digital

MAINDEC CHANGE NOTICE

CHANGE NO.
11-DZMMG-B - 1 (DIG-)
Sheet 1 of 1

AUTHOR	PROGRAM DATE	PRODUCT LINE	MAINDEC NUMBER
J. Lacey DATE EXT. 6/7/72 3142	MAY 72	PDP-11 V11 07433	11-DZMMG-B

PROGRAM NAME Worst Case Noise Test **DEVICE** Core Memory

ITEM

Ø. For information only:
Latest revision has been released to program library and will be phased-in when all copies of 11-DZMMG-A are gone. This program supersedes MAINDEC-11-DIGC.

1. 6/30/72 The following patch is used only with ACT-11. Its purpose is to reduce the run time required for worst case noise.

ADDRESS	FROM	TO
000456	177750	177776
000606	177750	177776
000756	177750	177776
001106	177750	177776

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZMMG-B-D

PRODUCT NAME: WORST CASE NOISE TEST

DATE REVISED: MAY 19,1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JOHN RODENHISER/ JIM LACEY

M. C. N. REQUIRED
THIS PROGRAM REQUIRES MCNGS
IN ORDER TO WORK PROPERLY

1, ABSTRACT

THIS TEST GENERATES THE MAXIMUM AMOUNT OF PLANE NOISE
POSSIBLE DURING THE EXECUTION OF MEMORY REFERENCE IN-
STRUCTIONS. THE NOISE GENERATED IS DISTRIBUTED ACROSS
THE CORE PLANE AS AN ALGEBRAIC SUPPLEMENT TO THE NOR-
MAL DYNAMIC NOISE PRESENT ON THE SENSE LINES DURING
MEMORY READ-REGENERATE OPERATIONS. DATA MODIFICATION
AS A RESULT OF NOISE AMPLITUDES IS FLAGGED AS AN ERROR,
WITH THE LOCATION AND CONTENTS RECORDED ON THE TELETYPE.
PROVISIONS HAVE BEEN ADDED TO INCLUDE WORST CASE PATTERNS
FOR INTERLEAVED MEMORIES.

2, REQUIREMENTS

2,1 EQUIPMENT

PDP-11 WITH MINIMUM 4K OF MEMORY

2,2 STORAGE

2,2,1 PROGRAM STORAGE - THE ROUTINE USES MEMORY FROM 200 TO 2466.

3, LOADING PROCEDURE

3,1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

1. ABSOLUTE LOADER MUST BE IN MEMORY.
2. PLACE BINARY TAPE IN READER.
3. LOAD ADDRESS #7500. (* DETERMINED BY ADDRESS OF LOADER)
4. PRESS "START" (PROGRAM WILL LOAD).

4, STARTING PROCEDURE

4,1 CONTROL SWITCH SETTING

STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO.

4,2 STARTING ADDRESSES

200 = START FOR AUTOMATIC TEST LIMITS
202 = START FOR SELECTED TEST LIMITS

4,3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY.
SET SWITCH REGISTER TO STARTING ADDRESS.
LOAD ADDRESS 200.
PRESS START.
THE PROGRAM WILL RUN THROUGH THE SELECTED ADDRESS FIELD AND LOOP.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200, WITH ALL SWITCHES DOWN, THE PROGRAM WILL PRINT OUT ON ERRORS AND CONTINUE IN TEST.

5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP ... HALT ON ERROR
SW13 = 1 OR UP ... INHIBIT PRINTOUT
SW12 = 1 OR UP ... HALT ON END OF PROGRAM
SW11 = 1 OR UP ... LOOP ON PRESENT PART OF PROGRAM
SW10 = 1 OR UP ... RING TTY BELL ON ERROR

5.1.3 AUTOMATIC TEST LIMITS

IF THE PROGRAM IS STARTED AT ADDRESS 200 THE PROGRAM WILL TEST ALL AVAILABLE MEMORY, CARE SHOULD BE TAKEN TO SELECT THE PROPER OPERATIONAL SWITCH SETTINGS (REFER TO 5.1.2) BEFORE STARTING.

5.1.4 SELECTED TEST LIMITS

IF THE PROGRAM IS STARTED AT ADDRESS 202 A MESSAGE WILL BE PRINTED ON THE TELETYPE INSTRUCTING THE OPERATOR ON THE METHOD OF SELECTING OTHER MEMORY TEST AREAS AND THEN THE PROGRAM WILL STOP AT THE FIRST OF THREE HALTS.

A. SET THE LOW TEST LIMIT IN THE SWITCH REGISTER AND PRESS CONTINUE.

B. THEN SET THE HIGH TEST LIMIT IN THE SWITCH REGISTER AND PRESS CONTINUE.

C. THEN SET THE OPERATIONAL SWITCH SETTINGS (REF 5.1.2) AND PRESS CONTINUE.

THE PROGRAM WILL NOW BEGIN TESTING THE SELECTED AREA. THE PROGRAM WILL NOT ALLOW LIMITS TO BE SELECTED THAT WOULD CAUSE ITSELF TO BE DESTROYED. AN EXCEPTION TO THIS WOULD BE THE INSTRUCTIONAL TEXT STORAGE AREA. STARTING THE PROGRAM AT 200 OR SELECTING LIMITS WHICH OVERLAP THIS AREA WILL CAUSE THE TEXT MESSAGE TO BE DESTROYED.

5.2 SUBROUTINE ABSTRACTS

5.2.1 XORCK

SUBROUTINE XORCK IS A CALL TO THE EXCLUSIVE OR ADDRESS CHECKER. THE ADDRESS CHECKER EXAMINES ADDRESS BITS 1 AND 8 AS EACH TEST LOCATION IS ADDRESSED TO DETERMINE IF THE EXCLUSIVE OR CONDITION BETWEEN THE TWO BITS IS PRESENT. THE SUBROUTINE IS CONCLUDED BY SETTING THE XORFLG IF THE EXCLUSIVE OR IS PRESENT OR LEAVING XORFLG RESET IF THE CONVERSE IS TRUE.

5.2.2 XORCKA

SUBROUTINE XORCKA IS THE SAME AS XORCK EXCEPT IT USES BITS 8 AND 13 INSTEAD OF BITS 1 AND 8.

5.2.3 XORCKB

SUBROUTINE XORCKB IS THE SAME AS XORCK AND XORCKA EXCEPT IT USES BITS 3 AND 9.

5.2.4 ERRORA & B

SUBROUTINE ERRORA & B IS CALLED BY EITHER ERRORA OR ERRORB WITH THE TWO CALLS TERMINATING INTO A COMMON SUBROUTINE AFTER SELECTING THE STARTING ADDRESS OF THEIR RESPECTIVE ERROR MESSAGES. SINCE NO SCOPE LOOP PROVISIONS ARE SUPPLIED WITHIN THIS TEST, THREE OPTIONS ARE AVAILVAILABLE IN THE EVENT OF AN NOISE ERROR; RING TTY BELL, INHIBIT PRINTING AND HALT ON ERROR. IF THE INHIBIT PRINT SWITCH IS NOT PRESENT THE ERROR MESSAGE WILL BE PRINTED. THE SUBROUTINE UPON THE END OF PRINTING, OR IN THE EVENT SW13 IS PRESENT, CHECKS THE HALT SWITCH (SW15). IF THE SWITCH IS NOT SET THE PROGRAM IMMEDIATELY RETURN TO THE MAIN TEST PROGRAM. IF SW15 IS SET THE PROGRAM WILL HALT. A CONTINUE FROM THE HALT RETURNS TO THE MAIN TEST PROGRAM.

6, ERRORS

6,1 ERROR PRINTOUT

PRINTS ALL ERRORS UNLESS INHIBITED BY SW13.

6,2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION OR RELOAD
STARTING ADDRESS AND START.

7, RESTRICTIONS

7,1 STARTING RESTRICTION

NONE

7,2 OPERATIONAL RESTRICTION

NONE

8, MISCELLANEOUS

8,1 EXECUTION TIME

THE EXECUTION TIME OF THIS PROGRAM IS APPROXIMATELY
1.5 MINUTE WITH 4K OF CORE, THE TTY BELL WILL RING
AT THE END OF EACH PASS.

9, PROGRAM DESCRIPTION

9.1 THE WORST CASE NOISE TEST IS DESIGNED TO PRODUCE THE GREATEST AMOUNT OF PLANE NOISE POSSIBLE DURING MEMORY WRITING AND READING CYCLES. THE NOISE PARAMETERS ARE EFFECTED BY A NUMBER OF FACTORS, HENCE THE TEST IS DESIGNED TO TEST ALL CASE CONDITIONS ARISING OUT OF THE FOUR STANDARD MEMORY CONSTRUCTION CONFIGURATIONS. THE NOISE GENERATED IS DISTRIBUTED ACROSS THE CORE PLANE ALGEBRAICALLY ADDS TO THE NORMAL DYNAMIC NOISE PRESENT ON THE SENSE LINES TO EFFECTIVELY CAUSE MISREADING OF DATA (WITHIN THE PLANE) THAT IS IN THE LOW "1" OR HIGH "0" CATAGORY. THE SENSE WINDINGS OF MOST MEMORIES ARE SUCH THAT WORST CASE PATTERNS CAN BE CAUSED BY ALTERNATELY WRITING -1 AND 0 DATA CONFIGURATIONS THROUGHOUT MEMORY. UNDER THESE CONDITIONS WORST CASE NOISE IS GENERATED BY READ, WRITE COMPLEMENT, READ, WRITE COMPLEMENT, AT EACH LOCATION. THE TEST IS REPEATED AFTER COMPLEMENTING ALL OF THE PATTERN DATA STORED IN THE MEMORY TEST ZONE. THIS TEST DEVIATES FROM THIS FORM OF TESTING ONLY IN THE DISTRIBUTION OF THE -1 AND 0 DATA PATTERNS WITHIN THE MEMORY TEST ZONE. THE CONSTRAINT PLACED ON THIS TEST REQUIRES WRITING THE COMPLEMENT OF THE DATA PATTERN AS DESCRIBED BY THE EXCLUSIVE OR OF THE SECOND BIT OF THE X AND Y SELECTION LINES. THESE BITS CORRESPOND TO ADDRESS BITS 1 AND 8. THEREFORE, THE PATTERN OR ITS COMPLEMENT WILL BE WRITTEN INTO THE MEMORY TEST ZONE AS DETERMINED BY THE XOR BETWEEN ADDRESS BITS 1 AND 8.

THE PROGRAM IS COMPOSED OF TWO PARTS. PART 1 IS RUN FIRST AND DURING THIS SECTION OF THE PROGRAM A -1 CONFIGURATION IS WRITTEN INTO ALL LOCATIONS HAVING AN ADDRESS WITH AN XOR STATE BETWEEN BITS 1 AND 8. ALL OTHER LOCATIONS ARE LOADED WITH THE ZERO CONFIGURATION. AFTER THE TEST ZONE HAS BEEN LOADED, THE MEMORY IS RESCANNED. THIS TIME EACH LOCATION IS READ, COMPLEMENTED, READ, AND COMPLEMENTED (RCRC). ANY LOCATION DETECTED AS BEING DISTURBED BY A PREVIOUS RCRC OPERATION WILL BE FLAGGED AS AN ERROR. UPON THE CONCLUSION OF THE READ SCAN LOOP, THE PROGRAM AUTOMATICALLY SWITCHES TO PART 2.

IN PART 2, THE DATA PATTERNS STORED IN MEMORY ARE COMPLEMENTED, I.E., ZERO PATTERNS ARE STORED IN LOCATIONS HAVING ADDRESSES WITH AN XOR BETWEEN BITS 1 AND 8. ALL OTHER LOCATIONS ARE LOADED WITH THE -1 CONFIGURATION. THE XOR PATTERN DISTRIBUTION FOR PART 1 AND 2 IS SUMMARIZED FOR REFERENCE AS FOLLOWS:

PART 1
XOR (1 & 8) = -1 PATTERN
NO XOR (1 & 8) = 0 PATTERN

PART 2
XOR (1 & 8) = 0 PATTERN
NO XOR (1 & 8) = -1 PATTERN

(9',1 CONT'D)

AFTER MEMORY IS LOADED IT IS SCANNED AGAIN WITH A READ, COMPLEMENT, READ, COMPLEMENT LOOP AS IN PART 1, ANY LOCATION DETECTED AS BEING DISTURBED BY A PREVIOUS RCRC OPERATION IS FLAGGED AS AN ERROR, BEFORE WRITING OR READING ANY LOCATION (IN EITHER PART) THE PROGRAM ISSUES A CALL TO SUBROUTINE XORCK WHICH TESTS BITS 1 AND 8 AND SETS THE XORFLG IF THE XOR CONDITION IS PRESENT, SUBROUTINE ERRORA IS CALLED FOR ANY LOCATION DISTURBED FROM THE -1 CONFIGURATION, AND ERRORB IS CALLED FOR ANY LOCATION DISTURBED FROM THE 0 CONFIGURATION.

THE PROGRAM WILL PRINT OUT ERRORS AND REPEAT WHEN COMPLETE WITHOUT INTERRUPTION, UPON COMPLETION THE PROGRAM WILL RING THE TELETYPE BELL AND THEN HALT IF SWITCH 12 IS PRESENT, A CONTINUE FROM THE HALT WILL INITIATE ANOTHER PASS.

9',2 THIS PROGRAM HAS BEEN MODIFIED TO PROVIDE WORST CASE NOISE FOR INTERLEAVED MEMORIES, THE TEST DESCRIPTION IS SUBSTANTIALLY THE SAME EXCEPT IN ADDITION TO A MEMORY PATTERN DETERMINED BY THE XOR OF BITS 1 AND 8 A SECOND PATTERN IS DETERMINED BY THE XOR OF BITS 13 AND 8.

9',3 THIS PROGRAM HAS BEEN MODIFIED TO PROVIDE WORST CASE NOISE FOR MM11S, MM11L, MM11K, AND MM11M, THE TEST DESCRIPTION IS SUBSTANTIALLY THE SAME EXCEPT IN ADDITION TO THE MEMORY PATTERNS DETERMINED BY THE XOR OF BITS 1 AND 8 AND BITS 8 AND 13 A THIRD PATTERN IS DETERMINED BY THE XOR OF BITS 3 AND 9'.

10', LISTING

```
                                ;NLIST  SEQ
                                ;TITLE  WORST CASE NOISE TEST   MAINDEC-11-DZMMG-B
                                ;COPYRIGHT 1970, 1971, 1972 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                                ;PROGRAMMER: JOHN RODENHISER/JIM LACEY
            000240              NOP=240
                                        ;ENABL  ABS
            000000                      ;.=0

                                ;TRAP CATCHER 0-176

            000004                      ;.=4
    000004  000264                      TLG
            000200                      ;.=200
    000200  000423                      BR      START1
    000202  000405                      BR      START
    000204  002156              LOLMT:  QEFA
    000206  017470              HILMT:  17470
    000210  177564              TCSR:   177564
    000212  177566              TDBR:   177566
    000214  177570              SWREG:  177570

    000216  012767  002156  177760  START:  MOV     #QEFA,LOLMT
    000224  012767  017470  177754          MOV     #17470,HILMT
    000232  012706  002154                  MOV     #BUFFER,%6
    000236  012702  002160                  MOV     #MSG2,%2
    000242  004767  001450                  JSR     %7,TOP
    000246  000432                          BR      ST1
    000250  012706  002154          START1: MOV     #BUFFER,%6      ;SET THE SP
    000254  010603                          MOV     %6,%3           ;TEST POINT IN LOWER BANK
    000256  005723              SEE:        TST     (%3)+           ;TEST
    000260  000240                          NOP                     ;PRECAUTIONARY DELAY
    000262  000775                          BR      SEE             ;NO TRAP, CONTINUE
    000264  162703  000004          TLG:    SUB     #4,%3           ;TRAPPED, SETUP CORE LIMITS
    000270  005737  000042                  TST     @#42            ;LOADED BY A MONITOR?
    000274  001407                          BEQ     $1              ;BR IF NO
    000276  023727  000042  001316          CMP     @#42,#ENDADR    ;YES---WAS IT DDP1?
    000304  001405                          BEQ     $2              ;BR IF NO
    000306  162703  002734                  SUB     #1500.,%3       ;YES---SAVE CORE FOR CHAIN MODE
    000312  000402                          BR      $2              ;SKIP NEXT I/N
    000314  162703  000300          $1:     SUB     #300,%3         ;PROTECT THE LOADERS
    000320  010367  177662          $2:     MOV     %3,HILMT        ;SET HILMT
    000324  012767  002156  177652          MOV     #QEFA,LOLMT     ;SET LOLMT
    000332  000427                          BR      PART            ;BRANCH TO TEST START
    000334  000000              ST1:        HALT                    ;WAIT FOR CONTINUE
    000336  005777  177652                  TST     @SWREG          ;LOOK FOR LOLMT
    000342  001407                          BEQ     HISET           ;DEFERRED
    000344  027767  177644  177632          CMP     @SWREG,LOLMT    ;CKN LOLMT INPUT>TERMNL
    000352  103403                          BLO     HISET           ;LOLMIT<TERMNL USE LMT DEFINED
    000354  017767  177634  177622          MOV     @SWREG,LOLMT    ;LOLMT>TERMNL STORE INPUT
    000362  000000              HISET:      HALT                    ;WAIT FOR CONTINUE
    000364  005777  177624                  TST     @SWREG          ;LOOK FOR HILMT
    000370  001407                          BEQ     CONSET          ;DEFERRED
    000372  027767  177616  177604          CMP     @SWREG,LOLMT
    000400  103403                          BLO     CONSET
```

```
000472  017767  177606  177576          MOV     @SWREG,HILMT    ;STORE INPUT
000476  000000                  CONSET: HALT                    ;SET UP CONTROL SWITCHES
000500  012767  001332  001170  PART:   MOV     #XORCK,XOR      ;INIT 1 AND 8
                              ;PART 1 TEST = WRITE CHECKERBOARD
000506  016700  177560          PART1:  MOV     LOLMT,%0        ;INITIALIZE TEST AREA
000512  004777  001160          P1A:    JSR     %7,@XOR         ;GET XOR OF ADDRESS BITS 1 AND 8 (13,8)
000516  005767  001156                  TST     XORFLG          ;TEST XOR BIT TO DETERMINE IF ALL 1'S OR ALL 0'S WRITTEN
000522  001403                          BEQ     P1B
000524  012720  177777                  MOV     #-1,(0)+
000530  000401                          BR      .+4
000532  005020                  P1B:    CLR     (0)+
000534  020067  177534                  CMP     %0,HILMT        ;DONE ALL MEMORY?
000540  101764                          BLOS    P1A             ;NO
000542  012767  177750  001122          MOV     #-30,CNTR       ;INITIALIZE LOOP COUNTER
000550  016700  177516          P1C:    MOV     LOLMT,%0        ;REINITIALIZE TEST AREA
000554  004777  001116          P1D:    JSR     %7,@XOR         ;GET XOR OF BITS 1,8 (13,8)
000560  005767  001114                  TST     XORFLG
000564  001403                          BEQ     P1E
000566  021027  177777                  CMP     @%0,#-1         ;READ CHECK
000572  000401                          BR      .+4
000574  005710                  P1E:    TST     @%0             ;READ CHECK
000576  001404                          BEQ     P1F
000600  004767  000756                  JSR     %7,ERROR        ;HAVE ERROR
000604  000167  177676                  JMP     PART1
000610  005720                  P1F:    TST     (0)+            ;INCREMENT TEST AREA
000612  020067  177456                  CMP     %0,HILMT        ;DONE ALL TEST AREA?
000616  101756                          BLOS    P1D             ;NO
000620  032777  004000  177454          BIT     #4000,@SWREG    ;TEST SW11 TO LOOP ON PART 1 READ CHECK
000626  001350                          BNE     P1C
000630  005267  001036                  INC     CNTR            ;+1 TO LOOP COUNT
000634  001345                          BNE     P1C             ;REPEAT TEST
                              ;2ND HALF OF PART 1 TEST

                              ;READ, WRITE COMP, READ, WRITE COMP
000640  016700  177430          P1G:    MOV     LOLMT,%0
000644  004777  001030          P1H:    JSR     %7,@XOR
000650  005767  001026                  TST     XORFLG
000654  001403                          BEQ     P1K
000656  012720  177777                  MOV     #-1,(0)+
000662  000401                          BR      .+4
000664  005020                  P1K:    CLR     (0)+
000666  020067  177404                  CMP     %0,HILMT
000672  101764                          BLOS    P1H
000674  012767  177750  000772          MOV     #-30,CNTR
000702  016700  177366          P1L:    MOV     LOLMT,%0
000706  004777  000766          P1M:    JSR     %7,@XOR
000712  005767  000764                  TST     XORFLG
000716  001406                          BEQ     P1N
000720  021010                          CMP     @%0,@%0         ;READ
000722  005110                          COM     @%0             ;WRITE COMPLEMENT
000724  005710                          TST     @%0             ;READ CHECK
000726  001011                          BNE     P1P
000730  005120                          COM     (0)+            ;WRITE COMPLEMENT, +2 TO LIMIT
000732  000413                          BR      P1R
```

```
000644  021010                  P1N:    CMP     @%0,@%0         ;READ
000646  005110                          COM     @%0             ;WRITE COMPLEMENT
000650  021027  177777                  CMP     @%0,#-1
000654  001002                          BNE     P1P
000656  005120                          COM     (0)+            ;WRITE COMPLEMENT, +2 TO LIMIT
000660  000404                          BR      P1R
000662  004767  000606          P1P:    JSR     %7,ERROR
000666  000167  177656                  JMP     P1G
000672  020067  177310          P1R:    CMP     %0,HILMT        ;DONE ALL TEST AREA?
000676  101747                          BLOS    P1M             ;NO
000700  032777  004000  177306          BIT     #4000,@SWREG    ;TEST SW 11 TO LOOP ON PART 1 READ, WRITE COMP, READ CH
000706  001341                          BNE     P1L
000710  005267  000670                  INC     CNTR            ;+1 TO LOOP COUNTER
000714  001336                          BNE     P1L             ;REPEAT TEST.
000716  000400                          BR      PART2
                                ;NOW DO PART 2 WHICH IS COMPLEMENT PATTERN OF PART 1
000720  016700  177260          PART2:  MOV     LOLMT,%0        ;INITIALIZE TEST AREA
000724  004777  000660          P2A:    JSR     %7,@XOR         ;GET XOR OF ADDRESS BITS 1 AND 8 (13 AND 8)
000730  005767  000656                  TST     XORFLG          ;TEST XOR BIT TO DETERMINE IF ALL 1'S OR 0'S WRITTEN
000734  001402                          BEQ     P2B
000736  005020                          CLR     (0)+
000740  000402                          BR      .+6
000742  012720  177777          P2B:    MOV     #-1,(0)+
000746  020067  177234                  CMP     %0,HILMT        ;DONE ALL MEMORY?
000752  101764                          BLOS    P2A             ;NO

000754  012767  177750  000622          MOV     #-30,CNTR       ;INITIALIZE LOOP COUNTER
000762  016700  177216          P2C:    MOV     LOLMT,%0        ;REINITIALIZE TEST AREA
000766  004777  000616          P2D:    JSR     %7,@XOR         ;GET XOR OF BITS 1 AND 8 (13,8)
000772  005767  000614                  TST     XORFLG
000776  001402                          BEQ     P2E
001000  005710                          TST     @%0             ;READ CHECK
001002  000402                          BR      .+6
001004  021027  177777          P2E:    CMP     @%0,#-1         ;READ CHECK
001010  001404                          BEQ     P2F
001012  004767  000456                  JSR     %7,ERROR        ;HAVE ERROR
001016  000167  177676                  JMP     PART2
001022  005720                  P2F:    TST     (0)+            ;INCREMENT TEST AREA
001024  020067  177156                  CMP     %0,HILMT        ;DONE ALL TEST AREA?
001030  101756                          BLOS    P2D             ;NO
001032  032777  004000  177154          BIT     #4000,@SWREG    ;TEST SW10 TO LOOP ON PART 2 READ CHECK
001040  001350                          BNE     P2C
001042  005267  000536                  INC     CNTR            ;+1 TO LOOP COUNT
001046  001345                          BNE     P2C             ;REPEAT TEST
                                ;SECOND HALF OF PART 2 TEST
                                ;READ, WRITE COMP, READ, WRITE COMP
001050  016700  177130          P2G:    MOV     LOLMT,%0
001054  004777  000530          P2H:    JSR     %7,@XOR
001060  005767  000526                  TST     XORFLG
001064  001402                          BEQ     P2K
001066  005020                          CLR     (0)+
001070  000402                          BR      .+6
001072  012720  177777          P2K:    MOV     #-1,(0)+
001076  020067  177104                  CMP     %0,HILMT
```

```
001102  101764                          BLOS    P2H
001104  012767  177750  000472          MOV     #-30,CNTR
001112  016700  177066          P2L:    MOV     LOLMT,%0
001116  004777  000466          P2M:    JSR     %7,@XOR
001122  005767  000464                  TST     XORFLG
001126  001407                          BEQ     P2N
001130  021010                          CMP     @%0,@%0         ;READ
001132  005110                          COM     @%0             ;WRITE COMPLEMENT
001134  021027  177777                  CMP     @%0,#-1
001140  001010                          BNE     P2P
001142  005120                          COM     (0)+            ;WRITE COMPLEMENT, +2 TO LIMIT
001144  000412                          BR      P2R
001146  021010          P2N:    CMP     @%0,@%0         ;READ
001150  005110                          COM     @%0             ;WRITE COMPLEMENT
001152  005710                          TST     @%0
001154  001002                          BNE     P2P
001156  005120                          COM     (0)+            ;WRITE COMPLEMENT, +2 TO LIMIT
001160  000404                          BR      P2R
001162  004767  000306          P2P:    JSR     %7,ERROR
001166  000167  177656                  JMP     P2G
001172  020067  177010          P2R:    CMP     %0,HILMT        ;DONE ALL TEST AREA?
001176  101747                          BLOS    P2M             ;NO

001200  032777  004000  177006          BIT     #4000,@SWREG    ;TEST SW 11 TO LOOP ON PART 2 READ, WRITE COMP, READ CHE
001206  001341                          BNE     P2L
001210  005267  000370                  INC     CNTR            ;INCREMENT LOOP COUNT
001214  001336                          BNE     P2L             ;REPEAT TEST
001216  022767  001332  000364          CMP     #XORCK,XOR      ;WAS THIS PASS WITH 1 AND 8?
001224  001005                          BNE     P1              ;BR IF NO
001226  012767  001370  000354          MOV     #XORCKA,XOR     ;SET XOR FOR 8 AND 13
001234  000167  177160                  JMP     PART1
001240  022767  001370  000342  P1:     CMP     #XORCKA,XOR     ;WAS IT WITH 8 AND 13
001246  001005                          BNE     RBPC            ;BR IF NO
001250  012767  001434  000332          MOV     #XORCKB,XOR     ;SET XOR FOR 3 AND 9
001256  000167  177136                  JMP     PART1
001262  105777  176722          RBPC:   TSTB    @TCSR
001266  100375                          BPL     .-4
001270  012777  000207  176714          MOV     #207,@TDBR      ;RING BELL
001276  032777  010000  176710  CONTCK: BIT     #10000,@SWREG   ;TEST SW 12 TO HALT AT END OF TEST
001304  001401                          BEQ     .+4
001306  000000                          HALT
001310  013702  000042                  MOV     @#42,%2
001314  001404                          BEQ     DOAGN
001316  004712                  ENDADR: JSR     %7,(2)
001320  000240                          NOP
001322  000240                          NOP
001324  000240                          NOP
001326  000167  177060          DOAGN:  JMP     PART
                                ;SUBROUTINE XOR CHECK BETWEEN ADDRESS BITS 1 & 8
001332  005067  000254          XORCK:  CLR     XORFLG          ;RESET THE XOR FLAG
001336  010005                          MOV     %0,%5           ;SETUP MASK FOR 3RD OCTAL DIGIT
001340  042705  177377                  BIC     #177377,%5      ;MASK BIT 8
001344  000305                          SWAB    %5
001346  006105                          ROL     %5              ;ROTATE BIT 8 INTO POSITION TO XOR WITH BIT 1
```

```
001350  060005                          ADD     %0,%5           ;ADD BIT 1 TO BIT 8
001352  032705  000002                  BIT     #2,%5           ;TEST FOR XOR
001356  001403                          BEQ     XORCK1
001360  012767  177777  000224          MOV     #-1,XORFLG      ;XORFLG = -1 IF XOR WAS MET
001366  000207                  XORCK1: RTS     %7
                                ;XOR CHECK FOR BITS 13,8
001370  005067  000216          XORCKA: CLR     XORFLG
001374  010005                          MOV     %0,%5
001376  042705  157777                  BIC     #157777,%5
001402  006005                          ROR     %5
001404  006005                          ROR     %5
001406  006005                          ROR     %5
001410  006005                          ROR     %5
001412  006005                          ROR     %5
001414  060005                          ADD     %0,%5
001416  032705  000400                  BIT     #400,%5
001422  001403                          BEQ     XORA
001424  012767  177777  000160          MOV     #-1,XORFLG
001432  000207                  XORA:   RTS     %7
                                ;XOR BITS 3 & 9
001434  005067  000152          XORCKB: CLR     XORFLG
001440  010005                          MOV     %0,%5
001442  042705  176777                  BIC     #176777,%5
001446  000305                          SWAB    %5
001450  006105                          ROL     %5
001452  006105                          ROL     %5
001454  060005                          ADD     %0,%5
001456  032705  000010                  BIT     #10,%5
001462  001403                          BEQ     XORB
001464  012767  177777  000120          MOV     #-1,XORFLG
001472  000207                  XORB:   RTS     %7
                                ;ERROR A & B SUBROUTINE
001474  032777  002000  176512  ERROR:  BIT     #2000,@SWREG    ;TEST SWITCH 9 TO RING BELL ON ERROR
001502  001406                          BEQ     ERRORA
001504  105777  176500                  TSTB    @TCSR
001510  100375                          BPL     .-4
001512  012777  000207  176472          MOV     #207,@TDBR
001520  012702  002010          ERRORA: MOV     #MSG1,%2
001524  032777  020000  176462          BIT     #20000,@SWREG   ;SET UP INHIBIT PRINT MASK IN PTSWCK
001532  001016                          BNE     SWCHK           ;UP, GO CK ERROR CONTROLS
001534  004767  000156                  JSR     %7,TOP          ;OK, GO PRINT ERROR MSG
001540  010002                          MOV     %0,%2           ;GET ERROR ADDRESS
001542  004767  000046                  JSR     %7,PRTAB        ;GO TO PRINT ROUTINE

001546  012702  000006                  MOV     #6,%2
001552  004767  000204                  JSR     %7,SPACE        ;GO PRINT 6 SPACES
001556  011002                          MOV     @%0,%2          ;GET ERROR DATA
001560  004767  000030                  JSR     %7,PRTAB        ;GO TO PRINT ROUTINE
001564  004767  000146                  JSR     %7,CRLF
001570  032777  100000  176416  SWCHK:  BIT     #100000,@SWREG  ;AND WITH HALT ON ERROR SWITCH
001576  001401                          BEQ     .+4             ;SKIP NX IF NO SWITCH
001600  000000                          HALT
001602  000207                          RTS     %7              ;RETURN TO PROGRAM
001604  000000                  CNTR:   0
```

```
001606  000000                  BELLCT: 0
001610  001332                  XOR:    XORCK
001612  000000                  XORFLG: 0
                                ;**********PRINTING SUBROUTINES**********
                                ;PRINT THE CONTENTS OF %2 AS AN OCTAL NUMBER
001614  012767  000006  000072  PRTAB:  MOV     #6,BINCT        ;SETUP FOR SIX DIGITS
001622  010267  000064          MOV     %2,TOODLE       ;SAVE THE INPUT
001626  005002                  CLR     %2
001630  006167  000056          ROL     TOODLE          ;POSITION THE SIGN BIT
001634  000410                  BR      FORM
001636  006167  000050          MKNUM:  ROL     TOODLE          ;POSITION NEXT DIGIT
001642  006167  000044          ROL     TOODLE
001646  006167  000040          ROL     TOODLE
001652  016702  000034          MOV     TOODLE,%2
001656  006102                  FORM:   ROL     %2              ;FORM THE DIGIT
001660  042702  177770          BIC     #177770,%2      ;EXTRACT THIS DIGIT
001664  052702  000260          BIS     #260,%2         ;CONVERT TO TTY CODE
001670  105777  176314          WAIT1:  TSTB    @TCSR           ;WAIT ON THE TTY
001674  100375                  BPL     WAIT1
001676  110277  176310          MOVB    %2,@TDBR        ;TYPE THIS DIGIT
001702  005367  000006          DEC     BINCT           ;LAST DIGIT TYPED?
001706  001353                  BNE     MKNUM           ;BR IF NO
001710  000207                  RTS     %7
001712  000000                  TOODLE: 0
001714  000000                  BINCT:  0
                                ;PRINT A MESSAGE--%2 POINTS TO THE FIRST CHARACTER
                                ;NOTE: THIS ROUTINE WILL ALWAYS START WITH A CR & LF
001716  142777  000177  176264   TOP:   BICB    #177,@TCSR      ;CLR INT, FLAG
001724  010246                  MOV     %2,-(6)         ;SAVE THE MESSAGE POINTER
001726  004767  000004          JSR     %7,CRLF         ;GO DO A CARRIAGE RETURN & LINE FEED
001732  012602                  MOV     (6)+,%2         ;GET THE MESSAGE POINTER
001734  000402                  BR      TOP1            ;GO PRINT THE MESSAGE
                                ;ENTER HERE FOR A CARRIAGE RETURN & LINE FEED
001736  012702  002005          CRLF:   MOV     #$CRLF,%2
                                ;LIKE "TOP" BUT DOES NOT START WITH A CR & LF
001742  105777  176242          TOP1:   TSTB    @TCSR           ;WAIT ON TTY DONE FLAG
001746  100375                  BPL     TOP1
001750  112277  176236          MOVB    (2)+,@TDBR      ;SEND A CHARACTER
001754  105712                  TSTB    (2)             ;CHECK FOR THE END MARKER
001756  001371                  BNE     TOP1
001760  000207                  RTS     %7
                                ;TYPE SPACES AS DETERMINED BY %2
001762  105777  176222          SPACE:  TSTB    @TCSR
001766  100375                  BPL     SPACE
001770  116777  000010  176214          MOVB    $SPACE,@TDBR
001776  005302                  DEC     %2              ;DO MORE?
002000  001370                  BNE     SPACE           ;BR IF YES
002002  000207                  RTS     %7
002004     240                  $SPACE: .BYTE   240
002005     015  000012          $CRLF:  .ASCIZ  <15><12>
002010  051105  047522  020122  MSG1:   .ASCII  /ERROR       BAD/<15><12>
002016  020040  020040  020040
002024  040502  006504     012
002031     114  041517  052101          .ASCIZ  /LOCATION    DATA/<15><12>
```

```
002036  047511  020116  020040
002044  042040  052101  006501
002052  000012
                                        ;EVEN
002054  000000                  QEF:    0
        002154                          ;.=QEF+100
002154  000000                  BUFFER: 0
002156  000000                  QEFA:   0
002160  042523  020124  042515  MSG2:   ;ASCII  /SET MEMORY ADDRESS LIMITS VIA SWITCH REGISTER/<15><12>
002166  047515  054522  040440
002174  042104  042522  051523
002202  046040  046511  052111
002210  020123  044526  020101
002216  053523  052111  044103
002224  051040  043505  051511
002232  042524  006522     012
002237     123  052105  046040          ;ASCII  /SET LOWER LIMIT IN SW-REG AND PRESS CONTINUE/<15><12>
002244  053517  051105  046040
002252  046511  052111  044440
002260  020116  053523  051055
002266  043505  040440  042116
002274  050040  042522  051523
002302  041440  047117  044524
002310  052516  006505     012
002315     123  052105  052440          ;ASCII  /SET UPPER LIMIT IN SW-REG AND PRESS CONTINUE/<15><12>
002322  050120  051105  046040
002330  046511  052111  044440
002336  020116  053523  051055
002344  043505  040440  042116
002352  050040  042522  051523
002360  041440  047117  044524
002366  052516  006505     012
002373     125  042116  043105          ;ASCII  /UNDEFINED AND/OR DESTRUCTIVE LIMITS WILL BE SET 2156-17470/
002400  047111  042105  040440
002406  042116  047457  020122
002414  042504  052123  052522
002422  052103  053111  020105
002430  044514  044515  051524
002436  053440  046111  020114
002444  042502  051440  052105
002452  031040  032461  026460
002460  033461  033464     060
002465     015  000012                  ;ASCIZ  <15><12>
        000001                          ;END
```

BELLCT	001600	BINCT	001714	BUFFER	002154	CNTR	001604
CONSET	000410	CONTCK	001276	CRLF	001736	DOAGN	001326
ENDADR	001310	ERROR	001474	ERRORA	001520	FORM	001656
HILMT	000206	HISET	000362	LOLMT	000204	MKNUM	001636
MSG1	002010	MSG2	002160	NOP =	000240	PART	000412
PART1	000420	PART2	000720	PRTAB	001614	P1	001240
P1A	000424	P1B	000444	P1C	000462	P1D	000466
P1E	000500	P1F	000522	P1G	000550	P1H	000554
P1K	000574	P1L	000612	P1M	000616	P1N	000644
P1P	000662	P1R	000672	P2A	000724	P2B	000742
P2C	000762	P2D	000766	P2E	001004	P2F	001022
P2G	001050	P2H	001054	P2K	001072	P2L	001112
P2M	001110	P2N	001146	P2P	001162	P2R	001172
QEF	002054	QEFA	002156	RBPC	001262	SEE	000256
SPACE	001762	START	000216	START1	000250	STI	000334
SWCHK	001570	SWREG	000214	TCSR	000210	TDBR	000212
TLB	000264	TOODLE	001712	TOP	001716	TOP1	001742
WAIT1	001670	XOR	001610	XORA	001432	XORB	001472
XORCK	001332	XORCKA	001370	XORCKB	001434	XORCK1	001366
XORFLG	001612	$CRLF	002005	$SPACE	002004	$1	000314
$2	000320	. =	002470				

ERRORS DETECTED: 0

```
*DZMMGB,DZMMGB=DZMMGB/SOL
RUN-TIME:  1 3 0 SECONDS
CORE USED:  3K
```