	U	00100000	1	914	987 990 999 1002 1011 1014 1023 1026
MSG	I	00000EB8	4	858	792
MSGCMD	C	00000F02	9	884	871 872
MSGMSG	C	00000F0B	95	885	865 882 863
MSGMVC	I	00000EFC	6	882	869
MSGOK	I	00000ECE	2	867	864
MSGRET	I	00000EE8	4	878	875
MSGSAVE	F	00000EF0	4	881	861 878
NUMLOOPS	F	00000FB4	4	916	206 424
OP1DATA	A	00000004	4	942	188
OP1LEN	F	00000008	4	943	186 189
OP1WHERE	A	00000018	4	949	185
OP1WLEN	F	0000001C	4	950	187
OP2DATA	A	0000000C	4	944	194
OP2LEN	F	00000010	4	945	193 195
OP2WHERE	A	00000014	4	948	192
OPSPERF	D	00000D90	8	757	202 211 213 218 220 222 224 226 228 230 232 234 236 238 240 242 244 246 248 250 252 254 256 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292 294 296 298 300 302 304 306 308 310 312 314 316 318 320 322 324 326 328 330 332 334 336 338 340 342 344 346 348 350 352 354 356 358 360 362 364 366 368 370 372 374 376 378 380 382 384 386 388 390 392 394 396 398 400 402 404 406 408 412 414 428 431 438 441 444 447 450 453 456 459 462 465 469 472 475 478 481 484 487 490 493 496 500 503 506 509 512 515 518 521 524 527 531 534 537 540 543 546 549 552 555 558 562 565 568 571 574 577 580 583 586 589 593 596 599 602 605 608 611 614 617 620 624 627 630 633 636 639 642 645 648 651 655 658 661 664 667 670 673 676 679 682 684 687 690 693 696 699 702 705 708 711 715 718 721 724 727 730 735 738
OPSWHERE	U	00000014	1	947	198

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
OVERHEAD	D	00000FD0	8	921	420 768
PAGE	U	00001000	1	912	
PRTLINE	C	00000FF0	38	927	929 746 783 784 791
PRTLNG	U	00000044	1	929	790
R0	U	00000000	1	1084	82 789 790 793 858 861 863 865 867 878
R1	U	00000001	1	1085	198 202 208 211 213 218 220 222 224 226 228 230 232 234 236 238 240 242 244 246 248 250 252 254 256 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292 294 296 298 300 302 304 306 308 310 312 314 316 318 320 322 324 326 328 330 332 334 336 338 340 342 344 346 348 350 352 354 356 358 360 362 364 366 368 370 372 374 376 378 380 382 384 386 388 390 392 394 396 398 400 402 404 406 408 412 414 428 431 438 441 444 447 450 453 456 459 462 465 469 472 475 478 481 484 487 490 493 496 500 503 506 509 512 515 518 521 524 527 531 534 537 540 543 546 549 552 555 558 562 565 568 571 574 577 580 583 586 589 593 596 599 602 605 608 611 614 617 620 624 627 630 633 636 639 642 645 648 651 655 658 661 664 667 670 673 676 679 682 684 687 690 693 696 699 702 705 708 711 715 718 721 724 727 730 735 738 745 791 838 840 845 848 872 882
R10	U	0000000A	1	1094	185 190 192 196 773 774 776
R11	U	0000000B	1	1095	186 187 193 773 777
R12	U	0000000C	1	1096	
R13	U	0000000D	1	1097	117 120 121 122 124
R14	U	0000000E	1	1098	130 172 755
R15	U	0000000F	1	1099	419 747 763 766 771 796 797 808 824 827 828 849
R2	U	00000002	1	1086	199 200 201 429 432 439 442 445 448 451 454 457 460 463 466 470 473 476 479 482 485 488 491 494 497 501 504 507 510 513 516 519 522 525 528 532 535 538 541 544 547 550 553 556 559 563 566 569 572 575 578 581 584 587 590 594 597 600 603 606 609 612 615 618 621 625 628 631 634 637 640 643 646 649 652 656 659 662 665 668 671 674 677 680 685 688 691 694 697 700 703 706 709 713 716 719 722 725 728 731 736 739 789 792 793 840 842 859 861 867 868 869 871 878 879
R3	U	00000003	1	1087	199 841 844 845 846
R4	U	00000004	1	1088	198 202 208 211 213 218 220 222 224 226 228 230 232 234 236 238 240 242 244 246 248 250 252 254 256 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292 294 296 298 300 302 304 306 308 310 312 314 316 318 320 322 324 326 328 330 332 334 336 338 340 342 344 346 348 350 352 354 356 358 360 362 364 366 368 370 372 374 376 378 380 382 384 386 388 390 392 394 396 398 400 402 404 406 408 412 414 428 429 431 432 438 439 441 442 444 445 447 448 450 451 453 454 456 457 459 460 462 463 465 466 469 470 472 473 475 476 478 479 481 482 484 485 487 488 490 491 493 494 496 497 500 501 503 504 506 507 509 510 512 513 515 516 518 519 521 522 524 525 527 528 531 532 534 535 537 538 540 541 543 544 546 547 549 550 552 553 555 556 558 559 562 563 565 566 568 569 571 572 574 575 577 578 580 581 583 584 586 587 589 590 593 594 596 597 599 600 602 603 605 606 608 609 611 612 614 615 617 618 620 621 624 625 627 628 630 631 633

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
					634 636 637 639 640 642 643 645 646 648 649 651 652 655 656 658 659 661 662 664 665 667 668 670 671 673 674 676 677 679 680 682 684 685 687 688 690 691 693 694 696 697 699 700 702 703 705 706 708 709 711 713 715 716 718 719 721 722 724 725 727 728 730 731 735 736 738 739 745 838 841 842 846 848
R5	U	00000005	1	1089	174 175 178 751 752 753 764 768 795 809 821 826 840
R6	U	00000006	1	1090	180 181 188 190 194 196 209 417 426 742 769 811 812 813 814 816 817 818 819 822 841
R7	U	00000007	1	1091	189 195 206 417 424 742 770 809 811 814 816 819 823 826 846
R8	U	00000008	1	1092	
R9	U	00000009	1	1093	118 124 125
REG2LOW	U	000000DD	1	963	
REG2PATT	U	AABBCCDD	1	962	988 1000 1012 1024
RPTDWSAV	D	00000E20	8	802	789 793
RPTSAVE	F	00000E18	4	799	763 796
RPTSPEED	I	00000DB0	4	763	747
RPTSVR5	F	00000E1C	4	800	764 795
SAVE1T4	F	00000410	4	161	208 745
SAVER2	F	00000420	4	162	
SAVER5	F	00000424	4	163	178 751
SUBDWORD	I	00000E84	4	838	771 824
SUBDWSAV	D	00000EA8	8	851	838 848
SUBTEST	X	00000401	1	155	142
TEST91	I	00000528	4	171	130
TESTADDR	D	00000400	8	153	
TESTNUM	X	00000400	1	154	139 181
TICKSAAA	P	00000FD8	8	923	776 779
TICKSBBB	P	00000FE0	8	924	777 781
TICKSTOT	P	00000FE8	8	925	779 780 781 784
TIMEOPT	X	00000408	1	158	136 171
TNUM	X	00000000	1	937	180
TRTO1L0	X	00001618	4	1043	
TRTO1L11	X	00001E18	4	1045	
TRTO1LF0	X	00002618	4	1047	1008 1020
TRTOP10	X	00001118	4	1035	
TRTOP111	X	00001218	4	1037	
TRTOP1F0	X	00001318	4	1039	
TRTOP1F1	X	00001418	4	1041	984 996
TRTOP20	X	00002E18	1	1053	
TRTOP211	X	00022F18	1	1056	
TRTOP2F0	X	00023018	1	1058	
TRTOP411	X	00023118	1	1060	
TRTOP4F0	X	00023318	1	1062	
TRTOP811	X	00023518	1	1064	
TRTOP8F0	X	00043618	1	1067	
TRTOP8F1	X	00063718	1	1070	
TRTOPCF0	X	00083818	1	1073	1009 1021
TRTOPCF1	X	000A3A16	1	1077	985 997
TRTRE2TST	J	00000000	801814	81	84 88 92 82
TRTRENEXT	U	00000034	1	960	752
TRTREPERF	A	00001040	4	969	174
TRTRETEST	4	00000000	52	936	175
TST91LOP	U	00000532	1	177	754
=AL2(L'MSGMSG)	R	00000FA6	2	907	863

```
     SYMBOL         TYPE    VALUE      LENGTH    DEFN  REFERENCES

=CL5'TRTRE'          C     00000FA8          5   908   746
=F'0'                F     00000F9C          4   904   753
=F'1'                F     00000FA0          4   905   844
=H'0'                H     00000FA4          2   906   858
=P'4294967296'       P     00000FAD          6   909   780
```

MACRO DEFN REFERENCES

No defined macros

```
   DESC       SYMBOL     SIZE        POS           ADDR

Entry: 0

Image       IMAGE       801814  00000-C3C15  00000-C3C15
  Region                801814  00000-C3C15  00000-C3C15
    CSECT   TRTRE2TST   801814  00000-C3C15  00000-C3C15
```

```
   STMT                    FILE NAME
1     /devstor/dev/tests/TRTRE-02-performance.asm

** NO ERRORS FOUND **
```