digital

MAINDEC CHANGE NOTICE

65MCN

CHANGE NO.
11-DZQMB-A - 1
Sheet 1 of 1

AUTHOR	PROGRAM DATE	PRODUCT LINE	MAINDEC NUMBER
John Adams DATE EXT. 8/31/72 2349	MAY 1972	PDP-11/45 V6706199	11-DCQMB-A

PROGRAM NAME Memory Exerciser **DEVICE** PDP-11

ITEM

1. Problem: Program erroneously types error type outs when a non multiple of 4K of memory is installed i.e. 34K. The following patch is necessary only for non multiples of 4K of memory if the total memory size exceeds 28K.
Solution: Patch the following locations:

LOCATION	FROM	TO	COMMENT
3132	427Ø2	527Ø3	BIS #20000,R3
3134	16ØØØØ	2ØØØØ	
3136	1Ø2Ø3	2Ø7	RTS PC

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZQMB-A-D
PRODUCT NAME:	0-124K MEMORY EXERCISER
DATE CREATED:	15 MAY 1972
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JOHN ADAMS

1.0 ABSTRACT

PROGRAM DZQMB TESTS CONTIGUOUS MEMORY ADDRESS FROM 000000 TO 757776. IT VERIFIES THAT EACH ADDRESS IS UNIQUE (AN ADDRESS TEST) AND THAT EACH MEMORY LOCATION CAN BE READ/WRITTEN RELIABLY (WORST CASE NOISE TESTS).

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 FAMILY PROCESSOR
OPTIONAL...
KT11-C OR KT11-D MEMORY MANAGEMENT OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINE USES MEMORY 0-17777

2.3 PRELIMINARY PROGRAMS

BASIC PROCESSOR TESTS
KT11-C/KT11-D LOGIC TESTS

3.0 LOADING AND STARTING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER
LOAD ADDRESS 200
PRESS START.
THE PROGRAM WILL LOOP AND RING BELL ON COMPLETION.
PASS COUNT MAY BE MONITORED IN THE DISPLAY REGISTER OR LOC 1000.
NOTE: THIS PROGRAM RELOCATES THE LOADERS (BOOT AND ABS). TO RESTORE RESTART AT 210. BEFORE RESTARTING INSURE THAT THE PROGRAM HAS NOT BEEN RELOCATED. EXAMINE LOCATION 200. IT SHOULD CONTAIN 137. IF NOT THE PC WILL INDICATE WHICH BANK CONTAINS THE PROGRAM. NEXT START THE PROGRAM AT #17400,WHERE # = BITS 13-15 OF THE PC. THE PROGRAM WILL RELOCATE BACK TO 0-4K AND HALT AT 176. PRESS CONTINUE TO RESUME TESTING.

4.0 SWITCH SETTINGS

SW15 = 1 OR UP.... HALT ON ERROR
NOTE: IF SW15=1 WHEN AN ERROR OCCURS THE PROGRAM WILL HALT; IF SW15 IS RAISED AFTER THE ERROR TYPEOUT BEGINS THE PROGRAM WILL HALT WHEN THE TYPEOUT COMPLETES.
SW14 = 1 OR UP.... LOOP SUBTEST
SW13 = 1 OR UP.... INHIBIT ERROR TYPEOUT
SW11 = 1 OR UP.... INHIBIT SUBTEST ITERATION
SW10 = 1 OR UP.... RING BELL ON ERROR
SW8 = 1 OR UP .... LOAD PDP11/45 MICRO BREAK REGISTER
SW7-SW0..... VALUE TO BE LOADED

5.0 SUBROUTINE ABSTRACTS

5.1 SCOPE

THE SCOPE (EMT) SERVICE ROUTINE STORES IN R1 THE PC OF THE LAST TEST SUCCESSFULLY EXECUTED AND MAY BE USED AS AN AID IN DEBUGGING IF THE PROGRAM 'BOMBS' BECAUSE OF A HARDWARE FAILURE. ADDITIONALLY THE SCOPE ROUTINE SETS THE STACK POINTER TO ITS INITIAL SETTING (SEE SEC 8.2) AND ENTERS EACH TEST IN KERNEL MODE, PREVIOUS KERNEL MODE. THE SCOPE ROUTINE ALSO CONTAINS INSTRUCTIONS TO LOAD THE MICRO BREAK REGISTER (SEE SEC 4.0 FOR SWITCH SETTINGS).

6.0 ERRORS

THESE TESTS PRINT OUT THE PC WHERE THE ERROR HAS DETECTED, THE FAILING ADDRESS, THE GOOD DATA, AND THE BAD DATA I.E.

PC=XXXXXX ADDRESS XXXXXX GOOD DATA XXXXXX BAD DATA XXXXXX

THE ADDRESS OF THE FAILING LOCATION IS THE TRUE 18 BIT PHYSICAL ADDRESS.

NOTE: WHEN TESTING MEMORY LOCATIONS 0-17776 THE PC TYPED WILL BE 20000 GREATER THAN RELFECTED IN THE PROGRAM LISTING

6.1 ERROR RECOVERY

PRESS CONTINUE OR RESTART AT 200 OR PREVIOUS SCOPE.

CAUTION! DO NOT ATTEMPT TO RESTART IF THE PROGRAM IS IN A RELOCATED POSITION. LISTING REFLECTS RESTART PROCEDURES FOR RELOCATED PROGRAM.

7.0 RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7,2 OPERATIONAL RESTRICTION

PROGRAM CHECKS CONTIGUOUS MEMORY

SPECIAL NOTICE: THIS PROGRAM TESTS ALL OF CONTIGUOUS MEMORY,HOWEVER IF ALL MEMORY IS NOT CONTIGUOUS THE PROGRAM MAY BE PATCHED SO AS TO TEST MEMORY BEYOND THE MISSING ADDRESSES. FOR EXAMPLE, IF NO MEMORY EXISTS BETWEEN 140000-200000 THE FOLLOWING PATCHES WILL ALLOW WORST CASE NOISE TESTING ABOVE 200000.

LOCATION	PATCH	
3602	137	JUMPS AROUND 1 XOR 8 PATTERN
3604	3630	FROM 140000-200000
4426	137	JUMPS AROUND 3 XOR 9 PATTERN
4430	4454	FROM 140000-200000
5164	137	JUMPS AROUND 8 XOR 13 PATTERN
5166	5200	FROM 140000-200000

NOTE: THE PROGRAM IN THE ABOVE CASE WILL REPORT THE LAST MEMORY ADDRESS AS BEING 140000, AND WILL NOT INSURE UNIQUE ADDRESSING BEYOND 140000.

8,0 MISCELLANEOUS

IF THE PROGRAM HALTS IN THE TRAP/INTERRUPT VECTOR AREA (0-1000), EXAMINE REGISTER 6 (THE KERNEL STACK PTR), R6 CONTAINS THE ADDRESS WHERE THE PC OF THE INSTRUCTION THAT CAUSED THE TRAP ABORT IS STORED. SEE ALSO R1 (R1 SPECIFIES THE LAST TEST COMPLETED).

NOTE: THE PDP11/45 WILL DISPLAY THE TRAP VECTOR ADDRESS+4 IN THE ADDRESS LIGHTS. THUS A TRAP TO 4 (BUS ERROR) WILL DISPLAY 10 IN THE ADDRESS LIGHTS.

8,1 STACK POINTER

THE STACK POINTER IS INITIALLY SET TO THE FOLLOWING VALUE:
KERNEL #500
AND IS RESET TO THIS VALUE AT THE START OF EACH SUBTEST.

8,2 PASS COUNT

200(8) PASSES ARE REQUIRED FOR COMPLETION OF THIS PROGRAM; AT WHICH TIME THE BELL WILL RING AT THE TTY. THE PASS COUNT MAY BE OBSERVED BY TURNING THE SWITCH TO THE DISPLAY POSITION. THE PASS COUNT SHOULD BE MONITORED IN THE EVENT THAT THE PROGRAM ENTERS AN UNDEFINED LOOP. THE PASS COUNT IS ALSO STORED IN LOCATION 1000.

9.0 PROGRAM DESCRIPTION

THE PROGRAM VERIFIES EACH ADDRESS BY WRITING THE VALUE OF EACH ADDRESS INTO ITSELF STARTING AT LOCATION 20000 AND ENDING AT THE LAST LOCATION IN MEMORY. THE VALUE OF THE LAST LOCATION +2 IS TYPED ON THE TTY. NEXT THE VALUES WRITTEN ARE VERIFIED. TO COMPLETE THE ADDRESS TEST THE COMPLEMENT VALUE OF EACH MEMORY ADDRESS IS WRITTEN STARTING AT THE LAST MEMORY ADDRESS AND ENDING AT ADDRESS 20000. THE WRITTEN COMPLEMENT VALUES ARE THEN VERIFIED. THIS COMPLETES THE MEMORY ADDRESS PORTION OF THE TEST. THE NEXT PHASE OF TESTING INCLUDES READING, WRITING AND CHECKING MEMORY USING SEVERAL WORST CASE NOISE TEST PATTERNS (1 XOR 8, 3 XOR 9, AND 8 XOR 13). A SUBTEST IS DEDICATED TO CHECKING EACH BANK (4K OR 8K) OF MEMORY. THE TEST PROCEEDS BY EXERCISING EACH BANK OF MEMORY USING THE TEST PATTERNS NOTED ABOVE. NOTE THAT WITH THE MEMORY MANAGEMENT OPTION INSTALLED THAT ALL ADDRESSES ARE WRITTEN, READ AND CHECKED WITH THE MEMORY MANAGEMENT ENABLED. AFTER ALL MEMORY FROM 20000 TO THE LAST ADDRESS HAS BEEN TESTED AS DESCRIBED ABOVE THE PROGRAM RELOCATES ITSELF AND BEGINS TESTING 0-17776 (1 XOR 8) AND 0-37776 (3 XOR 9, 8 XOR 13). WHEN ALL TESTING IS COMPLETE THE PROGRAM RELOCATES TO ITS ORIGINAL POSITION (0-1776), INCREMENTS THE PASS COUNT (LOCATION 1000) AND RESTARTS BEGINNING WITH THE WORST CASE NOISE TESTS. THE BELL WILL RING AT THE TELETYPE WHEN 200 (8) PASSES HAVE COMPLETED, AND THE PROGRAM WILL RESTART BEGINNING WITH THE MEMORY ADDRESS TESTS.

```
                        .NLIST  SEQ,MD,MC
                        .LIST   ME
                        .ABS
                        .TITLE  TEST DZQMB-A 0-124K MEMORY EXERCISER
                    ;COPYRIGHT 1972 DIGITAL EQUIPMENT CORP., MAYNARD,MASS.

                    ;THIS TEST CHECKS THAT ALL MEMORY ADDRESSES ARE UNIQUE VIA ADDRESS TESTS
                    ;AND CHECKS DATA RELIABILITY OF MEMORY VIA WORST CASE NOISE TEST PATTERNS

                    ;LOADING AND STARING INSTRUCTIONS
                    ;LOAD ADDRESS 200 AND START
                    ;NOTE:  PROGRAM WILL RUN WORST CASE TEST PATTERNS IN LOWEST 4K
                    ;THUS THE PROGRAM CANNOT BE RESTARTED AT 200 IF RELOCATED, TO PREVENT
                    ;RELOCATION FROM OCCURING DEPOSIT 1 INTO LOCATION 42 (NOT NECESSARY
                    ;IF LOADED VIA ACT11), THIS ACTION WILL PREVENT RELOCATION AND ALSO
                    ;INHIBIT TESTING MEMORY IN LOWEST 4K.
                    ;THIS PROGRAM ALSO RELOCATES THE ABS AND BOOT LOADERS TO ALLOW TESTING
                    ;OF MEMORY, TO RESTORE THE LOADERS RESTART AT 210.
                    ;       STACK POINTER IS SET AT 500
                    ;       BELL WILL RING WHEN TEST IS COMPLETE

                    ;GENERAL REGISTER ASSIGNMENTS
000000                      R0=%0
000001                      R1=%1
000002                      R2=%2
000003                      R3=%3
000004                      R4=%4
000005                      R5=%5
000006                      SP=%6
000007                      PC=%7
000000                      R10=%0
000001                      R11=%1
000002                      R12=%2
000003                      R13=%3
000004                      R14=%4
000005                      R15=%5

                    ;FLOATING POINT REGISTERS
000000                      AC0=%0
000001                      AC1=%1
000002                      AC2=%2
000003                      AC3=%3
000004                      AC4=%4
000005                      AC5=%5

                    ;STACK POINTER REGISTERS
000006                      KSP=%6                  ;KERNEL STACK POINTER
000006                      SSP=%6                  ;SUPERVISOR STACK POINTER
000006                      USP=%6                  ;USER STACK POINTER

                    ;STATUS REGISTER (PSW) BIT ASSIGNMENTS
000001                      C=1                     ;C BIT
000002                      V=2                     ;V BIT
000004                      Z=4                     ;Z BIT
```

```
000010                  N=10                    ;N BIT
000020                  T=20                    ;'T' BIT
000340                  PRTY7=340               ;PRIORITY LEVEL 7
000200                  PRTY4=200               ;PRIORITY LEVEL 4
000000                  KM=000000               ;KERNEL MODE
040000                  SM=040000               ;SUPERVISORY MODE
140000                  UM=140000               ;USER MODE
000000                  PKM=000000              ;PREVIOUS KERNEL MODE
010000                  PSM=010000              ;PREVIOUS SUPERVISORY MODE
030000                  PUM=030000              ;PREVIOUS USER MODE
004000                  REG=004000              ;SELECT R10-R15

                        ;VECTOR ADDRESSES
000004                  ERRVEC=4                ;ADDRESS OF ERROR VECTOR
000010                  RESVEC=10               ;ADDRESS OF RESERVED INST. TRAP VECTOR
000014                  TBITVEC=14              ;ADDRESS OF 'T' BIT TRAP VECTOR
000014                  TRTVEC=14               ;ADDRESS OF 'TRACE' TRAP VECTOR
000014                  BPTVEC=14               ;ADDRESS OF 'BREAKPOINT' TRAP VECTOR
000020                  IOTVEC=20               ;ADDRESS OF IOT TRAP VECTOR
000024                  PFVEC=24                ;ADDRESS OF POWER FAIL TRAP VECTOR
000030                  EMTVEC=30               ;ADDRESS OF EMT VECTOR
000034                  TRAPVEC=34              ;ADDRESS OF TRAP VECTOR
000064                  TPVEC=64                ;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
000240                  PIRVEC=240              ;ADDRESS OF PIRQ VECTOR
000244                  FPEVEC=244              ;ADDRESS OF FLOATING POINT INT. VECTOR
000250                  MMVEC=250               ;ADDRESS OF MEM MGMT ERROR TRAP VECTOR

                        ;REGISTER ADDRESSES
177776                  PSW=177776              ;ADDRESS OF STATUS REGISTER
177774                  SLR=177774              ;ADDRESS OF STACK LIMIT REGISTER
177772                  PIRQ=177772             ;ADDRESS OF PROGRAM INTERRUPT REQUEST
177770                  UBREAK=177770           ;ADDRESS OF MICRO BREAK REGISTER
177560                  TKS=177560              ;ADDRESS OF KEYBOARD CSR
177562                  TKB=177562              ;ADDRESS OF KEYBOARD BUFFER
177564                  TPS=177564              ;ADDRESS OF TELEPRINTER CSR
177566                  TPB=177566              ;ADDRESS OF TELEPRINTER BUFFER
177570                  SWR=177570              ;ADDRESS OF CONSOL SWITCH REGISTER
177570                  DISPLAY=177570          ;ADDRESS OF CONSOL DISPLAY REGISTER

                        ;INITIAL STACK POINTER SETTING
000500                  STKPTR=500              ;

                        ;MISCELLANEOUS BIT ASSIGNMENTS
100000                  BIT15=100000
040000                  BIT14=40000
020000                  BIT13=20000
000400                  BIT8=400
000100                  BIT6=100
010000                  PIR4=10000              ;LEVEL 4 PROGRAM INT. RQST. (FOR PIRQ)

                        ;MEM MGMT REGISTER SSR0 BIT ASSIGNMENTS
000001                  ENMM=1                  ;ENABLE MEMORY MANAGEMENT
000000                  VP0=0                   ;'VIRTUAL' PAGE 0
000002                  VP1=2                   ;ETC
```

```
000004                          VP2=4
000006                          VP3=6
000010                          VP4=10
000012                          VP5=12
000014                          VP6=14
000016                          VP7=16
000020                          DS=20                   ;'D' SPACE PAGE
000000                          IS=00                   ;'I' SPACE PAGE
000140                          UPG=140                 ;USER PAGE
000040                          SPG=40                  ;SUPERVISOR PAGE
000000                          KPG=000                 ;KERNEL PAGE
000200                          IC=200                  ;INSTRUCTION COMPLETE
000400                          DM=400                  ;DESTINATION MODE
001000                          TE=1000                 ;TRAP ENABLE
004000                          OSTF=4000               ;OST ABORT FLAG
010000                          MMTF=10000              ;MEMORY MANAGEMENT TRAP FLAG
020000                          AVA=20000               ;ACCESS VIOLATION ABORT
040000                          PLA=40000               ;PAGE LENGTH ABORT FLAG
100000                          NRA=100000              ;NON-RESIDENT ABORT

                        ;SR1 BIT ASSIGNMENTS
000010                          S1=10
000020                          S2=20
000040                          S4=40
000060                          S6=60
000100                          S8=100
000370                          SM1=370
000360                          SM2=360
000340                          SM4=340
000320                          SM6=320
000300                          SM8=300
000000                          D0=0
004000                          D1=4000
010000                          D2=10000
174000                          DM1=174000
170000                          DM2=170000
000000                          DR0=000
000400                          DR1=400
001000                          DR2=1000
001400                          DR3=1400
002000                          DR4=2000
002400                          DR5=2400
003000                          DR6=3000
003400                          DR7=3400

                        ;SR3 BIT ASSIGNMENTS
000001                          UDE=1                   ;USER 'D' SPACE ENABLE
000002                          SDE=2                   ;SUPERVISOR 'D' SPACE ENABLE
000004                          KDE=4                   ;KERNEL 'D' SPACE ENABLE

                        ;MEMORY MANAGEMENT REGISTER ADDRESS ASSIGNMENTS
177572                          SR0=177572              ;ADDRESS OF MEM MGMT REGISTER SR0
177574                          SR1=177574              ;    "    "       "          "   SR1
177576                          SR2=177576              ;    "    "       "          "   SR2
```

```
172516                          SR3=172516              ;ADDRESS OF MEM MGMT REGISTER SR3

177600                          UIPDR0=177600           ;ADDRESS OF USER 'I' PAGE DESCRIPTOR
177602                          UIPDR1=177602           ;REGISTERS
177604                          UIPDR2=177604
177606                          UIPDR3=177606
177610                          UIPDR4=177610
177612                          UIPDR5=177612
177614                          UIPDR6=177614
177616                          UIPDR7=177616

177620                          UDPDR0=177620           ;ADDRESS OF USER 'D' PAGE DESCRIPTOR
177622                          UDPDR1=177622           ;REGISTERS
177624                          UDPDR2=177624
177626                          UDPDR3=177626
177630                          UDPDR4=177630
177632                          UDPDR5=177632
177634                          UDPDR6=177634
177636                          UDPDR7=177636

177640                          UIPAR0=177640           ;ADDRESS OF USER 'I' PAGE ADDRESS
177642                          UIPAR1=177642           ;REGISTERS
177644                          UIPAR2=177644
177646                          UIPAR3=177646
177650                          UIPAR4=177650
177652                          UIPAR5=177652
177654                          UIPAR6=177654
177656                          UIPAR7=177656

177660                          UDPAR0=177660           ;ADDRESS OF USER 'D' PAGE ADDRESS
177662                          UDPAR1=177662           ;REGISTERS
177664                          UDPAR2=177664
177666                          UDPAR3=177666
177670                          UDPAR4=177670
177672                          UDPAR5=177672
177674                          UDPAR6=177674
177676                          UDPAR7=177676

172200                          SIPDR0=172200           ;ADDRESS OF SUPERVISOR 'I' PAGE
172202                          SIPDR1=172202           ;DESCRIPTOR REGISTERS
172204                          SIPDR2=172204
172206                          SIPDR3=172206
172210                          SIPDR4=172210
172212                          SIPDR5=172212
172214                          SIPDR6=172214
172216                          SIPDR7=172216

172220                          SDPDR0=172220           ;ADDRESS OF SUPERVEISOR 'D' PAGE
172222                          SDPDR1=172222           ;DESCRIPTOR REGISTERS
172224                          SDPDR2=172224
172226                          SDPDR3=172226
172230                          SDPDR4=172230
172232                          SDPDR5=172232
172234                          SDPDR6=172234
```

```
172236                          SDPDR7=172236

172240                          SIPAR0=172240           ;ADDRESS OF SUPERVISOR 'I' PAGE
172242                          SIPAR1=172242           ;ADDRESS REGISTERS
172244                          SIPAR2=172244
172246                          SIPAR3=172246
172250                          SIPAR4=172250
172252                          SIPAR5=172252
172254                          SIPAR6=172254
172256                          SIPAR7=172256

172260                          SDPAR0=172260           ;ADDRESS OF SUPERVISOR 'D' PAGE
172262                          SDPAR1=172262           ;ADDRESS REGISTERS
172264                          SDPAR2=172264
172266                          SDPAR3=172266
172270                          SDPAR4=172270
172272                          SDPAR5=172272
172274                          SDPAR6=172274
172276                          SDPAR7=172276

172300                          KIPDR0=172300           ;ADDRESS OF KERNEL 'I' PAGE
172302                          KIPDR1=172302           ;DESCRIPTOR REGISTERS
172304                          KIPDR2=172304
172306                          KIPDR3=172306
172310                          KIPDR4=172310
172312                          KIPDR5=172312
172314                          KIPDR6=172314
172316                          KIPDR7=172316

172320                          KDPDR0=172320           ;ADDRESSES OF KERNEL 'D' PAGE
172322                          KDPDR1=172322           ;DESCRIPTOR REGISTERS
172324                          KDPDR2=172324
172326                          KDPDR3=172326
172330                          KDPDR4=172330
172332                          KDPDR5=172332
172334                          KDPDR6=172334
172336                          KDPDR7=172336

172340                          KIPAR0=172340           ;ADDRESSES OF KERNEL 'I' PAGE
172342                          KIPAR1=172342           ;ADRESS REGISTERS
172344                          KIPAR2=172344
172346                          KIPAR3=172346
172350                          KIPAR4=172350
172352                          KIPAR5=172352
172354                          KIPAR6=172354
172356                          KIPAR7=172356

172360                          KDPAR0=172360           ;ADDRESSES OF KERNEL 'D' PAGE
172362                          KDPAR1=172362           ;ADDRESS REGISTERS
172364                          KDPAR2=172364
172366                          KDPAR3=172366
172370                          KDPAR4=172370
172372                          KDPAR5=172372
172374                          KDPAR6=172374
```

```
172376                          KDPAR7=172376

                        ;ACCESS CONTROL FIELD DEFINITIONS (IN PDR)
000000                          NR0=0                   ;NON-RESIDENT ABORT ALL REFS.
000001                          RDOT=1                  ;TRAP ON READ,ABORT ON WRITE
000002                          RDO=2                   ;READ,ABORT ON WRITE
000003                          NR3=3                   ;UNUSED ABORT ALL
000004                          RWT=4                   ;TRAP ON READ & WRITE
000005                          RWTW=5                  ;READ,TRAP ON WRITE
000006                          RW=6                    ;READ & WRITE
000007                          NR7=7                   ;ABORT ALL
000010                          ED=10                   ;EXPANSION DIRECTION BIT
000000                          UP=00                   ;EXPAND UP
000010                          DWN=10                  ;EXPAND DOWN
000100                          W=100                   ;'W' BIT
000200                          A=200                   ;'A' BIT

                        ;INSTRUCTION EQUATES
104400                          HLT=TRAP
104000                          SCOPE=EMT               ;SCOPE IS AN EMT TRAP
```

```
                    000000                          .=0
000000  000000                          .WORD   0               ;SPECIAL TRAP/INTERRUPT CATCHER IF PRO-
000002  000000                          .WORD   0               ;GRAM HALTS AT 0 THEN ADDRESS WAS NOT
                                                                ;LOADED PROPERLY FROM VECTOR.
                    000030                          .=EMTVEC
000030  001004                          .WORD   SCOPEA
                    000200                          .=200
000200  000137  002506                  JMP     @#START         ;GO TO START OF TEST
000204  000137  002532                  JMP     @#START1        ;RESTART ADDRESS
                    001000                          .=1000
001000  000000              ICNT:   0                           ;CONTAINS PASS COUNT
001002  000000              MMAVA:  0                           ;MEM MGMT AVAILABLE INDICATOR
                                                                ;0=NOT AVAIL,-1=AVAIL

                                    ;SCOPE (EMT) SERVICE ROUTINE
                                    ;THIS ROUTINE ALLOWS THE SUBTEST TO BE CONTINUOUSLY LOOPED, ITERATED
                                    ;(OR NOT ITERATED) BEFORE BEGINNING NEXT SUBTEST
001004  032737  040000  177570  SCOPEA: BIT     #BIT14,@#SWR    ;CONTINUOUSLY LOOP TEST?
001012  001412                          BEQ     SCOPEC
001014  012706  000500          SCOPEB: MOV     #STKPTR,SP      ;SET INITIAL STACK POINTER
001020  032737  000400  177570          BIT     #400,@#SWR      ;LOAD PDP11/45 MICRO BREAK REG?
001026  001403                          BEQ     .+10
001030  113737  177570  177770          MOVB    @#SWR,@#UBREAK  ;LOAD MICRO BREAK REG WITH SR0-7
001036  000111                          JMP     (R1)            ;RETURN
001040  032737  004000  177570  SCOPEC: BIT     #4000,@#SWR     ;SUBTEST ITERATION DESIRED?
001046  001006                          BNE     SCOPEE          ;BRANCH IF NO ITERATION DESIRED?
001050  005327                          DEC     (PC)+           ;DECREMENT SUBTEST ITERATION COUNT
001052  000010              SCOPED: 10                          ;CONTAINS SUBTEST ITERATION COUNT
001054  001357                          BNE     SCOPEB
001056  012767  000010  177766          MOV     #10,SCOPED      ;RESET ITERATION COUNT
001064  011601              SCOPEE: MOV     (SP),R1         ;GET ADDRESS OF NEXT TEST
001066  000752                          BR      SCOPEB
```

```
                001070                          RETURN=.
                000034                          .=TRAPVEC
000034          001070                          .WORD   ERROR
000036          000340                          .WORD   PRTY7
                001070                          .=RETURN

                                        ;ERROR SERVICE CALLED BY TRAP (HLT) INSTRUCTION
001070  005737  177570          ERROR:  TST     @#SWR           ;HALT ON ERROR?
001074  100002                          BPL     .+6
001076  000000                          HALT
001100  000471                          BR      3$
001102  032737  020000  177570          BIT     #20000,@#SWR    ;PRINT OUT DESIRED?
001110  001052                          BNE     1$              ;BRANCH IF NO PRINTOUT
001112  011667  000156                  MOV     (SP),SAVPC      ;GET PC OF ERROR CALL
001116  004767  000212                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001122  010267  000150                  MOV     R2,SAVR2        ;SAVE R2
001126  016702  000142                  MOV     SAVPC,R2
001132  004767  000310                  JSR     PC,$FORM0       ;GO TO FORMAT ROUTINE
001136  004567  000242                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001142  001300                          ERRPC
001144  004567  000234                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001150  002473                          DIGITS
001152  016702  000120                  MOV     SAVR2,R2
001156  004767  000264                  JSR     PC,$FORM0       ;GO TO FORMAT ROUTINE
001162  004567  000216                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001166  002451                          ADRESS
001170  004567  000210                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001174  001304                          XMTDAT
001176  010067  000470                  MOV     R0,D2BTYP
001202  004767  000466                  JSR     PC,O2A          ;GO PRINT VALUE
001206  004567  000172                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001212  001317                          RECDAT
001214  010367  000452                  MOV     R3,D2BTYP
001220  004767  000450                  JSR     PC,O2A
001224  004567  000154                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001230  002503                          $CRLF
001232  004767  000122                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001236  032737  002000  177570  1$:     BIT     #2000,@#SWR     ;RING BELL ON ERROR
001244  001403                          BEQ     2$
001246  004567  000132                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
001252  001332                          BELL
001254  005737  177570          2$:     TST     @#SWR           ;HALT AFTER PRINT OUT
001260  100001                          BPL     .+4
001262  000000                          HALT
001264  010042                  3$:     MOV     R0,-(R2)        ;RESTORE CORRECT DATA TO ADDRESS
001266  062702  000002                  ADD     #2,R2
001272  000002                          RTI

001274  000000          SAVPC:  0
001276  000000          SAVR2:  0
001300  041520  000075  ERRPC:  .ASCIZ  'PC='
001304  047507  042117  042040  XMTDAT: .ASCIZ  'GOOD DATA='
001312  052101  036501  000
001317  040040  040502  020104  RECDAT: .ASCIZ  ' BAD DATA='
```

```
001324  040504  040524  000075
001332  000007                  BELL:   .ASCIZ  <7>
                                        .EVEN
                                ;ROUTINE TO SAVE REGISTERS ON THE STACK
                                ;CALLED BY SAVE MACRO OR JSR    PC,$SAVR
001334  012667  000016          $SAVR:  MOV     (SP)+,1$        ;SAVE RETURN PC
001340  010546                          MOV     %5,-(SP)
001342  010446                          MOV     %4,-(SP)
001344  010346                          MOV     %3,-(SP)
001346  010246                          MOV     %2,-(SP)
001350  010146                          MOV     %1,-(SP)
001352  010046                          MOV     %0,-(SP)
001354  012707                          MOV     (PC)+,PC        ;RETURN
001356  000000                  1$:     0                       ;CONTAINS RETURN ADDRESS

                                ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
                                ;CALLED BY RESTORE MACRO OR JSR  PC,$RESTR
001360  012667  000016          $RESTR: MOV     (SP)+,1$        ;SAVE RETURN PC
001364  012600                          MOV     (SP)+,%0
001366  012601                          MOV     (SP)+,%1
001370  012602                          MOV     (SP)+,%2
001372  012603                          MOV     (SP)+,%3
001374  012604                          MOV     (SP)+,%4
001376  012605                          MOV     (SP)+,%5
001400  012707                          MOV     (PC)+,PC        ;RETURN
001402  000000                  1$:     0                       ;CONTAINS RETURN ADDRESS

                                ;ROUTINE TO PRINT ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
001404  000240                  $PRINT: NOP
001406  010546                          MOV     R5,-(SP)
001410  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN PC
001414  011505                          MOV     (R5),R5         ;GET MESSAGE ADDRESS
001416  066705  004300                  ADD     RELOCF,R5       ;ADD RELOCATION FACTOR
001422  105715                  1$:     TSTB    (R5)            ;CHECK FOR TERMINATOR
001424  001002                          BNE     2$
001426  012605                          MOV     (SP)+,R5        ;GET RETURN ADDRESS
001430  000205                          RTS     R5              ;RETURN
001432  112537  177566          2$:     MOVB    (R5)+,@#TPB     ;PRINT CHARACTER
001436  105737  177564                  TSTB    @#TPS           ;WAIT UNTIL
001442  100375                          BPL     .-4             ;DONE
001444  000766                          BR      1$

                                ;ROUTINE TO PLACE ASCII VALUE OF AN ADDRESS IN TO ADDRESS MESSAGE
001446  000240                  $FORM0: NOP
001450  066767  004246  000014          ADD     RELOCF,11$+2
001456  066767  004240  000134          ADD     RELOCF,41$+2
001464  004767  177644                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001470  012704  002473          11$:    MOV     #DIGITS,R4      ;ADDRESS WHERE ASCII VALUES ARE STORED
001474  005003                          CLR     R3              ;WORKING & INDEX REGISTER
001476  162702  000002                  SUB     #2,R2           ;ADJUST ADDRESS
001502  010205                          MOV     R2,R5           ;SAVE
001504  010501                          MOV     R5,R1
001506  005767  177270                  TST     MMAVA           ;CHECK IF MEM MGMT IS AVAILABLE
001512  001426                          BEQ     1$              ;BRANCH IF NOT AVAILABLE
```

```
001514  032737  000001  177572          BIT     #1,@#SR0        ;IS MEM MGMT ENABLED
001522  001422                          BEQ     1$
001524  042701  017777                  BIC     #17777,R1       ;SAVE PAR SELECTOR BITS
001530  000301                          SWAB    R1              ;SWAP BYTES
001532  006001                          ROR     R1
001534  006001                          ROR     R1              ;FORM INDEX VALUE
001536  006001                          ROR     R1
001540  006001                          ROR     R1
001542  017102  001652                  MOV     @PARTAB(1),R2   ;GET CONTENTS OF PAR
001546  012700  000006                  MOV     #6,R0           ;SHIFT COUNT
001552  006302                          ASL     R2              ;SHIFT KIPAR1 6 PLACES LEFT
001554  006103                          ROL     R3              ;2 MSB'S GO INTO R3
001556  077003                          SOB     R0,.-4
001560  042705  160000                  BIC     #160000,R5      ;CLEAR PAR SELECTOR BITS
001564  060502                          ADD     R5,R2           ;FORM 18 BIT ADDRESS
001566  005503                          ADC     R3              ;IN R2 & R3
001570  006302                  1$:     ASL     R2              ;FIRST DIGIT TO R3
001572  006103                          ROL     R3
001574  012700  000006                  MOV     #6,R0           ;DIGIT COUNT
001600  000404                          BR      3$              ;PRINT FIRST DIGIT
001602  006302                  2$:     ASL     R2
001604  006103                          ROL     R3
001606  005305                          DEC     R5
001610  001374                          BNE     2$
001612  012705  000003          3$:     MOV     #3,R5           ;DIGIT SHIFT COUNT
001616  116324  002434          41$:    MOVB    DIGTAB(3),(4)+  ;LOAD DIGIT INTO MESSAGE
001622  005003                          CLR     R3              ;CLEAR INDEX
001624  005300                          DEC     R0              ;DEC DIGIT COUNT
001626  001365                          BNE     2$
001630  004767  177524                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001634  046767  004062  177630          BIC     RELOCF,1$+2
001642  046767  004054  177750          BIC     RELOCF,41$+2
001650  000207                          RTS     PC              ;RETURN

001652  172340          PARTAB: KIPAR0
001654  172342                  KIPAR1
001656  172344                  KIPAR2
001660  172346                  KIPAR3
001662  172350                  KIPAR4
001664  172352                  KIPAR5
001666  172354                  KIPAR6
001670  172356                  KIPAR7
                        ;ROUTINE TO TYPE OCTAL VALUE PLACED IN D2BTYP
001672  000000          D2BTYP: 0               ;OCTAL VALUE TO BE TYPED
001674  013746  177564  02A:    MOV     @#TPS,-(SP)     ;SAVE  TPCSR
001700  004767  177430          JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001704  016700  177762          MOV     D2BTYP,R0       ;GET VALUE
001710  012703  000006          MOV     #6,R3           ;COUNTER
001714  005002                  CLR     R2              ;WORKING REGISTER
001716  006100                  ROL     R0
001720  006102                  ROL     R2
001722  062702  000260  1$:     ADD     #260,R2         ;FORM ASCII VALUE
001726  105737  177564          TSTB    @#TPS           ;AND TYPE
001732  100375                  BPL     .-4
```

```
001734  010237  177566                  MOV     R2,@#TPB
001740  005002                          CLR     R2
001742  006100                          ROL     R0
001744  006102                          ROL     R2
001746  006100                          ROL     R0
001750  006102                          ROL     R2
001752  006100                          ROL     R0
001754  006102                          ROL     R2
001756  005303                          DEC     R3
001760  001360                          BNE     1$
001762  004767  177372                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001766  105737  177564                  TSTB    @#TPS           ;WAIT FOR MESSAGE TO FINISH
001772  100375                          BPL     .-4
001774  012637  177564                  MOV     (SP)+,@#TPS
002000  000207                          RTS     PC

                                ;ROUTINE TO SAVE ABS LOADER
002002  012700  017776          $LDR:   MOV     #17776,R0
002006  012737  002020  000004          MOV     #2$,@#ERRVEC    ;SET TIME OUT TRAP VECTOR
002014  005720                          TST     (R0)+
002016  000776                          BR      .-2
002020  022626                  2$:     CMP     (SP)+,(SP)+
002022  162700  000302                  SUB     #302,R0         ;POINT R0 BACK TO LOADER
002026  010067  000052                  MOV     R0,$LDR1        ;SAVE FOR RESTORE ROUTINE
002032  012702  000140                  MOV     #140,R2         ;WORD COUNT
002036  012703  002106                  MOV     #$LDR1+2,R3     ;WHERE LOADER IS TO BE STORED
002042  012023                  1$:     MOV     (R0)+,(R3)+     ;STORE LOADER
002044  005302                          DEC     R2
002046  001375                          BNE     1$
002050  000207                          RTS     PC              ;RETURN

                                ;ROUTINE TO RESTORE LOADER
002052  016705  000026          $RLDR:  MOV     $LDR1,R5        ;GET FIRST ADDRESS OF WHERE LOADER IS
                                                                ;TO BE RESTORED
002056  012704  002106                  MOV     #$LDR1+2,R4     ;ADDRESS WHERE LOADER IS STORED
002062  012702  000140                  MOV     #140,R2         ;WORD COUNT
002066  012425                  1$:     MOV     (R4)+,(R5)+     ;RESTORE
002070  005302                          DEC     R2
002072  001375                          BNE     1$
002074  004567  177304                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
002100  002406                          $LDRM
002102  000000                          HALT

002104  000000                  $LDR1:  0                       ;FIRST ADDRESS
        002406                          .=.+300                 ;SAVE 300 LOCATIONS
002406  047514  042101  051105  $LDRM:  .ASCIZ  'LOADER IS RESTORED'<15><12>
002414  044440  020123  042522
002422  052123  051117  042105
002430  005015     000
        002434
                                        .EVEN
                                ;DIGIT TABLE
002434  030460                  DIGTAB: "01
002436  031462                          "23
002440  032464                          "45
```

```
002442  033466                          "67

                                        ;MESSAGES
002444  040514  052123      040   LST:      .ASCII  'LAST '
002451     115  046505  051117   ADRESS:   .ASCII  'MEMORY ADDRESS IS '
002456  020131  042101  051104
002464  051505  020123  051511
002472     040
002473     060  030060  030060   DIGITS:   .ASCIZ  '000000 '
002500  028060     000
002503     015  000012           SCRLF:    .ASCIZ  <15><12>
                                           .EVEN
```

```
                .TITLE  MEMORY ADDRESS TESTS

;THIS TEST ADDRESS MEMORY UP TO 128K AND PROVES 'UNIQNESS' OF ALL
;MEMORY ADDRESS IN A 32K SEGMENT. THE TEST WRITES INTO EACH MEMORY
;ADDRESS THE VALUE OF THAT ADDRESS AND THEN CHECKS FOR THE CORRECT
;DATA IN EACH ADDRESS.
;THE TWELVE MOST SIGNIFICANT BITS OF THE LAST AVAILABLE MEMORY ADDRESS
;IS STORED IN R5.
;STARTING INTRUCTIONS
;       LOAD ADDRESS=200
;       PRESS START
;       STACK POINTER IS AT 500
;**************RESTART AT 210 TO RESTORE LOADER*******************
```

```
                                        ;MEMORY ADDRESS TEST
002506  012737  002532  000202  START:  MOV     #START1,@#202   ;CHANGE START ADDRESS
002514  012706  000500          ;       MOV     #STKPTR,SP      ;SET UP STACK PTR
002520  004767  177256          ;       JSR     PC,$LDR         ;GO SAVE LOADER
        002524                  ;       $RET=.
        000210                  ;       .=210
000210  000167  001636          ;       JMP     $RLDR           ;RESTORE LOADER
        002524                  ;       .=$RET
002524  004567  176654          ;       JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
002530  002503                  ;       $CRLF
002532  012706  000500          START1: MOV     #STKPTR,SP      ;SET STACK PTR
002536  005067  176236          ;       CLR     ICNT            ;CLEAR PASS COUNT
002542  012706  000500          BEGIN:  MOV     #STKPTR,SP      ;SET STACK PTR
002546  016737  176226  177570  ;       MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT
002554  005067  003142          ;       CLR     RELOC           ;CLEAR RELOCATION FACTOR

                                        ;CHECK IF MEMORY MANAGEMENT IS AVAILABLE
002560  012737  002602  000004  ;       MOV     #NOMM,@#ERRVEC  ;SET TIME OUT TRAP
002566  005067  176210          ;       CLR     MMAVA           ;CLEAR INDICATOR
002572  005037  177572          ;       CLR     @#SR0           ;REFERENCE MEM MGMT
002576  005167  176200          ;       COM     MMAVA           ;SET INDICATOR TO -1 IF AVAILABLE
002602  005037  002124          NOMM:   JMP     WRTUP

                                        ;ROUTINE TO SET UP MEMMOMY REGISTERS
002606  000240                  LDMM0:  NOP
002610  005767  176160          ;       TST     MMAVA
002614  001427                  ;       BEQ     1$
002616  012737  077406  172300  ;       MOV     #<100*256.>-<60*UP*RW>,@#KIPDR0     ;SET KIPDR0=RW UP 100 BLOCKS
002624  012737  077406  172302  ;       MOV     #<200*256.>-<60*UP*RW>,@#KIPDR1     ;SET KIPDR1=RW UP 200 BLOCKS
002632  005037  172304  ;       CLR     @#KIPDR2
002636  012737  077406  172316  ;       MOV     #<200*256.>-<60*UP*RW>,@#KIPDR7     ;SET KIPDR7=RW UP 200 BLOCKS
002644  005037  172340  ;       CLR     @#KIPAR0
002650  012737  000200  172342  ;       MOV     #200,@#KIPAR1
002656  012737  007600  172356  ;       MOV     #7600,@#KIPAR7
002664  012737  000001  177572  ;       MOV     #1,@#SR0            ;ENABLE MEM MGMT
002672  000240                  ;       NOP
002674  000207                  1$:     RTS     PC

                                        ;MEMORY MANAGEMENT ABORT ROUTINE FOR WRITE UP
002676  000240                  MMABT0: NOP
002700  012702  020000          ;       MOV     #20000,R2           ;RESET R2
002704  062737  000200  172342  ;       ADD     #200,@#KIPAR1       ;ADVANCE TO NEXT 4K
002712  013716  177776          ;       MOV     @#SR2,(SP)          ;RETURN TO INSTRUCTION THAT
002716  005037  177572          ;       CLR     @#SR0               ;DISABLE MEM MGMT
002722  012737  000001  177572  ;       MOV     #1,@#SR0            ;ENABLE MEM MGMT
002730  000002                  ;       RTI                         ;CAUSED THE ABORT

                                        ;ROUTINE TO WRITE VALUE OF MEMORY ADDRESS INTO MEMORY ADDRESS
                                        ;FOR EXAMPLE ROUTINE WRITES 20000 INTO LOCATION 20000
002732  012737  002770  000004  WRTUP: MOV     #DONE0,@#ERRVEC ;SET TIME OUT TRAP VECTOR
002740  004767  177642          ;       JSR     PC,LDMM0
002744  012737  002676  000250  ;       MOV     #MMABT0,@#MMVEC ;SET MEM MGMT ABORT VECTOR
002752  012702  020000          ;       MOV     #20000,R2       ;FIRST ADDRESS
002756  010203                  ;       MOV     R2,R3           ;LOAD CONSTANT
```

```
002760  010322                          MOV     R3,(R2)+        ;WRITE VALUE OF ADDRESS INTO ADDRESS
002762  062703  000002                   ADD     #2,R3           ;NEXT VALUE
002766  000774                          BR      .-6             ;WRITE UNTIL DONE

002770  012706  000500          DONE0:  MOV     #STKPTR,SP      ;SET STACK PTR
002774  004767  176446                  JSR     PC,$FORM0       ;GO TO FORMAT ROUTINE
003000  004567  176400                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
003004  002444                          LST
003006  004567  176372                  JSR     R5,$PRINT       ;GO TO PRINT ROUTINE
003012  002503                          $CRLF

                                ;ROUTINE TO CHECK THAT VALUE OF MEMORY ADDRESS WAS WRITTEN CORRECTLY
003014  012702  020000                  MOV     #20000,R2       ;SET R2
003020  012737  003056  000004          MOV     #DONE1,@#ERRVEC ;SET TIME OUT TRAP
003026  010200                          MOV     R2,R0
003030  162700  000002                  SUB     #2,R0           ;SUBTRACT 2
003034  004767  177546                  JSR     PC,LDMM0
003040  062700  000002          1$:     ADD     #2,R0
003044  012203                          MOV     (R2)+,R3        ;GET WRITTEN VALUE
003046  020003                          CMP     R0,R3           ;CHECK
003050  001773                          BEQ     1$
003052  104400                          HLT                     ;ERROR! TO DETERMINE WHICH ADDRESS WAS
                                ;WRITTEN IMPROPERLY EXAMINE R2, NEXT EXAMINE MEM MGMT REGISTER KIPAR1
                                ;(IF MEM MGMT IS AVAILABLE), ADD R2 AND KIPAR1 TOGETHER AS SHOWN BELOW

                                ;       R2=2                    0 000 XXX XXX XXX XXX XXX
                                ;       KIPAR1(772342)  0 000 YYY YYY YYY YYY YYY YYY
                                ;       ADDRESS                 ZZZ ZZZ ZZZ ZZZ ZZZ ZZZ

003054  000771                          BR      1$
003056  012706  000500          DONE1:  MOV     #STKPTR,SP      ;SET STACK PTR

                                ;ROUTINE TO WRITE 1'S COMPLEMENT VALUE OF ADDRESS INTO ADDRESS
                                ;FOR EXAMPLE ROUTINE WRITES 157777 INTO ADDRESS 20000

003062  162702  000002                  SUB     #2,R2           ;R2=LAST ADDRESS
003066  010203                          MOV     R2,R3
003070  005103                          COM     R3              ;COMPLEMENT VALUE IN R3
003072  004767  000024                  JSR     PC,LDMM1
003076  012737  003170  000250          MOV     #MMABT1,@#MMVEC ;SET ABORT VECTOR
003104  062703  000002          1$:     ADD     #2,R3
003110  010342                          MOV     %3,-(R2)        ;WRITE COMPLIMENT VALUE INTO ADDRESS
003112  020227  017776                  CMP     R2,#17776
003116  001372                          BNE     1$
003120  000442                          BR      DONE3
                                ;ROUTINE TO LOAD MEM MGMT FOR WRITING DOWN
003122  000240                  LDMM1:  NOP
003124  005767  175652                  TST     MMAVA
003130  001416                          BEQ     1$
003132  042702  160000                  BIC     #160000,R2      ;CLEAR PAR SELECTOR BITS
003136  010203                          MOV     R2,R3
003140  013702  172342                  MOV     @#KIPAR1,R2     ;START FORMING ADDRESS IN R2
003144  072227  000006                  ASH     #6,R2
003150  060203                          ADD     R2,R3
```

```
003152  005103                          COM     R3
003154  012702  040000                  MOV     #40000,R2
003160  162737  000200  172342          SUB     #200,@#KIPAR1   ;ADJUST PAR
003166  000207                  1$:     RTS     PC

                                ;MEM MGMT ABORT SERVICE FOR WRITE DOWN
003170  000240                  MMABT1: NOP
003172  012702  040000                  MOV     #40000,R2       ;RESET R2
003176  162737  000200  172342          SUB     #200,@#KIPAR1
003204  001406                          BEQ     2$
003206  013716  177576                  MOV     @#SR2,(SP)
003212  012737  000001  177572          MOV     #1,@#SR0        ;ENABLE MEM MGMT
003220  000002                          RTI
003222                          2$:
003222  005037  177572                  CLR     @#SR0           ;DISABLE MEM MGMT
003226  000240                  DONE3:  NOP

                                ;SET UP TO CHECK COMPLEMENT DATA WRITTEN DOWN
003230  005767  175546                  TST     MMAVA   ;CHECK IF MM IS AVAIL
003234  001406                          BEQ     1$
003236  012737  000200  172342          MOV     #200,@#KIPAR1   ;INIT KIPAR1
003244  012737  002676  000250          MOV     #MMABT0,@#MMVEC ;SET ABORT VECTOR
003252  012737  003312  000004  1$:     MOV     #DONE4,@#ERRVEC
003260  012702  020000                  MOV     #20000,R2       ;FIRST ADDRESS
003264  010200                          MOV     R2,R0
003266  005100                          COM     R0              ;FIRST DATA (COM OF ADDRESS)
003270  062700  000002                  ADD     #2,R0
003274  162700  000002          2$:     SUB     #2,R0
003300  012203                          MOV     (R2)+,R3        ;GET VALUE
003302  020003                          CMP     R0,R3           ;CHECK
003304  001773                          BEQ     2$
003306  104400                          HLT
003310  000771                          BR      2$
003312  000240                  DONE4:  NOP
003314  000240                          NOP
```

```
                                        .TITLE  MEMORY WORST CASE NOISE TESTS



                                ;THIS TEST WRITES MEMORY WORST CASE NOISE TEST PATTERNS THROUGHOUT
                                ;MEMORY AND CHECKS THAT THEY CAN BE WRITTEN AND READ.
                                ;SET UP TRAP VECTORS
003316  012706  000500          BEGIN1: MOV     #STKPTR,SP      ;SET STACK PTR
003322  016737  175452  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT
003330  004767  177252                  JSR     PC,LDMM0        ;GO SET UP MEM MGMT
003334  012737  002676  000250          MOV     #MMABT0,@#MMVEC ;SET MEM MGMT ABORT VECTOR
003342  012737  003460  000004          MOV     #DONE5,@#ERRVEC ;SET UP TIME OUT TRAP
003350  012737  005724  000020          MOV     #STMM1,@#IOTVEC ;SET UP IOT TRAP VECTOR

                                ;LOAD CONSTANTS
003356  012700  177777                  MOV     #-1,R0
003362  005005                          CLR     R5
003364  012702  020000                  MOV     #20000,R2

                                ;WRITE WORST CASE NOISE TEST PATTERN (I XOR 8)
003370  005100                  REVPAT: COM     R0
003372  005105                          COM     R5
003374  012703  000006                  MOV     #6,R3
003400  010022                  WRT20:  MOV     R0,(2)+
003402  010522                          MOV     R5,(2)+
003404  010022                          MOV     R0,(2)+
003406  010522                          MOV     R5,(2)+
003410  010022                          MOV     R0,(2)+
003412  010522                          MOV     R5,(2)+
003414  010022                          MOV     R0,(2)+
003416  010522                          MOV     R5,(2)+
003420  010022                          MOV     R0,(2)+
003422  010522                          MOV     R5,(2)+
003424  010022                          MOV     R0,(2)+
003426  010522                          MOV     R5,(2)+
003430  010022                  WRT8:   MOV     R0,(2)+
003432  010522                          MOV     R5,(2)+
003434  010022                          MOV     R0,(2)+
003436  010522                          MOV     R5,(2)+
003440  010022                          MOV     R0,(2)+
003442  010522                          MOV     R5,(2)+
003444  010022                          MOV     R0,(2)+
003446  010522                          MOV     R5,(2)+
003450  005303                          DEC     R3
003452  003352                          BGT     WRT20
003454  001765                          BEQ     WRT8
003456  000744                          BR      REVPAT

003460  012737  003472  000004  DONE5:  MOV     #1$,@#ERRVEC
003466  005037  177572                  CLR     @#SR0           ;DISABLE MEM MGMT
```

```
003472  012706  002500          1$:     MOV     #STKPTR,SP      ;SET STACK PTR
003476  012737  004176  000004          MOV     #DONE6,@#ERRVEC ;SET TIME OUT TRAP VECTOR
003504  010701                          MOV     PC,R1           ;SET SCOPE PTR

                                ;TEST MEMORY ADDRESSES FROM 20000-37776 USING 1 XOR 8 PATTERN
003506  012702  020000          B1:     MOV     #20000,R2       ;GET STARTING ADDRESS
003512  005712                          TST     (R2)            ;CHECK IF AVAILABLE
003514  004767  002360                  JSR     PC,.1XOR8       ;GO TO COMMON SUBROUTINE TO CHECK
003520  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 40000-57776 USING 1 XOR 8 PATTERN
003522  012702  040000          B2:     MOV     #40000,R2       ;GET STARTING ADDRESS
003526  005712                          TST     (R2)            ;CHECK IF AVAILABLE
003530  004767  002344                  JSR     PC,.1XOR8       ;GO TO COMMON SUBROUTINE TO CHECK
003534  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 60000-77776 USING 1 XOR 8 PATTERN
003536  012702  060000          B3:     MOV     #60000,R2       ;GET STARTING ADDRESS
003542  005712                          TST     (R2)            ;CHECK IF AVAILABLE
003544  004767  002330                  JSR     PC,.1XOR8       ;GO TO COMMON SUBROUTINE TO CHECK
003550  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 100000-117776 USING 1 XOR 8 PATTERN
003552  012702  100000          B4:     MOV     #100000,R2      ;GET STARTING ADDRESS
003556  005712                          TST     (R2)            ;CHECK IF AVAILABLE
003560  004767  002314                  JSR     PC,.1XOR8       ;GO TO COMMON SUBROUTINE TO CHECK
003564  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 120000-137776 USING 1 XOR 8 PATTERN
003566  012702  120000          B5:     MOV     #120000,R2      ;GET STARTING ADDRESS
003572  005712                          TST     (R2)            ;CHECK IF AVAILABLE
003574  004767  002300                  JSR     PC,.1XOR8       ;GO TO COMMON SUBROUTINE TO CHECK
003600  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 140000-157776 USING 1 XOR 8 PATTERN
003602  012702  140000          B6:     MOV     #140000,R2      ;GET STARTING ADDRESS
003606  005712                          TST     (R2)            ;CHECK IF AVAILABLE
003610  004767  002264                  JSR     PC,.1XOR8       ;GO TO COMMON SUBROUTINE TO CHECK
003614  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 160000-177776 USING 1 XOR 8 PATTERN
003616  000004          B7:     IOT                             ;IOT TRAP SETS MEM MGMT
003620  001600                  1600
003622  004767  002252                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003626  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 200000-217776 USING 1 XOR 8 PATTERN
003630  000004          B10:    IOT                             ;IOT TRAP SETS MEM MGMT
003632  002000                  2000
003634  004767  002240                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003640  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 220000-237776 USING 1 XOR 8 PATTERN
003642  000004          B11:    IOT                             ;IOT TRAP SETS MEM MGMT
```

```
003644  002200                          2200
003646  004767  002226                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003652  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 240000-257776 USING 1 XOR 8 PATTERN
003654  000004                  B12:    IOT                     ;IOT TRAP SETS MEM MGMT
003656  002400                          2400
003660  004767  002214                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003664  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 260000-277776 USING 1 XOR 8 PATTERN
003666  000004                  B13:    IOT                     ;IOT TRAP SETS MEM MGMT
003670  002600                          2600
003672  004767  002202                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003676  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 300000-317776 USING 1 XOR 8 PATTERN
003700  000004                  B14:    IOT                     ;IOT TRAP SETS MEM MGMT
003702  003000                          3000
003704  004767  002170                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003710  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 320000-337776 USING 1 XOR 8 PATTERN
003712  000004                  B15:    IOT                     ;IOT TRAP SETS MEM MGMT
003714  003200                          3200
003716  004767  002156                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003722  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 340000-357776 USING 1 XOR 8 PATTERN
003724  000004                  B16:    IOT                     ;IOT TRAP SETS MEM MGMT
003726  003400                          3400
003730  004767  002144                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003734  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 360000-377776 USING 1 XOR 8 PATTERN
003736  000004                  B17:    IOT                     ;IOT TRAP SETS MEM MGMT
003740  003600                          3600
003742  004767  002132                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003746  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 400000-417776 USING 1 XOR 8 PATTERN
003750  000004                  B20:    IOT                     ;IOT TRAP SETS MEM MGMT
003752  004000                          4000
003754  004767  002120                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003760  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 420000-437776 USING 1 XOR 8 PATTERN
003762  000004                  B21:    IOT                     ;IOT TRAP SETS MEM MGMT
003764  004200                          4200
003766  004767  002106                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
003772  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 440000-457776 USING 1 XOR 8 PATTERN
003774  000004                  B22:    IOT                     ;IOT TRAP SETS MEM MGMT
```

```
003776  004400                          4400
004000  004767  002074                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004004  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 460000-477776 USING 1 XOR 8 PATTERN
004006  000004                B23:    IOT                     ;IOT TRAP SETS MEM MGMT
004010  004600                          4600
004012  004767  002062                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004016  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 500000-517776 USING 1 XOR 8 PATTERN
004020  000004                B24:    IOT                     ;IOT TRAP SETS MEM MGMT
004022  005000                          5000
004024  004767  002050                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004030  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 520000-537776 USING 1 XOR 8 PATTERN
004032  000004                B25:    IOT                     ;IOT TRAP SETS MEM MGMT
004034  005200                          5200
004036  004767  002036                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004042  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 540000-557776 USING 1 XOR 8 PATTERN
004044  000004                B26:    IOT                     ;IOT TRAP SETS MEM MGMT
004046  005400                          5400
004050  004767  002024                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004054  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 560000-577776 USING 1 XOR 8 PATTERN
004056  000004                B27:    IOT                     ;IOT TRAP SETS MEM MGMT
004060  005600                          5600
004062  004767  002012                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004066  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 600000-617776 USING 1 XOR 8 PATTERN
004070  000004                B30:    IOT                     ;IOT TRAP SETS MEM MGMT
004072  006000                          6000
004074  004767  002000                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004100  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 620000-637776 USING 1 XOR 8 PATTERN
004102  000004                B31:    IOT                     ;IOT TRAP SETS MEM MGMT
004104  006200                          6200
004106  004767  001766                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004112  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 640000-657776 USING 1 XOR 8 PATTERN
004114  000004                B32:    IOT                     ;IOT TRAP SETS MEM MGMT
004116  006400                          6400
004120  004767  001754                  JSR     PC,.1XOR8       ;GO TO COMMON CHECK ROUTINE
004124  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 660000-677776 USING 1 XOR 8 PATTERN
004126  000004                B33:    IOT                     ;IOT TRAP SETS MEM MGMT
```

```
004130  006600                          6600
004132  004767  001742                  JSR     PC,,1XOR8       ;GO TO COMMON CHECK ROUTINE
004136  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 700000-717776 USING 1 XOR 8 PATTERN
004140  000004                  B34:    IOT                     ;IOT TRAP SETS MEM MGMT
004142  007000                          7000
004144  004767  001730                  JSR     PC,,1XOR8       ;GO TO COMMON CHECK ROUTINE
004150  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 720000-737776 USING 1 XOR 8 PATTERN
004152  000004                  B35:    IOT                     ;IOT TRAP SETS MEM MGMT
004154  007200                          7200
004156  004767  001716                  JSR     PC,,1XOR8       ;GO TO COMMON CHECK ROUTINE
004162  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 740000-757776 USING 1 XOR 8 PATTERN
004164  000004                  B36:    IOT                     ;IOT TRAP SETS MEM MGMT
004166  007400                          7400
004170  004767  001704                  JSR     PC,,1XOR8       ;GO TO COMMON CHECK ROUTINE
004174  104000                          SCOPE

                                ;SET UP TO START WRITING 3 XOR 9 PATTERN
                                ;LOAD CONSTANTS
004176  012706  000500          DONE6:  MOV     #STKPTR,SP      ;SET STACK PTR
004202  012700  177777                  MOV     #-1,R0
004206  005005                          CLR     R5
004210  012702  020000                  MOV     #20000,R2
004214  012737  004304  000004          MOV     #DONE7,@#ERRVEC ;SET TIME OUT TRAP VECTOR
004222  004767  176360                  JSR     PC,LDMM0        ;GO SET UP MEM MGMT

                                ;ROUTINE TO WRITE WORST CASE NOISE PATTERN (3 XOR 9)
004226  005100                  COMDAT: COM     R0
004230  005105                          COM     R5
004232  012703  000020                  MOV     #20,R3          ;LOAD COUNT
004236                          WRT16:
004236  010022                          MOV     R0,(R2)+
004240  010022                          MOV     R0,(R2)+
004242  010022                          MOV     R0,(R2)+
004244  010022                          MOV     R0,(R2)+
004246  010522                          MOV     R5,(R2)+
004250  010522                          MOV     R5,(R2)+
004252  010522                          MOV     R5,(R2)+
004254  010522                          MOV     R5,(R2)+
004256  010022                          MOV     R0,(R2)+
004260  010022                          MOV     R0,(R2)+
004262  010022                          MOV     R0,(R2)+
004264  010022                          MOV     R0,(R2)+
004266  010522                          MOV     R5,(R2)+
004270  010522                          MOV     R5,(R2)+
004272  010522                          MOV     R5,(R2)+
004274  010522                          MOV     R5,(R2)+
004276  005303                          DEC     R3
004300  001356                          BNE     WRT16
```

```
004302  000751                          BR      COMDAT

004304  012737  004316  000004  DONE7:  MOV     #1$,@#ERRVEC
004312  005037  177572                  CLR     @#SR0           ;DISABLE MEM MGMT
004316  012706  000500          1$:     MOV     #STKPTR,SP      ;SET STACK PTR
004322  012737  005022  000004          MOV     #DONE8,@#ERRVEC         ;SET TIME OUT TRAP
004330  010701                          MOV     PC,R1           ;SET SCOPE PTR

                                ;TEST MEMORY ADDRESSES FROM 20000-37776 USING 3 XOR 9 PATTERN
004332  012702  020000          B37:    MOV     #20000,R2       ;GET STARTING ADDRESS
004336  005712                          TST     (R2)            ;CHECK AVAILABILITY
004340  004767  001672                  JSR     PC,.3XOR9       ;GO CHECK
004344  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 40000-57776 USING 3 XOR 9 PATTERN
004346  012702  040000          B40:    MOV     #40000,R2       ;GET STARTING ADDRESS
004352  005712                          TST     (R2)            ;CHECK AVAILABILITY
004354  004767  001656                  JSR     PC,.3XOR9       ;GO CHECK
004360  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 60000-77776 USING 3 XOR 9 PATTERN
004362  012702  060000          B41:    MOV     #60000,R2       ;GET STARTING ADDRESS
004366  005712                          TST     (R2)            ;CHECK AVAILABILITY
004370  004767  001642                  JSR     PC,.3XOR9       ;GO CHECK
004374  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 100000-117776 USING 3 XOR 9 PATTERN
004376  012702  100000          B42:    MOV     #100000,R2      ;GET STARTING ADDRESS
004402  005712                          TST     (R2)            ;CHECK AVAILABILITY
004404  004767  001626                  JSR     PC,.3XOR9       ;GO CHECK
004410  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 120000-137776 USING 3 XOR 9 PATTERN
004412  012702  120000          B43:    MOV     #120000,R2      ;GET STARTING ADDRESS
004416  005712                          TST     (R2)            ;CHECK AVAILABILITY
004420  004767  001612                  JSR     PC,.3XOR9       ;GO CHECK
004424  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 140000-157776 USING 3 XOR 9 PATTERN
004426  012702  140000          B44:    MOV     #140000,R2      ;GET STARTING ADDRESS
004432  005712                          TST     (R2)            ;CHECK AVAILABILITY
004434  004767  001576                  JSR     PC,.3XOR9       ;GO CHECK
004440  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 160000-177776 USING 3 XOR 9 PATTERN
004442  000004                  B45:    IOT
004444  001600                          1600
004446  004767  001564                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004452  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 200000-217776 USING 3 XOR 9 PATTERN
004454  000004                  B46:    IOT
004456  002000                          2000
004460  004767  001552                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
```

```
004464  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 220000-237776 USING 3 XOR 9 PATTERN
004466  000004                  B47:    IOT
004470  002200                          2200
004472  004767  001540                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004476  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 240000-257776 USING 3 XOR 9 PATTERN
004500  000004                  B50:    IOT
004502  002400                          2400
004504  004767  001526                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004510  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 260000-277776 USING 3 XOR 9 PATTERN
004512  000004                  B51:    IOT
004514  002600                          2600
004516  004767  001514                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004522  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 300000-317776 USING 3 XOR 9 PATTERN
004524  000004                  B52:    IOT
004526  003000                          3000
004530  004767  001502                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004534  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 320000-337776 USING 3 XOR 9 PATTERN
004536  000004                  B53:    IOT
004540  003200                          3200
004542  004767  001470                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004546  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 340000-357776 USING 3 XOR 9 PATTERN
004550  000004                  B54:    IOT
004552  003400                          3400
004554  004767  001456                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004560  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 360000-377776 USING 3 XOR 9 PATTERN
004562  000004                  B55:    IOT
004564  003600                          3600
004566  004767  001444                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004572  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 400000-417776 USING 3 XOR 9 PATTERN
004574  000004                  B56:    IOT
004576  004000                          4000
004600  004767  001432                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
004604  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 420000-437776 USING 3 XOR 9 PATTERN
004606  000004                  B57:    IOT
004610  004200                          4200
004612  004767  001420                  JSR     PC,.3XOR9       ;GO TO COMMON CHECK ROUTINE
```

```
004616  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 440000-457776 USING 3 XOR 9 PATTERN
004620  000004                  B60:    IOT
004622  004400                          4400
004624  004767  001406                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004630  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 460000-477776 USING 3 XOR 9 PATTERN
004632  000004                  B61:    IOT
004634  004600                          4600
004636  004767  001374                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004642  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 500000-517776 USING 3 XOR 9 PATTERN
004644  000004                  B62:    IOT
004646  005000                          5000
004650  004767  001362                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004654  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 520000-537776 USING 3 XOR 9 PATTERN
004656  000004                  B63:    IOT
004660  005200                          5200
004662  004767  001350                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004666  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 540000-557776 USING 3 XOR 9 PATTERN
004670  000004                  B64:    IOT
004672  005400                          5400
004674  004767  001336                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004700  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 560000-577776 USING 3 XOR 9 PATTERN
004702  000004                  B65:    IOT
004704  005600                          5600
004706  004767  001324                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004712  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 600000-617776 USING 3 XOR 9 PATTERN
004714  000004                  B66:    IOT
004716  006000                          6000
004720  004767  001312                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004724  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 620000-637776 USING 3 XOR 9 PATTERN
004726  000004                  B67:    IOT
004730  006200                          6200
004732  004767  001300                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004736  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 640000-657776 USING 3 XOR 9 PATTERN
004740  000004                  B70:    IOT
004742  006400                          6400
004744  004767  001266                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
```

```
                                        SCOPE

                                ;TEST MEMORY ADDRESSES FROM 660000-677776 USING 3 XOR 9 PATTERN
004752  000004                  B71:    IOT
004754  006600                          6600
004756  004767  001254                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004762  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 700000-717776 USING 3 XOR 9 PATTERN
004764  000004                  B72:    IOT
004766  007000                          7000
004770  004767  001242                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
004774  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 720000-737776 USING 3 XOR 9 PATTERN
004776  000004                  B73:    IOT
005000  007200                          7200
005002  004767  001230                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
005006  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 740000-757776 USING 3 XOR 9 PATTERN
005010  000004                  B74:    IOT
005012  007400                          7400
005014  004767  001216                  JSR     PC,,3XOR9       ;GO TO COMMON CHECK ROUTINE
005020  104000                          SCOPE

                                ;SET UP TO WRITE 8 XOR 13 TEST PATTERN STARTING AT ADDRESS 40000
005022  012706  000500          DONE8:  MOV     #STKPTR,SP              ;SET STACK PTR
005026  012737  005106  000004          MOV     #DONE9,@#ERRVEC         ;SET TIME OUT TRAP VECTOR
005034  012702  040000                  MOV     #40000,R2               ;FIRST ADDRESS
005040  004767  175542                  JSR     PC,LDMM0
005044  005000                  4$:     CLR     R0                      ;FIRST DATA
005046  012746  000002                  MOV     #2,-(SP)                ;PASS COUNT
005052  012705  000040          1$:     MOV     #32.,R5                 ;PATTERN REVERSE COUNT
005056  005100                          COM     R0
005060  012703  000200          2$:     MOV     #200,R3
005064  005100                          COM     R0
005066  010022                  3$:     MOV     R0,(R2)+
005070  005303                          DEC     R3
005072  001375                          BNE     3$
005074  005305                          DEC     R5
005076  001370                          BNE     2$
005100  005316                          DEC     (SP)
005102  001363                          BNE     1$
005104  000757                          BR      4$

005106  004767  000654          DONE9:  JSR     PC,LDMM2        ;GO SET UP MEM MGMT
005112  012737  006042  000250          MOV     #MMABT2,@#MMVEC ;SET ABORT VECTOR
005120  012706  000500                  MOV     #STKPTR,SP      ;SET STACK PTR
005124  012737  005370  000004          MOV     #DONE10,@#ERRVEC        ;SET TIME OUT TRAP
005132  010701                          MOV     PC,R1           ;SET SCOPE PTR

                                ;TEST MEMORY ADDRESSES FROM 40000-60000 USING 8 XOR 13 PATTERN
005134  012702  040000          B75:    MOV     #40000,R2       ;GET STARTING ADDRESS
```

```
005140  005712                          TST     (R2)            ;CHECK AVAILABILITY
005142  004767  001254                  JSR     PC,,8XOR13      ;GO TO CHECK ROUTINE
005146  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 100000-120000 USING 8 XOR 13 PATTERN
005150  012702  100000          B76:    MOV     #100000,R2      ;GET STARTING ADDRESS
005154  005712                          TST     (R2)            ;CHECK AVAILABILITY
005156  004767  001240                  JSR     PC,,8XOR13      ;GO TO CHECK ROUTINE
005162  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 140000-160000 USING 8 XOR 13 PATTERN
005164  012702  140000          B77:    MOV     #140000,R2      ;GET STARTING ADDRESS
005170  005712                          TST     (R2)            ;CHECK AVAILABILITY
005172  004767  001224                  JSR     PC,,8XOR13      ;GO TO CHECK ROUTINE
005176  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 200000-377776 USING 8 XOR 13 PATTERN
005200  000004                  B100:   IOT
005202  002000                          2000
005204  004767  001212                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005210  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 240000-437776 USING 8 XOR 13 PATTERN
005212  000004                  B101:   IOT
005214  002400                          2400
005216  004767  001200                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005222  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 300000-477776 USING 8 XOR 13 PATTERN
005224  000004                  B102:   IOT
005226  003000                          3000
005230  004767  001166                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005234  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 340000-537776 USING 8 XOR 13 PATTERN
005236  000004                  B103:   IOT
005240  003400                          3400
005242  004767  001154                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005246  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 400000-577776 USING 8 XOR 13 PATTERN
005250  000004                  B104:   IOT
005252  004000                          4000
005254  004767  001142                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005260  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 440000-637776 USING 8 XOR 13 PATTERN
005262  000004                  B105:   IOT
005264  004400                          4400
005266  004767  001130                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005272  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 500000-677776 USING 8 XOR 13 PATTERN
005274  000004                  B106:   IOT
```

```
005276  005000                          5000
005300  004767  001116                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005304  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 540000-737776 USING 8 XOR 13 PATTERN
005306  000004                  B107:   IOT
005310  005400                          5400
005312  004767  001104                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005316  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 600000-777776 USING 8 XOR 13 PATTERN
005320  000004                  B110:   IOT
005322  006000                          6000
005324  004767  001072                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005330  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 640000-1037776 USING 8 XOR 13 PATTERN
005332  000004                  B111:   IOT
005334  006400                          6400
005336  004767  001060                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005342  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 700000-1077776 USING 8 XOR 13 PATTERN
005344  000004                  B112:   IOT
005346  007000                          7000
005350  004767  001046                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005354  104000                          SCOPE

                                ;TEST MEMORY ADDRESSES FROM 740000-1137776 USING 8 XOR 13 PATTERN
005356  000004                  B113:   IOT
005360  007400                          7400
005362  004767  001034                  JSR     PC,,8XOR13      ;GO TO COMMON CHECK ROUTINE
005366  104000                          SCOPE


                                ;CHECK ADDRESSES FROM 000000-017776 USING 1 XOR 8 PATTERN
005370  005737  000042          DONE10: TST     @#42            ;CHECK IF PROGRAM LOADED VIA ACT11
005374  001402                          BEQ     13$
005376  000137  006602                  JMP     @#END           ;DO NOT RELOCATE IF ACT11
005402  012737  005414  000004  13$:    MOV     #12$,@#ERRVEC   ;SET TIME OUT TRAP
005410  005037  177572                  CLR     @#SR0           ;DISABLE MEM MGMT
005414  012706  000500          12$:    MOV     #STKPTR,SP      ;SET STACK PTR
005420  004567  000226                  JSR     5,RELOC         ;RELOCATE PROGRAM CODE
005424  000000                          000000                  ;FROM 000000 TO
005426  020000                          020000                  ;020000
005430  005704                          TST     R4              ;WAS RELOCATION SUCCESSFUL?
005432  001402                          BEQ     11$             ;BRANCH IF SUCCESSFUL
005434  000167  001142                  JMP     END             ;END OF TEST IF NOT
005440  012767  020000  000254  11$:    MOV     #20000,RELOCF   ;SET RELOCATION FACTOR
005446  000137  025452                  JMP     @#,+4+20000     ;GO EXECUTE RELOCATED CODE AT .+20000
                                ;**********************IMPORTANT NOTE**********************************
                                ;PROGRAM IS NOW EXECUTING CODE FROM PC AS SHOWN BELOW +20000.
                                ;CAUTION: DO NOT ATTEMPT TO RESTART PROGRAM AT 200
                                ;RESTART PROGRAM AT PC+20000 OF IS BELOW TO CONTINUE TESTING
```

```
                                ;TO RESTORE PROGRAM CODE TO ORIGINAL POSITION RESTART AT 8$ +20000

005452  010701                  1$:     MOV     PC,R1           ;RESTART ADDRESS TO LOOP TEST

                                ;WRITE 1 XOR 8 TEST PATTERN IN LOCATIONS 000000-017776
005454  005002                          CLR     R2              ;FIRST ADDRESS
005456  012700  177777                  MOV     #-1,R0          ;CONSTANT FOR WRITING
005462  005004                          CLR     R4              ;       "
005464  005100                  2$:     COM     R0
005466  005104                          COM     R4
005470  012703  000100                  MOV     #100,R3         ;PATTERN REVERSE COUNT
005474  010022                  3$:     MOV     R0,(R2)+        ;WRITE 1 XOR 8 TEST PATTERN
005476  010422                          MOV     R4,(R2)+
005500  005303                          DEC     R3
005502  001374                          BNE     3$
005504  022702  020000                  CMP     #20000,R2       ;FINISHED
005510  001365                          BNE     2$
005512  000240                          NOP

005514  012706  020500                  MOV     #STKPTR+20000,SP        ;SET STACK PTR

                                ;CHECK 1 XOR 8 TEST PATTERN AS WRITTEN ABOVE
005520  012767  000377  001052          MOV     #000377,COUNT   ;TEST PATTERN INDICATOR (IF MSB=0
                                                                ;THEN NORMAL PATTERN, IF MSB=1 THEN
                                                                ;COMPLEMENT PATTERN)

005526  005000                          CLR     R0              ;FIRST DATA
005530  005002                  30$:    CLR     R2              ;FIRST ADDRESS
005532  012705  020000                  MOV     #20000,R5       ;LAST ADDRESS
005536  012704  000200          4$:     MOV     #200,R4         ;PATTERN REVERSE COUNT
005542  005100                          COM     R0
005544  005100                  5$:     COM     R0
005546  011212                          MOV     (R2),(R2)       ;READ
005550  012203                          MOV     (R2)+,R3        ;SAVE
005552  020003                          CMP     R0,R3           ;CHECK
005554  001403                          BEQ     50$
005556  005046                          CLR     -(SP)           ;FAKE STATUS TO STACK FOR RTI RETURN
005560  004767  173304                  JSR     PC,ERROR        ;ERROR! R2=ADDRESS,R0=GOOD DATA,R3=BAD DATA
005564  005304                  50$:    DEC     R4
005566  001366                          BNE     5$
005570  020205                          CMP     R2,R5           ;DONE?
005572  001361                          BNE     4$
005574  006367  001000                  ASL     COUNT           ;SHIFT PATTERN INDICATOR
005600  001410                          BEQ     7$              ;BRANCH IF TEST IS DONE
005602  102352                          BVC     30$             ;DO NOT COMPLEMENT IF V IS CLEAR
005604  005002                          CLR     R2              ;COMPLEMENT 1 XOR 8 TEST PATTERN
005606  005122                  6$:     COM     (R2)+
005610  020205                          CMP     R2,R5
005612  001375                          BNE     6$
005614  012700  177777                  MOV     #-1,R0
005620  000743                          BR      30$             ;BEGIN TEST WITH COMPLEMENT TEST PAT.
005622  032737  040000  177570  7$:     BIT     #BIT14,@#SWR    ;LOOP TEST
005630  001310                          BNE     1$
005632  012706  000500          8$:     MOV     #STKPTR,SP      ;SET STACK PTR
```

```
005636  004567  000010                  JSR     R5,RELOC
005642  020000                          020000
005644  000000                          000000
005646  000137  006602                  JMP     @#END
```

```
                                        .TITLE PROGRAM SUBROUTINES
                                ;ROUTINE TO RELOCATE PROGRAM CODE
005652  012500                RELOC:  MOV     (R5)+,R0        ;GET FROM ADDRESS
005654  012502                        MOV     (R5)+,R2        ;GET TO ADDRESS
005656  010203                        MOV     R2,R3
005660  010367  000036                MOV     R3,RELOCF       ;SET RELOCATION FACTOR
005664  062703  017776                ADD     #17776,R3       ;MOVES 4K
005670  012737  005712  000004        MOV     #2$,@#ERRVEC    ;SET TIME OTU TRAP
005676  005004                        CLR     R4              ;CLEAR RELOCATION SUCCESSFUL INDICATOR
005700  005723                        TST     (R3)+           ;CHECK IF MEMORY IS AVAILABLE
005702  012022                1$:     MOV     (R0)+,(R2)+     ;RELOCATE
005704  020203                        CMP     R2,R3           ;PROGRAM CODE
005706  001375                        BNE     1$
005710  000402                        BR      3$
005712  022626                2$:     CMP     (SP)+,(SP)+     ;RESTORE STACK PTR
005714  005104                        COM     R4
005716  000240                3$:     NOP
005720  000205                        RTS     5               ;RETURN, R4=-1 IF NO RELOACATION
005722  000000                RELOCF: 0                       ;CONTAINS RELOCATION FACTOR

                                ;ROUTINE TO SET UP MEMORY MANAGEMENT FOR WORST CASE NOISE TESTS
                                ;CALLED BY IOT TRAP
005724  013637  172342        STMM1:  MOV     @(SP)+,@#KIPAR1 ;SET PAGE BASE ADDRESS
005730  013737  172342  172344        MOV     @#KIPAR1,@#KIPAR2
005736  062737  000200  172344        ADD     #200,@#KIPAR2
005744  062746  000002                ADD     #2,-(SP)        ;SET RETURN ADDRESS
005750  012702  020000                MOV     #20000,R2       ;SET ADDRESS
005754  005712                        TST     (R2)            ;CHECK IF MEMORY IS AVAILABLE
005756  012737  000001  177572        MOV     #1,@#SR0        ;ENABLE MEM MGMT
005764  000002                        RTI                     ;RETURN & START TEST

                                ;ROUTINE TO SET UP MEM MGMT FOR 8 XOR 13 PATTERN
005766  000240                LDMM2:  NOP
005770  005767  173006                TST     MMAVA           ;CHECK IF MEM MGMT IS AVAILABLE
005774  001421                        BEQ     1$
005776  012737  077406  172304        MOV     #200*256.-400+UP+RW,@#KIPDR2    ;SET KIPDR2=RW UP 200 BLOCKS
006004  012737  077406  172306        MOV     #200*256.-400+UP+RW,@#KIPDR3    ;SET KIPDR3=RW UP 200 BLOCKS
006012  005037  172310                CLR     @#KIPDR4
006016  012737  000400  172344        MOV     #400,@#KIPAR2
006024  012737  000600  172346        MOV     #600,@#KIPAR3
006032  012737  000001  177572        MOV     #1,@#SR0        ;ENABLE MEM MGMT
006040  000207                1$:     RTS     PC

                                ;ROUTINE TO SERVICE 8 XOR 13 ABORTS
006042  000240                MMABT2: NOP
006044  012702  040000                MOV     #40000,R2
006050  062737  000400  172344        ADD     #400,@#KIPAR2
006056  062737  000400  172346        ADD     #400,@#KIPAR3
006064  013716  177576                MOV     @#SR2,(SP)      ;SET RETURN TO INSTRUCTION THAT ABORTED
006070  012737  000001  177572        MOV     #1,@#SR0        ;ENABLE MEM MGMT
006076  000002                        RTI

                                ;ROUTINE TO CHECK 1 XOR 8 MEMORY WORST CASE NOISE TEST PATTERN
006100  000240                .1XOR8: NOP
```

```
006102  012700  177777            MOV     #-1,R0
006106  010205                    MOV     R2,R5
006110  062705  020000            ADD     #20000,R5       ;FORM LAST ADDRESS
006114  012746  000002            MOV     #2,-(SP)        ;LOOP COUNT
006120  012704  000100    1$:     MOV     #100,R4         ;PATTERN REVERSE COUNT

006124  005100            2$:     COM     R0              ;COMPLEMENT TEST DATA
006126  011212                    MOV     (R2),(R2)
006130  011212                    MOV     (R2),(R2)
006132  012203                    MOV     (R2)+,R3        ;SAVE MEMORY DATA
006134  020003                    CMP     R0,R3           ;CHECK
006136  001401                    BEQ     .+4
006140  104400                    HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                        ;ADDRESS = (R2)-2
006142  005100                    COM     R0              ;NEXT DATA
006144  011212                    MOV     (R2),(R2)
006146  011212                    MOV     (R2),(R2)
006150  012203                    MOV     (R2)+,R3        ;SAVE
006152  020003                    CMP     R0,R3           ;CHECK
006154  001401                    BEQ     .+4
006156  104400                    HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                        ;ADDRESS = (R2)-2
006160  005304                    DEC     R4              ;DECREMENT PATTERN COUNT
006162  001360                    BNE     2$
006164  005100                    COM     R0              ;REVERSE PATTERN
006166  020205                    CMP     R2,R5           ;LAST ADDRESS?
006170  001353                    BNE     1$              ;LOOP
006172  005316                    DEC     (SP)            ;2 PASSES?
006174  001411                    BEQ     4$
006176  162702  020000            SUB     #20000,R2       ;RESET ADDRESS TO START OF BANK
006202  005122            3$:     COM     (R2)+
006204  020205                    CMP     R2,R5           ;DONE COMPLEMENTING?
006206  001375                    BNE     3$
006210  162702  020000            SUB     #20000,R2       ;RESET ADDRESS
006214  005000                    CLR     R0
006216  000740                    BR      1$              ;AND GO THROUGH ANOTHER LOOP
006220  005726            4$:     TST     (SP)+
006222  162702  020000            SUB     #20000,R2       ;RESET TO STARTING ADDRESS OF BANK
006226  005122            5$:     COM     (R2)+           ;RESET PATTERN
006230  020205                    CMP     R2,R5           ;DONE?
006232  001375                    BNE     5$
006234  000207                    RTS     PC              ;RETURN

                            ;ROUTINE TO CHECK 3 XOR 9 WORST CASE NOISE PATTERN
006236  000240            .3XOR9: NOP
006240  012700  177777            MOV     #-1,R0
006244  010205                    MOV     R2,R5
006246  062705  020000            ADD     #20000,R5       ;FORM LAST ADDRESS
006252  012746  000002            MOV     #2,-(SP)        ;LOOP COUNT
006256  012704  000100    1$:     MOV     #100,R4         ;PATTERN REVERSE COUNT

006262  005100            2$:     COM     R0              ;COMPLEMENT TEST DATA
006264  011212                    MOV     (R2),(R2)       ;READ MEMORY DATA
006266  011212                    MOV     (R2),(R2)
```

```
006270  012203                          MOV     (R2)+,R3        ;SAVE
006272  020003                          CMP     R0,R3           ;CHECK
006274  001401                          BEQ     .+4
006276  104400                          HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                                ;ADDRESS = (R2)-2
006300  011212                          MOV     (R2),(R2)       ;READ MEMORY DATA
006302  011212                          MOV     (R2),(R2)
006304  012203                          MOV     (R2)+,R3        ;SAVE
006306  020003                          CMP     R0,R3           ;CHECK
006310  001401                          BEQ     .+4
006312  104400                          HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                                ;ADDRESS = (R2)-2
006314  011212                          MOV     (R2),(R2)       ;READ MEMORY DATA
006316  011212                          MOV     (R2),(R2)
006320  012203                          MOV     (R2)+,R3        ;SAVE
006322  020003                          CMP     R0,R3           ;CHECK
006324  001401                          BEQ     .+4
006326  104400                          HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                                ;ADDRESS = (R2)-2
006330  011212                          MOV     (R2),(R2)       ;READ MEMORY DATA
006332  011212                          MOV     (R2),(R2)
006334  012203                          MOV     (R2)+,R3        ;SAVE
006336  020003                          CMP     R0,R3           ;CHECK
006340  001401                          BEQ     .+4
006342  104400                          HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                                ;ADDRESS = (R2)-2
006344  005304                          DEC     R4              ;DECREMENT TEST PATTERN REVERSE COUNT
006346  001345                          BNE     2$
006350  005100                          COM     R0              ;REVERSE PATTERN
006352  020205                          CMP     R2,R5           ;END OF MEMORY?
006354  001340                          BNE     1$
006356  005316                          DEC     (SP)            ;2 LOOPS?
006360  001411                          BEQ     4$
006362  162702  020000                  SUB     #20000,R2       ;RESET STARTING ADDRESS TO FIRST ADDRESS
006366  005122                  3$:     COM     (R2)+           ;COMPLEMENT WORST CASE PATTERN
006370  020205                          CMP     R2,R5
006372  001375                          BNE     3$
006374  005000                          CLR     R0
006376  162702  020000                  SUB     #20000,R2
006402  000725                          BR      1$

006404  005726                  4$:     TST     (SP)+
006406  162702  020000                  SUB     #20000,R2       ;SET R2 = STARTING ADDRESS OF BANK
006412  005122                  5$:     COM     (R2)+           ;RESTORE WORST CASE PATTERN
006414  020205                          CMP     R2,R5
006416  001375                          BNE     5$
006420  000207                          RTS     PC              ;EXIT

                                ;ROUTINE TO CHECK 8 XOR 13 WORST CASE NOISE TEST PATTERN
006422  000240                  .8XOR13:        NOP
006424  012700  177777                  MOV     #-1,R0
006430  010205                          MOV     R2,R5
006432  062705  040000                  ADD     #40000,R5       ;FORM LAST ADDRESS
006436  012746  000002                  MOV     #2,-(SP)        ;LOOP COUNT
```

```
006442  012767  000041  000130  11$:    MOV     #33.,COUNT      ;PATTERN REVERSE COUNT
006450  012704  000100          1$:     MOV     #100,R4
006454  005367  000120                  DEC     COUNT           ;DECREMENT PATTERN REVERSE COUNT
006460  001004                          BNE     10$
006462  012767  000040  000110          MOV     #32.,COUNT      ;RESET COUNT
006470  005100                          COM     R0              ;COMPLEMENT PATTERN
006472  005100                  10$:    COM     R0
006474                          2$:
006474  011212                          MOV     (R2),(R2)       ;READ MEMORY DATA
006476  011212                          MOV     (R2),(R2)
006500  012203                          MOV     (R2)+,R3        ;SAVE
006502  020003                          CMP     R0,R3           ;CHECK
006504  001401                          BEQ     .+4
006506  104400                          HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                                ;ADDRESS = (R2)-2
006510  011212                          MOV     (R2),(R2)       ;READ MEMORY DATA
006512  011212                          MOV     (R2),(R2)
006514  012203                          MOV     (R2)+,R3        ;SAVE
006516  020003                          CMP     R0,R3           ;CHECK
006520  001401                          BEQ     .+4
006522  104400                          HLT                     ;ERROR! MEM DATA(R3)NOT=TEST DATA(R0)
                                                                ;ADDRESS = (R2)-2
006524  005304                          DEC     R4
006526  001362                          BNE     2$
006530  020205                          CMP     R2,R5
006532  001346                          BNE     1$
006534  005316                          DEC     (SP)
006536  001411                          BEQ     4$
006540  162702  040000                  SUB     #40000,R2       ;RESET R2 = STARTING ADDRESS
006544  005122                  3$:     COM     (R2)+           ;COMPLEMENT TEST PATTERN
006546  020205                          CMP     R2,R5
006550  001375                          BNE     3$
006552  005000                          CLR     R0
006554  162702  040000                  SUB     #40000,R2       ;RESET R2 TO STARTING ADDRESS
006560  000730                          BR      11$
006562  005726                  4$:     TST     (SP)+           ;POP LOOP COUNT OFF STACK
006564  162702  040000                  SUB     #40000,R2       ;RESET R2
006570  005122                  5$:     COM     (R2)+           ;RESTORE ORIGINAL PATTERN
006572  020205                          CMP     R2,R5
006574  001375                          BNE     5$
006576  000207                          RTS     PC              ;RETURN
006600  000000                  COUNT:  0                       ;CONTAINS PATTERN COUNT

006602  012706  000500          END:    MOV     #STKPTR,SP      ;SET STACK PTR
006606  005267  172166                  INC     ICNT
006612  022767  000200  172160          CMP     #200,ICNT       ;200 PASSES?
006620  001402                          BEQ     DONE
006622  000137  003316                  JMP     @#BEGIN1
006626  012737  000207  177566  DONE:   MOV     #207,@#TPB      ;RING BELL
006634  105737  177564                  TSTB    @#TPS           ;WAIT FOR BELL TO RING
006640  100375                          BPL     .-4
006642  013700  000042                  MOV     @#42,R0         ;GET DECTAPE MONITOR RETURN ADDRESS
006646  001404                          BEQ     FINISH
006650  004710                          JSR     PC,(R0)         ;GO TO DECTAPE MONITOR
```

```
006652  000240                  NOP
006654  000240                  NOP
006656  000240                  NOP
006660  000167  173646  FINISH: JMP     START1

                        ;ROUTINE TO RELOCATE PROGRAM CODE TO LOWEST 4K
                        ;START AT 37400 IF PROGRAM IS RELOCATED TO 20000-37776,
                        ;START AT 57400 IF PROGRAM IS RELOCATED TO 40000-57776
        017400                  .=17400
017400  010700          REL24K: MOV     PC,R0           ;FORM BASE ADDRESS WHERE CODE
017402  042700  017777          BIC     #17777,R0       ;IS RELOCATED
017406  010067  000004          MOV     R0,1$           ;PUT FROM ADDRESS INTO SUBROUTINE CALL
017412  004567  166234          JSR     R5,RELOC        ;RELOCATE CODE TO
017416  000000          1$:     0                       ;LOWEST 4K
017420  000000                  0
017422  012706  000500          MOV     #STKPTR,SP      ;SET STACK PTR
017426  005037  000176          CLR     @#176           ;PUT A HALT AT 176
017432  000137  000176          JMP     @#176           ;RETURN AND HALT AT 176

        000001                  .END
```

DZQMBA SYMBOL TABLE

```
A       = 000200        AC0     =%000000        AC1     =%000001        AC2     =%000002
AC3     =%000003        AC4     =%000004        AC5     =%000005        ADRESS    002451
AVA     = 020000        BEGIN     002542        BEGIN1    003316        BELL      001332
BIT13   = 020000        BIT14   = 040000        BIT15   = 100000        BIT6    = 000100
BIT8    = 000400        BPTVEC  = 000014        B1        003506        B10       003630
B100      005200        B101      005212        B102      005224        B103      005236
B104      005250        B105      005262        B106      005274        B107      005306
B11       003642        B110      005320        B111      005332        B112      005344
B113      005356        B12       003654        B13       003666        B14       003700
B15       003712        B16       003724        B17       003736        B2        003522
B20       003750        B21       003762        B22       003774        B23       004006
B24       004020        B25       004032        B26       004044        B27       004056
B3        003536        B30       004070        B31       004102        B32       004114
B33       004126        B34       004140        B35       004152        B36       004164
B37       004332        B4        003552        B40       004346        B41       004362
B42       004376        B43       004412        B44       004426        B45       004442
B46       004454        B47       004466        B5        003566        B50       004500
B51       004512        B52       004524        B53       004536        B54       004550
B55       004562        B56       004574        B57       004606        B6        003602
B60       004620        B61       004632        B62       004644        B63       004656
B64       004670        B65       004702        B66       004714        B67       004726
B7        003616        B70       004740        B71       004752        B72       004764
B73       004776        B74       005010        B75       005134        B76       005150
B77       005164        C       = 000001        COMDAT    004226        COUNT     006600
DIGITS    002473        DIGTAB    002434        DISPLA  = 177570        DM      = 000400
DM1     = 174000        DM2     = 170000        DONE      006626        DONE0     002770
DONE1     003056        DONE10    005370        DONE3     003226        DONE4     003312
DONE5     003460        DONE6     004176        DONE7     004304        DONE8     005022
DONE9     005106        DR0     = 000000        DR1     = 000400        DR2     = 001000
DR3     = 001400        DR4     = 002000        DR5     = 002400        DR6     = 003000
DR7     = 003400        DS      = 000020        DWN     = 000010        D0      = 000000
D1      = 004000        D2      = 010000        D2BTYP    001672        ED      = 000010
EMTVEC  = 000030        END       006602        ENMM    = 000001        ERROR     001070
ERRPC     001300        ERRVEC  = 000004        FINISH    006660        FPEVEC  = 000244
HLT     = 104400        IC      = 000200        ICNT      001000        IOTVEC  = 000020
IS      = 000000        KDE     = 000004        KDPAR0  = 172360        KDPAR1  = 172362
KDPAR2  = 172364        KDPAR3  = 172366        KDPAR4  = 172370        KDPAR5  = 172372
KDPAR6  = 172374        KDPAR7  = 172376        KDPDR0  = 172320        KDPDR1  = 172322
KDPDR2  = 172324        KDPDR3  = 172326        KDPDR4  = 172330        KDPDR5  = 172332
KDPDR6  = 172334        KDPDR7  = 172336        KIPAR0  = 172340        KIPAR1  = 172342
KIPAR2  = 172344        KIPAR3  = 172346        KIPAR4  = 172350        KIPAR5  = 172352
KIPAR6  = 172354        KIPAR7  = 172356        KIPDR0  = 172300        KIPDR1  = 172302
KIPDR2  = 172304        KIPDR3  = 172306        KIPDR4  = 172310        KIPDR5  = 172312
KIPDR6  = 172314        KIPDR7  = 172316        KM      = 000000        KPG     = 000000
KSP     =%000006        LDMM0     002606        LDMM1     003122        LDMM2     005766
LST       002444        MMABT0    002676        MMABT1    003170        MMABT2    006042
MMAVA     001002        MMTF    = 010000        MMVEC   = 000250        N       = 000010
NOMM      002602        NRA     = 100000        NR0     = 000000        NR3     = 000003
NR7     = 000007        OSTF    = 004000        O2A       001674        PARTAB    001652
PC      =%000007        PFVEC   = 000024        PIRQ    = 177772        PIRVEC  = 000240
PIR4    = 010000        PKM     = 000000        PLA     = 040000        PRTY4   = 000200
PRTY7   = 000340        PSM     = 010000        PSW     = 177776        PUM     = 030000
RD0     = 000002        RDOT    = 000001        RECDAT    001317        REG     = 004000
RELOC     005652        RELOCF    005722        REL24K    017400        RESVEC  = 000010
```

```
RETURN  = 001070        REVMAT  = 003370        RW      = 000006        RWT     = 000004
RWTM    = 000035        R0      =%000000        R1      =%000001        R10     =%000000
R11     =%000001        R12     =%000002        R13     =%000003        R14     =%000004
R15     =%000005        R2      =%000002        R3      =%000003        R4      =%000004
R5      =%000005        SAVPC   = 001274        SAVR2   = 001276        SCOPE   = 104000
SCOPEA  = 001004        SCOPEB  = 001014        SCOPEC  = 001040        SCOPED  = 001052
SCOPEE  = 001064        SDE     = 000002        SDPAR0  = 172260        SDPAR1  = 172262
SDPAR2  = 172264        SDPAR3  = 172266        SDPAR4  = 172270        SDPAR5  = 172272
SDPAR6  = 172274        SDPAR7  = 172276        SDPDR0  = 172220        SDPDR1  = 172222
SDPDR2  = 172224        SDPDR3  = 172226        SDPDR4  = 172230        SDPDR5  = 172232
SDPDR6  = 172234        SDPDR7  = 172236        SIPAR0  = 172240        SIPAR1  = 172242
SIPAR2  = 172244        SIPAR3  = 172246        SIPAR4  = 172250        SIPAR5  = 172252
SIPAR6  = 172254        SIPAR7  = 172256        SIPDR0  = 172200        SIPDR1  = 172202
SIPDR2  = 172204        SIPDR3  = 172206        SIPDR4  = 172210        SIPDR5  = 172212
SIPDR6  = 172214        SIPDR7  = 172216        SLR     = 177774        SM      = 040000
SM1     = 020370        SM2     = 000360        SM4     = 000340        SM6     = 000320
SM8     = 000300        SP      =%000006        SPG     = 000040        SR0     = 177572
SR1     = 177574        SR2     = 177576        SR3     = 172516        SSP     =%000006
START   = 002506        START1  = 002532        STKPTR  = 000500        STMMI   = 005724
SWR     = 177570        S1      = 000010        S2      = 000020        S4      = 000040
S6      = 000060        S8      = 000100        T       = 000020        TBITVE  = 000014
TE      = 001000        TKB     = 177562        TKS     = 177560        TPB     = 177566
TPS     = 177564        TPVEC   = 000064        TRAPVE  = 000034        TRTVEC  = 000014
UBREAK  = 177770        UDE     = 003301        UDPAR0  = 177660        UDPAR1  = 177662
UDPAR2  = 177664        UDPAR3  = 177666        UDPAR4  = 177670        UDPAR5  = 177672
UDPAR6  = 177674        UDPAR7  = 177676        UDPDR0  = 177620        UDPDR1  = 177622
UDPDR2  = 177624        UDPDR3  = 177626        UDPDR4  = 177630        UDPDR5  = 177632
UDPDR6  = 177634        UDPDR7  = 177636        UIPAR0  = 177640        UIPAR1  = 177642
UIPAR2  = 177644        UIPAR3  = 177646        UIPAR4  = 177650        UIPAR5  = 177652
UIPAR6  = 177654        UIPAR7  = 177656        UIPDR0  = 177600        UIPDR1  = 177602
UIPDR2  = 177604        UIPDR3  = 177606        UIPDR4  = 177610        UIPDR5  = 177612
UIPDR6  = 177614        UIPDR7  = 177616        UM      = 140000        UP      = 000000
UP0     = 000140        USP     =%000006        V       = 000002        VP0     = 000000
VP1     = 000002        VP2     = 000004        VP3     = 000006        VP4     = 000010
VP5     = 000012        VP6     = 000014        VP7     = 000016        W       = 000100
WRTUP   = 002732        WRT16   = 004236        WRT20   = 003400        WRT8    = 003430
X       = 010000        XMTDAT  = 001304        Y       = 011777        Z       = 000114
$CRLF   = 002503        $FORM0  = 001446        $LDR    = 002002        $LDRM   = 002406
$LDR1   = 002104        $PRINT  = 001404        $RESTR  = 001360        $RET    = 002524
$RLDR   = 002052        $SAVR   = 001334        .1XOR8  = 006100        .3XOR9  = 006236
.8XOR1  = 006422        .       = 017436
```

ERRORS DETECTED: 0