PDP11

RANDOM EXERCISER

MD-11-DCQKA-C

EP-DCQKA-C-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA

```
                                .REM %
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCQKA-C-D
PRODUCT NAME: MUL - DIV RANDOM EXERCISER
DATE RELEASED: SEPTEMBER, 1976
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BOB BRAIN

TABLE OF CONTENTS

CONTENTS

```
                        DESCRIPTION

                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.
```

```
                                DESCRIPTION

                        5.      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.
```

DESCRIPTION

5.2.4 TRAPCATCHER

A ".+2" - "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.

```
                                DESCRIPTION


                        8.2     STACK POINTER

                                STACK IS INITALLY SET TO 500


                        8.3     POWER FAIL

                                TO USE, START THE TEST AS USUAL AND POWER DOWN  THEN  UP  AT
                                ANY TIME.  THERE SHOULD BE NO ERROR TYPEOUTS.


                        9.      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  MULTIPLICAN SAVING  BOTH
                                ANSWERS.  THE ORIGINAL NUMBERS ARE HARDWARE MULTIPLIED  BOTH
                                WAYS  AND  COMPAIRED  TO  THE  ORIGINAL  ANSWER.  THE  TWO
                                DIVISIONS  ARE  PERFORMED  AND  COMPAIRED  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.

                        %
```

```
                                .TITLE  MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER
                                ;COPYRIGHT 1972, 1976 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  003640                                  .TRT
000016  000000                                  0
000020  003710                                  .IOT
000022  000340                                  340
                        000030          .=30
000030  002206                                  .EMT
000032  000000                                  0
000034  002372                                  .TRP
000036  000000                                  0
                        000046                  .=46
000046  002124                                  $ENDAD
                        000052                  .=52
000052  000000                                  0
                        000200                  .=200
000200  000167  000576                          JMP     BEGIN

                        000004          TYPE=   IOT
                        040000          SW14=   40000
                        020000          SW13=   20000
                        010000          SW12=   10000
                        004000          SW11=    4000
                        002000          SW10=    2000
                        000001          N=      1
                        177570          SWR=    177570
                        177570          DISPLAY=SWR
                        104000          SCOPE=  EMT
                        104400          HLT=    TRAP
                        000000          R0=     %0
                        000000          .AC=    R0
                        000001          .MQ=    %1
                        000002          R2=     %2
                        000003          TTY=    %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  002650          1$:     CLR     ERRORS          ;ERROR COUNT
001012  012767  000001  002632          MOV     #1,CNT
001020  026737  001100  000042          CMP     $ENDAD,@#42     ;ARE WE IN AUTOMATIC MODE?
001026  001006                          BNE     .+16            ;NO, BRANCH
001030  012767  000060  002614          MOV     #60,CNT         ;ALLOW SHORTER PASS TIME FOR AUTO MODE
001036  012767  000777  001142          MOV     #777,.EMT       ;BRANCH-SELF ON ERROR SO PGM. TIMES OUT IN AUTO MODE
001044  012767  123456  002574          MOV     #123456,LONUM   ;INITALIZE LONUM FOR RANDOM
001052  012767  176543  002570          MOV     #176543,HINUM   ;INITALIZE HINUM FOR RANDOM
                
001060  012704  003666          GO:     MOV     #AREA,R4        ;SET POINTER TO ANSWER AREA
001064  004767  002330                  JSR     PC,RANDOM       ;GET RANDOM NUMBERS
001070  016701  002552                  MOV     LONUM,.MQ       ;LOAD MULTIPLIER (SOFTWARE)
001074  016702  002550                  MOV     HINUM,R2        ;LOAD MULTIPLICAN (SOFTWARE)
001100  004767  003542                  JSR     PC,MUL1         ;MULTIPLY
001104  010124                          MOV     .MQ,(4)+        ;SAVE MQ
001106  010024                          MOV     .AC,(4)+        ;SAVE AC
001110  016724  003026                  MOV     SR1,(4)+        ;SAVE SR
                
001114  016702  002526                  MOV     LONUM,R2        ;LOAD DIVISOR
001120  004767  003022                  JSR     PC,DIVI         ;DIVIDE BY MULTIPLIER (SOFTWARE)
001124  010024                          MOV     .AC,(4)+        ;SAVE AC
001126  010124                          MOV     .MQ,(4)+        ;SAVE MQ
001130  016724  003006                  MOV     SR1,(4)+        ;SAVE SR
                
001134  005767  002506                  TST     LONUM
001140  001411                          BEQ     OK1
001142  005701                          TST     .MQ
001144  001402                          BEQ     .+6
001146  004767  002372                  JSR     PC,BAD
001152  020067  002472                  CMP     .AC,HINUM
001156  001402                          BEQ     .+6
001160  004767  002360                  JSR     PC,BAD
001164  012704  003666          OK1:    MOV     #AREA,R4        ;RESET POINTER
001170  012401                          MOV     (4)+,.MQ        ;RESTORE MQ
001172  012400                          MOV     (4)+,.AC        ;RESTORE AC
001174  062704  000010                  ADD     #10,R4          ;INCREMENT POINTER
                
001200  016702  002444                  MOV     HINUM,R2        ;LOAD DIVISOR
001204  004767  002736                  JSR     PC,DIVI         ;DIVIDE BY MULTIPLICAN (SOFTWARE)
001210  010024                          MOV     .AC,(4)+        ;SAVE AC
001212  010124                          MOV     .MQ,(4)+        ;SAVE MQ
001214  016714  002722                  MOV     SR1,(4)         ;SAVE SR
001220  005767  002424                  TST     HINUM
001224  001411                          BEQ     ITER
001226  005701                          TST     .MQ
001230  001402                          BEQ     .+6
001232  004767  002306                  JSR     PC,BAD
001236  020067  002404                  CMP     .AC,LONUM
001242  001402                          BEQ     .+6
001244  004767  002274                  JSR     PC,BAD
001250  005067  177524          ITER:   CLR     ICNT            ;SET UP COUNT
```

```
                                        ;*************************************************************
                                        ;TEST 1         LONUM X HINUM
                                        ;*************************************************************
001254  104000                                  SCOPE
001256  012704  003666                          MOV     #AREA,R4        ;RESTORE POINTER
001262  012467  003706                          MOV     (4)+,.MQX       ;RESTORE MQ
001266  012467  003700                          MOV     (4)+,.ACX       ;RESTORE AC
001272  011467  002644                          MOV     (4),SR1         ;RESTORE SR1

001276  016700  002344                          MOV     LONUM,.AC       ;LOAD MULTIPLIER (HARDWARE)
001302  070067  002342                          MUL     HINUM,.AC       ;LOAD MULTIPLICAN AND MULTIPLY
001306  004767  000626                          JSR     PC,FLOW
001312  000240                                  NOP
001314  026700  003652                          CMP     .ACX,.AC        ;CHECK HIGH ORDER PRODUCT
001320  001401                                  BEQ     .+4             ;SKIP IF OK
001322  104400                                  HLT
001324  026701  003644                          CMP     .MQX,.MQ        ;CHECK LOW ORDER PRODUCT
001330  001401                                  BEQ     .+4             ;SKIP IF OK
001332  104400                                  HLT
001334  126767  002602  000640                  CMPB    SR1,TEMP        ;CHECK STATUS REGISTER
001342  001401                                  BEQ     .+4             ;SKIP IF GOOD
001344  104400                                  HLT

                                        ;*************************************************************
                                        ;TEST 2         HINUM X LONUM
                                        ;*************************************************************
001346  104000                                  SCOPE
001350  012704  003666                          MOV     #AREA,R4        ;RESTORE POINTER
001354  012467  003614                          MOV     (4)+,.MQX       ;RESTORE MQ
001360  012467  003606                          MOV     (4)+,.ACX       ;RESTORE AC
001364  011467  002552                          MOV     (4),SR1         ;RESTORE SR1

001370  016700  002254                          MOV     HINUM,.AC       ;LOAD MULTIPLIER (HARDWARE)
001374  070067  002246                          MUL     LONUM,.AC       ;MULTIPLY
001400  004767  000534                          JSR     PC,FLOW
001404  000240                                  NOP
001406  026700  003560                          CMP     .ACX,.AC        ;CHECK HIGH ORDER PRODUCT
001412  001401                                  BEQ     .+4             ;SKIP IF OK
001414  104402                                  HLT+2
001416  026701  003552                          CMP     .MQX,.MQ        ;CHECK LOW ORDER PRODUCT
001422  001401                                  BEQ     .+4             ;SKIP IF OK
001424  104402                                  HLT+2
001426  126767  002510  000546                  CMPB    SR1,TEMP        ;CHECK STATUS REGISTER
001434  001401                                  BEQ     .+4             ;SKIP IF GOOD
001436  104402                                  HLT+2
```



```
                                        ;*************************************************************
                                        ;TEST 3         RESULT / LONUM
                                        ;*************************************************************
001440  104000                                  SCOPE
001442  012704  003666                          MOV     #AREA,R4        ;RESTORE POINTER
001446  012401                                  MOV     (4)+,.MQ        ;RESTORE MQ
001450  012400                                  MOV     (4)+,.AC        ;RESTORE AC
001452  005724                                  TST     (4)+            ;INTREMENT POINTER
001454  012467  003512                          MOV     (4)+,.ACX       ;RESTORE AC
001460  012467  003510                          MOV     (4)+,.MQX       ;RESTORE MQ
001464  011467  002452                          MOV     (4),SR1         ;RESTORE SR1
```

```
463  001470  071067  002152                    DIV     LONUM,.AC       ;DIVIDE BY MULTIPLIER (HARDWARE)
464  001474  004767  000440                    JSR     PC,FLOW
465  001500  000410                           BR      .SR1
466  001502  026700  003464                    CMP     .ACX,.AC        ;CHECK REMAINDER
467  001506  001401                           BEQ     .+4             ;SKIP IF OK
468  001510  104401                           HLT+1
469  001512  026701  003456                    CMP     .MQX,.MQ        ;CHECK QUOTENT
470  001516  001401                           BEQ     .+4             ;SKIP IF OK
471  001520  104401                           HLT+1
472  001522  126767  002414  000452  .SR1:    CMPB    SR1,TEMP        ;CHECK STATUS REGISTER
473  001530  001401                           BEQ     .+4             ;SKIP IF OK
474  001532  104401                           HLT+1
475
476                                   ;*****************************************************************
477                                   ;TEST 4         RESULT / HINUM
478                                   ;*****************************************************************
479  001534  104000                           SCOPE
480  001536  012704  003666                    MOV     #AREA,R4        ;RESTORE POINTER
481  001542  012401                           MOV     (4)+,.MQ        ;RESTORE MQ
482  001544  012400                           MOV     (4)+,.AC        ;RESTORE AC
483  001546  062704  000010                    ADD     #10,R4          ;INCREMENT POINTER
484  001552  012467  003414                    MOV     (4)+,.ACX       ;RESTORE AC
485  001556  012467  003412                    MOV     (4)+,.MQX       ;RESTORE MQ
486  001562  011467  002354                    MOV     (4),SR1         ;RESTORE SR1
487
488  001566  071067  002056                    DIV     HINUM,.AC       ;DIVIDE BY MULTIPLICAN (HARDWARE)
489  001572  004767  000342                    JSR     PC,FLOW
490  001576  000410                           BR      .SR2
491  001600  026700  003366                    CMP     .ACX,.AC        ;CHECK REMAINDER
492  001604  001401                           BEQ     .+4             ;SKIP IF OK
493  001606  104403                           HLT+3
494  001610  026701  003360                    CMP     .MQX,.MQ        ;CHECK QUOTENT
495  001614  001401                           BEQ     .+4             ;SKIP IF OK
496  001616  104403                           HLT+3
497  001620  126767  002316  000354  .SR2:    CMPB    SR1,TEMP        ;CHECK STATUS REGISTER
498  001626  001401                           BEQ     .+4             ;SKIP IF OK
499  001630  104403                           HLT+3
```



```
500  001632  104000                           SCOPE                           ;CALCULATE NEW RANDOM DIVISOR
501  001634  016767  002006  002024           MOV     LONUM,AREA              ;GET MQ
502  001642  016767  002002  002020           MOV     HINUM,AREA+2            ;GET AC
503  001650  016701  001772                    MOV     LONUM,.MQ               ;LOAD MQ (SOFTWARE)
504  001654  016700  001770                    MOV     HINUM,.AC               ;LOAD AC (SOFTWARE)
505  001660  004767  001534                    JSR     PC,RANDOM               ;GET 1 RANDOM NUMBER
506  001664  016767  001756  002000           MOV     LONUM,AREA+4            ;GET DIVISOR
507  001672  016702  001750                    MOV     LONUM,R2                ;LOAD THE DIVISOR
508  001676  004767  002244                    JSR     PC,DIVI                 ;DIVIDE
509  001702  010067  003264                    MOV     .AC,.ACX
510  001706  010167  003262                    MOV     .MQ,.MQX
511  001712  012767  001720  001736           MOV     #LOCX,LAD               ;FUDGE LAD
512
513
514                                   ;*****************************************************************
515                                   ;TEST 5         LONUM HINUM / NEW LONUM
516                                   ;*****************************************************************
517  001720  016701  001742           LOCX:   MOV     AREA,.MQ                ;RESTORE MQ
518  001724  016700  001740                   MOV     AREA+2,.AC              ;RESTORE AC
519  001730  071067  001736                   DIV     AREA+4,.AC              ;DIVIDE
520  001734  004767  000200                   JSR     PC,FLOW
521  001740  000410                           BR      .SR3
522  001742  026700  003224                   CMP     .ACX,.AC                ;CHECK AC
```

```
                                                                 .401
  523  001746  001401                          BEQ
  524  001750  104407                          HLT+7                    ;REMAINDER WRONG
  525  001752  026701  003216                  CMP     .MQX,.MQ         ;CHECK QUOTIENT
  526  001756  001401                          BEQ     .+4
  527  001760  104407                          HLT+7                    ;QUOTIENT WRONG
  528  001762  126767  002154  000212  .SR3:   CMPB    SR1,TEMP         ;CHECK STATUS
  529  001770  001401                          BEQ     .+4
  530  001772  104407                          HLT+7
  531

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 12
DCQKAC.SRC      T BIT AND LOOPING

  532  001774  104000                          SCOPE
  533
  534  001776  005046                          CLR     -(6)             ;CLEAR TRACE TRAP
  535  002000  032767  010000  175562          BIT     #SW12,SWR        ;RUN WITH TRT?  *SW12*
  536  002006  001010                          BNE     2$
  537  002010  005167  000170                  COM     TRPB
  538  002014  100005                          BPL     2$
  539  002016  052716  000020                  BIS     #20,(6)          ;SET TRACE TRAP
  540  002022  012746  001060                  MOV     #GO,-(6)         ;JUMP TO START OF TEST
  541  002026  000006                          RTT
  542  002030  012746  002036          2$:     MOV     #RTIX,-(6)       ;JUMP TO START OF TEST
  543  002034  000006                          RTT
  544  002036  062767  000001  001606  RTIX:   ADD     #1,CNT           ;COUNT NUMBER OF PASSES
  545  002044  016737  001602  177570          MOV     CNT,@#DISPLAY
  546  002052  032767  000177  001572          BIT     #177,CNT         ;CHECK FOR MULTIPLES OF 200
  547  002060  001402                          BEQ     .+6
  548  002062  000167  176772          DOIT:   JMP     GO               ;RERUN
  549  002066  032767  002000  175474          BIT     #2000,SWR
  550  002074  001372                          BNE     DOIT
  551  002076  005067  001550                  CLR     CNT
  552  002102  112767  000007  001550          MOVB    #BELL,.TYPE      ;TYPE A BELL
  553  002110  000004  003660                  TYPE    ,.TYPE
  554  002114  013700  000042                  MOV     @#42,R0          ;GET MONITOR ADDRESS (IF ANY)
  555  002120  001405                          BEQ     AGAIN
  556  002122  000005                          RESET
  557  002124  004710                  $ENDAD: JSR     7,(0)            ;CALL THE MONITOR
  558  002126  000240                          NOP
  559  002130  000240                          NOP
  560  002132  000240                          NOP
  561  002134  000167  176720          AGAIN:  JMP     GO               ;RERUN
  562
  563
  564  002140  113767  177776  000034  FLOW:   MOVB    @#177776,TEMP    ;GET THE PS BITS
  565  002146  042767  177760  000026          BIC     #177760,TEMP
  566  002154  032767  000002  000020          BIT     #2,TEMP
  567  002162  001003                          BNE     EQZ
  568  002164  062716  000002                  ADD     #2,(6)
  569  002170  000207                          RTS     PC
  570  002172  042767  000014  000002  EQZ:    BIC     #14,TEMP
  571  002200  000207                          RTS     PC
  572
  573  002202  000000                  TEMP:   0
  574  002204  000000                  TRPB:   0

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 13
DCQKAC.SRC      SCOPE ROUTINE

  575  002206  032737  004000  177570  .EMT:   BIT     #SW11,@#SWR      ;KILL ITERATIONS
  576  002214  001053                          BNE     SAVLAD
  577  002216  005767  176556                  TST     ICNT
  578  002222  001445                          BEQ     ICINT            ;BRANCH IF FIRST
```

```
                                                                MQ1
 579  002224  032737  040000  177570          BIT     #SW14,@#SWR     ;KILL ITERATIONS
 580  002232  001047                          BNE     KIT             ;YES
 581  002234  026767  001424  176536          CMP     TIMES,ICNT      ;DONE?
 582  002242  003043                          BGT     KIT             ;BRANCH IF NOT
 583  002244  005767  001404                  TST     ERCNT           ;ANY ERRORS?
 584  002250  001432                          BEQ     ICINT
 585  002252  032737  020000  177570          BIT     #SW13,@#SWR     ;INHIBIT TYPEOUTS?
 586  002260  001024                          BNE     NOTY
 587  002262  000004  003202                  TYPE    ,ER6
 588  002266  016703  001362                  MOV     ERCNT,TTY       ;TYPE ERCNT IN OCTAL
 589  002272  004767  001510                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
 590  002276  000004  003226                  TYPE    ,ER7
 591  002302  016703  176472                  MOV     ICNT,TTY        ;TYPE ICNT IN OCTAL


^A^HPRINTS      ;AND SUPRESS LEADING ZERO'S
 593  002312  000004  002316                  TYPE    .+2             ;.ASCIZ <15><12>
 594  002322  005737  177570                  TST     @#SWR           ;HALT ON ERROR  *SW15*
 595  002326  100001                          BPL     .+4
 596  002330  000000                          HALT
 597  002332  005067  001316          NOTY:   CLR     ERCNT
 598  002336  012767  000001  176434  ICINT:  MOV     #1,ICNT         ;FIRST ITERATION
 599  002344  011667  001306          SAVLAD: MOV     (6),LAD         ;SAVE LOOP ADDRESS
 600  002350  000002                          RTI                     ;RETURN
 601
 602  002352  005267  176422          KIT:    INC     ICNT
 603  002356  005767  001274          OVR:    TST     LAD             ;FIRST ONE?
 604  002362  001770                          BEQ     SAVLAD
 605  002364  016716  001266                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
 606  002370  000002                          RTI                     ;FIXES PS

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 14
DCQKAC.SRC      ERROR ROUTINES

 607  002372  032737  002000  177570  .TRP:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
 608  002400  001405                          BEQ     .ET
 609  002402  112767  000007  001250          MOVB    #BELL,.TYPE     ;TYPE A BELL
 610  002410  000004  003660                  TYPE    .TYPE
 611  002414  005767  001234          .ET:    TST     ERCNT           ;CHECK FOR FIRST ERROR
 612  002420  001402                          BEQ     .+6
 613  002422  000167  000412          NOH:    JMP     NOHEAD
 614  002426  011667  000422                  MOV     (6),HLTAD       ;SAVE ADDRESS OF HLT
 615  002432  005267  001224                  INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
 616  002436  132737  000001  000041          BITB    #1,@#41         ;AUTO RUN?
 617  002444  001412                          BEQ     1$              ;NO!
 618  002446  022767  000010  001206          CMP     #10,ERRORS      ;TOO MANY ERRORS
 619  002454  001006                          BNE     1$              ;NOT YET!
 620  002456  013700  000042                  MOV     @#42,R0         ;GET ADDRESS
 621  002462  001403                          BEQ     1$              ;NONE???
 622  002464  005037  000042                  CLR     @#42            ;ZERO IT
 623  002470  004710                          JSR     PC,(0)          ;RETURN TO MONITOR
 624  002472  032767  020000  175070  1$:     BIT     #SW13,SWR       ;SKIP TYPEOUT IF SET
 625  002500  001350                          BNE     NOH
 626  002502  011646                          MOV     (6),-(6)        ;PUT ADDRESS OF INSTRUCTION ON STACK
 627  002504  162716  000002                  SUB     #2,(6)          ;GET THE INSTRUCTION
 628  002510  032776  000001  000000          BIT     #1,@(6)         ;TEST FOR MULTIPLY OR DIVIDE
 629  002516  001034                          BNE     DIVP            ;GOTO DIVIDE PRINT IF ON A 1
```

NQ1

```
002520  000004  003101                    TYPE    ,ER1
002524  032776  000002  000000            BIT     #2,@(6)         ;TEST FOR WHICH MULTIPLY
002532  001413                            BEQ     SECO
002534  016703  001106                    MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
002540  004767  001232                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002544  000004  003122                    TYPE    ,ER2
002550  016703  001074                    MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
002554  004767  001216                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002560  000462                            BR      PSOFT
002562                            SECO:
002562  016703  001062                    MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
002566  004767  001204                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002572  000004  003122                    TYPE    ,ER2
002576  016703  001044                    MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
002602  004767  001170                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002606  000447                            BR      PSOFT
```



```
002610  000004  003060            DIVP:   TYPE    ,ER1D
002614  016703  001050                    MOV     AREA+2,TTY      ;TYPE AREA+2 IN OCTAL
002620  004767  001152                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002624  112767  000040  001026            MOVB    #40,.TYPE       ;TYPE A 40
002632  000004  003660                    TYPE    ,.TYPE
002636  016703  001024                    MOV     AREA,TTY        ;TYPE AREA IN OCTAL
002642  004767  001130                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002646  000004  003126                    TYPE    ,ER2A
002652  032776  000004  000000            BIT     #4,@(6)
002660  001405                            BEQ     DIVZ
002662  016703  001004                    MOV     AREA+4,TTY      ;TYPE AREA+4 IN OCTAL
002666  004767  001104                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002672  000415                            BR      PSOFT
002674  032776  000002  000000    DIVZ:   BIT     #2,@(6)         ;TEST FOR WHICH DIVIDE
002702  001005                            BNE     SECI
002704  016703  000736                    MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
002710  004767  001062                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002714  000404                            BR      PSOFT
002716                            SECI:
002716  016703  000726                    MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
002722  004767  001050                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002726  000004  003132            PSOFT:  TYPE    ,ER3
002732  016703  002234                    MOV     .ACX,TTY        ;TYPE .ACX IN OCTAL
002736  004767  001034                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002742  000004  003165                    TYPE    ,ER4
002746  016703  002222                    MOV     .MQX,TTY        ;TYPE .MQX IN OCTAL
002752  004767  001020                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
002756  000004  003172                    TYPE    ,ER5
002762  016703  001154                    MOV     SR1,TTY         ;TYPE SR1 IN OCTAL
002766  004767  001014                    JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
002772  000004  003152                    TYPE    ,ER3B
002776  000004  003145                    TYPE    ,ER3A
003002  010003                            MOV     .AC,TTY         ;TYPE .AC IN OCTAL
003004  004767  000766                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003010  000004  003165                    TYPE    ,ER4
003014  010103                            MOV     .MQ,TTY         ;TYPE .MQ IN OCTAL
003016  004767  000754                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003022  000004  003172                    TYPE    ,ER5
003026  016703  177150                    MOV     TEMP,TTY        ;TYPE TEMP IN OCTAL
003032  004767  000750                    JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
003036  005726                            TST     (6)+            ;RESTORE POINTER
003040  021667  000010            NOHEAD: CMP     (6),HLTAD       ;CHECK FOR SAME HLT
003044  001002                            BNE     .+6
003046  005267  000602                    INC     ERCNT
```

```
   690  003052  000002                                RTI

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 16
DCQKAC.SRC      ERROR ROUTINES

   691  003054  000000                        MLTAD:  0
   692  003056  000000                        SER:    0
   693  003060  005015  042012  053111        ER1D:   .ASCIZ  <15><12><12>"DIV ERROR:   "
   694  003066  042440  051122  051117
   695  003074  020072  020040      000
   696  003101     015  005012  052515        ER1M:   .ASCIZ  <15><12><12>"MUL ERROR:   "
   697  003106  020114  051105  047522
   698  003114  035122  020040  000040
   699  003122  054040  000040                ER2:    .ASCIZ  " X "
   700  003126  027440  000040                ER2A:   .ASCIZ  " / "
   701  003132  005015  051412  043117        ER3:    .ASCII  <15><12><12>"SOFTWARE"
   702  003140  053524  051101      105
   703  003145     040  020040  000040        ER3A:   .ASCIZ  "    "
   704  003152  005015  040510  042122        ER3B:   .ASCIZ  <15><12>"HARDWARE"
   705  003160  040527  042522      000
   706  003165     040  020040  000040        ER4:    .ASCIZ  "    "
   707  003172  020040  020040  051520        ER5:    .ASCIZ  "    PS="
   708  003200  000075
   709  003202  005015  042412  051122        ER6:    .ASCIZ  <15><12><12>"ERRORS OCCURRED "
   710  003210  051117  020123  041517
   711  003216  052503  051122  042105
   712  003224  000040
   713  003226  052040  046511  051505        ER7:    .ASCIZ  " TIMES OUT OF "
   714  003234  047440  052125  047440
   715  003242  020106     000
   716  003245     015  000012                CRL:    .ASCIZ  <15><12>
   717  003250  005015  047523  052106        ERNG:   .ASCII  <15><12>"SOFTWARE DIVIDE DOESN'T MATCH REAL ANSWER.  "
   718  003256  040527  042522  042040
   719  003264  053111  042111  020105
   720  003272  047524  051505  023516
   721  003300  020124  040515  041524
   722  003306  020110  042522  046101
   723  003314  040440  051516  042527
   724  003322  027122  020040
   725  003326  047514  020103  020075        E1:     .ASCIZ  "LOC = "
   726  003334     000
   727  003335     015  027012  041501        E2:     .ASCIZ  <15><12>".AC = "
   728  003342  036440  000040
   729  003346  020040  020040  046456        E3:     .ASCIZ  "    .MQ = "
   730  003354  020121  020075      000
   731  003361     040  020040  046040        E4:     .ASCIZ  "    LONUM = "
   732  003366  047117  046525  036440
   733  003374  000040
   734  003376  020040  020040  044510        E5:     .ASCIZ  "    HINUM = "
   735  003404  052516  020115  020075
   736  003412     000
   737  003413     015  005012      000        E6:     .ASCIZ  <15><12><12>
   738          003420                              .EVEN

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 17
DCQKAC.SRC      RANDOM NUMBER GENERATOR

   739  003420  010046                        RANDOM: MOV     .AC,-(6)        ;SAVE R0
   740  003422  010146                                MOV     .MQ,-(6)        ;SAVE R1
   741  003424  010246                                MOV     R2,-(6)         ;SAVE R2
   742  003426  010446                                MOV     R4,-(6)         ;SAVE R4
   743  003430  016700  000212                        MOV     LONUM,.AC       ;SET R0 WITH LOW
   744  003434  016701  000210                        MOV     HINUM,.MQ       ;SET R1 WITH HIGH
   745  003440  012704  177771                        MOV     #-7,R4          ;SET SHIFT COUNT
```

C02

```
003444  005002                   CLR     R2
003446  006300          1$:      ASL     .AC             ;SHIFT R0 LEFT AND
003450  006101                   ROL     .MQ             ;ROTATE CARRY INTO R1 AND
003452  006102                   ROL     R2              ;ROTATE CARRY INTO R2
003454  005204                   INC     R4              ;CHECK FOR DONE
003456  001373                   BNE     1$              ;CONTINUE SHIFT LOOP
003460  066702  000162           ADD     LONUM,R2        ;ADD NUMBER TO MAKE X 129
003464  005501                   ADC     .MQ             ;PROPOGATE CARRY
003466  066701  000156           ADD     HINUM,.MQ       ;ADD NUMBER TO MAKE X 129
003472  005502                   ADC     R2              ;PROPOGATE CARRY
003474  062700  001057           ADD     #1057,.AC       ;ADD LOW CONSTANT
003500  005501                   ADC     .MQ             ;PROPOGATE CARRY
003502  005502                   ADC     R2              ;PROPOGATE CARRY
003504  062701  047401           ADD     #47401,.MQ      ;ADD HIGH CONSTANT
003510  005502                   ADC     R2              ;PROPOGATE CARRY
003512  062702  000006           ADD     #6,R2           ;ADD HIGHEST CONSTART
003516  060200                   ADD     R2,.AC          ;REPRIME R0 WITH HIGHEST DIGIT
003520  005501                   ADC     .MQ             ;PROPOGATE CARRY
003522  010067  000120           MOV     .AC,LONUM       ;SAVE R0
003526  010167  000116           MOV     .MQ,HINUM       ;SAVE R1
003532  012604                   MOV     (6)+,R4         ;RESTORE R4
003534  012602                   MOV     (6)+,R2         ;RESTORE R2
003536  012601                   MOV     (6)+,.MQ        ;RESTORE R1
003540  012600                   MOV     (6)+,.AC        ;RESTORE R0
003542  000207                   RTS     PC              ;RETURN


003544  000004  003250  BAD:     TYPE    ,ERNG
003550  011603                   MOV     (6),TTY         ;TYPE (6) IN OCTAL
003552  004767  000230           JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
003556  000004  003335           TYPE    ,E2
003562  010003                   MOV     .AC,TTY         ;TYPE .AC IN OCTAL
003564  004767  000206           JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003570  000004  003346           TYPE    ,E3
003574  010103                   MOV     .MQ,TTY         ;TYPE .MQ IN OCTAL
003576  004767  000174           JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003602  000004  003361           TYPE    ,E4
003606  016703  000034           MOV     LONUM,TTY       ;TYPE LONUM IN OCTAL
003612  004767  000160           JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003616  000004  003376           TYPE    ,E5
003622  016703  000022           MOV     HINUM,TTY       ;TYPE HINUM IN OCTAL
003626  004767  000144           JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
003632  000004  003413           TYPE    ,E6
003636  000207                   RTS     PC
```



```
003640  000006          .TRT:    RTT
003642  000000          TAD:     0               ;TRAP ADDRESS
003644  000000          TLOC:    0               ;TRAP LOCATION
003646  000000          LONUM:   0               ;RANDOM NUMBER #1
003650  000000          HINUM:   0               ;RANDOM NUMBER #2
003652  000000          CNT:     0               ;PASS NUMBER
003654  000000          ERCNT:   0               ;ERROR COUNT
003656  000000          LAD:     0               ;LOOP ADDRESS
003660  000000          .TYPE:   0               ;CHARACTER TYPING LOCATION
003662  000000          ERRORS:  0               ;ERRORS ENCOUNTERED
003664  000400          TIMES:   256.

003666  000000          AREA:    0               ;MQ (MULTIPLY)          DIVIDEND RIGHT
003670  000000                   0               ;AC (MULTIPLY)          DIVIDEND LEFT
003672  000000                   0               ;SR (MULTIPLY)          DIVISOR
003674  000000                   0               ;AC (DIVIDE BY LONUM)
```

```
        003676  000000                                  0                       ;MQ (DIVIDE BY LONUM)
        003700  000000                                  0                       ;SR (DIVIDE BY LONUM)
        003702  000000                                  0                       ;AC (DIVIDE BY HINUM)
        003704  000000                                  0                       ;MQ (DIVIDE BY HINUM)
        003706  000000                                  0                       ;SR (DIVIDE BY HINUM)

        003710  010346                          .IOT:   MOV     TTY,-(6)        ;SAVE TTY
        003712  017603  000002                          MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
        003716  105713                          1$:     TSTB    (TTY)           ;TERMINATOR?
        003720  001406                                  BEQ     2$
        003722  112337  177566                          MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
        003726  105737  177564                          TSTB    @#177564        ;IS THE PRINTER READY
        003732  100375                                  BPL     .-4
        003734  000770                                  BR      1$              ;GET THE NEXT CHARACTER

        003736  017646  000002                  2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
        003742  062766  000002  000004                  ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
        003750  022666  000002                          CMP     (6)+,2(6)       ;IS IT .+2?
        003754  001006                                  BNE     3$              ;NO
        003756  062703  000002                          ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
        003762  042703  000001                          BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
        003766  010366  000002                          MOV     TTY,2(6)        ;RESTORE ADDRESS
        003772  012603                          3$:     MOV     (6)+,TTY        ;RESTORE TTY
        003774  000002                                  RTI                     ;RETURN
MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 19
DCQKAC.SRC      OCTAL DUMP OF A WORD

        003776  112767  000001  000130  PRINTR: MOVB    #1,A4$          ;SET ZERO FILL SWITCH
        004004  000402                                  BR      .+6
        004006  005067  000122          PRINTS: CLR     A4$             ;SUPRESS LEADING ZERO'S
        004012  112767  177772  000115          MOVB    #-6,A4$+1       ;SET COUNT
        004020  010446                                  MOV     R4,-(6)         ;SAVE R4
        004022  012704  004124                          MOV     #3$,R4          ;SET POINTER TO FIRST ASCII CHAR.
        004026  105014                                  CLRB    (4)             ;CLEAR FIRST BYTE
        004030  000405                                  BR      2$              ;ROTATE FIRST BIT
        004032  105014                          1$:     CLRB    (4)             ;CLEAR BYTE OF CHARACTER
        004034  006103                                  ROL     TTY             ;ROTATE BIT INTO C
        004036  106114                                  ROLB    (4)             ;PACK IT
        004040  006103                                  ROL     TTY             ;ROTATE BIT INTO C
        004042  106114                                  ROLB    (4)             ;PACK IT
        004044  006103                          2$:     ROL     TTY             ;ROTATE BIT INTO C
        004046  106114                                  ROLB    (4)             ;PACK IT
        004050  105714                                  TSTB    (4)
        004052  001402                                  BEQ     .+6
        004054  105267  000054                          INCB    A4$
        004060  105767  000050                          TSTB    A4$             ;CHECK FILL SWITCH
        004064  001402                                  BEQ     .+6
        004066  152724  000060                          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
        004072  105267  000037                          INCB    A4$+1
        004076  001355                                  BNE     1$              ;REPEAT
        004100  022704  004124                          CMP     #3$,R4
        004104  001002                                  BNE     .+6
        004106  112724  000060                          MOVB    #'0,(4)+
        004112  105014                                  CLRB    (4)
        004114  000004  004124                          TYPE    ,3$             ;TYPE IT
        004120  012604                                  MOV     (6)+,R4         ;RESTORE R4
        004122  000207                                  RTS     PC

        004124  000004                          3$:     .BLKW   4
        004134  000000                          A4$:    0
MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 20
DCQKAC.SRC      MULTIPLY AND DIVIDE ROUTINES
```

```
                                        ;       DIVIDE SUBROUTINE
                                        ;
004136  000000                  Z:     .WORD   0
004140  000000                  COUNT: .WORD   0
004142  000000                  SR1:   .WORD   0
004144  000000                  SINE:  .WORD   0
                                        ;
004146  005702                  DIVI:  TST     R2
004150  001004                         BNE     DIVL
004152  012767  000003  177762         MOV     #3,SR1
004160  000207                         RTS     PC
004162  012767  177760  177750 DIVL:  MOV     #-16.,COUNT     ;SET UP STEP COUNTER
004170  005067  177746                 CLR     SR1
004174  005067  177744                 CLR     SINE
004200  005700                         TST     .AC
004202  100002                         BPL     .+6
004204  105167  177734                 COMB    SINE
004210  005067  177722         STEP1:  CLR     Z               ;INITIALIZE Z INDICATORS
004214  000241                         CLC
004216  005700                         TST     .AC             ;CHECK REMAINDER SIGN
004220  100002                         BPL     .+6
004222  105167  177711                 COMB    Z+1
004226  005702                         TST     R2              ;CHECK DIVISOR SIGN
004230  100002                         BPL     .+6
004232  105167  177701                 COMB    Z+1
004236  105767  177675                 TSTB    Z+1             ;-1 IF DIFFERENT, 0 IF SAME
004242  100403                         BMI     .+8.
004244  005267  177666                 INC     Z               ;SET Z=1 IF SAME
004250  000261                         SEC                     ;SET C TO A 1
004252  006101                         ROL     .MQ             ;ROTATE AC AND MQ ONCE LEFT  STEP2
004254  006100                         ROL     .AC
004256  105767  177654                 TSTB    Z               ;STEP3
004262  001406                         BEQ     .+14.
004264  160200                         SUB     R2,.AC          ;SUBTRACT IF Z = 1
004266  106167  177653                 ROLB    SINE+1
004272  105167  177647                 COMB    SINE+1
004276  000403                         BR      .+8.
004300  060200                         ADD     R2,.AC          ;ADD IF Z = 0
004302  106167  177637                 ROLB    SINE+1
004306  022767  177760  177624         CMP     #-16.,COUNT     ;CHECK FOR FIRST TIME THROUGH    STEP4
004314  001014                         BNE     STEP8
```



```
004316  005700          STEP6:  TST     .AC
004320  001412                  BEQ     STEP8
004322  106067  177617          RORB    SINE+1
004326  103404                  BCS     .+10.           ;IF C AND DIVIDEND ARE
004330  105767  177610          TSTB    SINE
004334  100431                  BMI     OFLW            ;       DIFFERENT, AN OVERFLOW
004336  000403                  BR      .+8.            ;       HAS OCCURRED
004340  105767  177600          TSTB    SINE
004344  100025                  BPL     OFLW
                                ;
004346  005267  177566  STEP8:  INC     COUNT
004352  001402                  BEQ     .+6
004354  000167  177630          JMP     STEP1
                                ;
004360  005700                  TST     .AC             ;BEGINNING OF CORRECTION ROUTINE
004362  001022                  BNE     STEP9           ;IF THE REMAINDER IS ZERO,
004364  000261          STEP10: SEC                     ;       SHIFT THE QUOTIENT ONCE
```

```
 922  004366  006101                          ROL     .MQ             ;       FILLING WITH "1"
 923  004370  116705  177550          STP9:   MOVB    SINE,R5
 924  004374  005702                          TST     R2
 925  004376  100001                          BPL     .+4
 926  004400  005105                          COM     R5
 927  004402  005705                          TST     R5
 928  004404  100403                          BMI     .+8.
 929  004406  005701                          TST     .MQ
 930  004410  100403                          BMI     OFLW
 931  004412  000465                          BR      DONX
 932  004414  005701                          TST     .MQ
 933  004416  003463                          BLE     DONX
 934  004420  052767  000002  177514  OFLW:   BIS     #2,SR1
 935  004426  000207                          RTS     PC
 936
 937
 938  004430  010267  177504          STEP9:  MOV     R2,COUNT
 939  004434  100402                          BMI     .+6
 940  004436  005467  177476                  NEG     COUNT           ;IF THE DIVISOR WAS (+) NEGATE IT
 941  004442  020067  177472                  CMP     .AC,COUNT       ;IF THE SAME AS REMAINDER, GO
 942  004446  001416                          BEQ     STEP11          ;       TO STEP 11
 943
 944  004450  105067  177463                  CLRB    Z+1
 945  004454  005700                          TST     .AC             ;COMPARE THE SIGNS OF THE
 946  004456  100002                          BPL     .+6             ;       REMAINDER AND THE
 947  004460  105167  177453                  COMB    Z+1             ;       DIVIDEND.  IF THEY
 948  004464  105767  177454                  TSTB    SINE            ;       ARE THE SAME, GO TO
 949  004470  100002                          BPL     .+6             ;       STEP 10
 950  004472  105167  177441                  COMB    Z+1
 951  004476  105767  177435                  TSTB    Z+1
 952  004502  001730                          BEQ     STEP10
```



```
 953  004504  005067  177426          STEP11: CLR     Z
 954  004510  005700                          TST     .AC             ;SET Z=1 IF REMAINDER AND
 955  004512  100002                          BPL     .+6             ;       THE DIVISOR HAVE THE
 956  004514  105167  177417                  COMB    Z+1             ;       SAME SIGN
 957  004520  005702                          TST     R2
 958  004522  100002                          BPL     .+6
 959  004524  105167  177407                  COMB    Z+1
 960  004530  105767  177403                  TSTB    Z+1
 961  004534  001003                          BNE     .+8.
 962  004536  005267  177374                  INC     Z
 963  004542  005201                          INC     .MQ             ;ADD Z TO QUOTIENT
 964  004544  000241                          CLC
 965  004546  006101                          ROL     .MQ             ;SHIFT QUOTIENT LEFT ONCE
 966                                                                  ;       FILLING WITH "0"
 967  004550  105767  177362                  TSTB    Z
 968  004554  001402                          BEQ     .+6
 969  004556  160200                          SUB     R2,.AC          ;SUBTRACT DIVISOR FROM REMAINDER
 970  004560  000703                          BR      STP9
 971  004562  060200                          ADD     R2,.AC          ;ADD DIVISOR TO REMAINDER
 972  004564  000701                          BR      STP9
 973
 974  004566  010046                  DONX:   MOV     .AC,-(6)
 975  004570  010100                          MOV     .MQ,.AC
 976  004572  012601                          MOV     (6)+,.MQ
 977  004574  005700                          TST     .AC
 978  004576  001003                          BNE     .+8.
 979  004600  052767  000004  177334          BIS     #4,SR1
 980  004606  005700                  DONE:   TST     .AC
 981  004610  100003                          BPL     .+8.
```

```
 982  004612  052767  000010  177322          BIS     #GO2
 983  004620  005700                          TST     .AC
 984  004622  001005                          BNE     NXT
 985  004624  005701                          TST     .MQ
 986  004626  001003                          BNE     NXT
 987  004630  052767  000004  177304          BIS     #4,SR1
 988  004636  000207                  NXT:    RTS     PC
```



```
 989                                    ;       MULTIPLY SUBROUTINE
 990
 991  004640  000000                    CONT:   .WORD   0
 992  004642  000000                    ONE:    .WORD   0
 993  004644  000000                    CHEK:   .WORD   0
 994
 995  004646  005067  177770            MUL1:   CLR     ONE             ;CLEAR Z STEP1
 996  004652  005067  177264                    CLR     SR1
 997  004656  005067  177762                    CLR     CHEK
 998  004662  005000                            CLR     .AC             ;CLEAR AC
 999  004664  005701                            TST     .MQ
1000  004666  100002                            BPL     .+6
1001  004670  005167  177750                    COM     CHEK
1002  004674  005702                            TST     R2
1003  004676  100002                            BPL     .+6
1004  004700  005167  177740                    COM     CHEK
1005  004704  012767  177760  177726            MOV     #-16.,CONT      ;INITIALIZE COUNT
1006  004712  022701  100000                    CMP     #100000,.MQ
1007  004716  001011                            BNE     CNXT
1008  004720  005702                            TST     R2
1009  004722  100023                            BPL     STEP2
1010  004724  010200                            MOV     R2,.AC
1011  004726  005400                            NEG     .AC
1012  004730  000241                            CLC
1013  004732  005001                            CLR     .MQ
1014  004734  006000                            ROR     .AC
1015  004736  006001                            ROR     .MQ
1016  004740  000451                            BR      DONE2
1017  004742  022702  100000            CNXT:   CMP     #100000,R2
1018  004746  001011                            BNE     STEP2
1019  004750  005701                            TST     .MQ
1020  004752  100007                            BPL     STEP2
1021  004754  010100                            MOV     .MQ,.AC
1022  004756  005400                            NEG     .AC
1023  004760  000241                            CLC
1024  004762  005001                            CLR     .MQ
1025  004764  006000                            ROR     .AC
1026  004766  006001                            ROR     .MQ
1027  004770  000435                            BR      DONE2
1028  004772  105067  177645            STEP2:  CLRB    ONE+1
1029  004776  132767  000001  177636            BITB    #1,ONE
1030  005004  001402                            BEQ     .+6
1031  005006  105167  177631                    COMB    ONE+1           ;COMP ONE+1 IF Z=1
1032  005012  032701  000001                    BIT     #1,.MQ
1033  005016  001402                            BEQ     .+6
1034  005020  105167  177617                    COMB    ONE+1           ;COMP ONE+1 IF R1 IS ODD
1035  005024  105767  177613                    TSTB    ONE+1           ;IF THE LOW ORDER BIT IS THE SAME
1036  005030  001406                            BEQ     STEP4           ;AS Z, GO TO STEP4
```



```
1037  005032  032701  000001            STEP3:  BIT     #1,.MQ
```

```
1038  005036  001002                          BNE     .+8.
1039  005040  060200                          ADD     R2,.AC          ;ADD MULTIPLICAND IF MQ IS EVEN
1040  005042  000401                          BR      .+4
1041  005044  160200                          SUB     R2,.AC          ;SUBTRACT MULTIPLICAND IF MQ IS ODD
1042  005046  006200                  STEP4:  ASR     .AC
1043  005050  006001                          ROR     .MQ
1044  005052  106167  177564                  ROLB    ONE
1045
1046  005056  005267  177556          STEP5:  INC     CONT
1047  005062  001343                          BNE     STEP2           ;GO TO STEP2 IF NOT 16TH PASS
1048  005064  005700                  DONE2:  TST     .AC
1049  005066  001416                          BEQ     DO2
1050  005070  005767  177550                  TST     CHEK
1051  005074  100406                          BMI     DON1
1052  005076  005700                          TST     .AC
1053  005100  100003                          BPL     .+8.
1054  005102  005401                          NEG     .MQ
1055  005104  005500                          ADC     .AC
1056  005106  005400                          NEG     .AC
1057  005110  000405                          BR      DO2
1058  005112  005700                  DON1:   TST     .AC
1059  005114  100403                          BMI     .+8.
1060  005116  005401                          NEG     .MQ
1061  005120  005500                          ADC     .AC
1062  005122  005400                          NEG     .AC
1063  005124  005700                  DO2:    TST     .AC
1064  005126  001412                          BEQ     DO3
1065  005130  022700  177777                  CMP     #-1,.AC
1066  005134  001011                          BNE     DO4X
1067  005136  005701                          TST     .MQ
1068  005140  100403                          BMI     .+8.
1069  005142  052767  000001  176772          BIS     #1,SR1
1070  005150  000167  177432                  JMP     DONE
1071  005154  005701                  DO3:    TST     .MQ
1072  005156  100003                          BPL     .+8.
1073  005160  052767  000001  176754  DO4X:   BIS     #1,SR1
1074  005166  000167  177414          DO4:    JMP     DONE
1075
1076
1077  005172  000000                  .ACX:   0
1078  005174  000000                  .MQX:   0
1079
1080          000001                  .END
```

```
DIVP    002610          629     646#
DIVZ    002674          655     659#
DOIT    002062          548#    550
DONE    004606          980#   1070    1074
DONE2   005064         1016    1027    1048#
DONX    004566          931     933     974#
DON1    005112         1051    1058#
DO2     005124         1049    1057    1063#
DO3     005154         1064    1071#
DO4     005166         1074#
DO4X    005160         1066    1073#
EQZ     002172          567     570#
ERCNT   003654          583     588     597*    611     689*    796#
ERNG    003250          717#    773
ERRORS  003662          355*    615*    618     799#
ER1D    003060          646     693#
ER1M    003101          630     696#
ER2     003122          635     642     699#
ER2A    003126          653     700#
ER3     003132          667     701#
ER3A    003145          677     703#
ER3B    003152          676     704#
ER4     003165          670     680     706#
ER5     003172          673     683     707#
ER6     003202          587     709#
ER7     003226          590     713#
E1      003326          725#
E2      003335          727#    776
E3      003346          729#    779
E4      003361          731#    782
E5      003376          734#    785
E6      003413          737#    788
FLOW    002140          417     440     464     489     520     564#
GO      001060          364#    540     548     561

HINUM   003650          362*    367     384     392     397     416     438     488     502     504     636     640     665
                        744     754     765*    786     794#

HLT   = 104400          340#    421     424     427     444     447     450     468     471     474     493     496     499
                        524     527     530
HLTAD   003054          614*    687     691#
ICINT   002336          578     584     598#

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 27
DCQKAC.SRC      CROSS REFERENCE TABLE -- USER SYMBOLS

ICNT    001000          353#    405*    577     581     591     598*    602*
ITER    001250          398     405#
KIT     002352          580     582     602#
LAD     003656          511*    599*    603     605     797#
LOCX    001720          511     517#

LONUM   003646          361*    366     373     379     402     415     439     463     501     503     506     507     633
                        643     661     743     752     764*    783     793#
MUL1    004646          368     995#
N     = 000006          336#    406     409#    429     432#    451     454#    476     479#    514     517#
NOH     002422          613#    625
NOHEAD  003040          613     687#
NOTY    002332          586     597#
NXT     004636          984     986     988#
OFLW    004420          910     913     930     934#
OK1     001164          380     387#
ONE     004642          992#    995*   1028*   1029    1031*   1034*   1035    1044*
OVR     002356          603#
```

```
*       PC      =%000007        349#    365*    368*    374*    383*    386*    393*    401*    404*    417*    440*    464*    489
*                               505*    508*    520*    569*    571*    589*    592*    623*    634*    637*    641*    644*    648
*                               652*    657*    662*    666*    669*    672*    675*    679*    682*    685*    770*    775*    778
                                781*    784*    787*    789*    859*    874*    935*    988*
        PRINTR  003776          634     637     641     644     648     652     657     662     666     669     672     679     682
                                778     781     784     787     830#
        PRINTS  004006          589     592     675     685     775     832#
        PSOFT   002726          638     645     658     663     667#
        RANDOM  003420          365     505     739#
        RTIX    002036          542     544#
        R0      =%000000        341#    342     554*    620*
*       R2      =%000002        344#    367*    373*    392*    507*    741     746*    749*    752*    755*    758*    760*    761
                                762     767*    871     886     897     901     924     938     957     969     971     1002    1008
                                1010    1017    1039    1041
*       R4      =%000004        346#    364*    387*    390*    410*    433*    455*    480*    483*    742     745*    750*    766
                                834     835*    853     858#
        R5      =%000005        347#    923*    926*    927
        SAVLAD  002344          576     599#    604
        SCOPE   = 104000        339#    409     432     454     479     500     532
        SECI    002716          660     664#
        SECO    002562          632     639#
        SER     003056          692#
        SINE    004144          868#    877*    880*    898*    899*    902*    907*    909     912     923     948
        SP      =%000006        348#    354*
        SR1     004142          371     377     396     413*    425     436*    448     461*    472     486*    497     528     674
                                867#    873*    876*    934*    979*    982*    987*    996*    1069*   1073*
        STEP1   004210          881#    917
        STEP10  004364          921#    952
        STEP11  004504          942     953#
        STEP2   004772          1009    1018    1020    1028#   1047
        STEP3   005032          1037#
        STEP4   005046          1036    1042#
        STEP5   005056          1046#
        STEP6   004316          905#
        STEP8   004346          904     906     915#
        STEP9   004430          920     938#
        STP9    004370          923#    970     972
        SWR     = 177570        337#    338     535     549     575     579     585     594     607     624
        SW10    = 002000        335#    607
```



```
        SW11    = 004000        334#    575
        SW12    = 010000        333#    535
        SW13    = 020000        332#    585     624
        SW14    = 040000        331#    579
        TAD     003642          791#
        TEMP    002202          425     448     472     497     528     564*    565*    566     570*    573#    684
        TIMES   003664          581     800#
        TLOC    003644          792#
        TRPB    002204          537*    574#
*       TTY     =%000003        345#    588*    591*    633*    636*    640*    643*    647*    651*    656*    661*    665*    668
                                671*    674*    678*    681*    684*    774*    777*    780*    783*    786*    812     813*    814
```

K02

```
                       816    825*   826*   827    838*   839*   841*   843*
 TYPE   = 000004       330*   553    587    590    593    610    630    635    642    646    650    653    667
                       670    673    676    677    680    683    773    776    779    782    785    788    857
*  Z      004136       865*   881*   885*   888*   889    891*   895    944*   947*   950*   951    953*   956
                       959*   960    962*   967
 $ENDAD 002124         324    357    557*
 .      = 005176       312*   313*   318*   323*   325*   327*   351*   358    382    385    400    403    420
                       423    426    443    446    449    467    470    473    492    495    498    523    526
                       529    547    593    594*   595    612    688    738*   818    831    846    849    854
                       861*   879    884    887    890    896    900    908    911    916    925    928    939
                       946    949    955    958    961    968    978    981    1000   1003   1030   1033   1038
                       1040   1053   1059   1068   1072
 .AC    =%000000       342*   370    375    384    389*   394    402    415*   416*   419    438*   439*   442
                       457*   463*   466    482*   488*   491    504*   509    518*   519*   522    678    739
                       743*   747*   756*   762*   764    769*   777    878    883    894*   897*   901*   905
*                      919    941    945    954    969*   971*   974    975*   977    980    983    998*   1010
                       1011*  1014*  1021*  1022*  1025*  1039*  1041*  1042*  1048   1052   1055*  1056*  1058
                       1061*  1062*  1063   1065
 .ACX   005172         412*   419    435*   442    459*   466    484*   491    509*   522    668    1077*
 .EMT   002206         319    360*   575*
 .ET    002414         608    611*
 .IOT   003710         316    812*
*  .MQ  =%000001       343*   366*   369    376    381    388*   395    399    422    445    456*   469    481
*                      494    503*   510    517*   525    681    740    744*   748*   753*   754*   757*   759
                       763*   765    768*   780    893*   922*   929    932    963*   965*   975    976*   985
*                      999    1006   1013*  1015*  1019   1021   1024*  1026*  1032   1037   1043*  1054*  1060
                       1067   1071
 .MQX   005174         411*   422    434*   445    460*   469    485*   494    510*   525    671    1078*
 .SR1   001522         465    472*
 .SR2   001620         490    497*
 .SR3   001762         521    528*
 .TRP   002372         321    607*
 .TRT   003640         314    790*
 .TYPE  003660         552*   553    609*   610    649*   650    798*

 MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 30
 DCQKAC.SRC       CROSS REFERENCE TABLE -- MACRO NAMES

 DUMP       351*   633    636    639    643    647    651    656    661    664    668    671    678    681    777
            780    783    786
 NUMBER     351*   406    429    451    476    514
 PRINT      351*   593
 SCOPEX     351*   406    429    451    476
 SDUMP      351*   588    591    674    684    774
 TYPEM      351*   552    609    649
```

```
ADC     753  755  757  758  760  763 1055 1061
ADD     390  483  544  568  752  754  756  759  761  762  822  825  901  971 1039
ASL     747
ASR    1042
BCS     908
BEQ     380  382  385  398  400  403  420  423  426  443  446  449  467  470  473
        492  495  498  523  526  529  547  555  578  584  604  608  612  617  621
        632  655  815  846  849  896  906  916  942  952  968 1030 1033 1036 1049
       1064
BGT     582
BIC     565  570  826
BIS     539  934  979  982  987 1069 1073
BISB    850
BIT     535  546  549  566  575  579  585  607  624  628  631  654  659 1032 1037
BITB    616 1029
BLE     933
BMI     890  910  928  930  939 1051 1059 1068
BNE     358  536  550  567  576  580  586  619  625  629  660  688  751  824  852
        854  872  904  920  961  978  984  986 1007 1018 1038 1047 1066
BPL     538  595  818  879  884  887  913  925  946  949  955  958  981 1000 1003
       1009 1020 1053 1072
BR      465  490  521  638  645  658  663  819  831  837  900  911  931  970  972
       1016 1027 1040 1057
CLC     882  964 1012 1023
CLR     355  405  534  551  597  622  746  832  876  877  881  953  995  996  997
        998 1013 1024
CLRB    836  838  856  944 1028
CMP     357  384  402  419  422  442  445  466  469  491  494  522  525  581  618
        687  823  853  903  941 1006 1017 1065
CMPB    425  448  472  497  528
COM     537  926 1001 1004
COMB    880  885  888  899  947  950  956  959 1031 1034
DIV     463  488  519
EMT     339
HALT    313  596
INC     602  615  689  750  891  915  962  963 1046
INCB    847  851
IOT     330
JMP     328  548  561  613  917 1070 1074
JSR     365  368  374  383  386  393  401  404  417  440  464  489  505  508  520
        557  589  592  623  634  637  641  644  648  652  657  662  666  669  672
        675  679  682  685  775  778  781  784  787
MOV     354  356  359  360  361  362  364  366  367  369  370  371  373  375  376
        377  387  388  389  392  394  395  396  410  411  412  413  415  433  434
        435  436  438  455  456  457  459  460  461  480  481  482  484  485  486
```

M02

```
             501     502     503     504     506     507     508     510     511     517     518     540     542     545     554
             588     591     598     599     605     614     620     626     633     636     640     643     647     651     656
             661     665     668     671     674     678     681     684     739     740     741     742     743     744     749
             764     765     766     767     768     769     774     777     780     783     786     812     813     821     827
             828     834     835     858     873     875     938     974     975     976    1005    1010    1021
MOVB         552     564     609     649     816     830     833     855     923
MUL          416     439
NEG          940    1011    1022    1054    1056    1060    1062
NOP          418     441     558     559     560
RESET        556
ROL          748     749     839     841     843     893     894     922     965

MAINDEC-11-DCQKA-C  MUL - DIV RANDOM EXERCISER  MACY11 27(732)  21-JUN-76  13:39  PAGE 33
DCQKAC.SRC       CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ROLB         840     842     844     898     902    1044
ROR         1014    1015    1025    1026    1043
RORB         907
RTI          600     606     690     829
RTS          569     571     770     789     859     874     935     988
RTT          541     543     790
SEC          892     921
SUB          627     897     969    1041
TRAP         340

TST          379     381     397     399     458     577     583     594     603     611     686     871     878     883     886
             905     919     924     927     929     932     945     954     957     977     980     983     985     999    1002
            1008    1019    1048    1050    1052    1058    1063    1067    1071
TSTB         814     817     845     848     889     895     909     912     948     951     960     967    1035
.ASCII       701     717

.ASCIZ       594     693     696     699     700     703     704     706     707     709     713     716     725     727     729
             731     734     737
.BLKW        861
.ENABL       298
.END        1080
.EVEN        594     738

.LIST        296     313     351     406     409     429     432     451     454     476     479     514     517     532     575
             594     607     739     830     863
.MACR        351
.MACRO       351

.NLIST       296     213     351     406     409     429     432     451     454     476     479     514     517     532     575
             594     607     739     830     863
.REM           1
.REPT        313
.SBTTL       296     351     406     429     451     476     514     532     575     607     739     830     863
.TITLE       295
.WORD        865     866     867     868     991     992     993


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

#DCQKAC,DCQKAC/SOL/PAGNUM/CRF=DCQKAC.SRC
RUN-TIME: 3 6 1 SECONDS
RUN-TIME RATIO: 136/12=10.8
```

CORE USED: 7K (13 PAGES)

Spooler runtime 4 Seconds, 23 KCS, 116 disk reads, 4 disk writes, 26 pages