IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DCQKA-B-D
PRODUCT NAME:	MUL - DIV RANDOM EXERCISER
DATE CREATED:	NOVEMBER 20, 1973
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	BOB BRAIN

CONTENTS

1. Abstract

This test is to be used as an exerciser for the PDP11/45 MUL and DIV instructions. It tests the divide and multiply with random numbers.

2. Requirements

2.1 Equipment

PDP11/45 standard computer

2.2 Storage

Program Storage - the routines use memory 0 - 17776

2.3 Preliminary programs

MAINDEC-11-DCKBI and DCKBJ

3. Loading procedure

Use standard procedure for ABS tapes.

4. Starting Procedure

4.1 Control switch settings

See 5.1.1 (all down for worst case testing)

4.2 Starting address

The program should always be started at 200.

4.3 Program and/or operator action

1) Load program into memory using ABS loader.
2) Load address 200.
3) Set switches (see sec 5.1.1) All down for worst case
4) Press start.
5) The program will loop and bell will ring once every pass
6) A minimum of two passes should always be run.
7) The display on the 11/45 will show the pass count. Set the data display switch to the DISPLAY position.

5.0 Operating procedure

5.1 Operational switch settings

At SA 200 ,, all switches down will test all functions of MUL and DIV and print out on errors and continue in test. (bell will ring at completion of 512 passes)

5.1.1 Switch settings are:

```
SW15 = 1 .....  HALT ON ERROR
SW14 = 1 .....  SCOPE LOOP
SW13 = 1 .....  INHIBIT PRINTOUT
SW12 = 1 .....  INHIBIT TRACE TRAPPING
SW11 = 1 .....  INHIBIT ITERATION LOOP
SW10 = 1 .....  BELL ON ERROR
       0 .....  BELL ON 512 PASSES
```

5.2 Subroutine Abstracts

5.2.1 SCOPE

This subroutine call is placed between each subtest in the instruction section. It records the starting address of each subtest as it is being entered in location "LAD". If a scope loop is requested, the current subtest will be looped upon. SW<11> on a 1 inhibits iteration of subtests. The contents of LAD may be used to determine the last subtest successfully completed.

5.2.2 HLT

This routine prints out an error message (See 6.1.) To inhibit typeouts, put SW<13> on a 1.

5.2.3 TRTRAP

If SW<12> is on a 0, the T bit will be set on alternate passes. When set, it causes a trap after each instruction. The first instruction executed upon trapping is an "RTT" which returns to the interrupted sequence of instructions. This sequence is continued until the end of the program is reached.

5.2.4 TRAPCATCHER

A "CONTINUE" & "HALT" sequence is repeated from 0 - 776 to catch any unexpected traps. Thus any unexpected traps or interupts will HALT at the vector + 2.

6. Errors

Loading and starting at 200 with all switches down is worst case testing. When an error is detected and it is necessary to scope on it, place SW15 up to halt on error, Hit continue with SW14 up to loop on test, and SW13 up to delete printouts.

6.1 Error printout

The format is as follows:

```
DIV ERROR:   040000 000000 / 100000

SOFTWARE  000000  100000  PS=22
HARDWARE  000000  100000  PS=122

ERRORS OCCURED 75 TIMES OUT OF 400
```

The line starting with 'DIV ERROR' is the operation being performed when the failure was detected. The 'SOFTWARE' line shows the calculated result and the 'HARDWARE' line is the result by the hardware. The last line reflects the number of errors out of the number of tries. This line will only be typed if the iterations are on.

6.2 Error recovery

Restart at 200

7. Restrictions

None

8. Miscellaneous

8.1 Execution time

A bell will ring within 1 minute with all switches down.

6.2 Stack Pointer

Stack is initially set to 500

6.3 Power Fail

To use, start the test as usual and power down then up at any time. There should be no error typeouts.

7. Program Description

This program is an exerciser of the multiply and divide instructions. It first generates 2 random numbers and multiplies them together. It saves this answer and divides it by the multiplier and the multiplicand saving both answers. The original numbers are hardware multiplied both ways and compared to the original answer. The two divisions are performed and compared to the original software answers. One more random number is generated and this number is divided into the original two numbers and the answer checked against the hardware answers. Each hardware multiply and divide is done 256 times. If SW11 is up, they will only be done once. SW14 will cause the test that you are in to be looped upon.

TABLE OF CONTENTS


```
                                ;TITLE  MAINDEC-11-DCQKA-B  MUL & DIV RANDOM EXERCISER
                                ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD,MASS.
                                ;PROGRAM BY BOB BRAIN
                                ;ENABL  ABS
                                ;SWITCH SETTINGS

                                ;       SWITCH                  USE
                                ;       ******          ********************
                                ;         10            0 = BELL ON PASS COMPLETE
                                ;                       1 = BELL ON ERROR
                                ;         11            INHIBIT ITERATIONS
                                ;         12            INHIBIT TRACE TRAP
                                ;         13            INHIBIT ERROR TYPEOUTS
                                ;         14            LOOP ON TEST
                                ;         15            HALT ON ERROR


        000000                          .=0                     ;TRAP CATCHER AT 0 = 776
        000014                          .=14
000014  003612                          .TRT
000016  000000                          0
000020  003662                          .IOT
000022  000340                          340
        000030                  .=30
000030  002160                          .EMT
000032  000000                          0
000034  002344                          .TRP
000036  000000                          0
        000200                          .=200
000200  000167  000270                  JMP     BEGIN

        000004                  TYPE=   IOT
        040000                  SWI4=   40000
        020000                  SWI3=   20000
        010000                  SWI2=   10000
        004000                  SWI1=    4000
        002000                  SWI0=    2000
        000001                  N=      1
        177570                  SW=     177570
        177570                  DISPLAY=SWR
        104000                  SCOPE=  EMT
        104400                  HLT=    TRAP
        000000                  R0=     %0
        000000                  .AC=    R0
        000001                  .MQ=    %1
        000002                  R2=     %2
        000003                  TT=     %3
        000004                  R4=     %4
        000005                  R5=     %5
        000006                  SP=     %6
        000007                  PC=     %7
        000007                  BELL=   7
```

```
                 001000                                .=1000

        001000   000000                    ICNT:   0
        001002   012706  000500            BEGIN:  MOV     #500,SP          ;SET PUSH DOWN LIST POINTER
        001006   005067  002622            1$:     CLR     ERRORS           ;ERROR COUNT
        001012   012767  000001  002604            MOV     #1,CNT
        001020   012767  123456  002572            MOV     #123456,LONUM    ;INITALIZE LONUM FOR RANDOM
        001026   012767  176543  002566            MOV     #176543,HINUM    ;INITALIZE HINUM FOR RANDOM

        001034   012704  003640            GO:     MOV     #AREA,R4         ;SET POINTER TO ANSWER AREA
        001040   004767  002326                    JSR     PC,RANDOM        ;GET RANDOM NUMBERS
        001044   016701  002550                    MOV     LONUM,.MQ        ;LOAD MULTIPLIER (SOFTWARE)
        001050   016702  002546                    MOV     HINUM,R2         ;LOAD MULTIPLICAN (SOFTWARE)
        001054   004767  003540                    JSR     PC,MUL1          ;MULTIPLY
        001060   010124                            MOV     .MQ,(4)+         ;SAVE MQ
        001062   010024                            MOV     .AC,(4)+         ;SAVE AC
        001064   016724  003024                    MOV     SR1,(4)+         ;SAVE SR

        001070   016702  002524                    MOV     LONUM,R2         ;LOAD DIVISOR
        001074   004767  003020                    JSR     PC,DIVI          ;DIVIDE BY MULTIPLIER (SOFTWARE)
        001100   010024                            MOV     .AC,(4)+         ;SAVE AC
        001102   010124                            MOV     .MQ,(4)+         ;SAVE MQ
        001104   016724  003004                    MOV     SR1,(4)+         ;SAVE SR

        001110   005767  002504                    TST     LONUM
        001114   001411                            BEQ     OK1
        001116   005701                            TST     .MQ
        001120   001402                            BEQ     .+6
        001122   004767  002370                    JSR     PC,BAD
        001126   020067  002470                    CMP     .AC,HINUM
        001132   001402                            BEQ     .+6
        001134   004767  002356                    JSR     PC,BAD
        001140   012704  003640            OK1:    MOV     #AREA,R4         ;RESET POINTER
        001144   012401                            MOV     (4)+,.MQ         ;RESTORE MQ
        001146   012400                            MOV     (4)+,.AC         ;RESTORE AC
        001150   062704  000010                    ADD     #10,R4           ;INCREMENT POINTER

        001154   016702  002442                    MOV     HINUM,R2         ;LOAD DIVISOR
        001160   004767  002734                    JSR     PC,DIVI          ;DIVIDE BY MULTIPLICAN (SOFTWARE)
        001164   010024                            MOV     .AC,(4)+         ;SAVE AC
        001166   010124                            MOV     .MQ,(4)+         ;SAVE MQ
        001170   016714  002720                    MOV     SR1,(4)          ;SAVE SR
        001174   005767  002422                    TST     HINUM
        001200   001411                            BEQ     ITER
        001202   005701                            TST     .MQ
        001204   001402                            BEQ     .+6
        001206   004767  002304                    JSR     PC,BAD
        001212   020067  002402                    CMP     .AC,LONUM
        001216   001402                            BEQ     .+6
        001220   004767  002272                    JSR     PC,BAD
        001224   005067  177550            ITER:   CLR     ICNT             ;SET UP COUNT
```



```
                                           ;*****************************************************************
                                           ;TEST 1          LONUM X HINUM
                                           ;*****************************************************************
        001230   104000                            SCOPE
        001232   012704  003640                    MOV     #AREA,R4         ;RESTORE POINTER
        001236   012467  003704                    MOV     (4)+,.MQX        ;RESTORE MQ
        001242   012467  003676                    MOV     (4)+,.ACX        ;RESTORE AC
        001246   011467  002342                    MOV     (4),SR1          ;RESTORE SR1

        001252   016700  002342                    MOV     LONUM,.AC        ;LOAD MULTIPLIER (HARDWARE)
        001256   070067  002340                    MUL     HINUM,.AC        ;LOAD MULTIPLICAN AND MULTIPLY
        001262   004767  000324                    JSR     PC,FLOW
        001266   000240                            NOP
        001270   026700  003650                    CMP     .ACX,.AC         ;CHECK HIGH ORDER PRODUCT
        001274   001401                            BEQ     .+4              ;SKIP IF OK
        001276   104400                            HLT
        001300   026701  003642                    CMP     .MQX,.MQ         ;CHECK LOW ORDER PRODUCT
        001304   001401                            BEQ     .+4              ;SKIP IF OK
        001306   104400                            HLT
        001310   126767  002300  000636            CMPB    SR1,TEMP         ;CHECK STATUS REGISTER
        001316   001401                            BEQ     .+4              ;SKIP IF GOOD
        001320   104400                            HLT

                                           ;*****************************************************************
                                           ;TEST 2          HINUM X LONUM
                                           ;*****************************************************************
        001322   104000                            SCOPE
        001324   012704  003640                    MOV     #AREA,R4         ;RESTORE POINTER
        001330   012467  003612                    MOV     (4)+,.MQX        ;RESTORE MQ
        001334   012467  003604                    MOV     (4)+,.ACX        ;RESTORE AC
        001340   011467  002250                    MOV     (4),SR1          ;RESTORE SR1

        001344   016700  002252                    MOV     HINUM,.AC        ;LOAD MULTIPLIER (HARDWARE)
        001350   070067  002244                    MUL     LONUM,.AC        ;MULTIPLY
        001354   004767  000232                    JSR     PC,FLOW
        001360   000240                            NOP
        001362   026700  003556                    CMP     .ACX,.AC         ;CHECK HIGH ORDER PRODUCT
        001366   001401                            BEQ     .+4              ;SKIP IF OK
        001370   104402                            HLT+2
        001372   026701  003550                    CMP     .MQX,.MQ         ;CHECK LOW ORDER PRODUCT
        001376   001401                            BEQ     .+4              ;SKIP IF OK
        001400   104402                            HLT+2
        001402   126767  002206  000544            CMPB    SR1,TEMP         ;CHECK STATUS REGISTER
        001410   001401                            BEQ     .+4              ;SKIP IF GOOD
        001412   104402                            HLT+2
```

```
                                        ;*****************************************************************
                                        ;TEST 3         RESULT / LONUM
                                        ;*****************************************************************
001414  104000                          $SCOPE
001416  012704  003040                  MOV     #AREA,R4        ;RESTORE POINTER
001422  012401                          MOV     (4)+,MQ         ;RESTORE MQ
001424  012400                          MOV     (4)+,AC         ;RESTORE AC
001426  005724                          TST     (4)+            ;INCREMENT POINTER
001430  012467  003010                  MOV     (4)+,ACX        ;RESTORE AC
001434  012467  003006                  MOV     (4)+,MQX        ;RESTORE MQ
001440  011467  002490                  MOV     (4),SR1         ;RESTORE SR1
001444  071067  002150                  DIV     LONUM,AC        ;DIVIDE BY MULTIPLIER (HARDWARE)
001450  004767  000436                  JSR     PC,FLOW
001454  000410                          BR      .+SR1
001456  026700  003002                  CMP     ACX,AC          ;CHECK REMAINDER
001462  001401                          BEQ     .+4             ;SKIP IF OK
001464  104401                          HLT+1
001466  026701  003004                  CMP     MQX,MQ          ;CHECK QUOTIENT
001472  001401                          BEQ     .+4             ;SKIP IF OK
001474  104401                          HLT+1
001476  126767  002412  000430  ,SR1:  CMPB    SR1,TEMP        ;CHECK STATUS REGISTER
001504  001401                          BEQ     .+4             ;SKIP IF OK
001506  104401                          HLT+1

                                        ;*****************************************************************
                                        ;TEST 4         RESULT / MINUM
                                        ;*****************************************************************
001510  104000                          $SCOPE
001512  012704  003040                  MOV     #AREA,R4        ;RESTORE POINTER
001516  012401                          MOV     (4)+,MQ         ;RESTORE MQ
001520  012400                          MOV     (4)+,AC         ;RESTORE AC
001522  062704  000010                  ADD     #10,R4          ;INCREMENT POINTER
001526  012467  003412                  MOV     (4)+,ACX        ;RESTORE AC
001532  012467  003410                  MOV     (4)+,MQX        ;RESTORE MQ
001536  011467  002392                  MOV     (4),SR1         ;RESTORE SR1
001542  071067  002094                  DIV     MINUM,AC        ;DIVIDE BY MULTIPLICAN (HARDWARE)
001546  004767  000340                  JSR     PC,FLOW
001552  000410                          BR      .+SR2
001554  026700  003304                  CMP     ACX,AC          ;CHECK REMAINDER
001560  001401                          BEQ     .+4             ;SKIP IF OK
001562  104403                          HLT+3
001564  026701  003306                  CMP     MQX,MQ          ;CHECK QUOTIENT
001570  001401                          BEQ     .+4             ;SKIP IF OK
001572  104403                          HLT+3
001574  126767  002314  000332  ,SR2:  CMPB    SR1,TEMP        ;CHECK STATUS REGISTER
001602  001401                          BEQ     .+4             ;SKIP IF OK
001604  104403                          HLT+3
```



```
001606  104000                          $SCOPE                  ;CALCULATE NEW RANDOM DIVISOR
001610  016767  002004  002022          MOV     LONUM,AREA      ;GET MQ
001616  016767  002000  002014          MOV     MINUM,AREA+2    ;GET AC
001624  016701  001770                  MOV     LONUM,MQ        ;LOAD MQ (SOFTWARE)
001630  016700  001766                  MOV     MINUM,AC        ;LOAD AC (SOFTWARE)
001634  004767  001332                  JSR     PC,RANDOM       ;GET A RANDOM NUMBER
001640  016767  001754  001776          MOV     LONUM,AREA+44   ;GET DIVISOR
001646  016702  001746                  MOV     LONUM,R2        ;LOAD THE DIVISOR
001652  004767  002242                  JSR     PC,DIV1         ;DIVIDE
001656  010067  003202                  MOV     ,AC,,ACX
001662  010167  003200                  MOV     ,MQ,,MQX
001666  012767  001974  001734          MOV     #LOCX,LAD       ;PUDGE LAD

                                        ;*****************************************************************
                                        ;TEST 5         LONUM MINUM / NEW LONUM
                                        ;*****************************************************************
001674  016701  001740  LOCX1:  MOV     AREA,MQ         ;RESTORE MQ
001700  016700  001736          MOV     AREA+2,AC       ;RESTORE AC
001704  071067  001734          DIV     AREA+4,AC       ;DIVIDE
001710  004767  000170          JSR     PC,FLOW
001714  000410                  BR      .+R3
001716  026700  003222          CMP     ACX,AC          ;CHECK AC
001722  001401                  BEQ     .+4
001724  104407                  HLT+7                   ;REMAINDER WRONG
001726  026701  003214          CMP     MQX,MQ          ;CHECK QUOTIENT
001732  001401                  BEQ     .+4
001734  104407                  HLT+7                   ;QUOTIENT WRONG
001736  126767  002132  000210  ,SR3:  CMPB    SR1,TEMP        ;CHECK STATUS
001744  001401                  BEQ     .+4
001746  104407                  HLT+7
```

```
001750  104000                                  SCOPE

001752  005046                                  CLR     -(6)              ;CLEAR TRACE TRAP
001754  032767  010000  175606                  BIT     #SW12,SWR         ;RUN WITH TRT?  -SW12-
001762  001010                                  BNE     2$
001764  005167  000166                          COM     TRPB
001770  100005                                  BPL     2$
001772  052716  000020                          BIS     #20,(6)           ;SET TRACE TRAP
001776  012746  001034                          MOV     #GO,-(6)          ;JUMP TO START OF TEST
002002  000006                                  RTT
002004  012746  002012                  2$:     MOV     #RTIX,-(6)        ;JUMP TO START OF TEST
002010  000006                                  RTT
002012  062767  000001  001604          RTIX:   ADD     #1,CNT            ;COUNT NUMBER OF PASSES
002020  016737  001600  177570                  MOV     CNT,@#DISPLAY
002026  032767  000177  001570                  BIT     #177,CNT          ;CHECK FOR MULTIPLES OF 200
002034  001402                                  BEQ     .+6
002036  000167  176772                  DOIT:   JMP     GO                ;RERUN
002042  032767  002000  175520                  BIT     #2000,SWR
002050  001372                                  BNE     DOIT
002052  005067  001546                          CLR     CNT
002056  112767  000007  001546                  MOVB    #BELL,.TYPE       ;TYPE A BELL
002064  000004  003632                          TYPE    ,.TYPE
002070  013700  000042                          MOV     @#42,R0           ;GET MONITOR ADDRESS (IF ANY)
002074  001404                                  BEQ     1$
002076  004710                                  JSR     7,(0)             ;CALL THE MONITOR
002100  000240                                  NOP
002102  000240                                  NOP
002104  000240                                  NOP
002106  000167  176722                  1$:     JMP     GO                ;RERUN


002112  113767  177776  000034          FLOW:   MOVB    @#177776,TEMP     ;GET THE PS BITS
002120  042767  177760  000026                  BIC     #177760,TEMP
002126  032767  000002  000020                  BIT     #2,TEMP
002134  001003                                  BNE     EQ2
002136  062716  000002                          ADD     #2,(6)
002142  000207                                  RTS     PC
002144  042767  000014  000002          EQ2:    BIC     #14,TEMP
002152  000207                                  RTS     PC

002154  000000                          TEMP:   0
002156  000000                          TRPB:   0
```



```
002160  032737  004000  177570          .EMT:   BIT     #SW11,@#SWR       ;KILL ITERATIONS
002166  001053                                  BNE     SAVLAD
002170  005767  176604                          TST     ICNT
002174  001445                                  BEQ     ICINT             ;BRANCH IF FIRST
002176  032737  040000  177570                  BIT     #SW14,@#SWR       ;KILL ITERATIONS
002204  001047                                  BNE     KIT               ;YES
002206  026767  001424  176564                  CMP     TIMES,ICNT        ;DONE?
002214  003043                                  BGT     KIT               ;BRANCH IF NOT
002216  005767  001404                          TST     ERCNT             ;ANY ERRORS?
002222  001432                                  BEQ     ICINT
002224  032737  020000  177570                  BIT     #SW13,@#SWR       ;INHIBIT TYPEOUTS?
002232  001024                                  BNE     NOTY
002234  000004  003194                          TYPE    ,ER6
002240  016703  001362                          MOV     ERCNT,TTY         ;TYPE ERCNT IN OCTAL
002244  004767  001510                          JSR     PC,PRINTS         ;AND SUPRESS LEADING ZERO'S
002250  000004  003200                          TYPE    ,ER7
002254  016703  176520                          MOV     ICNT,TTY          ;TYPE ICNT IN OCTAL
002260  004767  001474                          JSR     PC,PRINTS         ;AND SUPRESS LEADING ZERO'S
002264  000004  002270                          TYPE    ,.+2              ;.ASCIZ <15><12>
002274  005737  177570                          TST     @#SWR             ;HALT ON ERROR  -SW15-
002300  100001                                  BPL     .+4
002302  000000                                  HALT
002304  005067  001316                  NOTY:   CLR     ERCNT
002310  012767  000001  176462          ICINT:  MOV     #1,ICNT           ;FIRST ITERATION
002316  011667  001306                  SAVLAD: MOV     (6),LAD           ;SAVE LOOP ADDRESS
002322  000002                                  RTI                       ;RETURN

002324  005267  176450                  KIT:    INC     ICNT
002330  005767  001274                  OVR:    TST     LAD               ;FIRST ONE?
002334  001770                                  BEQ     SAVLAD
002336  016716  001266                          MOV     LAD,(6)           ;FUDGE RETURN ADDRESS
002342  000002                                  RTI                       ;FIXES PS
```

```
   304  002344  032737  002000  177570  ,TRP1   BIT     #SW10,@#SWR     ;BELL ON ERROR?
   305  002352  001405                  BEQ     ,E?
   306  002354  112767  000007  001250          MOVB    #BELL,,TYPE     ;TYPE A BELL
   307  002362  000004  003032                  TYPE    ,,TYPE
   308  002366  005767  001234          ,E?:    TST     ERCNT           ;CHECK FOR FIRST ERROR
   309  002372  001402                  BEQ     ,+6
   310  002374  000167  000412          NOR:    JMP     NOHEAD
   311  002400  011667  000422                  MOV     (6),HLTAD       ;SAVE ADDRESS OF HLT
   312  002404  005267  001224                  INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
   313  002410  132737  000001  000041          BITB    #1,@#41         ;ADTD RUN?
   314  002416  001412                  BEQ     1$              ;NO!
   315  002420  022767  000010  001206          CMP     #10,ERRORS      ;TOO MANY ERRORS
   316  002426  001006                  BNE     1$              ;NOT YET!
   317  002430  013700  000042                  MOV     @#42,R0         ;GET ADDRESS
   318  002434  001403                  BEQ     1$              ;NONE?!?!
   319  002436  005037  000042                  CLR     @#42            ;ZERO IT
   320  002442  004710                  JSR     PC,(0)          ;RETURN TO MONITOR
   321  002444  032767  020000  179116  1$:     BIT     #SW13,SWR       ;SKIP TYPEOUT IF SET
   322  002452  001350                  BNE     NOH
   323  002454  011646                  MOV     (6),-(6)        ;PUT ADDRESS OF INSTRUCTION ON STACK
   324  002456  162716  000002                  SUB     #2,(6)          ;GET THE INSTRUCTION
   325  002462  032776  000001  000000          BIT     #1,0(6)         ;TEST FOR MULTIPLY OR DIVIDE
   326  002470  001034                  BNE     DIVP            ;GOTO DIVIDE PRINT IF ON A 1
   327  002472  000004  003093                  TYPE    ,ER1M
   328  002476  032776  000002  000000          BIT     #2,0(6)         ;TEST FOR WHICH MULTIPLY
   329  002504  001413                  BEQ     SECO
   330  002506  016703  001106                  MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
   331  002512  004767  001232                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   332  002516  000004  003074                  TYPE    ,ER2
   333  002522  016703  001074                  MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
   334  002526  004767  001216                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   335  002532  000402                  BR      PSOFT
   336  002534                          SECO:
   337  002534  016703  001062                  MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
   338  002540  004767  001204                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   339  002544  000004  003074                  TYPE    ,ER2
   340  002550  016703  001044                  MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
   341  002554  004767  001170                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   342  002560  000447                  BR      PSOFT
```



```
   343  002562  000004  003032          DIVP:   TYPE    ,ER1D
   344  002566  016703  001030                  MOV     AREA+2,TTY      ;TYPE AREA+2 IN OCTAL
   345  002572  004767  001132                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   346  002576  112767  000040  001026          MOVB    #40,,TYPE       ;TYPE A 40
   347  002604  000004  003032                  TYPE    ,,TYPE
   348  002610  016703  001024                  MOV     AREA,TTY        ;TYPE AREA IN OCTAL
   349  002614  004767  001130                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   350  002620  000004  003100                  TYPE    ,ER2A
   351  002624  032776  000004  000000          BIT     #4,0(6)
   352  002632  001405                  BEQ     DIVE
   353  002634  016703  001004                  MOV     AREA+4,TTY      ;TYPE AREA+4 IN OCTAL
   354  002640  004767  001104                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   355  002644  000415                  BR      PSOFT
   356  002646  032776  000002  000000  DIVE:   BIT     #2,0(6)         ;TEST FOR WHICH DIVIDE
   357  002654  001005                  BNE     SECI
   358  002656  016703  000736                  MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
   359  002662  004767  001042                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   360  002666  000404                  BR      PSOFT
   361  002670                          SECI:
   362  002670  016703  000730                  MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
   363  002674  004767  001050                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   364  002700  000004  003104          PSOFT:  TYPE    ,ER3
   365  002704  016703  002234                  MOV     ,ACX,TTY        ;TYPE 'ACX IN OCTAL
   366  002710  004767  001034                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   367  002714  000004  003137                  TYPE    ,ER4
   368  002720  016703  002222                  MOV     ,MQX,TTY        ;TYPE 'MQX IN OCTAL
   369  002724  004767  001020                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   370  002730  000004  003144                  TYPE    ,ER5
   371  002734  016703  001164                  MOV     SR1,TTY         ;TYPE SR1 IN OCTAL
   372  002740  004767  001014                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
   373  002744  000004  003124                  TYPE    ,ER3B
   374  002750  000004  003117                  TYPE    ,ER3A
   375  002754  010003                  MOV     ,AC,TTY         ;TYPE 'AC IN OCTAL
   376  002756  004767  000766                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   377  002742  000004  003137                  TYPE    ,ER4
   378  002766  010103                  MOV     ,MQ,TTY         ;TYPE 'MQ IN OCTAL
   379  002770  004767  000754                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
   380  002774  000004  003144                  TYPE    ,ER5
   381  003000  016703  177130                  MOV     TEMP,TTY        ;TYPE TEMP IN OCTAL
   382  003004  004767  000730                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
   383  003010  005726                  TST     (6)+            ;RESTORE POINTER
   384  003012  021667  000010          NOREAD: CMP     (6),HLTAD       ;CHECK FOR SAME HLT
   385  003016  001002                  BNE     ,+6
   386  003020  005267  000002                  INC     ERCNT
   387  003024  000002                  RTI
```

MAINDEC-11-DCQKA-B MUL - DIV RANDOM EXERCISER MACY11.624 20-NOV-75 09:48 PAGE 10
DCQKAB.P11 ERROR ROUTINES

```
003026  000000                          HLTAD:  0
003030  000000                          SER:    0
003032  005015  042012  053111          ERID:   .ASCIZ  <15><12><12>"DIV ERROR!   "
003040  042440  051122  051117
003046  020072  020040     000
003053     015  005012  052515          ERIM:   .ASCIZ  <15><12><12>"MUL ERROR!   "
003060  020114  051105  047522
003066  035122  020040  000040
003074  054040  000040                  ER2:    .ASCIZ  " X "
003100  027440  000040                  ER2A:   .ASCIZ  " / "
003104  005015  051412  043117          ER3:    .ASCII  <15><12><12>"SOFTWARE"
003112  053524  051101     105
003117     040  020040  000040          ER3A:   .ASCIZ  "    "
003124  005015  040510  042122          ER3B:   .ASCIZ  <15><12>"HARDWARE"
003132  040527  042522     000
003137     040  020040  000040          ER4:    .ASCIZ  "    "
003144  020040  020040  051520          ER5:    .ASCIZ  "     PS="
003152  000075
003154  005015  042412  051122          ER6:    .ASCIZ  <15><12><12>"ERRORS OCCURRED "
003162  051117  020123  041517
003170  052503  051122  042105
003176  000040
003200  052040  046511  051505          ER7:    .ASCIZ  " TIMES OUT OF "
003206  047440  052125  047440
003214  020106     000
003217     015  000012                  CRL:    .ASCIZ  <15><12>
003222  005015  047523  052106          ERNG:   .ASCII  <15><12>"SOFTWARE DIVIDE DOESN'T MATCH REAL ANSWER.  "
003230  040527  042522  042040
003236  053111  042111  020105
003244  047504  051505  023516
003252  020124  040515  041524
003260  020110  042522  046101
003266  040440  051516  042527
003274  027122  020040
003300  047514  020103  020075          E1:     .ASCIZ  "LOC = "
003306     000
003307     015  027012  041501          E2:     .ASCIZ  <15><12>".AC = "
003314  036440  000040
003320  020040  020040  046456          E3:     .ASCIZ  "    .MQ = "
003326  020121  020075     000
003333     040  020040  046040          E4:     .ASCIZ  "    LONUM = "
003340  047117  046525  036440
003346  000040
003350  020040  020040  044510          E5:     .ASCIZ  "    HINUM = "
003356  052516  020115  020075
003364     000
003365     015  005012     000          E6:     .ASCIZ  <15><12><12>
        003372                                  .EVEN
```

MAINDEC-11-DCQKA-B MUL - DIV RANDOM EXERCISER MACY11.624 20-NOV-75 09:48 PAGE 11
DCQKAB.P11 RANDOM NUMBER GENERATOR

```
003372  010046                  RANDOM: MOV     .AC,-(6)        ;SAVE R0
003374  010146                          MOV     .MQ,-(6)        ;SAVE R1
003376  010246                          MOV     R2,-(6)         ;SAVE R2
003400  010446                          MOV     R4,-(6)         ;SAVE R4
003402  016700  000212                  MOV     LONUM,.AC       ;SET R0 WITH LOW
003406  016701  000210                  MOV     HINUM,.MQ       ;SET R1 WITH HIGH
003412  012704  177771                  MOV     #-7,R4          ;SET SHIFT COUNT
003416  005002                          CLR     R2
003420  006300                  1$:     ASL     .AC             ;SHIFT R0 LEFT AND
003422  006101                          ROL     .MQ             ;ROTATE CARRY INTO R1 AND
003424  006102                          ROL     R2              ;ROTATE CARRY INTO R2
003426  005204                          INC     R4              ;CHECK FOR DONE
003430  001373                          BNE     1$              ;CONTINUE SHIFT LOOP
003432  066702  000182                  ADD     LONUM,R2        ;ADD NUMBER TO MAKE X 129
003436  005501                          ADC     .MQ             ;PROPOGATE CARRY
003440  066701  000156                  ADD     HINUM,.MQ       ;ADD NUMBER TO MAKE X 129
003444  005502                          ADC     R2              ;PROPOGATE CARRY
003446  062700  001057                  ADD     #1057,.AC       ;ADD LOW CONSTANT
003452  005501                          ADC     .MQ             ;PROPOGATE CARRY
003454  005502                          ADC     R2              ;PROPOGATE CARRY
003456  062701  047401                  ADD     #47401,.MQ      ;ADD HIGH CONSTANT
003462  005502                          ADC     R2              ;PROPOGATE CARRY
003464  062702  000006                  ADD     #6,R2           ;ADD HIGHEST CONSTART
003470  060200                          ADD     R2,.AC          ;REPRIME R0 WITH HIGHEST DIGIT
003472  005501                          ADC     .MQ             ;PROPOGATE CARRY
003474  010067  000120                  MOV     .AC,LONUM       ;SAVE R0
003500  010167  000116                  MOV     .MQ,HINUM       ;SAVE R1
003504  012604                          MOV     (6)+,R4         ;RESTORE R4
003506  012602                          MOV     (6)+,R2         ;RESTORE R2
003510  012601                          MOV     (6)+,.MQ        ;RESTORE R1
003512  012600                          MOV     (6)+,.AC        ;RESTORE R0
003514  000207                          RTS     PC              ;RETURN


003516  000004  003222          BAD:    TYPE    ,ERNG
003522  011603                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
003524  004767  000230                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
003530  000004  003307                  TYPE    ,E2
003534  010003                          MOV     .AC,TTY         ;TYPE .AC IN OCTAL
003536  004767  000206                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003542  000004  003320                  TYPE    ,E3
003546  010103                          MOV     .MQ,TTY         ;TYPE .MQ IN OCTAL
003550  004767  000174                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003554  000004  003333                  TYPE    ,E4
003560  016703  000034                  MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
003564  004767  000160                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003570  000004  003350                  TYPE    ,E5
003574  016703  000022                  MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
003600  004767  000144                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003604  000004  003365                  TYPE    ,E6
003610  000207                          RTS     PC
```

```
003612  000006                    .TRT:   RTT
003614  000000                    TAD:    0                         ;TRAP ADDRESS
003616  000000                    TLOC:   0                         ;TRAP LOCATION
003620  000000                    LONUM:  0                         ;RANDOM NUMBER #1
003622  000000                    HINUM:  0                         ;RANDOM NUMBER #2
003624  000000                    CNT:    0                         ;PASS NUMBER
003626  000000                    ERCNT:  0                         ;ERROR COUNT
003630  000000                    LAD:    0                         ;LOOP ADDRESS
003632  000000                    .TYPE:  0                         ;CHARACTER TYPING LOCATION
003634  000000                    ERRORS: 0                         ;ERRORS ENCOUNTERED
003636  000400                    TIMES:  256.

003640  000000                    AREA:   0                         ;MQ (MULTIPLY)          DIVIDEND RIGHT
003642  000000                            0                         ;AC (MULTIPLY)          DIVIDEND LEFT
003644  000000                            0                         ;SR (MULTIPLY)          DIVISOR
003646  000000                            0                         ;AC (DIVIDE BY LONUM)
003650  000000                            0                         ;MQ (DIVIDE BY LONUM)
003652  000000                            0                         ;SR (DIVIDE BY LONUM)
003654  000000                            0                         ;AC (DIVIDE BY HINUM)
003656  000000                            0                         ;MQ (DIVIDE BY HINUM)
003660  000000                            0                         ;SR (DIVIDE BY HINUM)

003662  010346                    .IOT:   MOV     TTY,-(6)          ;SAVE TTY
003664  017603  000002                    MOV     @2(6),TTY         ;GET ADDRESS TO BE TYPED
003670  105713                    1$:     TSTB    (TTY)             ;TERMINATOR?
003672  001406                            BEQ     2$
003674  112337  177566                    MOVB    (TTY)+,@#177566   ;LOAD AND TYPE THE CHARACTER
003700  105737  177564                    TSTB    @#177564          ;IS THE PRINTER READY
003704  100375                            BPL     .-4
003706  000770                            BR      1$                ;GET THE NEXT CHARACTER

003710  017646  000002            2$:     MOV     @2(6),-(6)        ;GET ADDRESS TO BE TYPED
003714  062766  000002  000004            ADD     #2,4(6)           ;ADD 2 TO THE ADDRESS
003722  022666  000002                    CMP     (6)+,2(6)         ;IS IT .+2?
003726  001006                            BNE     3$                ;NO
003730  062703  000002                    ADD     #2,TTY            ;ADD 2 TO THE ADDRESS
003734  042703  000001                    BIC     #1,TTY            ;BACK UP TO AN EVEN BYTE
003740  010366  000002                    MOV     TTY,2(6)          ;RESTORE ADDRESS
003744  012603                    3$:     MOV     (6)+,TTY          ;RESTORE TTY
003746  000002                            RTI                       ;RETURN
```



```
003750  112767  000001  000130   PRINTR: MOVB    #1,A4$            ;SET ZERO FILL SWITCH
003756  000402                           BR      .+6
003760  005067  000122           PRINTS: CLR     A4$               ;SUPPRESS LEADING ZERO'S
003764  112767  177772  000115           MOVB    #-6,A4$+1         ;SET COUNT
003772  010446                           MOV     R4,-(6)           ;SAVE R4
003774  012704  004076                   MOV     #3$,R4            ;SET POINTER TO FIRST ASCII CHAR.
004000  105014                           CLRB    (4)               ;CLEAR FIRST BYTE
004002  000405                           BR      2$                ;ROTATE FIRST BIT
004004  105014                   1$:     CLRB    (4)               ;CLEAR BYTE OF CHARACTER
004006  006103                           ROL     TTY               ;ROTATE BIT INTO C
004010  106114                           ROLB    (4)               ;PACK IT
004012  006103                           ROL     TTY               ;ROTATE BIT INTO C
004014  106114                           ROLB    (4)               ;PACK IT
004016  006103                   2$:     ROL     TTY               ;ROTATE BIT INTO C
004020  106114                           ROLB    (4)               ;PACK IT
004022  105714                           TSTB    (4)
004024  001402                           BEQ     .+6
004026  105267  000054                   INCB    A4$
004032  105767  000050                   TSTB    A4$               ;CHECK FILL SWITCH
004036  001402                           BEQ     .+6
004040  152724  000060                   BISB    #60,(4)+          ;MAKE INTO ASCII CHAR
004044  105267  000037                   INCB    A4$+1
004050  001355                           BNE     1$                ;REPEAT
004052  022704  004076                   CMP     #3$,R4
004056  001002                           BNE     .+6
004060  112724  000060                   MOVB    #60,(4)+
004064  105014                           CLRB    (4)
004066  000004  004076                   TYPE    ,3$               ;TYPE IT
004072  012604                           MOV     (6)+,R4           ;RESTORE R4
004074  000207                           RTS     PC

004076  000004                   3$:     .BLKW   4
004106  000000                   A4$:    0
```

```
                                        ;       DIVIDE SUBROUTINE
                                        ;
004110  000000                  EI:    ;WORD   0
004112  000000                  COUNT1: ;WORD   0
004114  000000                  SR1:    ;WORD   0
004116  000000                  SIGE1:  ;WORD   0
                                        ;
                                        ;
004120  005702                  DIVI1:  TST     R2
004122  001004                          BNE     DIVL
004124  012767  000003  177762          MOV     #3,SR1
004132  000207                          RTS     PC
004134  012767  177760  177750  DIVL1:  MOV     #-16.,COUNT     ;SET UP STEP COUNTER
004142  005067  177746                  CLR     SR1
004146  005067  177744                  CLR     SIGE
004152  005700                          TST     AC
004154  100002                          BPL     .+6
004156  105167  177734                  COMB    SIGE
004162  005067  177722          STEP11: CLR     E               ;INITIALIZE E INDICATORS
004166  000241                          CLC
004170  005700                          TST     AC              ;CHECK REMAINDER SIGN
004172  100002                          BPL     .+6
004174  105167  177711                  COMB    E+1
004200  005702                          TST     R2              ;CHECK DIVISOR SIGN
004202  100002                          BPL     .+6
004204  105167  177701                  COMB    E+1
004210  105767  177675                  TSTB    E+1             ;1s IF DIFFERENT, 0 IF SAME
004214  100403                          BMI     .+8.
004216  005267  177666                  INC     E               ;SET E=1 IF SAME
004222  000261                          SEC                     ;SET C TO A 1
004224  006101                  STEP2:  ROL     .MQ             ;ROTATE AC AND MQ ONCE LEFT
004226  006100                          ROL     .AC
004230  105767  177654                  TSTB    E               ;STEP3
004234  001406                          BEQ     .+14.
004236  160200                          SUB     R2,.AC          ;SUBTRACT IF E = 1
004240  106167  177653                  ROLB    SIGE+1
004244  105167  177647                  COMB    SIGE+1
004250  000403                          BR      .+8.
004252  060200                          ADD     R2,.AC          ;ADD IF E = 0
004254  106167  177637                  ROLB    SIGE+1
004260  022767  177760  177624          CMP     #-16.,COUNT     ;CHECK FOR FIRST TIME THROUGH
004266  001014                          BNE     STEP8
```



```
004270  005700                  STEP61: TST     .AC
004272  001412                          BEQ     STEP8
004274  106067  177617                  RORB    SIGE+1
004300  103404                          BCS     .+10.           ;IF C AND DIVIDEND ARE
004302  105767  177610                  TSTB    SIGE            ;       DIFFERENT, AN OVERFLOW
004306  100431                          BMI     OFLW            ;       HAS OCCURRED
004310  000403                          BR      .+8.
004312  105767  177600                  TSTB    SIGE
004316  100025                          BPL     OFLW
                                        ;
004320  005267  177580          STEP81: INC     COUNT
004324  001402                          BEQ     .+6
004326  000167  177430                  JMP     STEP1
                                        ;
004332  005700                          TST     .AC             ;BEGINNING OF CORRECTION ROUTINE
004334  001022                          BNE     STEP9           ;IF THE REMAINDER IS ZERO,
004336  000261                  STEP101 SEC                     ;       SHIFT THE QUOTIENT ONCE
004340  006101                          ROL     .MQ             ;       FILLING WITH "1"
004342  116705  177530          STP91:  MOVB    SIGE,R5
004346  005702                          TST     R2
004350  100001                          BPL     .+4
004352  005105                          COM     R5
004354  005705                          TST     R5
004356  100403                          BMI     .+8.
004360  005701                          TST     .MQ
004362  100403                          BMI     OFLW
004364  000465                          BR      DONX
004366  005701                          TST     .MQ
004370  003463                          BLE     DONX
004372  052767  000002  177514  OFLW1:  BIS     #2,SR1
004400  000207                          RTS     PC
                                        ;
                                        ;
004402  010267  177504          STEP91: MOV     R2,COUNT
004406  100402                          BMI     .+6
004410  005467  177476                  NEG     COUNT           ;IF THE DIVISOR WAS (+) NEGATE IT
004414  020067  177472                  CMP     .AC,COUNT       ;IF THE SAME AS REMAINDER, GO
004420  001416                          BEQ     STEP11          ;       TO STEP 11
                                        ;
004422  105067  177463                  CLRB    E+1
004426  005700                          TST     .AC             ;COMPARE THE SIGNS OF THE
004430  100002                          BPL     .+6             ;       REMAINDER AND THE
004432  105167  177453                  COMB    E+1             ;       DIVIDEND.  IF THEY
004436  105767  177454                  TSTB    SIGE            ;       ARE THE SAME, GO TO
004442  100002                          BPL     .+6             ;       STEP 10
004444  105167  177441                  COMB    E+1
004450  105767  177435                  TSTB    E+1
004454  001730                          BEQ     STEP10
```

```
004456  005067  177426          STEP11: CLR     Z
004462  005700                          TST     @AC             ;SET Z=1 IF REMAINDER AND
004464  100002                          BPL     .+6             ;       THE DIVISOR HAVE THE
004466  105167  177417                  COMB    Z+1             ;       SAME SIGN
004472  005702                          TST     R2
004474  100002                          BPL     .+6
004476  105167  177407                  COMB    Z+1
004502  105767  177403                  TSTB    Z+1
004506  001003                          BNE     .+8.
004510  005267  177374                  INC     Z
004514  005201                          INC     @MQ             ;ADD Z TO QUOTIENT
004516  000241                          CLC
004520  006101                          ROL     @MQ             ;SHIFT QUOTIENT LEFT ONCE
                                                                ;       FILLING WITH "0"
004522  105767  177362                  TSTB    Z
004526  001402                          BEQ     .+6
004530  160200                          SUB     R2,@AC          ;SUBTRACT DIVISOR FROM REMAINDER
004532  000703                          BR      STP9
004534  060200                          ADD     R2,@AC          ;ADD DIVISOR TO REMAINDER
004536  000701                          BR      STP9

004540  010046                  DONX:   MOV     @AC,-(6)
004542  010100                          MOV     @MQ,@AC
004544  012601                          MOV     (6)+,@MQ
004546  005700                          TST     @AC
004550  001003                          BNE     .+8.
004552  052767  000004  177334          BIS     #4,SR1
004560  005700                  DONE:   TST     @AC
004562  100003                          BPL     .+8.
004564  052767  000010  177322          BIS     #10,SR1
004572  005700                          TST     @AC
004574  001005                          BNE     NXT
004576  005701                          TST     @MQ
004600  001003                          BNE     NXT
004602  052767  000004  177304          BIS     #4,SR1
004610  000207                  NXT:    RTS     PC
```



```
                                ;       MULTIPLY SUBROUTINE

004612  000000                  CONT:   .WORD   0
004614  000000                  ONE:    .WORD   0
004616  000000                  CHEK:   .WORD   0

004620  005067  177770          MUL1:   CLR     ONE             ;CLEAR Z STEP1
004624  005067  177294                  CLR     SR1
004630  005067  177762                  CLR     CHEK
004634  005000                          CLR     @AC             ;CLEAR AC
004636  005701                          TST     @MQ
004640  100002                          BPL     .+6
004642  005167  177750                  COM     CHEK
004646  005702                          TST     R2
004650  100002                          BPL     .+6
004652  005167  177740                  COM     CHEK
004656  012767  177760  177726          MOV     #-16.,CONT      ;INITIALIZE COUNT
004664  022701  100000                  CMP     #100000,@MQ
004670  001011                          BNE     CNXT
004672  005702                          TST     R2
004674  100023                          BPL     STEP2
004676  010200                          MOV     R2,@AC
004700  005400                          NEG     @AC
004702  000241                          CLC
004704  005001                          CLR     @MQ
004706  006000                          ROR     @AC
004710  006001                          ROR     @MQ
004712  000451                          BR      DONE2
004714  022702  100000          CNXT:   CMP     #100000,R2
004720  001011                          BNE     STEP2
004722  005701                          TST     @MQ
004724  100007                          BPL     STEP2
004726  010100                          MOV     @MQ,@AC
004730  005400                          NEG     @AC
004732  000241                          CLC
004734  005001                          CLR     @MQ
004736  006000                          ROR     @AC
004740  006001                          ROR     @MQ
004742  000435                          BR      DONE2
004744  105067  177645          STEP2:  CLRB    ONE+1
004750  132767  000001  177636          BITB    #1,ONE
004756  001402                          BEQ     .+6
004760  105167  177631                  COMB    ONE+1           ;COMP ONE+1 IF Z=1
004764  032701  000001                  BIT     #1,@MQ
004770  001402                          BEQ     .+6
004772  105167  177617                  COMB    ONE+1           ;COMP ONE+1 IF R1 IS ODD
004776  105767  177613                  TSTB    ONE+1           ;IF THE LOW ORDER BIT IS THE SAME
005002  001406                          BEQ     STEP4           ;AS Z, GO TO STEP4
```

```
734  005004  032701  000001          STEP3:  BIT     #1,MQ
735  005010  001002                          BNE     .+6
736  005012  060200                          ADD     R2,AC           ;ADD MULTIPLICAND IF MQ IS EVEN
737  005014  000401                          BR      .+4
738  005016  160200                          SUB     R2,AC           ;SUBTRACT MULTIPLICAND IF MQ IS ODD
739  005020  006200                  STEP4:  ASR     AC
740  005022  006001                          ROR     MQ
741  005024  106167  177544                  ROLB    ONE
742
743  005030  005267  177556          STEP5:  INC     CONT
744  005034  001343                          BNE     STEP2           ;GO TO STEP2 IF NOT 16TH PASS
745  005036  005700                  DONE2:  TST     AC
746  005040  001416                          BEQ     DO2
747  005042  005767  177550                  TST     CHEK
748  005046  100406                          BMI     DON1
749  005050  005700                          TST     AC
750  005052  100003                          BPL     .+8.
751  005054  005401                          NEG     MQ
752  005056  005500                          ADC     AC
753  005060  005400                          NEG     AC
754  005062  000405                          BR      DO2
755  005064  005700                  DON1:   TST     AC
756  005066  100403                          BMI     .+8.
757  005070  005401                          NEG     MQ
758  005072  005500                          ADC     AC
759  005074  005400                          NEG     AC
760  005076  005700                  DO2:    TST     AC
761  005100  001412                          BEQ     DO3
762  005102  022700  177777                  CMP     #-1.,AC
763  005106  001011                          BNE     DO4X
764  005110  005701                          TST     MQ
765  005112  100403                          BMI     .+8.
766  005114  052767  000001  176772          BIS     #1,SR1
767  005122  000167  177432                  JMP     DONE
768  005126  005701                  DO3:    TST     MQ
769  005130  100003                          BPL     .+8.
770  005132  052767  000001  176754  DO4X:   BIS     #1,SR1
771  005140  000167  177414          DO4:    JMP     DONE
772
773
774  005144  000000                  .ACX:   0
775  005146  000000                  .MQX:   0
776
777          000001                  .END
```



```
AREA        003640     62     85    108    131    153    178    199*   200*   204*   215    216    217    344
                      348    353    499#
A4S         004106    527*   529*   530*   544*   545    548*   559#
BAD         003516     81     84     99    102    470#
BEGIN       001002     30     56#
BELL      = 000007     52#   250    306
CHEK        004616    690#   694*   698*   701*   747
CNT         003624     58*   242*   243    244    249*   492#
CNXT        004714    704    714#
CONT        004612    688#   702*   743*
COUNT       004112    563#   572*   600    612*   635*   637*   638
CRL         003217    413#
DISPLA =  177570     40#   243*
DIVI        004120     72     91    206    568#
DIVL        004134    569    572#
DIVP        002562    326    343#
DIVE        002646    352    356#
DOIT        002036    246#   248
DONE        004563    677#   767    771
DONE2       005036    713    724    745#
DONX        004543    628    630    671#
DON1        005064    748    755#
DO2         005076    746    754    760#
DO3         005126    761    768#
DO4         005140    771#
DO4X        005132    763    770#
EQ2         002144    264    267#
ERCNT       003626    280    285    294*   308    384*   493#
ERNG        003022    414#   478
ERRORS      003634     27*   312*   315    496#
ER1D        003032    343    390#
ER1M        003053    327    393#
ER2         003074    332    339    396#
ER2A        003103    350    397#
ER3         003104    364    398#
ER3A        003117    374    400#
ER3B        003124    373    401#
ER4         003137    367    377    403#
ER5         003144    370    380    404#
ER6         003154    284    406#
ER7         003200    287    410#
E1          003300    422#
E2          003307    424#   473
E3          003323    426#   476
E4          003333    428#   479
E5          003350    431#   482
E6          003365    434#   485
FLOW        002112    115    138    162    187    218    261#
GO          001034     62#   238    246    258
HINUM       003622     60*    65     82     98     95    114    136    189    280    282    333    337    362
                      441    451    462*   483    491#
HLT       = 104400     42#   119    122    125    142    145    148    160    169    172    191    194    197
                      222    225    228
HLTAD       003026    311*   384    388#
```

```
ICINT     002310            275    281    295#
ICNT      001000             55#   103*   274    278    288    295*   299*
ITER      001224             96    103#
KIT       002324            277    279    299#
LAD       003630            209*   296*   300    302    494#
LOCX      001674            209    215#
LONUM     003620             59*    64     71     77    100    113    137    161    199    201    204    205    330
                            340    350    440    449    461*   480    490#
MUL1      004620             66    692#
N         = 000005           38#   104    107#   127    130#   149    152#   174    177#   210    215#
NOH       002374            310#   322
NOHEAD    003012            310    384#
NOTY      002304            283    294#
NXT       004610            681    683    685#
OFLW      004372            607    610    627    631#
OK1       001140             78     85#
ONE       004614            689#   692*   725*   726    728*   731*   732    741*
OVR       002330            500#
PC        =%000007           51#    63*    66*    72*    81*    84*    91*    99*   102*   115*   138*   162*   187*
                            203*   206*   218*   266*   268*   286*   289*   320*   331*   334*   338*   341*   345*
                            349*   354*   359*   363*   366*   369*   372*   376*   379*   382*   467*   472*   475*
                            478*   481*   484*   486*   556*   571*   632*   685*
PRINTR    003750            331    334    338    341    345    349    354    359    363    366    369    376    379
                            475    478    481    484    527#
PRINTS    003760            280    289    372    382    472    529#
PSOFT     002700            335    342    355    360    364#
RANDOM    003372             63    203    436#
RTIX      002012            240    242#
R0        =%000000           43#    44    252*   317*
R2        =%000002           46#    65*    71*    90*   209*   438    445*   446*   449*   452*   455*   457*   458*
                            459    464*   568    583    594    598    621    635    654    666    668    699    705
                            707    714    736    738
R4        =%000004           48#    62*    85*    88*   108*   131*   153*   178*   181*   439    448*   447*   463*
                            531    532*   550    555*
R5        =%000005           49#   620*   623*   624
SAVLAD    002316            273    296#   301
SCOPE     = 104000           41#   107    130    152    177    198    230
SECI      002670            327    361#
SECO      002534            329    336#
SER       003030            389#
SINE      004116            565#   574*   577*   595*   596*   599*   604*   606    609    620    645
SP        =%000006           50#    56*
SR1       004114             69     75     94    111*   123    134*   146    159*   170    184*   195    226    571
                            564#   570*   573*   631*   676*   679*   684*   693*   766*   770*
STEP1     004162            570#   614
STEP10    004336            618#   649
STEP11    004456            639    650#
STEP2     004744            700    715    717    725#   744
STEP3     005004            734#
STEP4     005020            733    739#
STEP5     005030            743#
STEP6     004270            602#
STEP8     004320            601    603    612#
STEP9     004402            617    635#
```



```
STP9      004342            620#   667    669
SWR       = 177570           39#    40    233    247    272    276    282    291    304    321
SW10      = 002000           37#   304
SW11      = 004000           36#   272
SW12      = 010000           35#   233
SW13      = 020000           34#   282    321
SW14      = 040000           33#   276
TAD       003614            488#
TEMP      002154            123    146    170    195    226    261*   262*   263    267*   270#   381
TIMES     003636            278    497#
TLOC      003616            489#
TRPB      002156            235*   271#
TTY       =%000003           47#   285*   288*   330*   333*   337*   340*   344*   348*   353*   358*   362*   365*
                            368*   371*   375*   378*   381*   471*   474*   477*   480*   483*   509    510*   511
                            513    522*   523*   524    525*   536*   538*   540*
TYPE      = 000004           32#   251    284    287    290    307    327    332    339    343    347    350    364
                            367    370    373    374    377    380    470    473    476    479    482    485    554
$         004110            562#   578*   582*   585*   586    588*   592    641*   644*   647*   648    650*   653*
                            626*   637    659*   664
.         = 005150           18#    19#    24#    29#    53#    80     83     98    101    118    121    124    141
                            144    147    165    168    171    190    193    196    221    224    227    245    290
                            291#   292    309    385    435#   515    528    543    546    551    558#   576    581
                            584    587    593    597    605    608    613    622    625    634    643    646    652
                            655    658    665    675    678    697    700    727    730    735    737    750    756
                            765    769
.AC       =%000000           44#    68     73     82     87*    92    100    113*   114*   117    136*   137*   140
                            155*   161*   164    180*   186*   189    202*   207    216*   217*   228    375    436
                            440*   444*   455*   459*   461    466*   474    575    580    591*   594*   598*   602
                            616    630    642    651    666*   668*   671    672*   674    677    680    695*   707*
                            708*   711*   718*   719*   722*   736*   738*   739*   745    749    752*   753*   755
                            758*   759*   760    762
.ACX      005144            110*   117    133*   140    157*   164    182*   189    207*   228    365    774#
.EMT      002160             25    272#
.ET       002366            305    308#
.IOT      003662             22    509#
.MQ       =%000001           45#    64*    67     74     79     86*    93     97    120    143    154*   167    179*
                            192    201*   208    215*   223    378    437    441*   445*   450*   451*   454*   456*
                            460*   462    465*   477    590*   619*   626    629    660*   662*   672    673*   682
                            696    703    710*   712*   716    718    721*   723*   729    734    748*   751*   757*
                            764    768
.MQX      005146            109*   120    132*   143    158*   167    183*   192    208*   223    368    779#
.SR1      001476            163    170#
.SR2      001574            188    195#
.SR3      001736            219    226#
.TRP      002344             27    304#
.TRT      003612             20    487#
.TYPE     003632            250*   251    306*   307    346*   347    495#
```

```
DUMP    53#   330   333   336   340   344   348   353   358   361   365   368   375   378   474
       477   480   483
NUMBER  53#   104   127   149   174   212
PRINT   53#   290
SCOPEX  53#   104   127   149   174
SDUMP   53#   285   288   371   381   471
TYPEM   53#   250   306   346
```



```
ADC     450   452   454   455   457   460   752   758
ADD      88   181   242   265   449   451   453   456   458   459   519   522   598   668   736
ASL     444
ASR     739
BCS     605
BEQ      78    80    83    96    98   101   118   121   124   141   144   147   165   168   171
        190   193   196   221   224   227   245   253   275   281   301   305   309   314   318
        329   352   512   543   546   593   603   613   639   649   665   727   730   733   746
        761
BGT     279
BIC     262   267   523
BIS     237   631   676   679   684   766   770
BISB    547
BIT     233   244   247   263   272   276   282   304   521   525   528   551   556   729   734
BITB    313   726
BLE     630
BMI     587   607   625   627   636   748   756   765
BNE     234   248   264   273   277   283   316   322   526   557   585   448   521   549   551
        569   601   617   658   675   681   683   704   715   735   744   763
BPL     236   292   515   576   581   584   610   422   643   646   652   655   678   697   700
        706   717   750   769
BR      103   188   219   335   342   355   360   516   528   534   597   608   628   667   669
        713   724   737   754
CLC     579   661   709   728
CLR      57   103   232   249   294   319   443   529   573   574   578   650   692   693   694
        695   710   721
CLRB    533   535   553   641   725
CMP      82   100   117   120   140   143   164   167   189   192   220   223   278   315   384
        528   550   600   638   703   714   762
CMPB    123   146   170   195   226
COM     235   623   698   701
COMB    577   582   585   596   644   647   653   656   728   731
DIV     101   186   217
EMT      41
HALT     19   293
INC     299   312   386   447   588   612   659   660   743
INCB    544   548
IOT      32
JMP      30   246   258   318   614   767   771
JSR      63    66    72    81    84    91    99   102   115   138   162   187   205   206   218
        254   286   289   320   331   334   338   341   345   349   354   359   363   366   369
        372   376   379   382   472   475   478   481   484
MOV      56    58    59    60    62    64    65    67    68    69    71    73    74    75    85
         86    87    90    92    93    94   108   109   110   111   113   131   132   133   134
        136   153   154   155   157   158   159   178   179   180   182   183   184   199   200
        201   202   204   205   207   208   209   215   216   238   240   243   252   285   288
        295   296   302   311   317   323   330   333   537   340   344   348   353   358   362
        365   368   371   375   378   381   436   437   438   439   440   441   442   461   462
        463   464   465   466   471   474   477   480   485   509   518   518   524   525   531
        532   555   570   572   635   671   672   675   702   707   718
MOVB    250   261   300   346   513   527   530   552   626
MUL     114   137
NEG     637   708   719   751   753   757   759
NOP     110   139   255   256   257
```

```
ROL      445   446   536   538   540   598   591   619   662
ROLB     537   539   541   595   599   741
ROR      711   712   722   723   748
   B     604
R+E      297   303   307   326
RTS      266   268   487   488   556   571   632   685
RTT      239   241   487
SEC      589   618
SUB      324   594   666   738
TRAP      42
TST       77    79    95    97   156   274   280   291   300   306   383   568   575   586   593
          602   616   621   624   626   629   642   651   654   674   677   680   682   696   699
          705   716   745   747   749   755   760   764   768
TST8     511   514   542   545   586   592   686   689   645   648   657   664   732
.ASCII   398   414
.ASCIZ   291   390   393   396   397   400   401   403   404   406   418   419   422   424   426
          428   431   434
.BLKW    558
.ENABL     4
.END     777
.EVEN    291   435
.LIST      2    19    53   104   107   127   130   149   152   174   177   212   215   236   272
          291   304   436   527   568
.MACR     53
.MACRO    53
.NLIST     2    19    53   104   107   127   130   149   152   174   177   212   215   236   272
          291   304   436   527   568
.REPT     19
.SBTTL     2    53   104   127   149   174   212   236   272   304   436   527   568
.TITLE     1
.WORD    562   563   564   565   688   689   698
```

ERRORS DETECTED: 0


```
 .DCQKAB,DCQKAB/SQL/CRF=DCQKAB,P11
 RUN-TIME:  2  5  1 SECONDS
 CORE USED:  5K
```