```
                                        .REM    *
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DCMFA-B-D
PRODUCT NAME:	COMBINED MS-11 (MOS PARITY) AND MF11-LP,MA11-P (CORE) PARITY MEMORY TESTS (SUPERSEDES DCMS-A-D)
DATE CREATED:	MAY 11, 1973
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA

1.0 ABSTRACT

THIS PROGRAM LOCATES THE PARITY MEMORY REGISTERS FOR BOTH THE CORE AND MOS PARITY MEMORIES AND PREFORMS A CHECK OF THE BITS IN EACH. IT THEN CREATES A MAP SHOWING THE MEMORY CONTROLLED BY EACH PARITY REGISTER. THE PARITY REGISTERS AND THE MEMORY ARE THEN TESTED USING THE INFORMATION IN THE MAP.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 WITH MF11-LP OR MA11-P PARITY MEMORY (CORE), MS-11 (MOS) PARITY MEMORY

2.2 STORAGE

THE PROGRAM REQUIRES 4K OF MEMORY.

3.0 LOADING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

200= NORMAL (WORST CASE) TESTING
210= ROUTINE TO RESTORE THE LOADER
220= ROUTINE TO SCAN FOR BAD PARITY
230= RESTART OF NORMAL TESTING- USES PREVIOUS MAP OF PARITY MEMORY

4.2.1 PROGRAM AND/OR OPERATOR ACTION

LOAD STARTING ADDRESS.
SET DESIRED SWITCH REGISTER SETTINGS (SEE 5.1- ALL DOWN FOR WORST CASE).
PRESS START.
IF SA 200 OR RESTART ADDRESS 230 IS USED, THE BELL WILL RING AT THE COMPLETION OF EACH PASS AND END PASS= XXX WILL BE TYPED (WHERE XXX IS THE NUMBER OF PASSES COMPLETED SINCE THE PROGRAM WAS LAST STARTED).
IF SA 210 OR SA 220 IS USED, THE PROGRAM WILL HALT WHEN DONE.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SW 15=1 OR UP -- HALT ON ERROR
SW 14=1 OR UP -- SCOPE LOOP
SW 13=1 OR UP -- INHIBIT PRINTOUT
SW 11=1 OR UP -- INHIBIT ITERATIONS
SW 10=1 OR UP -- HALT AFTER LOCATING BAD PARITY BEFORE CORRECTING IT
(USED IN PARITY SCAN ROUTINE ONLY)
SW 09=1 OR UP -- HALT AFTER THE PARITY MEMORY MAP HAS BEEN PRINTED
(ALLOWS MANUAL CHANGES TO FORCE TESTING OF MEMORY
THAT WAS NOT LOCATED)
SW 08=1 OR UP -- HALT AT END OF PASS (IF HALTED ELSEWHERE, THE
PROGRAM MAY BE RELOCATED TO BANK 1, BAD PARITY MAY
EXIST IN MEMORY, AND/OR WRITE WRONG PARITY MAY
BE SET)

5.2 SUBROUTINE ABSTRACTS

5.2.1 BEGIN SA 200, RESTART 230

5.2.2 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 64 ITERATIONS OF THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED (EXCEPT IN THOSE ROUTINES WHERE IMAX IS CHANGED). SWITCH 11 ON A ONE INHIBITS ITERATION OF SUBTESTS.

5.2.3 ERROR HANDLERS (ERRST,ERRP,ERR)

THESE ROUTINES ARE CALLED VIA EMTS TO PRINT OUT ERROR INFORMATION. (SEE 6.0 FOR DESCRIPTION OF ERROR INFORMATION)

5.2.4 PSCAN (SCAN MEMORY FOR BAD PARITY)

THIS ROUTINE READS ALL LOCATIONS IN MEMORY AND PRINTS OUT THE PHYSICAL ADDRESSES (18 BITS) OF THOSE LOCATIONS CONTAINING BAD PARITY. IT IS UTILIZED WITHIN THE PROGRAM WHILE EXERCISING MEMORY IF A PARITY ERROR OCCURS UNEXPECTEDLY, AND MAY ALSO BE CALLED USING STARTING ADDRESS 220.

5.2.5 $TYPE (ASCII MESSAGE TYPEOUT ROUTINE)

THIS IS THE STANDARD TYPEOUT ROUTINE, ALLOWING PATCHING TO UTILIZE OUTPUT DEVICES OTHER THAN THE ASR 33. $NULL CONTAINS THE VALUE TO BE USED AS A FILLER CHARACTER, AND $FILLS CONTAINS A NUMBER INDICATING THE NUMBER OF FILLER CHARACTERS REQUIRED. TPS AND TPB CONTAIN THE STATUS AND BUFFER REGISTER ADDRESSES OF THE OUTPUT DEVICE.

5.2.6 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH A HALT (000000). THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA, EXAMINE REGISTER SIX. IT WILL CONTAIN THE CURRENT STACK ADDRESS. THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VALUE OF THE LOCATION COUNTER WHEN THE TRAP OR INTERRUPT OCCURRED.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 ALTERING THE PARITY MEMORY MAP

IF THE MAP TYPED AT RUN TIME DOES NOT AGREE WITH THE HARDWARE PRESENT THE MAP CAN MANUALLY BE CHANGED TO ALLOW TESTING OF PARITY MEMORY THAT THE MAPPER DID NOT FIND. SETTING SWITCH 9 TO A 1 WILL CAUSE THE PROGRAM TO HALT AFTER THE MAP IS TYPED. AFTER THE HALT, MODIFY THE MAP AS DESIRED (SEE THE DESCRIPTION IN THE LISTING- THE MAP BEGINS AT LOCATION 600). THEN PRESS CONTINUE. THE NEW MAP WILL BE PRINTED, AND IF SW9 IS STILL SET THE PROCESS WILL BE REPEATED. IF SW9 IS NOT SET, THE PROGRAM WILL TEST PARITY MEMORY USING THE NEW MAP.

5.3.2 STOPPING THE PROGRAM

BECAUSE THE PROGRAM RELOCATES ITSELF TO BANK 1 WHILE TESTING BANK 0, A SWITCH IS PROVIDED TO HALT THE PROGRAM AT THE END OF A PASS. SETTING THIS SWITCH (SW8) WILL CAUSE THE PROGRAM TO HALT IN BANK 0 AT THE END OF THE CURRENT PASS (AFTER OUTPUTTING THE END OF PASS MESSAGE).

6.0 ERRORS

6.1 ERROR PRINTOUTS

THERE ARE THREE TYPES OF ERROR MESSAGES USING COMBINATIONS OF THE FOLLOWING ERROR TYPE ROUTINES.

PC=ZZZZZZ	PC OF FAILING ERROR CALL. REFER TO THIS ADDRESS IN THE LISTING FOR AN EXPLANATION OF THE ERROR.
ICNT=YYYYYY	CURRENT ITERATION COUNT OF FAILING TEST.
MPR=XXXXXX	ADDRESS OF PARITY REGISTER UNDER TEST.
MPR DATA=VVVVVV	CONTENTS OF PARITY REGISTER UNDER TEST.
TEST LOC=XXXXXX	MEMORY LOCATION UNDER TEST
S/B: XXXXXX	CONTENTS OF MEMORY LOCATION SHOULD BE.
WAS: XXXXXX	CONTENTS OF MEMORY LOCATION WAS.

6.2 DETERMINING ADDRESS OF TEST LOCATION WHEN KT11 IS PRESENT

IN MOST OF THE SUBTESTS, IF A KT11 IS PRESENT IT IS USED. IN ALL CASES IN THIS PROGRAM, WHEN THE KT11 IS ON, KERNEL PAGE 0 IS USED TO REFERENCE BANK 0 AND KERNEL PAGE 7 IS USED TO REFERENCE THE EXTERNAL BANK. IN MOST CASES, KERNEL PAGE 1 IS USED TO REFERENCE THE MEMORY CURRENTLY UNDER TEST. SINCE THE USE OF THE MEMORY MANAGEMENT OPTION IS SIMILAR THROUGHOUT THE PROGRAM, IT IS EASY TO DETERMINE THE ACTUAL (PHYSICAL) MEMORY ADDRESS BEING TESTED.

TO CALCULATE A PHYSICAL ADDRESS, ADD THE STARTING ADDRESS OF THE BANK BEING TESTED TO THE OFFSET WHICH GIVES THE ADDRESS WITHIN THE BANK. SINCE IN THIS PROGRAM ALL RELOCATED MEMORY TESTING IS DONE THRU KERNEL PAGE 1, KERNEL PAGE ADDRESS REGISTER 1 (ADDRESS 772342) WILL ALWAYS CONTAIN THE STARTING ADDRESS OF THE BANK. ACTUALLY, KERNEL PAGE ADDRESS REGISTER 1 (KPAR1) CONTAINS JUST THE TOP 12 BITS OF THE BANK STARTING ADDRESS. ADDING TWO ZEROES (OCTAL) TO THE RIGHT OF THIS VALUE WILL GIVE YOU THE FULL 18 BIT ADDRESS OF THE BANK. THE VIRTUAL ADDRESS USED TO REFERENCE THIS BANK UNDER TEST WILL ALWAYS START WITH 001 (BINARY, TOP 3 OF 16 BITS). THIS REFERENCES PAGE 1. THE LOWER 13 BITS GIVE THE ADDRESS WITHIN THE BANK- ADD THEM TO THE STARTING ADDRESS OF THE BANK TO GET THE FULL 18 BIT PHYSICAL ADDRESS.

FOR EXAMPLE, AN ERROR COMMENT MAY SAY "R1 CONTAINS THE ADDRESS OF THE TEST LOCATION (VIRTUAL THRU KERNEL PAGE 1 IF KT11 PRESENT)." R1 MIGHT CONTAIN 32000, AND KERNEL PAGE ADDRESS REGISTER 1 (LOCATION 772342) MIGHT CONTAIN 2400. FIRST GET THE STARTING ADDRESS OF THE BANK BY ADDING 2 ZEROES TO THE RIGHT OF THE NUMBER IN KPAR1. THUS THE VALUE 2400 INDICATES THAT THE BANK STARTS AT 240000. SECOND, CALCULATE THE OFFSET WITHIN THE BANK. THE VIRTUAL ADDRESS 32000 BREAKS DOWN INTO 1(TOP 3 BITS) WHICH REFERENCES KPAR1, AND 12000 (LOWER 13 BITS) WHICH IS THE OFFSET. ADD THE OFFSET (12000) TO THE BANK

```
                                        ADDRESS (240000) TO GET THE ACTUAL PHYSICAL ADDRESS BEING TESTED
                                        (252000).
```

6.3 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE HALT ON ERROR SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED. IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT.

6.4 ERRORS WHILE TESTING BANK ZERO (ERROR PC VALUES ABOVE 20000)

TEST20 AND TEST21 CHECK BANK 0 IF IT HAS PARITY MEMORY. TO DO THIS, THE CODE IS RELOCATED TO AND EXECUTED FROM BANK 1. THE ERROR PRINTOUTS WILL THUS GIVE THE PC IN BANK 1 OF THE ERROR CALL. SINCE ALL LOCATIONS HAVE BEEN MOVED UP 20000, SUBTRACT 20000 FROM THE ERROR PC TO GET THE ADDRESS IN THE LISTING WHICH CORRESPONDS TO THE PRINTOUT.

7.0 RESTRICTIONS

7.1 STARTING PROCEDURE

PROGRAM MUST BE LOADED INTO LOWER 4K OF MEMORY.

7.2 OPERATING RESTRICTION- AVOID USING THE "HALT" SWITCH

IF THE PROGRAM IS HALTED AT A RANDOM POINT DURING EXECUTION, SEVERAL PROBLEMS MAY ARISE. THE PROGRAM MAY BE RELOCATED TO BANK 1 AT THE TIME IT IS STOPPED, IN WHICH CASE NONE OF THE STANDARD STARTING ADDRESSES WILL WORK. WRITE WRONG PARITY MAY BE SET, IN WHICH CASE YOU MAY ENTER BAD PARITY WHILE PATCHING. AND MEMORY MAY CONTAIN BAD PARITY SINCE YOU MAY BE IN THE MIDDLE OF A TEST WHICH UTILIZES WRITE WRONG PARITY. IT IS THEREFORE STRONGLY RECOMMENDED THAT YOU HALT THE PROGRAM VIA THE "HALT AT END OF PASS" SWITCH (SW8) OR THE "HALT ON ERROR" SWITCH (SW15) RATHER THAN VIA THE HALT/ENABLE SWITCH.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

EXECUTION TIME DEPENDS ON THE AMOUNT OF PARITY MEMORY UNDER TEST.
IT TAKES ABOUT 1 MINUTE TO TEST 24K OF PARITY MEMORY (1 PASS).

8.2 STACK POINTERS

THE KERNEL STACK POINTER IS INITIALIZED TO 510.

9.0 PROGRAM DESCRIPTION

THIS PROGRAM FIRST LOCATES MA11 & MF11 CORE PARITY AND M8-11 MOS PARITY CONTROL
REGISTERS BY ADDRESSING EACH POSSIBLE REGISTER ADDRESS AND CHECKING THOSE WHICH
DO NOT TIME OUT. ON DETECTING THE PRESENCE OF A PARITY REGISTER
THE PROGRAM CHECKS IF IT IS A CORE PARITY OR A MOS PARITY REGISTER
AND ACCORDINGLY STORES THIS INFORMATION IN AN INDICATOR(INDC0-INDC15)
THE ADDRESSES OF THE REGISTERS ARE RECORDED
AND OUTPUT TO THE CONSOLE DEVICE, AND THEN THE REGISTERS ARE
CHECKED TO SEE THAT THE CORRECT BITS ARE R/W. RESET IS USED TO TEST
THE EFFECT OF INIT. PARITY MEMORY IS THEN LOCATED BY SETTING WRITE
WRONG PARITY IN ALL REGISTERS AND WRITING AND READING THE FIRST 4
ADDRESSES IN EACH 4K. EACH TIME A PARITY REGISTER RECORDS A PARITY
ERROR, THE MAP IS ALTERRED TO INDICATE THAT THAT REGISTER
CONTROLS THE MEMORY BEING ADDRESSED. THE FINAL MAP IS PRINTED AND
THEN THE PARITY CONTROL LOGIC IS CHECKED USING THE PARITY MEMORY
FOUND. SEVERAL PATTERNS ARE WRITTEN INTO EACH PARITY MEMORY
LOCATION TO SEE THAT NO PARITY ERRORS ARE CREATED. FINALLY, EACH
BYTE OF PARITY MEMORY IS WRITTEN WITH BOTH GOOD AND BAD PARITY TO
SHOW THAT THE PARITY BITS CAN BE TOGGLED AND SENSED.
SINCE THIS IS A COMBINED DIAGNOSTIC, AS FAR AS POSSIBLE
COMMON TESTS ARE USED FOR BOTH CORE AND MOS. ONLY WHERE THE
MOS CONTROLLER DEFERS FUNCTIONALLY FROM THE CORE, THE
INDICATOR IS CHECKED FOR MOS OR CORE AND THE MEMORY
IN QUESTION IS TESTED ACCORDINGLY.
A DETAILED EXPLANATION OF THE MAP IS GIVEN IN THE LISTING
(PAGE 9-12).
THE DISPLAY REGISTER CONTAINS THE NUMBER OF THE TEST
BEING EXECUTED.

*

```
                                ;MEMORY PARITY TEST
                                ;MAINDEC-11-DCMFA-B
                                ;COPYRIGHT 1972,1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                                ;AUTHOR:  JIM KAPADIA




                                ;SWITCH REGISTER SWITCH OPTIONS (SWITCH SET TO A 1)
                                ;SR15 - HALT ON ERROR
                                ;SR14 - SCOPE
                                ;SR13 - INHIBIT PRINTOUT
                                ;SR11 - INHIBIT ITERATIONS
                                ;SR10 - HALT AFTER LOCATING BAD PARITY BEFORE CORRECTING IT
                                ;SR09 - HALT AFTER TYPING PARITY MEMORY MAP (ALLOWS MANUAL
                                ;       CHANGES TO BE MADE TO THE MAP TO FORCE TESTING OF
                                ;       MEMORY THAT WAS NOT LOCATED)
                                ;SR08 - HALT AT END OF PASS (IF HALTED ELSEWHERE, THE PROGRAM
                                ;       MAY BE RELOCATED TO BANK1, WRITE WRONG PARITY MAY BE SET,
                                ;       AND/OR BAD PARITY MAY EXIST IN THE PARITY MEMORY).




                                ;SYMBOL DEFINITIONS
000001                          BIT0=1                          ;BIT DEFINITIONS
000002                          BIT1=2
000004                          BIT2=4
000010                          BIT3=10
000020                          BIT4=20
000040                          BIT5=40
000100                          BIT6=100
000200                          BIT7=200
000400                          BIT8=400
001000                          BIT9=1000
002000                          BIT10=2000
004000                          BIT11=4000
010000                          BIT12=10000
020000                          BIT13=20000
040000                          BIT14=40000
100000                          BIT15=100000
000001                          AE=1                            ;ACTION ENABLE
000004                          WWP=4                           ;WRITE WRONG PARITY
077400                          ADRS=77400                      ;ADDRESS OF ERROR
100000                          PERR=100000                     ;PARITY ERROR BIT
177570                          SR=177570                       ;PROCESSOR REGISTER DEFINITIONS
177570                          DISPLY=SR
177776                          PS=177776
000007                          PC=%7
000006                          SP=%6
177570                          DISPLAY=177570
000240                          NOP=240
000000                          OPEN=0
000510                          STKPT=510
```



```
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          R6=%6
000114                                  PARVEC=114              ;PARITY ERROR TRAP VECTOR
177572                                  SR0=177572              ;ADDRESS OF MEM MGMT REGISTER SR0




                                ;MACRO DEFINITIONS




                                ;TRAPCATCHER (.+2,HALT) LOADED INTO LOCATIONS 000-576

                                ;LOAD EMT VECTOR
000030                                  .=30
000030  015404                          EMTINT
000032  000340                          340

                                ;LOAD STARTING ADDRESS AREA
000200                                  .=200
000200  000167  001222                  JMP     START           ;GO TO START OF PROGRAM
000210                                  .=210
000210  000167  013556                  JMP     RSTLDR          ;GO RESTORE THE LOADERS
000220                                  .=220
000220  000167  001122                  JMP     SCAN            ;SCAN FOR BAD PARITY
000230                                  .=230
000230  000167  001032                  JMP     RSTART          ;RESTART WITHOUT RETYPING MAP INFORMATION

000510                                  .=510


                                ;GENERAL DATA AREA
000510  000000                TSTX:   0
000512  000000                FTITLE: 0                       ;TITLE PRINTED = 1
000514  000000                TEMPX:  0
000516  000000                ADRPT:  0                       ;MAPPING- ADDRESS POINTER
000520  000000                BITPT:  0                       ;MAPPING- BIT POINTER INDICATING BANK
000522  000000                TRFLG:  0                       ;MAPPING- TRANSISTION FLAG
000524  000000                TYFLG:  0                       ;MAPPING- TYPED FLAG
000526  000000                TYCOR:  0                       ;MAPPING- K CORE ACCUMULATOR
000530  000000                HIADR:  0                       ;USED TO CHECK WHEN DONE TESTING A BANK
000532  000000                TSTLOC: 0                       ;LOADED WITH ADDRESS OF LOCATION UNDER
                                                            ;TEST IN SOME SUBTESTS
000534  000000                SHDBE:  0                       ;VALUE EXPECTED
```

```
000536  000000          WAS:    0                       ;ACTUAL VALUE FOUND
000540  000000          TRDATA: 0
000542  000000          MPROK:  0
000544  000000          PASCNT: 0                       ;PASS COUNT
000546  000000          TBANK:  0
000550  000000          MEMUT:  0
000552  000000          NOKT:   0                       ;SET TO INDICATE NO KT11 PRESENT
000554  000000          HIWORD: 0
000556  000000          LOWFLG: 0
000560  000000          ODDFLG: 0                       ;IF SET INDICATES TESTING HIGH BYTE
                                                        ;OF MEMORY LOCATION
000562  000000          TEMP:   0
000564  000000          MTYFG:  0                       ;SET TO INDICATE MAP OF PARITY MEMORY
                                                        ;ALREADY TYPED
000566  000000          RELOC:  0


                        ;MEMORY PARITY CONTROL REGISTER ADDRESSES
                        ;THE LEAST SIGNIFICANT BIT IN THE DEVICE ADDRESS IS SET TO A ONE(1)
                        ;IF THE CONTROL IS FOUND NOT TO BE PRESENT, THE MEMORY PRESENT UNDER
                        ;CONTROL OF EACH CONTROLLER IS REPRESENTED BY 2 OCTAL WORDS, EACH BIT
                        ;REPRESENTS A 4K BLOCK, I.E. BIT0= 0-4K, BIT1= 4-8K, BIT15= 60-64K,
                        ;THE LOW BYTE OF THE LAST WORD FOR EACH REGISTER INDICATES THE OFFSET (0,2,4,OR 6)
                        ;FOR THE FIRST ADDRESS THAT ACTUALLY CORRESPONDED TO THE REGISTER. THE HIGH BYTE GETS
                        ;SET TO 1 TO INDICATE THAT A MEMORY ADDRESS HAS BEEN FOUND FOR THAT REGISTER.
                        ;FOR EXAMPLE, SAY THAT MPR0 AND MPR1 EXIST, CONTROLLING INTERLEAVED MEMORY
                        ;FROM 0 TO 16K, AND THAT MPR0 CONTROLS THE ADDRESSES ENDING IN 0 AND 4.
                        ;THE MAP WOULD THEN LOOK AS FOLLOWS:
                        ;       MPR0:   172100          ;BIT 0 IS CLEAR SINCE REGISTER IS PRESENT
                        ;               17              ;REGISTER CONTROLS 1ST 16K (=4 BANKS)
                        ;               0
                        ;               400             ;LOW BYTE SHOWS THAT FIRST ADDRESS
                        ;                               ;ENDS IN 0 (OCTAL)
                        ;                               ;HIGH BYTE CONTAINS A 1 TO INDICATE
                        ;                               ;THAT AN ADDRESS WAS FOUND
                        ;       MPR1:   172102          ;BIT 0 IS CLEAR SINCE REGISTER IS PRESENT
                        ;               17              ;REGISTER CONTROLS 1ST 16K
                        ;               0
                        ;               402             ;LOW BYTE INDICATES THAT THE FIRST
                        ;                               ;MEMORY ADDRESS ENDS IN 2 (OCTAL)
                        ;                               ;HIGH BYTE CONTAINS A 1 TO INDICATE
                        ;                               ;THAT AN ADDRESS WAS FOUND
                        ;THE REST OF THE MAP WOULD APPEAR AS IN THE LISTING

000570  172101          MPR0:   172100+1                ;PARITY STATUS REGISTERS
000572  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000574  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000576  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000600  172103          MPR1:   172102+1
000602  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000604  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000606  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000610  172105          MPR2:   172104+1
```



```
000612  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000614  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000616  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000620  172107          MPR3:   172106+1
000622  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000624  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000626  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000630  172111          MPR4:   172110+1
000632  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000634  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000636  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000640  172113          MPR5:   172112+1
000642  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000644  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000646  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000650  172115          MPR6:   172114+1
000652  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000654  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000656  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000660  172117          MPR7:   172116+1
000662  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000664  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000666  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000670  172121          MPR8:   172120+1
000672  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000674  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000676  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000700  172123          MPR9:   172122+1
000702  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000704  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000706  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000710  172125          MPR10:  172124+1
000712  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000714  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000716  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000720  172127          MPR11:  172126+1
000722  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000724  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000726  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000730  172131          MPR12:  172130+1
000732  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000734  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000736  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000740  172133          MPR13:  172132+1
000742  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000744  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000746  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000750  172135          MPR14:  172134+1
000752  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000754  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
000756  000000                  0                       ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000760  172137          MPR15:  172136+1
000762  000000                  0                       ;0-64K PARITY MEM UNDER THIS CONTROL
000764  000000                  0                       ;64-124K PARITY MEM UNDER THIS CONTROL
```

```
                                                0                               ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)

000770  000000                        TREG:   0                               ;PARITY REGISTER UNDER TEST



                                ;INDICATORS FOR CORE OR MOS PARITY REGISTER:
                                ;EACH INDICATOR REFERS TO A PARTICULAR PARITY REGISTER.  IF IT IS
                                ;A CORE PARITY REGISTER THEN A '1' IS STORED IN THE INDICATOR.
                                ;IF IT IS A MOS PARITY REGISTER THEN '-1' GETS STORED.
                                ;EX= IF MPR0 (172100) IS FOR CORE AND MPR1 (172102) IS FOR MOS
                                ;THEN THE INDICATOR MAP WILL LOOK AS FOLLOWING:
                                ;INDC0: 000001
                                ;INDC1: 177777

000772  000000                        INDC0:  0                               ;CORE=MOS PARITY INDICATOR FOR MPR
000774  000000                        INDC1:  0                               ;CORE=MOS PARITY IDCICATOR FOR MPR1
000776  000000                        INDC2:  0                               ;CORE=MOS PARITY INDICATOR FOR MPR2
001000  000000                        INDC3:  0                               ;CORE=MOS PARITY INDICATOR FOR MPR3
001002  000000                        INDC4:  0                               ;FOR MPR4
001004  000000                        INDC5:  0                               ;FOR MPR5
001006  000000                        INDC6:  0                               ;FOR MPR6
001010  000000                        INDC7:  0                               ;FOR MPR7
001012  000000                        INDC8:  0                               ;FOR MPR8
001014  000000                        INDC9:  0                               ;FOR MPR9
001016  000000                        INDC10: 0                               ;FOR MPR10
001020  000000                        INDC11: 0                               ;FOR MPR11
001022  000000                        INDC12: 0                               ;FOR MPR12
001024  000000                        INDC13: 0                               ;FOR MPR13
001026  000000                        INDC14: 0                               ;FOR MPR14
001030  000000                        INDC15: 0                               ;FOR MPR15
001032  000000                        RESRVD: 0
                                                                            ;BIT POSITIONS WHICH ARE RESERVED
                                                                            ;FOR FUTURE USE IN PARITY REGISTERS
001034  070032                        RESVC:  70032                           ;CORE PARITY
001036  077772                        RESVM:  77772                           ;MOS PARITY



                                ;PARITY PATTERNS
001040  125325                        PARPAT: 125325                          ;EVEN,ODD BYTES
001042  152652                                152652                          ;ODD,EVEN
001044  052452                                052452                          ;EVEN,ODD
001046  025125                                025125                          ;ODD,EVEN
001050  102070                                102070                          ;EVEN,EVEN
001052  072527                                072527                          ;ODD,ODD
001054  177777                                177777                          ;EVEN,EVEN
001056  107030                                107030                          ;ODD,ODD
001060  152525                                152525                          ;ODD,EVEN
001062  000000                                0                               ;EXTRA PATTERN AREA
001064  000000                                0                               ;TERMINATOR, DO NOT USE THIS LOC
```



```
                                ;THIS IS A MAP OF THE TOTAL MEMORY PRESENT IN THE SYSTEM.
001066  000000                        MEML:   0                               ;0-64K MEM PRESENT IN 4K CONTIGUOUS BLOCKS
001070  000000                        MEMH:   0                               ;64-124 MEM PRESENT IN 4K CONTIGOUS BLOCKS

                                ;THIS IS A MAP OF THE TOTAL PARITY MEMORY PRESENT IN THE SYSTEM.
001072  000000                        PMEML:  0                               ;0-64K PARITY MEMORY PRESENT
                                                                            ;(IN 4K CONTIGUOUS BLOCKS)
001074  000000                        PMEMH:  0                               ;64-124K PARITY MEMORY PRESENT
                                                                            ;(IN 4K CONTIGUOUS BLOCKS)
001076  000000                        PMEMX:  0                               ;TEMP TO HOLD CONTENTS OF EITHER
                                                                            ;LOW OR HIGH MAP



                                ;ROUTINE TO TYPE ASCII MESSAGES, MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.

                        001100          .=1100
001100  177564                  TPS:   177564                                  ;PRINTER STATUS REGISTER ADDRESS
001102  177566                  TPB:   177566                                  ;PRINTER BUFFER REGISTER ADDRESS
001104     000                  $NULL: .BYTE   0                               ;CONTAINS NULL CHARACTER FOR FILLS
001105     002                  $FILLS: .BYTE  2                               ;CONTAINS # OF FILLER CHARACTERS REQUIRED
001106     000                  $TPFLG: .BYTE  0                               ;"TERMINAL AVAILABLE" FLAG (0=YES)
001107     000                          .BYTE  0                               ;RESERVED

001110  105767  177772          $TYPE:  TSTB    $TPFLG                         ;IS THERE A TERMINAL?
001114  001401                          BEQ     6$                             ;BR IF YES
001116  000000                          HALT                                   ;HALT HERE IF NO TERMINAL
001120  010046                  6$:     MOV     R0,-(SP)                       ;SAVE R0
001122  017600  000002                  MOV     @2(SP),R0                      ;GET ADDRESS OF ASCIZ STRING
001126  112046                  1$:     MOVB    (R0)+,-(SP)                    ;PUSH CHARACTER TO BE TYPED ONTO STACK
001130  001005                          BNE     2$                             ;BR IF IT ISN'T THE TERMINATOR
001132  005726                          TST     (SP)+                          ;IF TERMINATOR POP IT OFF THE STACK
001134  012600                          MOV     (SP)+,R0                       ;RESTORE R0
001136  062716  000002          7$:     ADD     #2,(SP)                        ;ADJUST RETURN PC
001142  000002                          RTI                                    ;RETURN
001144  004767  000026          2$:     JSR     PC,5$                          ;GO TYPE THIS CHARACTER
001150  122726  000012          3$:     CMPB    #12,(SP)+                      ;CHECK IF THE CHARACTER TYPED
                                                                            ;WAS A LINE FEED
001154  001364                          BNE     1$                             ;GO GET NEXT CHARACTER IF NOT LINE FEED
001156  016746  177722                  MOV     $NULL,-(SP)                    ;GET # OF FILLER CHARACTERS NEEDED
                                                                            ;AND THE NULL CHARACTER
001162  105366  000001          4$:     DECB    1(SP)                          ;DOES A NULL NEED TO BE TYPED?
001166  002770                          BLT     3$                             ;BR IF NO--GO POP THE NULL OF THE STACK
001170  004767  000002                  JSR     PC,5$                          ;GO TYPE A NULL
001174  000772                          BR      4$                             ;LOOP
001176  105777  177676          5$:     TSTB    @TPS                           ;WAIT UNTIL PRINTER IS READY
001202  100375                          BPL     5$
001204  116677  000002  177670          MOVB    2(SP),@TPB                     ;LOAD CHARACTER TO BE
                                                                            ;TYPED INTO DATA REGISTER
001212  000207                          RTS     PC
```

```
                                        ;GENERAL DATA AREA
001214  000000                SCNFLG: 0                       ;SCNFLG GETS SET IF USING
                                                            ;SCAN ROUTINE (SA=220)
001216  000000                KTSTART:        0
001220  000000                ADRTYP: 0
001222  177600                PDRTAB: 177600
001224  172200                        172200
001226  172300                PDREND: 172300
001230  172300                KPDR0:  172300                  ;KERNEL PAGE DESCRIPTOR REGISTER ADDRESSES
001232  172302                KPDR1:  172302
001234  172304                KPDR2:  172304
001236  172316                KPDR7:  172316
001240  172340                KPAR0:  172340                  ;KERNEL PAGE ADDRESS REGISTER ADDRESSES
001242  172342                KPAR1:  172342
001244  172344                KPAR2:  172344
001246  172356                KPAR7:  172356
001250  000000                SPSAV:  0                       ;REGISTER SAVE LOCATIONS
001252  000000                R0SAV:  0
001254  000000                R1SAV:  0
001256  000000                R2SAV:  0
001260  000000                R3SAV:  0
001262  000000                R4SAV:  0
001264  000000                R5SAV:  0


                                        ;ROUTINE TO RESTART WITHOUT RETYPING MAP AFTER TEST HAS BEEN RUNNING
001266  012706  000510        RSTART: MOV     #STKPT,SP       ;SET UP STACK POINTER
001272  012767  000001  177264        MOV     #1,MTYFG        ;SET FLAG TO INDICATE MAP HAS BEEN TYPED
001300  005067  177240                CLR     PASCNT          ;INITIALIZE PASS COUNT
001304  012737  015304  000024        MOV     #PWRDN,@#24
001312  012737  000340  000026        MOV     #340,@#26
001320  005067  177164                CLR     TSTX
001324  005037  177776                CLR     @#PS            ;CLEAR PROCESSOR STATUS REGISTER
001330  012737  000006  000004        MOV     #6,@#4
001336  005037  000006                CLR     @#6
001342  000167  000416                JMP     BEGIN


                                        ;ROUTINE TO SCAN ALL MEMORY FOR BAD PARITY AND TYPE 18 BIT ADDRESSES OF BAD
                                        ;LOCATIONS
001346  012706  000510        SCAN:   MOV     #STKPT,SP       ;SETUP STACK POINTER
001352  005767  177134                TST     FTITLE          ;IF TITLE HAS BEEN PRINTED, REGISTERS
                                                            ;HAVE ALREADY BEEN LOCATED- GO
                                                            ;AND LOCATE IF NOT ALREADY DONE
001356  001006                        BNE     SCANB           ;BRANCH, REGISTERS HAVE ALREADY BEEN LOCATED
001360  005267  177630                INC     SCNFLG          ;INCREMENT SCNFLG
001364  000167  000120                JMP     START1          ;GO TO LOCATE THE REGISTERS
```



```
001370  005067  177620        SCANA:  CLR     SCNFLG          ;RETURN HERE AFTER LOCATING THE REGISTERS
001374  004767  007760        SCANB:  JSR     PC,MAPMEM       ;SETUP MEMORY MAP
001400  004767  012462                JSR     PC,PSCAN        ;SCAN FOR BAD PARITY
001404  104000                        TYPE                    ;TYPE MESSAGE "BAD PARITY SCAN COMPLETE"
001406  017055                        PSMSG
001410  005767  177136                TST     NOKT
001414  001002                        BNE     .+6
001416  005037  177572                CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
001422  000000                        HALT                    ;END OF PARITY SCAN
001424  000750                        BR      SCAN


                                        ;NORMAL STARTUP
001426  012706  000510        START:  MOV     #STKPT,SP       ;SET UP STACK POINTER
001432  005067  177126                CLR     MTYFG           ;CLEAR FLAG WHICH INDICATES MAP TYPED
001436  005067  177102                CLR     PASCNT          ;INITIALIZE PASS COUNT
001442  012737  015304  000024        MOV     #PWRDN,@#24     ;SETUP POWER FAIL RETURN
001450  012737  000340  000026        MOV     #340,@#26
001456  005067  177026                CLR     TSTX
001462  005767  177024        18:     TST     FTITLE          ;IS TITLE PRINTED YET?
001466  001010                        BNE     START1          ;YES, SKIP OVER
001470  004767  012160                JSR     PC,SAVLDR       ;COPY LOADER TO LOWER 4K
001474  005267  177012                INC     FTITLE          ;SET FLAG
001500  104000                        TYPE                    ;TYPE TITLE "MEMORY PARITY TEST
001502  016622                        MTIT                    ;MAINDEC-11-DCMFA"
001504  104000                        TYPE
001506  017110                        MLDRSV                  ;TYPE "LOADERS SAVED IN BANK 0,
                                                            ;TO RESTORE LOADERS, USE SA 210"


                                        ;SEARCH FOR PARITY REGISTERS PRESENT AND TYPE ADDRESSES OF THOSE FOUND
                                        ;FAILURE TO LOCATE A REGISTER INDICATES THAT THE ADDRESS TIMED OUT OR THAT
                                        ;BITS 5-7 IN THE REGISTER DID NOT SET
001510  104000        START1: TYPE                            ;TYPE "MEMORY PARITY REGISTERS PRESENT ARE:"
001512  016372                MMPRS
001514  005067  177022                CLR     MPROK           ;CLEAR MPR FLAG
001520  012702  000570                MOV     #MPRO,R2        ;SET UP POINTERS
001524  012703  000772                MOV     #INDCO,R3       ;POINTER TO CORE-MOS
001530  012737  001670  000004        MOV     #GMPRB,@#4      ;SET UP TIMEOUT TRAP RETURN
001536  005037  000006                CLR     @#6
001542  042712  000001        GMPRA:  BIC     #1,(2)          ;CLEAR FLAG BIT IN TABLE
001546  005062  000002                CLR     2(R2)           ;INITIALIZE LOCATIONS IN THE TABLE
001552  005062  000004                CLR     4(R2)
001556  005062  000006                CLR     6(R2)
001562  005772  000000                TST     @(2)            ;DOES THIS MPR EXIST? (IF NO, TIMES OUT)
001566  052772  000340  000000        BIS     #340,@(2)       ;YES- IS IT AN MF11-LP OR MA11-P CORE PARITY REG
001574  032772  000340  000000        BIT     #340,@(2)
001602  001414                        BEQ     18              ;NO, IS IT A MOS-11 PARITY REGISTER?  BRANCH
001604  011267  176704                MOV     (2),TEMPX       ;YES- PRINT REGISTER ADDRESS
001610  004567  013154                JSR     R5,OACNV        ;(GET ASCII)
```

```
                                TEMPX
                                MPRCOR
                                6
                                TYPE                            ;(TYPE ADDRESS)
                                MPRCOR
                                MOV     #1,(R3)                 ;SET INDICATOR FOR CORE PARITY
                                BR      2$
                        1$:     MOV     (2),TEMPX               ;IT IS A MOS REGISTER, PRINT ADDRESS
                                JSR     R5,OACNV                ;(GET ASCII)
                                TEMPX
                                MPRMOS
                                6
                                TYPE                            ;(TYPE ADDRESS)
                                MPRMOS
                                MOV     #-1,(R3)                ;SET INDICATOR FOR MOS PARITY
                        2$:     INC     MPROK                   ;SET MPR REGISTER PRESENT FLAG
                                BR      GMPRC                   ;SKIP NEXT
                        GMPRB:  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
                                BIS     #1,@R2                  ;SET FLAG INDICATING REGISTER NOT PRESENT
                        GMPRC:  ADD     #10,R2                  ;UPDATE POINTER
                                TST     (R3)+
                                CMP     R2,#TREG                ;DONE YET?
                                BLT     GMPRA                   ;NO, LOOP
                                MOV     #6,@#4                  ;YES, RESTORE TRAPCATCHER
                                TST     SCNFLG                  ;ARE YOU IN THE ROUTINE TO SCAN
                                                                ;MEMORY FOR BAD PARITY=(SCAN)
                                BEQ     GMPRD                   ;NO,BRANCH TO CARRY ON NORMALLY
                                JMP     SCANA                   ;YES, GO BACK TO THE MEMORY
                                                                ;SCAN ROUTINE
                        GMPRD:  TST     MPROK                   ;ANY PARITY REGISTERS PRESENT?
                                BNE     BEGIN                   ;YES= GO TEST CONTROLS PRESENT
                        NOREG:  TYPE                            ;NO= TYPE "NO PARITY REGISTER FOUND"
                                MTP
                                TST     @#42                    ;LOADED BY MONITOR?
                                BEQ     .+6                     ;NO, BRANCH
                                JMP     LOGICAL                 ;YES=  EXIT, NO REGISTERS PRESENT
                                HALT                            ;NO REGISTERS TO TEST
                                JMP     START                   ;IF CONTINUED, TRY AGAIN

                        BEGIN:  MOV     #TEST1+2,RETURN         ;SETUP SCOPE RETURN
                                MOV     #100,IMAX               ;MAXIMUM ITERATION COUNT
                                MOV     #6,@#4                  ;RESTORE TRAPCATCHER IN TIMEOUT VECTOR


                        ;*****************************************************************
                        ;SHOW THAT BITS 0,2,5-11, AND 15 OF EACH CORE PARITY REGISTER PRESENT
                        ;CAN BE SET AND CLEARED.  BITS 0,2,15 OF EACH MOS PARITY REGISTER
                        ;PRESENT CAN BE SET AND CLEARED
                        ;*****************************************************************
                        TEST1:  SCOPE
                                MOV     #1,@#DISPLY             ;LOAD TEST NUMBER INTO THE DISPLAY
                                MOV     #MPR0,R0                ;LOAD ADDRESS OF TABLE INTO R0
                                MOV     #INDC0,R4               ;LOAD ADDRESS OF INDICATOR IN R4
```



```
                        1$:     BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
                                BNE     4$                      ;NO= BRANCH TO GET NEXT ADDRESS
                                MOV     @R0,R1                  ;YES= LOAD R1 WITH ADDRESS OF
                                                                ;PARITY REGISTER
                                CMP     #1,(R4)                 ;IS THIS REGISTER CORE?
                                BNE     5$              ;NO
                                MOV     RESVC,RESRVD            ;YES, CORE, STORE RESERVED BITS
                                BR      .+10
                        5$:     MOV     RESVM,RESRVD            ;MOS, STORE RESERVED BITS
                                MOV     #1,R2                   ;LOAD R2 WITH VALUE OF FIRST BIT
                                                                ;TO BE TESTED
                                CLR     @R1                     ;INITIALIZE PARITY REGISTER
                                MOV     @R1,TREG                ;READ CONTENTS OF PARITY REGISTER
                                BIC     RESRVD,TREG             ;CLEAR BITS WHICH ARE RESERVED
                                BEQ     .+4                     ;CHECK OTHER BITS= BRANCH IF OK
                                ERROP                           ;CLEAR INSTRUCTION DID NOT INITIALIZE
                                                                ;ALL USED BITS IN PARITY REGISTER
                                                                ;TO ZERO (R1 CONTAINS ADDRESS OF
                                                                ;FAILING REGISTER)
                        2$:     BIT     R2,RESRVD               ;IS THIS BIT RESERVED?
                                BNE     3$                      ;YES= DON'T TEST IT SINCE IT
                                                                ;MAY BE ZERO OR ONE
                                MOV     R2,@R1                  ;NO= SET THIS BIT IN THE PARITY REGISTER
                                MOV     @R1,R3                  ;READ AND SAVE CONTENTS OF PARITY REGISTER
                                CLR     @R1                     ;CLEAR PARITY REGISTER
                                BIC     RESRVD,R3               ;CLEAR BIT LOCATIONS THAT ARE
                                                                ;RESERVED
                                CMP     R2,R3                   ;CHECK REST
                                BEQ     .+4                     ;BRANCH IF OK
                                ERROP                           ;PARITY REGISTER WHOSE ADDRESS IS IN R1
                                                                ;WAS INCORRECT AFTER THE VALUE IN R2
                                                                ;WAS WRITTEN INTO IT, ACTUAL CONTENTS
                                                                ;(WITH UNUSED BITS CLEARED) IS IN R3
                        3$:     ASL     R2                      ;ROTATE BIT TO BE TESTED
                                BCC     2$                      ;IF NOT DONE WITH ALL BIT POSITIONS
                                                                ;GO TEST THIS ONE
                        4$:     ADD     #10,R0                  ;MOVE R0 TO POINT TO NEXT POSSIBLE ADDRESS
                                TST     (R4)+
                                                                ;OF A PARITY REGISTER
                                CMP     R0,#TREG                ;AT END OF TABLE?
                                BLT     1$                      ;NO, BRANCH
                                CLR     TREG

                        ;*****************************************************************

                        ;SHOW THAT RESET CLEARS BITS 0,2, AND 15 OF EACH PARITY REGISTER
                        ;PRESENT.
                        ;*****************************************************************
                        TEST2:  SCOPE
                                MOV     #2,@#DISPLY             ;LOAD TEST NUMBER INTO THE DISPLAY
                                CLR     IMAX                    ;DON'T ITERATE TEST
                                MOV     #MPR0,R0                ;LOAD POINTER
                                MOV     #INDC0,R3               ;POINTER TO INDICATOR
                        1$:     BIT     #1,@R0                  ;IS THIS PARITY REGISTER PRESENT?
```

```
002210 001012                          BNE     5$                      ;NO BRANCH
002212 022713 000001                   CMP     #1,(R3)                 ;IS THIS CORE OR MOS PARITY REGISTER?
002216 001404                          BEQ     6$                      ;NO- BRANCH
002220 012770 100015 000000            MOV     #100015,@(R0)           ;MOS-SET ALL DEFINED BITS TO 1
002226 000403                          BR      .+10
002230 012770 107745 000000  6$:       MOV     #107745,@(R0)           ;CORE- SET ALL DEFINED BITS TO 1
002236 062700 000010         5$:       ADD     #10,R0                  ;MOVE POINTER TO POINT TO NEXT MPR ADDRESS
002242 005723                          TST     (R3)+                   ;INCREMENT POINTER TO INDICATOR
                                                                       ;OF A PARITY REGISTER
002244 020027 000770                   CMP     R0,#TREG                ;AT END OF TABLE?

002250 002755                          BLT     1$                      ;NO- CONTINUE
002252 105767 176630                   TSTB    $TPFLG                  ;YES- TERMINAL AVAILABLE?
002256 001003                          BNE     4$                      ;NO- BRANCH
002260 105777 176614                   TSTB    @TPS                    ;YES- WAIT FOR TERMINAL TO FINISH
002264 100375                          BPL     .-4
002266 000005                4$:       RESET                           ;ISSUE INIT
002270 012700 000570                   MOV     #MPR0,R0                ;LOAD ADDRESS OF THE TABLE
002274 012703 000772                   MOV     #INDC0,R3               ;POINTER TO INDICATOR
002300 032710 000001         2$:       BIT     #1,@R0                  ;IS THIS PARITY REGISTER PRESENT?
002304 001030                          BNE     3$                      ;NO- BRANCH
002306 022713 000001                   CMP     #1,(R3)                 ;IS THIS A CORE PAR REGISTER?
002312 001012                          BNE     7$                      ;NO, BRANCH
002314 017002 000000                   MOV     @(R0),R2                ;YES, GET CONTENTS OF REGISTER
002320 005070 000000                   CLR     @(R0)                   ;MAKE SURE THAT WWP AND AE ARE CLEAR
002324 042702 077772                   BIC     #77772,R2               ;MASK RESERVED BITS FOR CORE PAR
                                                                       ;-ITY REGISTER, BITS 5-11(ADDRS
                                                                       ; BITS) ARE ALSO MASKED
002330 005702                          TST     R2                      ;CHECK, IF REST WERE CLEARED
002332 001401                          BEQ     .+4
002334 104002                          ERROR                           ;CORE PARITY REGISTER WHOSE ADDRESS IS
                                                                       ;POINTED TO BY R0 WAS INCORRECT
                                                                       ;AFTER A RESET WAS ISSUED- CONTENTS
                                                                       ;SAVED IN R2 WITH UNUSED BITS MASKED
002336 000411                          BR      3$-4
002340 017002 000000         7$:       MOV     @(R0),R2                ;MOS, GET CONTENTS OF REGISTER
002344 005070 000000                   CLR     @(R0)                   ;MAKE SURE THAT WWP BAE ARE CLEAR
002350 046702 176462                   BIC     RESVM,R2                ;MASK RESERVED BITS FOR MOS PAR REG
002354 005702                          TST     R2                      ;CHECK REST
002356 001401                          BEQ     .+4                     ;RESET DID CLEAR ALL BITS
002360 104002                          ERROR                           ;MOS PARITY REGISTER WHOOSE ADDRESS IS
                                                                       ;POINTED BY R0 WAS INCORRECT,  AFTER
                                                                       ;ISSUING RESET CONTENTS OF PAR REG
                                                                       ;WERE AS SHOWN IN R2(UNUSED BITS
                                                                       ;HAVE BEEN MASKED)
002362 005070 000000                   CLR     @(R0)                   ;REINITIALIZE PARITY REGISTER
002366 062700 000010         3$:       ADD     #10,R0                  ;MOVE POINTER TO POINT TO ADDRESS
                                                                       ;OF NEXT REGISTER
002372 005723                          TST     (R3)+                   ;INCREMENT POINTER TO INDICATOR
002374 020027 000770                   CMP     R0,#TREG                ;DONE?
002400 002737                          BLT     2$                      ;NO- LOOP


                                       ;*************************************************************************
```



```
                                       ;MAP CORRESPONDENCE BETWEEN PARITY REGISTERS AND MEMORY, AND TYPE RESULTS
                                       ;NOTE THAT IF PARITY MEMORY IS NOT LOCATED CORRECTLY BY THIS SUBTEST
                                       ;IT IS DUE TO ONE OF THE FOLLOWING FAILURES:
                                       ;       -SETTING WRITE WRONG PARITY DID NOT CAUSE BAD PARITY TO BE WRITTEN
                                       ;       -PARITY GENERATE OR DETECT LOGIC FAILED
                                       ;       -PARITY ERROR BIT FAILED TO SET
                                       ;       -PARITY BITS IN MEMORY LOCATION FAILED (I.E. BIT STUCK AT GOOD PARITY VALUE)
                                       ;NOTE THAT SETTING SWITCH REGISTER SWITCH 9 WILL CAUSE A HALT AFTER THE MAP
                                       ;IS TYPED. IF YOU WISH TO CHANGE THE MAP TO ISOLATE THE CAUSE OF A MAPPING
                                       ;FAILURE, YOU CAN DO THIS ONCE THE PROCESSOR IS HALTED.  SEE THE DESCRIPTION
                                       ;IN THE LISTING (PRECEDING THE MAP TAG "MPR0" AT LOCATION 600) FOR THE MEANING
                                       ;OF THE MAP CONTENTS. AFTER MAKING THE DESIRED CHANGES, PRESS CONTINUE.  THE NEW
                                       ;MAP WILL BE TYPED AND IF SWITCH 9 IS LEFT SET THE PROCESS WILL BE REPEATED.
                                       ;IF SWITCH 9 IS NOT LEFT SET THE PROGRAM WILL PROCEED TO TEST THE PARITY MEMORY
                                       ;AND REGISTERS AS RECORDED IN THE NEW MAP.
                                       ;*************************************************************************
002402 104001                TEST3:    SCOPE
002404 012737 000003 177570            MOV     #3,@#DISPLY             ;LOAD TEST NUMBER INTO THE DISPLAY
002412 005767 176146                   TST     MTYFG                   ;IF MAPPING HAS ALREADY BEEN DONE
002416 001044                          BNE     TEST4                   ;SKIP SUBTEST
002420 004767 010740                   JSR     %7,CLRPAR
002424 004767 006730                   JSR     %7,MAPMEM               ;MAP MEMORY
002430 004767 007252                   JSR     %7,MAPREG               ;FIND PARITY MEMORY AND CORRESPONDING
                                                                       ;REGISTERS USING WRITE WRONG PARITY
                                                                       ;WITHOUT ACTION ENABLE SET
002434 005067 176432         CONT3:    CLR     PMEML                   ;INITIALIZE LOCATIONS INDICATING
002440 005067 176430                   CLR     PMEMH                   ;TOTAL PARITY MEMORY PRESENT
002444 012701 000570                   MOV     #MPR0,R1
002450 032711 000001         1$:       BIT     #1,@R1
002454 001006                          BNE     2$
002456 056167 000002 176406            BIS     2(R1),PMEML             ;FLAG EXISTING PARITY MEMORY (LOW 64K)
002464 056167 000004 176402            BIS     4(R1),PMEMH             ;FLAG EXISTING PARITY MEMORY (HIGH 64K)
002472 062701 000010         2$:       ADD     #10,R1
002476 020127 000770                   CMP     R1,#TREG
002502 103762                          BLO     1$
002504 004767 007712                   JSR     %7,TMAP                 ;TYPE MAP
002510 005267 176050                   INC     MTYFG                   ;INDICATE MAPPING DONE
002514 032737 001000 177570            BIT     #BIT9,@#SR              ;SWITCH 9 SET?
002522 001402                          BEQ     .+6                     ;NO- BRANCH
002524 000000                          HALT                            ;YES- SWITCH 9 SET INDICATING HALT
                                                                       ;AFTER TYPING PARITY MEMORY MAP
002526 000742                          BR      CONT3                   ;GO TYPE NEW MAP TO VERIFY USER'S INTENT


                                       ;*************************************************************************
                                       ;SHOW THAT ASSERT PB WORKS CORRECTLY FOR EACH REGISTER
                                       ;SHOW THAT NO TRAP OCCURS IF ACTION ENABLE (AE) IS NOT SET
                                       ;SHOW THAT SETTING AE WITH ERROR ALREADY SET DOESN'T CAUSE A TRAP
                                       ;NOTE THAT IF A KT11 IS PRESENT, IT IS USED DURING THIS SUBTEST
                                       ;*************************************************************************
002530 104001                TEST4:    SCOPE
002532 012737 000004 177570            MOV     #4,@#DISPLY             ;LOAD TEST NUMBER INTO THE DISPLAY
002540 012767 000100 013254            MOV     #100,1MAX
002546 004767 010612                   JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
```

```
 959  002552  005767  175774                TST     NOKT                    ;KT11 PRESENT?
 960  002556  001004                        BNE     1$                      ;NO=BRANCH
 961  002560  004767  010500                JSR     %7,MRALL                ;YES=MAP ALL PAGES NON RESIDENT
 962  002564  004767  010644                JSR     %7,MAP1                 ;THEN MAP KERNEL 0 TO BANK 0, KERNEL
 963                                                                        ;7 TO EXTERNAL BANK, SET KERNEL
 964                                                                        ;0,1, AND 7 RW, AND TURN ON KT11
 965  002570  012700  000570          1$:   MOV     #MPR0,R0                ;SETUP TO FIND REGISTERS PRESENT
 966  002574  012702  000772                MOV     #INDC0,R2
 967  002600  032710  000001          LOOP4: BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
 968  002604  001405                        BEQ     TST4                    ;YES=BRANCH TO TEST IT
 969  002606  062700  000010          LOP4:  ADD     #10,R0                  ;NO=CHECK FOR ANOTHER ONE
 970  002612  005722                        TST     (R2)+                   ;INCREMENT PTR9LT       CMP     R0,#TREG
 971  002614  103771                        BLO     LOOP4
 972  002616  000457                        BR      DONE4                   ;BRANCH WHEN ALL REGISTERS HAVE BEEN TESTED
 973  002620  004767  010654          TST4:  JSR     %7,LOCATM               ;LOCATE MEMORY CORRESPONDING TO
 974                                                                        ;THIS REGISTER- IF NO KT11, R1 SHOULD
 975                                                                        ;BE RETURNED CONTAINING THE ADDRESS
 976                                                                        ;OF THE 1ST LOCATION CONTROLLED BY THIS
 977                                                                        ;REGISTER (INCLUDING EXTERNAL INTERLEAVE
 978                                                                        ;OFFSET IF NEEDED)
 979                                                                        ;IF KT11 IS PRESENT, R1 SHOULD BE RETURNED
 980                                                                        ;POINTING TO THE 1ST LOCATION CONTROLLED
 981                                                                        ;BY THIS REGISTER, MAPPED THRU KERNEL
 982                                                                        ;PAGE 1, KERNEL PAGE 1 SHOULD BE
 983                                                                        ;MAPPED TO THE CORRECT BANK
 984  002624  032701  000001                BIT     #1,R1                   ;IS ERROR RETURN INDICATED?
 985  002630  001403                        BEQ     .+10                    ;NO= BRANCH
 986  002632  104002                        ERROR                           ;MAP INDICATES NO PARITY MEMORY
 987                                                                        ;IS CONTROLLED BY THIS REGISTER
 988                                                                        ;R0 POINTS TO THE ADDRESS OF THE
 989                                                                        ;PARITY REGISTER
 990  002634  000167  177746                JMP     LOP4
 991  002640  012737  002726  000114        MOV     #TRP4A,@#114            ;SETUP PARITY TRAP RETURN
 992  002646  012770  000004  000000        MOV     #WWP,@(R0)              ;SET WRITE WRONG PARITY
 993  002654  011111                        MOV     @R1,@R1                 ;WRITE CONTENTS OF LOCATION WITH
 994                                                                        ;WRONG PARITY
 995  002656  005070  000000                CLR     @(R0)                   ;CLEAR PARITY REGISTER
 996  002662  005711                        TST     @R1                     ;READ BAD PARITY WITH ACTION ENABLE
 997                                                                        ;CLEARED- NO TRAP EXPECTED
 998  002664  012737  002734  000114        MOV     #TRP4B,@#PARVEC         ;CHANGE PARITY TRAP RETURN
 999  002672  052770  000001  000000        BIS     #AE,@(R0)               ;SET ACTION ENABLE WITH PARITY ERROR
1000                                                                        ;ALREADY SET- SHOULDN'T TRAP YET
1001  002700  012737  002742  000114        MOV     #TRP4C,@#PARVEC         ;CHANGE PARITY TRAP RETURN
1002  002706  005711                        TST     @R1                     ;READ LOCATION AGAIN- SHOULD GET
1003                                                                        ;A PARITY TRAP DUE TO READING BAD
1004                                                                        ;PARITY WITH ACTION ENABLE SET
1005  002710  104002                        ERROR                           ;NO PARITY TRAP AFTER READING LOCATION
1006                                                                        ;WHICH SHOULD CONTAIN BAD PARITY-
1007                                                                        ;R1 CONTAINS ADDRESS OF MEMORY LOCATION
1008                                                                        ;(VIRTUAL, THRU KERNEL PAGE 1, IF KT11
1009                                                                        ;PRESENT). R0 POINTS TO THE ADDRESS OF
1010                                                                        ;THE PARITY REGISTER IN WHICH AE WAS SET
1011  002712  005070  000000          CONT4: CLR     @(R0)                   ;CLEAR PARITY REGISTER
1012  002716  005511                        ADC     @R1                     ;CLEAR BAD PARITY
```



```
1013  002720  005070  000000                CLR     @(R0)                   ;CLEAR PARITY ERROR BIT
1014  002724  000730                        BR      LOP4                    ;GO TO TEST NEXT REGISTER
1015  002726  104002          TRP4A: ERROR                                  ;PARITY TRAP OCCURRED WITH ACTION
1016                                                                        ;ENABLE CLEAR- R0 POINTS TO THE ADDRESS
1017                                                                        ;OF THE PARITY REGISTER UNDER TEST
1018                                                                        ;R1 CONTAINS THE ADDRESS OF THE MEMORY
1019                                                                        ;UNDER TEST (VIRTUAL IF KT11 IS PRESENT)
1020  002730  022626                        CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1021  002732  000767                        BR      CONT4
1022  002734  104002          TRP4B: ERROR                                  ;PARITY TRAP OCCURRED WHEN ACTION
1023                                                                        ;ENABLE WAS SET WITH PARITY ERROR
1024                                                                        ;ALREADY SET
1025                                                                        ;R0 POINTS TO THE ADDRESS OF THE
1026                                                                        ;PARITY REGISTER UNDER TEST
1027                                                                        ;R1 CONTAINS THE MEMORY ADDRESS UNDER
1028                                                                        ;TEST (VIRTUAL IF KT11 IS PRESENT)
1029  002736  022626                        CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1030  002740  000764                        BR      CONT4
1031  002742  005770  000000          TRP4C: TST     @(R0)                   ;ERROR BIT SET AFTER PARITY TRAP?
1032  002746  100401                        BMI     .+4                     ;YES= BRANCH
1033  002750  104002                        ERROR                           ;ERROR BIT NOT SET AFTER PARITY
1034                                                                        ;TRAP= R0 POINTS TO THE ADDRESS
1035                                                                        ;OF THE PARITY REGISTER UNDER TEST
1036  002752  022626                        CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1037  002754  000756                        BR      CONT4
1038  002756  012737  000116  000114  DONE4: MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
1039  002764  005767  175562                TST     NOKT
1040  002770  001002                        BNE     .+6
1041  002772  005037  177572                CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1042
1043
1044                                        ;*************************************************************************
1045                                        ;SHOW THAT READING GOOD PARITY AFTER BAD PARITY DOESN'T CLEAR PARITY ERROR BIT
1046                                        ;*************************************************************************
1047  002776  104001          TEST5: SCOPE
1048  003000  012737  000005  177570        MOV     #5,@#DISPLY             ;LOAD TEST NUMBER INTO THE DISPLAY
1049  003006  004767  010352                JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1050  003012  005767  175534                TST     NOKT                    ;KT11 PRESENT?
1051  003016  001004                        BNE     1$                      ;NO, BRANCH
1052  003020  004767  010240                JSR     PC,MRALL                ;YES, MAP ALL PAGES NON-RESIDENT
1053  003024  004767  010404                JSR     PC,MAP1                 ;THEN MAP KERNEL 0 TO BANK 0, MAP KERNEL
1054                                                                        ;7 TO THE EXTERNAL BANK, SET KERNEL
1055                                                                        ;0,1, AND 7 RW, AND TURN ON KT11
1056  003030  012700  000570          1$:   MOV     #MPR0,R0                ;SETUP TO FIND REGISTERS PRESENT
1057  003034  032710  000001          LOOP5: BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
1058  003040  001406                        BEQ     TST5                    ;YES= BRANCH TO TEST IT
1059  003042  062700  000010          LOP5:  ADD     #10,R0                  ;NO= CHECK FOR ANOTHER ONE
1060  003046  020027  000770                CMP     R0,#TREG
1061  003052  103770                        BLO     LOOP5
1062  003054  000431                        BR      DONE5                   ;EXIT WHEN ALL REGISTERS HAVE BEEN TESTED
1063  003056  004767  010416          TST5:  JSR     PC,LOCATM               ;LOCATE MEMORY CORRESPONDING TO  THIS
1064                                                                        ;REGISTER= R1 WILL BE RETURNED CONTAINING
1065                                                                        ;THE ADDRESS OF THE FIRST LOCATION
1066                                                                        ;CONTROLLED BY THIS REGISTER (MAPPED
```

```
                                                                ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
003062  032701  000001          BIT     #1,R1                 ;IS ERROR RETURN INDICATED?
003066  001403                  BEQ     .+10                  ;NO- BRANCH
003070  104002                  ERROR                         ;MAP INDICATES NO PARITY MEMORY IS
                                                              ;CONTROLLED BY THIS REGISTER, R0
                                                              ;POINTS TO THE ADDRESS OF THE
                                                              ;PARITY REGISTER
003072  000167  177744          JMP     LOP5
003076  012770  000004  000000  MOV     #WWP,@(R0)            ;SET WWP IN THIS REGISTER
003104  011111                  MOV     @R1,@R1               ;WRITE CONTENTS OF LOCATION WITH
                                                              ;WRONG PARITY
003106  005711                  TST     @R1                   ;DETECT WRONG PARITY
003110  042770  000004  000000  BIC     #WWP,@(R0)            ;CLEAR WWP IN PARITY REGISTER
003116  011111                  MOV     @R1,@R1               ;RESTORE GOOD PARITY
003120  005711                  TST     @R1                   ;REREAD LOCATION
003122  005770  000000          TST     @(R0)                 ;READ CONTENTS OF PARITY REGISTER
003126  100401                  BMI     .+4                   ;BRANCH IF PARITY ERROR IS STILL SET
003130  104002                  ERROR                         ;PARITY ERROR BIT CLEARED BY READING
                                                              ;GOOD PARITY (OR POSSIBLY WHILE DOING
                                                              ;THE BIC TO CLEAR WWP), R0 POINTS TO
                                                              ;THE PARITY REGISTER ADDRESS.
003132  005070  000000          CLR     @(R0)                 ;CLEAR THE PARITY REGISTER
003136  000741                  BR      LOP5
003140  005767  175406  DONE5:  TST     NOKT
003144  001002                  BNE     .+6
003146  005037  177572          CLR     @#SR0                 ;TURN OFF KT11 IF PRESENT


                        ;*********************************************************************
                        ;SHOW THAT PARITY GENERATE AND DETECT LOGIC WORKS CORRECTLY FOR EACH BYTE
                        ;SHOW THAT WRITE WRONG PARITY WORKS FOR HIGH AND LOW BYTES
                        ;SHOW THAT WRITING INTO LOCATION WHEN WRITE WRONG PARITY IS NOT SET
                        ;RESTORES GOOD PARITY
                        ;*********************************************************************
003152  104001  TEST6:  SCOPE
003154  012737  000006  177570  MOV     #6,@#DISPLY           ;LOAD TEST NUMBER INTO THE DISPLAY
003162  004767  010176          JSR     %7,CLRPAR             ;CLEAR ALL PARITY REGISTERS
003166  005767  175360          TST     NOKT                  ;KT11 PRESENT?
003172  001004                  BNE     1$                    ;NO, BRANCH
003174  004767  010064          JSR     %7,NRALL              ;YES- MAP IT (KERNEL 0 TO BANK 0, RW;
003200  004767  010230          JSR     %7,MAP1               ;KERNEL 7 TO EXTERNAL BANK, RW; KERNEL 1 RW)
                                                              ;AND TURN IT ON
003204  012700  000570  1$:     MOV     #MPR0,R0              ;SETUP TO FIND REGISTERS PRESENT
003210  032710  000001  LOOP6:  BIT     #1,@R0                ;IS THIS REGISTER PRESENT?
003214  001406                  BEQ     TST6                  ;YES- BRANCH TO TEST IT
003216  062700  000010  LOP6:   ADD     #10,R0                ;NO- CHECK FOR ANOTHER ONE
003222  020027  000770          CMP     R0,#TREG
003226  103770                  BLO     LOOP6
003230  000523                  BR      DONE6                 ;BRANCH TO DONE IF ALL REGISTERS
                                                              ;HAVE BEEN TESTED
003232  004767  010242  TST6:   JSR     %7,LOCATM             ;LOCATE MEMORY CORRESPONDING TO
                                                              ;THIS REGISTER- R1 SHOULD BE RETURNED
                                                              ;CONTAINING THE ADDRESS OF THE FIRST
                                                              ;LOCATION CONTROLLED BY THIS REGISTER
```



```
                                                              ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 PRESENT)
003236  032701  000001          BIT     #1,R1                 ;IF NO MEMORY WAS FOUND TO CORRESPOND
003242  001403                  BEQ     .+10                  ;ODD ADDRESS IS RETURNED-BRANCH IF OK
003244  104002                  ERROR                         ;MAP INDICATES NO PARITY MEMORY IS
                                                              ;CONTROLLED BY THIS REGISTER
                                                              ;R0 POINTS TO THE ADDRESS OF THE
                                                              ;PARITY REGISTER
003246  000167  177744          JMP     LOP6                  ;AFTER ERROR, CHECK FOR NEXT REGISTER


                        ;FIRST SHOW THAT IF THE PARITY REGISTER IS CLEARED INITIALLY,
                        ;PARITY ERROR DOESN'T SET
003252  005011                  CLR     @R1                   ;INITIALIZE LOCATION UNDER TEST
003254  005070  000000          CLR     @(R0)                 ;INITIALLY CLEAR PARITY REGISTER
003260  005002                  CLR     R2                    ;R2 CONTAINS VALUE TO BE LOADED
                                                              ;INTO MEMORY
003262  110211          1$:     MOVB    R2,@R1                ;WRITE VALUE INTO LOW BYTE
003264  005711                  TST     @R1                   ;READ WORD TO CHECK PARITY
003266  005770  000000          TST     @(R0)                 ;CHECK PARITY REGISTER
003272  100006                  BPL     5$                    ;BRANCH IF ERROR NOT SET
003274  104002                  ERROR                         ;PARITY ERROR SET WHEN VALUE IN R2 WAS
                                                              ;WRITTEN AND READ BACK FROM LOW BYTE OF
                                                              ;LOCATION WHOSE ADDRESS IS CONTAINED IN
                                                              ;R1 (WWP WAS NOT SET)
003276  005070  000000          CLR     @(R0)                 ;CLEAR ERROR BIT
003302  005011                  CLR     @R1                   ;REINITIALIZE TEST LOCATION
003304  005002                  CLR     R2                    ;REINITIALIZE VALUE TO BE USED
003306  000402                  BR      2$
003310  105202          5$:     INCB    R2                    ;INCREMENT VALUE TO BE LOADED
003312  001363                  BNE     1$                    ;LOOP UNTIL ALL VALUES HAVE BEEN USED
003314  110261  000001  2$:     MOVB    R2,1(R1)              ;WRITE ALL VALUES INTO HIGH BYTE
003320  005711                  TST     @R1                   ;READ WORD TO CHECK PARITY
003322  005770  000000          TST     @(R0)                 ;CHECK PARITY REGISTER
003326  100002                  BPL     .+6                   ;BRANCH IF ERROR NOT SET
003330  104002                  ERROR                         ;PARITY ERROR SET WHEN VALUE IN R2
                                                              ;WAS WRITTEN AND READ BACK FROM HIGH BYTE
                                                              ;OF LOCATION WHOSE ADDRESS IS CONTAINED
003332  000402                  BR      6$                    ;IN R1 (WWP WAS NOT SET)
003334  105202                  INCB    R2                    ;INCREMENT VALUE TO BE LOADED
003336  001366                  BNE     2$                    ;LOOP UNTIL ALL VALUES HAVE BEEN USED


                        ;TEST PARITY GENERATE AND DETECT LOGIC BY SETTING WRITE WRONG PARITY AND
                        ;WRITING EACH POSSIBLE VALUE TO THE LOW BYTE, THEN TO THE HIGH BYTE
003340  005011  6$:     CLR     @R1                   ;INITIALIZE LOCATION UNDER TEST
003342  005002          CLR     R2                    ;INITIALIZE VALUE TO BE WRITTEN
003344  012770  000004  000000  3$:     MOV     #WWP,@(R0)            ;SET WRITE WRONG PARITY
003352  110211          MOVB    R2,@R1                ;WRITE WRONG PARITY IN LOW BYTE
003354  005070  000000          CLR     @(R0)                 ;CLEAR WRITE WRONG PARITY, AND CLEAR
                                                              ;PARITY ERROR IF SET
003360  005711          TST     @R1                   ;READ BACK WRONG PARITY
003362  005770  000000          TST     @(R0)                 ;PARITY ERROR SET?
003366  100402                  BMI     .+6                   ;YES=BRANCH
003370  104002                  ERROR                         ;PARITY ERROR DID NOT SET WHEN THE
                                                              ;LOCATION UNDER TEST WAS WRITTEN
                                                              ;AND READ BACK WITH WRITE WRONG PARITY
```

```
                                                                        ;SET, R0 POINTS TO ADDRESS OF PARITY
                                                                        ;REGISTER, R1 CONTAINS ADDRESS OF LOCATION
                                                                        ;BEING TESTED (VIRTUAL, THRU KERNEL
                                                                        ;PAGE 1 IF KT11 IS PRESENT), R2
                                                                        ;CONTAINS THE VALUE WRITTEN
003372  000402                        BR      .+6                     ;EXIT LOOP AFTER ERROR
003374  105202                        INCB    R2                      ;INCREMENT DATA
003376  001362                        BNE     3$                      ;LOOP TILL DONE WITH ALL VALUES
003400  005011                        CLR     @R1                     ;REINITIALIZE LOCATION TO CLEAR BAD PARITY
003402  005070  000000                CLR     @(R0)                   ;CLEAR ERROR IF SET
003406  005711                        TST     @R1                     ;READ LOCATION, WHICH SHOULD NOW HAVE
                                                                        ;GOOD PARITY
003410  005770  000000                TST     @(R0)                   ;PARITY ERROR SET?
003414  100001                        BPL     .+4                     ;NO, BRANCH
003416  104002                        ERROR                           ;GOOD PARITY WAS NOT RESTORED BY
                                                                        ;WRITING INTO THE LOCATION WITH
                                                                        ;WRITE WRONG PARITY CLEARED
003420  012770  000004  000000  4$:   MOV     #4WP,@(R0)              ;SET WRITE WRONG PARITY
003426  110261  000001                MOVB    R2,1(R1)                ;WRITE WRONG PARITY IN HIGH BYTE
003432  005070  000000                CLR     @(R0)                   ;CLEAR WRITE WRONG PARITY AND PARITY
                                                                        ;ERROR IF SET
003436  005711                        TST     @R1                     ;READ BACK WRONG PARITY
003440  005770  000000                TST     @(R0)                   ;PARITY ERROR SET?
003444  100402                        BMI     .+6                     ;YES=BRANCH
003446  104002                        ERROR                           ;PARITY ERROR DID NOT SET WHEN THE LOCATION
                                                                        ;UNDER TEST WAS WRITTEN AND READ BACK WITH
                                                                        ;WRITE WRONG PARITY SET, R0 POINTS
                                                                        ;TO THE ADDRESS OF THE PARITY REGISTER,
                                                                        ;THE VALUE IN R2 WAS WRITTEN INTO THE HIGH
                                                                        ;BYTE OF THE LOCATION WHOSE ADDRESS IS IN R1
                                                                        ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 PRESENT)
                                                                        ;THEN THE PARITY REGISTER WAS
                                                                        ;CLEARED AND THE WORD WAS READ BACK
003450  000402                        BR      .+6                     ;EXIT LOOP AFTER ERROR
003452  105202                        INCB    R2                      ;INCREMENT DATA
003454  001361                        BNE     4$                      ;LOOP TILL DONE WITH ALL VALUES
003456  005011                        CLR     @R1                     ;CLEAR BAD PARITY IN TEST LOCATION
003460  005070  000000                CLR     @(R0)                   ;CLEAR PARITY ERROR BIT IF SET
003464  005711                        TST     @R1                     ;READ LOCATION, WHICH SHOULD NOW
                                                                        ;HAVE GOOD PARITY
003466  005770  000000                TST     @(R0)                   ;CHECK PARITY ERROR BIT
003472  100001                        BPL     .+4                     ;BRANCH IF CLEAR
003474  104002                        ERROR                           ;WRITING INTO LOCATION WHEN WRITE
                                                                        ;WRONG PARITY WAS NOT SET DID NOT
                                                                        ;WRITE GOOD PARITY
003476  000647                        BR      LUP6                    ;GO CHECK FOR ANOTHER PARITY REGISTER
003500  005767  175046        DONE6:  TST     NOKT
003504  001002                        BNE     .+6
003506  005037  177572                CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT


                                ;*************************************************************************
                                ;SHOW THAT SETTING PARITY ERROR AFTER SETTING ACTION ENABLE WON'T CAUSE A TRAP
```



```
                                ;*************************************************************************
003512  104001                TEST7:  SCOPE
003514  012737  000007  177570        MOV     #7,@#DISPLY             ;LOAD TEST NUMBER INTO THE DISPLAY
003522  004767  007636                JSR     %7,CLRPAR               ;INITIALLY CLEAR ALL PARITY REGISTERS
003526  012737  003622  000114        MOV     #TRP7,@#PARVEC          ;SETUP PARITY TRAP RETURN
003534  005037  000116                CLR     @#PARVEC+2
003540  012700  000570                MOV     #MPR0,R0                ;SETUP TO GET ADDRESS OF REGISTER PRESENT
003544  032710  000001        LUP7:   BIT     #1,@R0
003550  001406                        BEQ     TST7                    ;BRANCH TO TEST REGISTER
003552  062700  000010        LOOP7:  ADD     #10,R0
003556  020027  000770                CMP     R0,#TREG
003562  103770                        BLO     LUP7
003564  000412                        BR      DONE7                   ;BRANCH IF DONE TESTING ALL REGISTERS
003566  012770  000001  000000  TST7:  MOV     #AE,@(R0)               ;SET ACTION ENABLE
003574  052770  100000  000000        BIS     #PERR,@(R0)             ;SET PARITY ERROR
003602  000240                        NOP                             ;SHOULD NOT TRAP
003604  005070  000000                CLR     @(R0)                   ;CLEAR PARITY REGISTER
003610  000760                        BR      LOOP7                   ;GO CHECK NEXT REGISTER
003612  012737  000116  000114  DONE7:  MOV     #PARVEC+2,@#PARVEC
003620  000405                        BR      TEST10
003622  104002                TRP7:   ERROR                           ;TRAP OCCURRED WHEN PARITY ERROR BIT
                                                                        ;WAS SET VIA A BIS INSTRUCTION
                                                                        ;WITH ACTION ENABLE ALREADY SET.
                                                                        ;R0 POINTS TO THE ADDRESS OF THE
                                                                        ;PARITY REGISTER
003624  022626                        CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
003626  005070  000000                CLR     @(R0)                   ;CLEAR PARITY REGISTER
003632  000747                        BR      LOOP7


                                ;*************************************************************************
                                ;SHOW THAT REPEATED PARITY ERRORS WILL CAUSE REPEATED TRAPS IF ACTION
                                ;ENABLE IS SET AND PARITY ERROR IS LEFT SET.
                                ;SHOW THAT THE ERROR ADDRESS BITS (11-5) TRACK (ONLY FOR CORE PARITY REGISTERS)
                                ;*************************************************************************
003634  104001                TEST10: SCOPE
003636  012737  000010  177570        MOV     #10,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
003644  004767  007514                JSR     %7,CLRPAR               ;INITIALLY CLEAR ALL PARITY REGISTERS
003650  005767  174676                TST     NOKT                    ;KT11 PRESENT?
003654  001004                        BNE     1$                      ;NO= BRANCH
003656  004767  007402                JSR     %7,NRALL                ;YES= INITIALLY MAP ALL PAGES NR
003662  004767  007546                JSR     %7,MAP1                 ;MAP KERNEL 0 TO BANK 0,RW
                                                                        ;KERNEL 7 TO THE EXTERNAL BANK, RW
                                                                        ;MAKE KERNEL PAGE 1 RW AND TURN ON THE KT11
003666  012700  000570          1$:   MOV     #MPR0,R0
003672  012702  000772                MOV     #INDC0,R2
003676  032710  000001        LUP10:  BIT     #1,@R0
003702  001407                        BEQ     TST10                   ;BRANCH TO TEST REGISTER IF PRESENT
003704  062700  000010        LOOP10: ADD     #10,R0
003710  005722                        TST     (R2)+
003712  020027  000770                CMP     R0,#TREG
003716  103767                        BLO     LUP10
003720  000507                        BR      DONE10                  ;BRANCH IF ALL REGISTERS HAVE BEEN TESTED
```

```
1283  003722  004767  007552          TST10:  JSR     %7,LOCATM
1284  003726  032701  000001                  BIT     #1,R1                   ;ERROR RETURN INDICATED?
1285  003732  001403                          BEQ     .+10                    ;BRANCH IF NO
1286  003734  104002                          ERROR                           ;MAP INDICATES THERE IS NO MEMORY
1287                                                                          ;CORRESPONDING TO THIS REGISTER
1288                                                                          ;R0 POINTS TO THE ADDRESS OF
1289                                                                          ;THE PARITY REGISTER
1290  003736  000167  177742                  JMP     LOOP10
1291  003742  012770  000004  000000          MOV     #WWP,@(R0)              ;SET WRITE WRONG PARITY
1292  003750  011111                          MOV     @R1,@R1                 ;WRITE WRONG PARITY IN FIRST LOCATION
1293  003752  016161  004000  004000          MOV     4000(R1),4000(R1)       ;WRITE WRONG PARITY IN SECOND LOCATION
1294  003760  012770  000001  000000          MOV     #AE,@(R0)               ;SET ACTION ENABLE AND CLEAR REST
1295  003766  012737  004020  000114          MOV     #TRP10,@#PARVEC         ;SETUP PARITY TRAP RETURN
1296  003774  012767  000010  000152          MOV     #10,COUNT               ;SETUP COUNTER TO EXECUTE INSTRUCTION 1
1297                                                                          ;(INST1) TEN TIMES
1298  004002  005711                  INST1:  TST     @R1                     ;READ WRONG PARITY WITH AE SET- SHOULD
```



```
1299                                                                          ;TRAP TO TRP10
1300  004004  104002                          ERROR                           ;NO PARITY TRAP OCCURRED. R0 POINTS TO
1301                                                                          ;ADDRESS OF THE PARITY REGISTER BEING
1302                                                                          ;TESTED.
1303  004006  000441                          BR      CONT10
1304  004010  005761  004000          INST2:  TST     4000(R1)                ;READ WRONG PARITY FROM SECOND ADDRESS
1305                                                                          ;WITH AE SET- SHOULD TRAP TO TRP10A
1306  004014  104002                          ERROR                           ;NO PARITY TRAP OCCURRED. R0 POINTS TO
1307                                                                          ;THE ADDRESS OF THE PARITY REGISTER
1308                                                                          ;BEING TESTED
1309  004016  000435                          BR      CONT10
1310  004020  005367  000130          TRP10:  DEC     COUNT                   ;HAS PARITY TRAP OCCURRED TEN TIMES?
1311  004024  001413                          BEQ     1$                      ;YES- BRANCH
1312  004026  022712  000001                  CMP     #1,(R2)                 ;IS THIS A CORE PAR REG?
1313  004032  001005                          BNE     2$                      ;NO, BRANCH (NO ERROR
1314                                                                          ;ADDRESS BITS FOR MOS PAR
1315  004034  032770  000040  000000          BIT     #BIT5,@(R0)             ;IF ERROR ADDRESS BITS ARE TRACKING,
1316                                                                          ;BIT 5 SHOULD BE CLEAR (ONLY FOR CORE PARITY)
1317  004042  001401                          BEQ     .+4
1318  004044  104002                          ERROR                           ;PARITY ERROR ADDRESS BITS INCORRECT
1319                                                                          ;R0 POINTS TO THE ADDRESS OF THE PARITY
1320                                                                          ;REGISTER, R1 CONTAINS THE ADDRESS
1321                                                                          ;REFERENCED TO CAUSE A PARITY TRAP
1322                                                                          ;(VIRTUAL IF KT11 IS PRESENT)
1323  004046  012716  004002          2$:     MOV     #INST1,@SP              ;GO EXECUTE INSTRUCTION 1 AGAIN
1324  004052  000002                          RTI
1325  004054  012737  004070  000114  1$:     MOV     #TRP10A,@#PARVEC                ;CHANGE PARITY TRAP RETURN
1326  004062  012716  004010                  MOV     #INST2,@SP              ;GO EXECUTE INSTRUCTION 2
1327  004066  000002                          RTI
1328  004070  022712  000001          TRP10A: CMP     #1,(R2)                 ;IS THIS A CORE REG?
1329  004074  001005                          BNE     1$                      ;NO, BRANCH
1330  004076  032770  000040  000000          BIT     #BIT5,@(R0)             ;PARITY TRAP OCCURRED- CHECK PARITY
1331                                                                          ;ERROR ADDRESS BITS
1332  004104  001001                          BNE     .+4                     ;BRANCH IF OK (IF THE PARITY ERROR
1333                                                                          ;ADDRESS BITS TRACKED, BIT 5 WILL BE SET)
1334  004106  104002                          ERROR                           ;PARITY ERROR ADDRESS BITS INCORRECT
1335                                                                          ;R0 POINTS TO THE ADDRESS OF THE
1336                                                                          ;PARITY REGISTER. THE ADDRESS REFERENCED
1337                                                                          ;TO CAUSE THE ERROR WAS THAT IN
1338                                                                          ;R1 PLUS 4000 (OCTAL).
1339  004110  022626                  1$:     CMP     (SP)+,(SP)+
1340  004112  012737  000116  000114  CONT10: MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
1341  004120  005070  000000                  CLR     @(R0)                   ;CLEAR PARITY REGISTER
1342  004124  005511                          ADC     @R1                     ;CLEAR BAD PARITY
1343  004126  005561  004000                  ADC     4000(R1)
1344  004132  005070  000000                  CLR     @(R0)                   ;CLEAR PARITY ERROR BIT IF SET
1345  004136  000662                          BR      LOOP10
1346  004140  005767  174406          DONE10: TST     NOKT
1347  004144  001002                          BNE     .+6
1348  004146  005037  177572                  CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1349  004152  000401                          BR      TEST11
1350  004154  000000                  COUNT:  0
1351
1352
```

```
                                        ;*********************************************************************
                                        ;IF MULTIPLE PARITY ERRORS OCCUR DURING ONE INSTRUCTION (WITH ACTION ENABLE
                                        ;NOT SET) THE ERROR ADDRESS BITS WILL RECORD THE LAST ERROR (ONLY FOR CORE PARITY
                                        ;REGISTERS)
                                        ;*********************************************************************
004156  104001                        TEST11: SCOPE
004160  012737  000011  177570                MOV     #11,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
004166  004767  007172                        JSR     %7,CLRPAR               ;INITIALLY CLEAR ALL PARITY REGISTERS
004172  005767  174354                        TST     NOKT                    ;KT11 PRESENT?
004176  001004                                BNE     1$                      ;NO= BRANCH
004200  004767  007060                        JSR     %7,NRALL                ;YES, MAP KERNEL PAGE 0 TO BANK 0,RW
004204  004767  007224                        JSR     %7,MAP1                 ;MAP KERNEL PAGE 7 TO THE EXTERNAL BANK, RW
                                                                              ;SET KERNEL PAGE 1 RW AND TURN ON KT11
004210  012700  000570                1$:     MOV     #MPR0,R0                ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
004214  012703  000772                        MOV     #INDC0,R3
004220  032710  000001                LUP11:  BIT     #1,@R0
004224  001003                                BNE     LOOP11                  ;IF THIS REG NOT PRESENT, SKIP
004226  022713  000001                        CMP     #1,(R3)                 ;IS THIS A CORE PAR REG?
004232  001407                                BEQ     TST11                   ;YES, THEN TEST IT
                                                                              ;IF NOT CORE, SKIP THIS REGISTER
004234  062700  000010                LOOP11: ADD     #10,R0
004240  005723                                TST     (R3)+
004242  020027  000770                        CMP     R0,#TREG
004246  103764                                BLO     LUP11
004250  000443                                BR      DONE11                  ;BRANCH OUT IF ALL REGISTERS HAVE BEEN TESTED
004252  004767  007222                TST11:  JSR     %7,LOCATM               ;GET THE ADDRESS OF A MEMORY LOCATION
                                                                              ;CORRESPONDING TO THIS PARITY REGISTER
004256  032701  000001                        BIT     #1,R1                   ;ERROR RETURN INDICATED?
```



```
004262  001403                                BEQ     .+10                    ;BRANCH IF NOT
004264  104002                                ERROR                           ;NO MEMORY IN MAP CORRESPONDING TO
                                                                              ;THIS PARITY REGISTER, R0 POINTS
                                                                              ;TO THE ADDRESS OF THE PARITY REGISTER
004266  000167  177742                        JMP     LOOP11
004272  010102                                MOV     R1,R2                   ;SETUP SECOND TEST ADDRESS LOCATION
004274  062702  010000                        ADD     #10000,R2
004300  012770  000004  000000                MOV     #WWP,@(R0)              ;SET WRITE WRONG PARITY
004306  011111                                MOV     @R1,@R1                 ;WRITE WRONG PARITY IN FIRST TEST LOCATION
004310  011212                                MOV     @R2,@R2                 ;WRITE WRONG PARITY IN SECOND TEST LOCATION
004312  005070  000000                        CLR     @(R0)                   ;CLEAR PARITY REGISTER
004316  021112                                CMP     @R1,@R2                 ;READ FIRST TEST LOCATION, AND
                                                                              ;THEN READ SECOND TEST LOCATION
004320  005770  000000                        TST     @(R0)                   ;MAKE SURE PARITY ERROR SET
004324  100401                                BMI     .+4
004326  104002                                ERROR                           ;PARITY ERROR NOT SET AFTER
                                                                              ;READING TWO LOCATIONS WHICH
                                                                              ;SHOULD HAVE BAD PARITY
004330  032770  000100  000000                BIT     #BIT6,@(R0)             ;CHECK ERROR ADDRESS= IF THE LAST
                                                                              ;ADDRESS WAS RECORDED, BIT 6 WILL
                                                                              ;BE SET
004336  001001                                BNE     .+4
004340  104002                                ERROR                           ;PARITY ERROR ADDRESS BITS INCORRECT
                                                                              ;R0 POINTS TO ADDRESS OF PARITY REGISTER
                                                                              ;R2 CONTAINS ADDRESS OF LAST BAD PARITY
                                                                              ;LOCATION REFERENCED (IF KT11 PRESENT,
                                                                              ;ADDRESS IS VIRTUAL THRU KERNEL PAGE 1)
004342  005070  000000                        CLR     @(R0)                   ;CLEAR PARITY REGISTER
004346  005511                                ADC     @R1                     ;CLEAR BAD PARITY
004350  005512                                ADC     @R2
004352  005070  000000                        CLR     @(R0)                   ;CLEAR PARITY ERROR BIT
004356  000726                                BR      LOOP11
004360  005767  174166                DONE11: TST     NOKT
004364  001002                                BNE     .+6
004366  005037  177572                        CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT


                                        ;*********************************************************************
                                        ;SHOW THAT IF AN INSTRUCTION DOING A DATIP GETS A PARITY ERROR,
                                        ;THE ORIGINAL DATA IS REWRITTEN IF ACTION ENABLE IS SET, AND IS
                                        ;ALTERRED IF ACTION ENABLE IS CLEAR
                                        ;*********************************************************************
004372  104001                        TEST12: SCOPE
004374  012737  000012  177570                MOV     #12,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
004402  004767  006756                        JSR     %7,CLRPAR
004406  005767  174140                        TST     NOKT                    ;KT11 PRESENT?
004412  001004                                BNE     1$                      ;NO= BRANCH
004414  004767  006644                        JSR     %7,NRALL                ;YES, MAP KERNEL 0 TO BANK 0, RW
004420  004767  007010                        JSR     %7,MAP1                 ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
                                                                              ;SET KERNEL 1 RW AND TURN ON KT11
004424  012700  000570                1$:     MOV     #MPR0,R0                ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
004430  012702  000772                        MOV     #INDC0,R2
004434  032710  000001                LUP12:  BIT     #1,@R0
```

```
1436  004440  001003                          BNE     LOOP12          ;SKIP, IF THIS REG NOT PRESENT
1437  004442  022712  000001                  CMP     #1,(R2)         ;REG PRESENT, IS IT CORE?
1438  004446  001407                          BEQ     TST12           ;YES, DO* THIS TEST
1439                                                                  ;SKIP, IF THIS REG IS NOT CORE
1440  004450  062700  000010          LOOP12: ADD     #10,R0
1441  004454  005722                          TST     (R2)+
1442  004456  020027  000770                  CMP     R0,#TREG
1443  004462  103764                          BLO     LUP12
1444  004464  000474                          BR      DONE12          ;BRANCH TO DONE IF ALL REGISTERS
1445                                                                  ;HAVE BEEN TESTED
1446  004466  004767  007006          TST12:  JSR     %7,LOCATM       ;LOCATE MEMORY CORRESPONDING TO THIS
1447                                                                  ;REGISTER
1448  004472  032701  000001                  BIT     #1,R1           ;ERROR RETURN INDICATED?
1449  004476  001403                          BEQ     .+10            ;NO= BRANCH
1450  004500  104002                          ERROR                   ;NO MEMORY IN MAP CORRESPONDING TO
1451                                                                  ;THIS REGISTER, R0 POINTS TO
1452                                                                  ;THE ADDRESS OF THE PARITY REGISTER
1453  004502  000167  177742                  JMP     LOOP12
1454  004506  012737  004542  000114          MOV     #TRP12,@#PARVEC ;SET UP PARITY TRAP RETURN
1455  004514  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
1456  004522  012711  125252                  MOV     #125252,@R1     ;WRITE WRONG PARITY IN TEST LOCATION
1457  004526  012770  000001  000000          MOV     #AE,@(R0)       ;SET ACTION ENABLE AND CLEAR
1458                                                                  ;WRITE WRONG PARITY
1459  004534  005211                          INC     @R1             ;DO DATIP,DATO WITH ACTION ENABLE
1460                                                                  ;SET= SHOULD ABORT ON DATIP AND
1461                                                                  ;RESTORE ORIGINAL DATA
1462  004536  104002                          ERROR                   ;NO ABORT OCCURRED ON READING LOCATION
1463                                                                  ;WHICH SHOULD CONTAIN BAD PARITY
1464                                                                  ;(WITH AE SET).
1465                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER.
1466                                                                  ;R1 CONTAINS ADDRESS OF TEST LOCATION
1467                                                                  ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 IS
1468                                                                  ;PRESENT)
1469  004540  000440                          BR      CONT12
1470  004542  005070  000000          TRP12:  CLR     @(R0)           ;PARITY TRAP OCCURRED= CLEAR PARITY REGISTER
1471  004546  021127  125252                  CMP     @R1,#125252     ;ORIGINAL DATA RESTORED?
1472  004552  001401                          BEQ     .+4             ;YES, BRANCH
1473  004554  104002                          ERROR                   ;NO= DATIP WHICH GOT A PARITY ERROR
1474                                                                  ;TRAP ALTERRED CONTENTS OF LOCATION
1475                                                                  ;READ, ADDRESS OF TEST LOCATION IS IN R1
1476                                                                  ;(IF KT11 IS PRESENT, ADDRESS IN R1
1477                                                                  ;IS VIRTUAL THRU KERNEL PAGE 1)
1478                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER
1479  004556  005770  000000                  TST     @(R0)           ;MAKE SURE PARITY ERROR SET WHEN
1480  004562  100401                          BMI     .+4             ;DATA WAS REREAD IN THE ABOVE CMP
1481  004564  104002                          ERROR                   ;DATIP WHICH GOT A PARITY ERROR TRAP
1482                                                                  ;ALTERRED THE PARITY OF THE LOCATION READ
1483                                                                  ;R1 CONTAINS ADDRESS OF TEST LOCATION
1484                                                                  ;(VIRTUAL THRU KERNEL 1 IF KT11 PRESENT)
1485  004566  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1486  004570  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY AND CLEAR
1487                                                                  ;PARITY ERROR
1488  004576  012711  125252                  MOV     #125252,@R1     ;REWRITE DATA WITH WRONG PARITY
1489  004602  005070  000000                  CLR     @(R0)           ;CLEAR PARITY REGISTER
```



```
1490  004606  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAPCATCHER
1491  004614  005211                          INC     @R1             ;SINCE AE IS CLEAR, INSTRUCTION SHOULD
1492                                                                  ;COMPLETE AND SHOULD CLEAR BAD PARITY
1493  004616  005070  000000                  CLR     @(R0)           ;CLEAR PARITY ERROR BIT
1494  004622  022711  125253                  CMP     #125253,@R1     ;CHECK DATA
1495  004626  001401                          BEQ     .+4
1496  004630  104002                          ERROR                   ;DATIP, DATO TO A LOCATION CONTAINING BAD
1497                                                                  ;PARITY WITHOUT AE SET LEFT INCORRECT
1498                                                                  ;DATA, R0 POINTS TO THE ADDRESS OF
1499                                                                  ;THE PARITY REGISTER, R1 CONTAINS THE
1500                                                                  ;ADDRESS OF THE TEST LOCATION (VIRTUAL
1501                                                                  ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1502  004632  005770  000000                  TST     @(R0)           ;CHECK PARITY ERROR BIT
1503  004636  100001                          BPL     .+4
1504  004640  104002                          ERROR                   ;DATIP, DATO WITH AE CLEAR DID
1505                                                                  ;NOT CLEAR BAD PARITY IN LOCATION
1506                                                                  ;ADDRESSED, R0 POINTS TO THE ADDRESS
1507                                                                  ;OF THE PARITY REGISTER, R1 CONTAINS
1508                                                                  ;THE ADDRESS OF THE TEST LOCATION
1509                                                                  ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11
1510                                                                  ;IS PRESENT)
1511  004642  005070  000000          CONT12: CLR     @(R0)           ;CLEAR PARITY REGISTER
1512  004646  005011                          CLR     @R1             ;CLEAR LOCATION TO RESTORE GOOD PARITY
1513  004650  005070  000000                  CLR     @(R0)           ;CLEAR PARITY ERROR IF SET
1514  004654  000675                          BR      LOOP12          ;GO CHECK FOR ANOTHER PARITY
1515                                                                  ;REGISTER
1516  004656  012737  000116  000114  DONE12: MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAPCATCHER
1517  004664  005767  173662                  TST     NOKT
1518  004670  001002                          BNE     .+6
1519  004672  005037  177572                  CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1520
1521
1522
1523                                    ;*********************************************************************
1524                                    ;SHOW THAT IF AN INSTRUCTION DOING A DATI (BUT NO DATO TO THE SAME LOCATION)
1525                                    ;GETS A PARITY ERROR, THE ORIGINAL DATA IS UNALTERRED, WHETHER OR NOT ACTION
1526                                    ;ENABLE IS SET
1527                                    ;*********************************************************************
1528  004676  104001                    TEST13: SCOPE
1529  004700  012737  000013  177570            MOV     #13,@#DISPLY    ;LOAD TEST NUMBER INTO THE DISPLAY
1530  004706  004767  006452                    JSR     %7,CLRPAR
1531  004712  005767  173634                    TST     NOKT            ;KT11 PRESENT?
1532  004716  001004                            BNE     1$              ;NO= BRANCH
1533  004720  004767  006340                    JSR     %7,MRALL        ;YES, MAP KERNEL 0 TO BANK 0,KERNEL
1534  004724  004767  006504                    JSR     %7,MAP1         ;7 TO THE EXTERNAL BANK, AND KERNEL
1535                                                                    ;0,1,AND 7 RW
1536  004730  012700  000570            1$:     MOV     #MPR0,R0        ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1537  004734  032710  000001            LUP13:  BIT     #1,@R0
1538  004740  001406                            BEQ     TST13           ;IF THIS REGISTER IS PRESENT, GO
1539                                                                    ;TEST IT
1540  004742  062700  000010            LOOP13: ADD     #10,R0
1541  004746  020027  000770                    CMP     R0,#TREG
1542  004752  103770                            BLO     LUP13
1543  004754  000470                            BR      DONE13          ;BRANCH IF ALL REGISTERS HAVE BEEN
```

```
1544                                                                    ;TESTED
1545  004756  004767  006516          TST13:  JSR     %7,LOCATM         ;LOCATE MEMORY CORRESPONDING TO THIS
1546                                                                    ;REGISTER, AND IF KT11 IS PRESENT
1547                                                                    ;MAP KERNEL 1 TO THAT MEMORY
1548  004762  032701  000001                  BIT     #1,R1             ;ERROR RETURN INDICATED?
1549  004766  001403                          BEQ     .+10              ;NO- BRANCH
1550  004770  104002                          ERROR                     ;MAP INDICATES NO MEMORY WAS FOUND
1551                                                                    ;CORRESPONDING TO THIS REGISTER
1552                                                                    ;R0 POINTS TO THE ADDRESS OF THE
1553                                                                    ;PARITY REGISTER
1554  004772  000167  177744                  JMP     LOOP13
1555  004776  012737  005032  000114          MOV     #TRP13,@#PARVEC   ;SETUP PARITY TRAP RETURN
1556  005004  012770  000004  000000          MOV     #WWP,@(R0)        ;SET WRITE WRONG PARITY
1557  005012  012711  125252                  MOV     #125252,@R1       ;WRITE WRONG PARITY
1558  005016  012770  000001  000000          MOV     #AE,@(R0)         ;SET ACTION ENABLE AND CLEAR
1559                                                                    ;WRITE WRONG PARITY
1560  005024  005711                          TST     @R1               ;DATI WITH ACTION ENABLE SET SHOULD
1561                                                                    ;ABORT LEAVING DATA UNCHANGED
1562  005026  104002                          ERROR                     ;NO ABORT ON READING BAD PARITY
1563                                                                    ;WITH ACTION ENABLE SET, R0 POINTS
1564                                                                    ;TO THE ADDRESS OF THE PARITY REGISTER,
1565                                                                    ;R1 CONTAINS THE ADDRESS OF THE TEST
1566                                                                    ;LOCATION (VIRTUAL THRU KERNEL PAGE
1567                                                                    ;1 IF KT11 IS PRESENT).
1568  005030  000434                          BR      CONT13
1569  005032  005070  000000          TRP13:  CLR     @(R0)             ;ABORT OCCURRED AS EXPECTED- CLEAR REGISTER
1570  005036  021127  125252                  CMP     @R1,#125252       ;ORIGINAL DATA RESTORED?
1571  005042  001401                          BEQ     .+4               ;YES, BRANCH
1572  005044  104002                          ERROR                     ;DATI WHICH GOT A PARITY ERROR ALTERRED
1573                                                                    ;THE CONTENTS OF THE LOCATION ADDRESSED,
1574                                                                    ;(R1 CONTAINS THE ADDRESS OF
1575                                                                    ;MEMORY BEING TESTED- IF KT11 IS
1576                                                                    ;PRESENT, ADDRESS IN R1 IS VIRTUAL)
1577                                                                    ;R0 POINTS TO THE ADDRESS OF THE
1578                                                                    ;PARITY REGISTER
1579  005046  005770  000000                  TST     @(R0)             ;CHECK PARITY REGISTER
1580  005052  100401                          BMI     .+4               ;BRANCH IF PARITY ERROR SET
1581  005054  104002                          ERROR                     ;PARITY ERROR NOT SET AFTER READING
1582                                                                    ;DATA WITH BAD PARITY
1583                                                                    ;R0 POINTS TO THE ADDRESS OF THE PARITY
1584                                                                    ;REGISTER, R1 CONTAINS THE ADDRESS
1585                                                                    ;OF THE TEST LOCATION (VIRTUAL THRU
1586                                                                    ;KERNEL PAGE 1 IF KT11 PRESENT)
1587  005056  022626                          CMP     (SP)+,(SP)+       ;RESTORE STACK POINTER
1588  005060  012770  000004  000000          MOV     #WWP,@(R0)        ;SET WRITE WRONG PARITY, CLEAR PARITY ERROR
1589  005066  012711  125252                  MOV     #125252,@R1       ;REWRITE DATA WITH WRONG PARITY
1590  005072  005070  000000                  CLR     @(R0)             ;CLEAR PARITY REGISTER
1591  005076  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC          ;RESTORE TRAPCATCHER
1592  005104  005711                          TST     @R1               ;DATI TO LOCATION WITH BAD PARITY
1593                                                                    ;AE NOT SET-INSTRUCTION SHOULD COMPLETE
1594  005106  005070  000000                  CLR     @(R0)             ;CLEAR PARITY ERROR BIT
1595  005112  022711  125252                  CMP     #125252,@R1       ;CHECK DATA
1596  005116  001401                          BEQ     .+4
1597  005120  104002                          ERROR                     ;DATI TO LOCATION WITH BAD PARITY
```



```
1598                                                                    ;WITHOUT ACTION ENABLE SET LEFT INCORRECT DATA
1599                                                                    ;R1 CONTAINS THE ADDRESS OF THE TEST
1600                                                                    ;LOCATION (VIRTUAL THRU KERNEL PAGE 1
1601                                                                    ;IF KT11 IS PRESENT). R0 POINTS TO THE
1602                                                                    ;ADDRESS OF THE PARTIY REGISTER.
1603  005122  005070  000000          CONT13: CLR     @(R0)             ;CLEAR PARITY REGISTER
1604  005126  005011                          CLR     @R1               ;CLEAR LOCATION
1605  005130  005070  000000                  CLR     @(R0)             ;CLEAR PARITY ERROR IF SET
1606  005134  000702                          BR      LOOP13            ;GO CHECK FOR ANOTHER PARITY
1607                                                                    ;REGISTER
1608  005136  012737  000116  000114  DONE13: MOV     #PARVEC+2,@#PARVEC          ;RESTORE TRAPCATCHER
1609  005144  005767  173402                  TST     NOKT
1610  005150  001002                          BNE     .+6
1611  005152  005037  177572                  CLR     @#SR0             ;TURN OFF KT11 IF PRESENT
1612
1613
1614
1615                                          ;*********************************************************************
1616                                          ;CHECK PARITY MEMORY WITH SERIES OF PATTERNS FROM 4K TO 28K
1617                                          ;ENABLE PARITY TRAP
1618                                          ;*********************************************************************
1619  005156  104001          TEST14: SCOPE
1620  005160  012737  000014  177570          MOV     #14,@#DISPLY      ;LOAD TEST NUMBER INTO THE DISPLAY
1621  005166  005067  010630                  CLR     IMAX              ;DON'T ITERATE THE REST OF THE SUBTESTS
1622  005172  004767  006166                  JSR     PC,CLRPAR         ;CLEAR ALL PARITY REGISTERS
1623  005176  012737  005500  000114          MOV     #TRP14,@#PARVEC   ;SETUP PARITY TRAP RETURN
1624  005204  012767  000002  173306          MOV     #2,BITPT          ;INITIALIZE BANK INDICATOR TO BANK 1
1625  005212  012767  020000  173276          MOV     #20000,ADRPT      ;INITIALIZE MEMORY STARTING ADDRESS
1626  005220  036767  173274  173644  LOOP14: BIT     BITPT,PMEML       ;DOES THIS 4K HAVE PARITY?
1627  005226  001012                          BNE     TST14             ;YES, TEST IT
1628  005230  062767  020000  173260  LUP14:  ADD     #20000,ADRPT      ;NO- UPDATE MEMORY ADDRESS
1629  005236  006367  173256                  ASL     BITPT             ;UPDATE BIT POINTER
1630  005242  022767  000200  173250          CMP     #200,BITPT        ;THIS 28K DONE?
1631  005250  003363                          BGT     LOOP14            ;NO, BRANCH TO SEE IF NEXT 4K
1632                                                                    ;SHOULD BE TESTED
1633  005252  000443                          BR      DONE14            ;YES, EXIT
1634  005254  012704  001040          TST14:  MOV     #PARPAT,R4        ;INITIALIZE PATTERN POINTER
1635  005260  016767  173232  173242          MOV     ADRPT,HIADR       ;SET UPPER LIMIT FOR THIS 4K
1636  005266  062767  020000  173234          ADD     #20000,HIADR
1637  005274  016705  173216                  MOV     ADRPT,R5
1638  005300  005025                  2$:     CLR     (5)+              ;INITIALLY CLEAR CORE BLOCK UNDER TEST
1639  005302  020567  173222                  CMP     R5,HIADR
1640  005306  103774                          BLO     2$
1641  005310  012701  000570                  MOV     #MPR0,R1          ;INITIALIZE TO SET AE IN ALL REGISTERS
1642  005314  032711  000001          3$:     BIT     #1,@R1
1643  005320  001003                          BNE     .+10
1644  005322  012771  000001  000000          MOV     #AE,@(R1)         ;SET ACTION ENABLE IF REGISTER IS PRESENT
1645  005330  062701  000010                  ADD     #10,R1
1646  005334  020127  000770                  CMP     R1,#TREG
1647  005340  103765                          BLO     3$
1648  005342  004767  000024          4$:     JSR     %7,TPCORE         ;GO TO ROUTINE TO EXERCISE THIS 4K
1649                                                                    ;WITH THE CURRENT PATTERN
1650  005346  005724                          TST     (4)+              ;UPDATE PATTERN
1651  005350  005714                          TST     (4)               ;LAST PATTERN?
```

```
1652  005352  001373                          BNE     4$                      ;NO, LOOP
1653  005354  004767  006004                  JSR     PC,CLRPAR               ;YES, CLEAR ALL PARITY REGISTERS
1654  005360  000723                          BR      LUP14                   ;UPDATE AND CHECK NEXT 4K
1655  005362  012737  000116  000114  DONE14: MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAP CATCHER
1656  005370  000473                          BR      TEST15                  ;GO TO NEXT TEST
1657
1658                                  ;ROUTINE TO WRITE AND CHECK EACH LOCATION IN 4K (STARTING AT ADDRESS
1659                                  ;IN ADRPT) WITH VALUE POINTED TO BY R4
1660  005372  016705  173120          TPCORE: MOV     ADRPT,R5                ;SETUP R5 TO ADDRESS MEMORY
1661                                                                          ;LOCATION BEING CHECKED
1662  005376  011415                  1$:     MOV     (4),(5)                 ;WRITE PATTERN INTO MEMORY
1663  005400  011567  173132                  MOV     (5),WAS                 ;READ TEST LOCATION
1664  005404  021467  173126                  CMP     (4),WAS                 ;DATA OK?
1665  005410  001401                          BEQ     .+4                     ;YES- BRANCH
1666  005412  104002                          ERROR                           ;DATA INCORRECT IN LOCATION WHOSE
1667                                                                          ;ADDRESS IS IN R5, R4 POINTS TO THE
1668                                                                          ;DATA WRITTEN.
1669  005414  005725                          TST     (5)+                    ;UPDATE ADDRESS POINTER
1670  005416  020567  173106                  CMP     R5,HIADR                ;THIS 4K DONE?
1671  005422  103765                          BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
1672  005424  005067  173340                  CLR     TREG                    ;YES, DID ANY PARITY ERRORS OCCUR
1673                                                                          ;WITHOUT TRAPPING?
1674  005430  012701  000570                  MOV     #MPR0,R1
1675  005434  032711  000001          2$:     BIT     #1,(1)
1676  005440  001003                          BNE     .+10
1677  005442  005771  000000                  TST     @(R1)
1678  005446  100406                          BMI     3$                      ;YES- BRANCH
1679  005450  062701  000010                  ADD     #10,R1
1680  005454  020127  000770                  CMP     R1,#TREG
1681  005460  103765                          BLO     2$
1682  005462  000207                          RTS     %7                      ;NO, RETURN
1683  005464  011167  173300          3$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1684  005470  104004                          ERRORS                          ;PARITY ERROR SET (WITH AE SET) AND
1685                                                                          ;NO TRAP OCCURRED, TREG CONTAINS
1686                                                                          ;ADDRESS OF PARITY REGISTER WHICH
1687                                                                          ;HAS ERROR BIT SET.
1688  005472  004767  006370                  JSR     %7,PSCAN                ;SCAN FOR PARITY ERRORS AND PRINT
1689                                                                          ;18 BIT ADDRESSES OF THOSE FOUND,
1690                                                                          ;AFTER REPORTING EACH ERROR CLEAR IT
1691  005476  000207                          RTS     %7
1692
1693                                  ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
1694  005500  005067  173264  TRP14:  CLR     TREG
1695  005504  012701  000570          MOV     #MPR0,R1                ;FIND PARITY REGISTER INDICATING PARITY ERROR
1696  005510  032711  000001          1$:     BIT     #1,(R1)
1697  005514  001003                          BNE     .+10
1698  005516  005771  000000                  TST     @(R1)
1699  005522  100407                          BMI     2$                      ;BRANCH IF PARITY ERROR SET
1700  005524  062701  000010                  ADD     #10,R1
1701  005530  020127  000770                  CMP     R1,#TREG
1702  005534  103765                          BLO     1$
1703  005536  104002                          ERROR                           ;PARITY TRAP TO 114 OCCURRED DURING
1704  005540  000405                          BR      3$                      ;TEST 14 BUT NO REGISTERS HAVE
1705                                                                          ;PARITY ERROR SET
```



```
1706  005542  011167  173222          2$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1707  005546  104004                          ERRORS                          ;PARITY TRAP TO 114 OCCURRED DUE TO
1708                                                                          ;PARITY ERROR WHILE EXERCISING MEMORY
1709                                                                          ;R1 POINTS TO THE ADDRESS OF THE
1710                                                                          ;PARITY REGISTER HAVING PARITY ERROR
1711                                                                          ;BIT SET
1712  005550  004767  006312                  JSR     %7,PSCAN                ;SCAN FOR BAD PARITY AND TYPE 18 BIT ADDRESSES
1713                                                                          ;OF LOCATIONS FOUND BAD
1714  005554  022626                  3$:     CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1715  005556  000207                          RTS     %7                      ;RETURN (FROM JSR TO TPCORE) TO
1716                                                                          ;CHECK NEXT PATTERN
1717
1718
1719
1720                                  ;*************************************************************************
1721                                  ;CHECK PARITY MEMORY WITH SERIES OF PATTERNS ABOVE 28K
1722                                  ;ENABLE PARITY ERROR TRAPPING
1723                                  ;*************************************************************************
1724  005560  104001          TEST15: SCOPE
1725  005562  012737  000015  177570          MOV     #15,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
1726  005570  005767  172756                  TST     NOKT                    ;KT11 PRESENT?
1727  005574  001402                          BEQ     .+6                     ;YES- BRANCH
1728  005576  000167  000424                  JMP     TEST16                  ;NO, SKIP TEST
1729  005602  004767  005556                  JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1730  005606  004767  005452                  JSR     %7,NRALL                ;MAP KERNEL 0 TO BANK 0,RW
1731  005612  004767  005616                  JSR     %7,MAP1                 ;MAP KERNEL 7 TO THE EXTERNAL BANK
1732                                                                          ;SET KERNEL 1 RW AND TURN ON KT11
1733  005616  012777  001600  173416          MOV     #1600,@KPAR1            ;MAP KERNEL PAGE 1 TO BEGINNING OF 28-32K
1734  005624  005067  172726                  CLR     LOWFLG                  ;CLEAR FLAG TO INDICATE CHECKING LOWER 64K
1735  005630  016767  173236  173240          MOV     PMEML,PMEMX
1736  005636  012737  006146  000114          MOV     #TRP15,@#PARVEC         ;SETUP PARITY TRAP RETURN
1737  005644  012767  000200  172646          MOV     #200,BITPT              ;INITIALIZE BIT POINTER
1738  005652  036767  172642  173216  LOOP15: BIT     BITPT,PMEMX             ;DOES THIS 4K HAVE PARITY?
1739  005660  001022                          BNE     TST15                   ;YES, BRANCH TO TEST IT
1740  005662  062777  000200  173352  LOP15:  ADD     #200,@KPAR1             ;NO- MAP TO NEXT 4K
1741  005670  006367  172624                  ASL     BITPT                   ;UPDATE BIT POINTER
1742  005674  103366                          BCC     LOOP15                  ;BRANCH IF NOT DONE WITH 64K
1743  005676  005767  172654                  TST     LOWFLG                  ;DONE WITH 128K?
1744  005702  001051                          BNE     DONE15                  ;YES, BRANCH
1745  005704  005267  172646                  INC     LOWFLG                  ;NO, SET FLAG INDICATING UPPER 64K
1746  005710  016767  173160  173160          MOV     PMEMH,PMEMX             ;SETUP PARITY MAP WORD
1747  005716  012767  000001  172574          MOV     #1,BITPT                ;SETUP BIT POINTER FOR UPPER 64K
1748  005724  000752                          BR      LOOP15                  ;CONTINUE
1749  005726  012704  001040  TST15:  MOV     #PARPAT,R4              ;INITIALIZE PATTERN POINTER
1750  005732  012767  020000  172556          MOV     #20000,ADRPT            ;INITIALIZE VIRTUAL ADDRESS OF MEMORY
1751                                                                          ;BEING TESTED
1752  005740  012705  020000                  MOV     #20000,R5
1753  005744  005025                  2$:     CLR     (5)+                    ;INITIALLY CLEAR CORE BLOCK UNDER TEST
1754  005746  020527  040000                  CMP     R5,#40000
1755  005752  103774                          BLO     2$
1756  005754  012701  000570                  MOV     #MPR0,R1                ;INITIALIZE TO SET ACTION ENABLE IN ALL
1757                                                                          ;PARITY REGISTERS
1758  005760  032711  000001          3$:     BIT     #1,@R1
1759  005764  001003                          BNE     .+10
```

```
1760  005766  012771  000001  000000          MOV     #AE,@(R1)               ;SET ACTION ENABLE IF THIS REGISTER
1761                                                                          ;IS PRESENT
1762  005774  062701  000010                  ADD     #10,R1
1763  006000  020127  000770                  CMP     R1,#TREG
1764  006004  103765                          BLO     3$
1765  006006  004767  000030          4$:     JSR     %7,TPCORX               ;EXERCISE THIS 4K
1766  006012  005724                          TST     (4)+                    ;UPDATE PATTERN
1767  006014  005714                          TST     (4)                     ;LAST PATTERN?
1768  006016  001373                          BNE     4$                      ;NO, LOOP
1769  006020  004767  005340                  JSR     PC,CLRPAR               ;YES, CLEAR ALL PARITY REGISTERS
1770  006024  000716                          BR      LOP15                   ;UPDATE AND CHECK NEXT 4K
1771  006026  005037  177572          DONE15: CLR     @#SR0                   ;TURN OFF KT11 WHEN DONE
1772  006032  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
1773  006040  000472                          BR      TEST16                  ;GO TO NEXT TEST
1774
1775                                  ;PARITY MEMORY TEST ROUTINE USING KT11 AND TESTING MEMORY ABOVE 28K
1776                                  ;WRITES AND CHECKS EACH LOCATION IN 4K USING KERNEL PAGE 1 MAPPED TO CURRENT BANK
1777  006042  000240          TPCORX: NOP
1778  006044  012705  020000                  MOV     #20000,R5               ;SETUP R5 TO POINT TO THE LOCATION
1779                                                                          ;UNDER TEST (VIRTUAL ADDRESS)
1780  006050  011415          1$:     MOV     (4),(5)                 ;WRITE PATTERN
1781  006052  011567  172460                  MOV     (5),WAS                 ;READ TEST LOCATION
1782  006056  021467  172454                  CMP     (4),WAS                 ;DATA OK?
1783  006062  001401                          BEQ     .+4                     ;YES- BRANCH
1784  006064  104002                          ERROR                           ;NO- DATA INCORRECT IN LOCATION WHOSE
1785                                                                          ;VIRTUAL ADDRESS IS IN R1 (GOES THRU
1786                                                                          ;KERNEL PAGE 1), R4 POINTS TO
1787                                                                          ;THE VALUE WRITTEN.
1788  006066  005725                          TST     (5)+                    ;UPDATE ADDRESS POINTER
1789  006070  020527  040000                  CMP     R5,#40000               ;THIS 4K DONE?
1790  006074  103765                          BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
1791  006076  005067  172666                  CLR     TREG                    ;YES, CHECK TO SEE IF ANY PARITY
1792                                                                          ;ERRORS OCCURRED WITHOUT TRAPPING
1793  006102  012701  000570                  MOV     #MPR0,R1
1794  006106  032711  000001          2$:     BIT     #1,(1)                  ;IS THIS PARITY REGISTER PRESENT?
1795  006112  001003                          BNE     .+10                    ;NO, GET NEXT ONE
1796  006114  005771  000000                  TST     @(R1)                   ;YES- DID ERROR SET?
1797  006120  100406                          BMI     3$                      ;YES- BRANCH
1798  006122  062701  000010                  ADD     #10,R1                  ;NO- GET NEXT REGISTER
1799  006126  020127  000770                  CMP     R1,#TREG
1800  006132  103765                          BLO     2$
1801  006134  000207                          RTS     %7                      ;NO ERRORS- EXIT
1802  006136  011167  172626          3$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1803  006142  104004                          ERRORS                          ;PARITY ERROR SET (AE ALREADY SET)
1804                                                                          ;AND NO TRAP OR TIMEOUT OCCURRED
1805                                                                          ;R1 POINTS TO THE ADDRESS OF THE
1806                                                                          ;PARITY REGISTER
1807  006144  000207                          RTS     %7
1808
1809                                  ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
1810  006146  005067  172616          TRP15:  CLR     TREG
1811  006152  012701  000570                  MOV     #MPR0,R1                ;LOCATE PARITY REGISTER INDICATING ERROR
1812  006156  032711  000001          1$:     BIT     #1,(R1)
1813  006162  001003                          BNE     .+10
```



```
1814  006164  005771  000000                  TST     @(R1)
1815  006170  100407                          BMI     2$                      ;BRANCH IF PARITY ERROR IS SET
1816  006172  062701  000010                  ADD     #10,R1
1817  006176  020127  000770                  CMP     R1,#TREG
1818  006202  103765                          BLO     1$
1819  006204  104002                          ERROR                           ;TRAP TO 114 OCCURRED DURING TEST 15 BUT
1820                                                                          ;NO PARITY REGISTERS HAVE PARITY ERROR SET
1821  006206  000405                          BR      3$
1822  006210  011167  172554          2$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1823  006214  104004                          ERRORS                          ;PARITY TRAP TO 114 OCCURRED DUE TO
1824                                                                          ;PARITY ERROR WHILE EXERCISING MEMORY
1825                                                                          ;"TREG" CONTAINS ADDRESS OF PARITY REGISTER
1826                                                                          ;HAVING PARITY ERROR BIT SET
1827  006216  004767  005644                  JSR     %7,PSCAN                ;SCAN MEMORY FOR BAD PARITY AND PRINT 18
1828                                                                          ;BIT ADDRESSES OF LOCATIONS FOUND
1829                                                                          ;CLEAR BAD PARITY IN EACH AFTER
1830                                                                          ;REPORTING IT
1831  006222  022626          3$:     CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1832  006224  000207                          RTS     %7                      ;RETURN (FROM JSR TO TPCORX) TO
1833                                                                          ;TEST NEXT PATTERN
1834
1835
1836
1837                                  ;*************************************************************************
1838                                  ;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY FROM 4K TO 28K
1839                                  ;WRITE WRONG PARITY AND READ IT BACK WITH ACTION ENABLE SET, MAKING
1840                                  ;SURE THAT A TRAP OCCURS, THEN WRITE GOOD PARITY AND MAKE SURE THAT
1841                                  ;NO TRAP OCCURS WHEN IT IS READ. MAKE SURE THAT THE ERROR ADDRESS BITS
1842                                  ;(PARITY REGISTER BITS 5-11) ARE CORRECT.
1843                                  ;*************************************************************************
1844  006226  104001          TEST16: SCOPE
1845  006230  012737  000016  177570          MOV     #16,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
1846  006236  012737  006604  000114          MOV     #TRP16,@#PARVEC         ;SET UP TRAP RETURN
1847  006244  012767  000002  172246          MOV     #2,BITPT                ;INIT 4K BIT POINTER TO BANK 1
1848  006252  012767  020000  172236          MOV     #20000,ADRPT            ;INIT MEMORY STARTING ADDRESS
1849  006260  036767  172234  172604  1$:     BIT     BITPT,PMEML             ;DOES THIS 4K HAVE PARITY?
1850  006266  001012                          BNE     3$                      ;YES, BRANCH TO TEST IT
1851  006270  062767  020000  172220  2$:     ADD     #20000,ADRPT            ;NO, UPDATE MEMORY ADDRESS BY 4K
1852  006276  006367  172216                  ASL     BITPT                   ;UPDATE BIT POINTER
1853  006302  022767  000200  172210          CMP     #200,BITPT              ;THIS 28K DONE?
1854  006310  003363                          BGT     1$                      ;NO, CHECK NEXT 4K
1855  006312  000421                          BR      DONE16                  ;YES, EXIT
1856  006314  016767  172170  172206  3$:     MOV     ADRPT,HIADR             ;SET UPPER LIMIT THIS 4K
1857  006322  062767  020000  172200          ADD     #20000,HIADR
1858  006330  004767  005030                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1859  006334  016705  172156                  MOV     ADRPT,R5
1860  006340  005025                  5$:     CLR     (5)+                    ;CLEAR BANK UNDER TEST
1861  006342  020567  172162                  CMP     R5,HIADR
1862  006346  103774                          BLO     5$
1863  006350  004767  000020          6$:     JSR     %7,WRP16                ;GO WRITE WRONG PARITY IN EACH BYTE
1864  006354  000745                          BR      2$                      ;UPDATE AND CHECK NEXT 4K
1865  006356  004767  005002          DONE16: JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS IF DONE
1866  006362  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
1867  006370  000167  000472                  JMP     TEST17                  ;GO TO NEXT TEST
```

```
1868
1869                                    ;WRITE WRONG PARITY TEST ROUTINE - TESTS EACH BYTE IN 4K
1870                                    ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
1871                                    ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
1872  006374  016705  172116          WWP16$: MOV     ADRPT,R5                ;SET TEST ADDRESS POINTER
1873  006400  005067  172154                  CLR     ODDFLG                  ;INDICATE TESTING LOW BYTE
1874  006404  012767  125253  172122          MOV     #125253,SHDBE           ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
1875  006412  012715  125253          WWP16A$: MOV    #125253,@R5             ;INITIALIZE TEST LOCATION
1876  006416  012701  000570                  MOV     #MPR0,R1                ;SETUP TO LOAD PARITY REGISTERS
1877  006422  032711  000001          1$:     BIT     #1,(1)
1878  006426  001003                          BNE     .+10
1879  006430  012771  000005  000000          MOV     #WWP+AE,@(R1)           ;SET WRITE WRONG PARITY AND ACTION
1880                                                                          ;ENABLE IF THIS PARITY REGISTER
1881                                                                          ;IS PRESENT
1882  006436  062701  000010                  ADD     #10,R1
1883  006442  020127  000770                  CMP     R1,#TREG
1884  006446  103765                          BLO     1$
1885  006450  005767  172104                  TST     ODDFLG                  ;WRITING HIGH BYTE?
1886  006454  100425                          BMI     2$                      ;YES, BRANCH
1887  006456  112715  000253                  MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
1888  006462  012701  000570          5$:     MOV     #MPR0,R1                ;THIS CODE CLEARS WWP BIT IN ALL
1889  006466  032711  000001                  BIT     #1,(1)                  ;PARITY REGISTERS
1890  006472  001003                          BNE     .+10                    ;
1891  006474  042771  000004  000000          BIC     #WWP,@(R1)
1892  006502  062701  000010                  ADD     #10,R1
1893  006506  020127  000770                  CMP     R1,#TREG
1894  006512  103765                          BLO     5$+4
1895  006514  005767  172040                  TST     ODDFLG                  ;TESTING HIGH OR LOW BYTE?
1896  006520  100407                          BMI     6$                      ;BRANCH,IF HIGH BYTE
1897  006522  142715  000377                  BICB    #377,@R5                ;DETECT WRONG PARITY WITH DATIP-
1898                                                                          ;SHOULD TRAP TO TRP16 BEFORE DOING THE DATOB
1899  006526  000407                          BR      3$
1900  006530  112765  000252  000001  2$:     MOVB    #252,1(R5)              ;WRITE WRONG PARITY IN HIGH BYTE
1901  006536  000751                          BR      5$
1902  006540  142765  000377  000001  6$:     BICB    #377,1(R5)              ;DETECT WRONG PARITY WITH DATIP-
1903                                                                          ;SHOULD TRAP TO TRP16 BEFORE DOING THE DATOB
1904  006546  012701  000570          3$:     MOV     #MPR0,R1                ;NO TRAP OCCURRED- SETUP TO CLEAR
1905                                                                          ;AE AND WWP
1906  006552  032711  000001          4$:     BIT     #1,@R1
1907  006556  001003                          BNE     .+10
1908  006560  042771  000005  000000          BIC     #WWP+AE,@(R1)           ;CLEAR AE AND WWP IN ALL PARITY REGISTERS
1909  006566  062701  000010                  ADD     #10,R1
1910  006572  020127  000770                  CMP     R1,#TREG
1911  006576  103765                          BLO     4$
1912  006600  104002                          ERROR                           ;ERROR, NO TRAP AFTER WRITING AND
1913                                                                          ;READING WRONG PARITY IN LOCATION
1914                                                                          ;WHOSE ADDRESS IS IN R5 (AE AND WWP
1915                                                                          ;WERE SET IN ALL PARITY REGISTERS)
1916                                                                          ;TESTING LOW BYTE IF ODDFLG IS POSITIVE
1917                                                                          ;TESTING HIGH BYTE IF ODDFLG IS NEGATIVE
1918                                                                          ;NOTE THAT AE AND WWP WERE CLEARED
1919                                                                          ;BEFORE TYPING THE ERROR PRINTOUT
1920  006602  000517                          BR      CN16
1921
```



```
1922                                    ;WHEN WRONG PARITY DATA IS READ BACK SHOULD ENTER HERE VIA TRAP TO 114
1923  006604  012701  000570          TRP16$: MOV     #MPR0,R1                ;PARITY TRAP OCCURRED- FIRST CLEAR
1924  006610  032711  000001          TRP16A$: BIT    #1,@R1                  ;WWP AND AE IN ALL REGISTERS
1925  006614  001003                          BNE     .+10
1926  006616  042771  000005  000000          BIC     #AE+WWP,@(R1)
1927  006624  062701  000010                  ADD     #10,R1
1928  006630  020127  000770                  CMP     R1,#TREG
1929  006634  103765                          BLO     TRP16A
1930  006636  012701  000570                  MOV     #MPR0,R1                ;FIND THE REGISTER THAT SENSED THE ERROR
1931  006642  012703  000772                  MOV     #INDCO,R3
1932  006646  005067  172116                  CLR     TREG
1933  006652  032711  000001          1$:     BIT     #1,@R1                  ;DOES THIS CONTROL EXIST?
1934  006656  001017                          BNE     2$                      ;NO, BRANCH
1935  006660  005771  000000                  TST     @(R1)                   ;YES- IS ERROR SET?
1936  006664  100014                          BPL     2$                      ;NO, BRANCH
1937  006666  005767  172076                  TST     TREG                    ;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
1938  006672  001401                          BEQ     .+4                     ;NO- BRANCH
1939  006674  104002                          ERROR                           ;ERROR SET IN MORE THAN ONE PARITY REGISTER
1940                                                                          ;AFTER WRITING WRONG PARITY IN LOCATION
1941                                                                          ;WHOSE ADDRESS IS IN R5
1942  006676  036761  171616  000002          BIT     BITPT,2(R1)             ;DOES MAP INDICATE THIS PARITY REGISTER
1943                                                                          ;CONTROLS THIS MEMORY?
1944  006704  001001                          BNE     .+4                     ;YES, BRANCH
1945  006706  104002                          ERROR                           ;PARITY REGISTER RESPONDED TO MEMORY
1946                                                                          ;NOT INCLUDED IN ITS MAP
1947                                                                          ;PARITY REGISTER'S ADDRESS IS POINTED
1948                                                                          ;TO BY R1, ADDRESS OF LOCATION CAUSING
1949                                                                          ;PARITY ERROR IS IN R5
1950  006710  010304                          MOV     R3,R4
1951  006712  011167  172052                  MOV     @R1,TREG                ;STORE REGISTER ADDRESS
1952  006716  062701  000010          2$:     ADD     #10,R1
1953  006722  005723                          TST     (R3)+
1954  006724  020127  000770                  CMP     R1,#TREG
1955  006730  103750                          BLO     1$                      ;BRANCH UNTIL ALL THE PARITY
1956                                                                          ;REGISTERS HAVE BEEN CHECKED
1957  006732  011567  171600                  MOV     (5),WAS                 ;SAVE DATA FROM LOCATION UNDER TEST
1958  006736  022715  125253                  CMP     #125253,@R5             ;DID BICB CHANGE DATA?
1959  006742  001401                          BEQ     .+4                     ;NO, CONTINUE
1960  006744  104003                          ERRORP                          ;DATA WAS MODIFIED BY THE BICB WHICH
1961                                                                          ;GOT A PARITY ERROR TRAP- SINCE PARITY ERROR
1962                                                                          ;TRAP OCCURRED, CONTENTS SHOULD NOT HAVE
1963                                                                          ;BEEN MODIFIED, R5 CONTAINS ADDRESS
1964                                                                          ;OF TEST LOCATION, "TREG" CONTAINS
1965                                                                          ;ADDRESS OF PARITY REGISTER SENSING
1966                                                                          ;ERROR
1967  006746  005767  172016                  TST     TREG                    ;WAS PARITY ERROR SET IN ANY REGISTERS?
1968  006752  001002                          BNE     3$                      ;YES- BRANCH
1969  006754  104002                          ERROR                           ;PARITY TRAP OCCURRED ON READING
1970                                                                          ;WRONG PARITY (WITH AE SET) BUT NO
1971                                                                          ;REGISTERS HAD PARITY ERROR BIT SET.
1972                                                                          ;R5 CONTAINS THE ADDRESS OF THE
1973                                                                          ;TEST LOCATION.
1974  006756  000420                          BR      4$
1975  006760  022714  000001          3$:     CMP     #1,(R4)
```

```
                        BNE     4$                      
006766  017701  171776         MOV     @TREG,R1        ;GET PARITY REGISTER CONTENTS
006772  042701  170037         BIC     #170037,R1      ;MASK OFF ALL BUT ERROR ADDRESS BITS
006776  010502                 MOV     R5,R2           ;GET ADDRESS OF LOCATION UNDER TEST
007000  042702  003777         BIC     #3777,R2        ;POSITION BITS IN R2
007004  000302                 SWAB    R2
007006  006302                 ASL     R2
007010  006302                 ASL     R2
007012  020102                 CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
007014  001401                 BEQ     .+4
007016  104003                 ERRORP                  ;ERROR ADDRESS BITS (PARITY REGISTER
                                                       ;BITS 11-5) ARE INCORRECT. R5 CONTAINS
                                                       ;THE ADDRESS OF THE TEST LOCATION.
                                                       ;"TREG" CONTAINS THE ADDRESS OF THE
                                                       ;PARITY REGISTER DETECTING THE ERROR
007020  022626         4$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
007022  011515         CNT16:  MOV     (5),(5)         ;RESTORE TEST LOCATION TO FIX BAD PARITY
007024  005077  171740         CLR     @TREG           ;CLEAR ERROR BIT IN PARITY REGISTER
007030  005715                 TST     @R5             ;READ LOCATION TO SEE IF PARITY IS GOOD
007032  005777  171732         TST     @TREG           ;IS PARITY ERROR SET?
007036  100001                 BPL     .+4             ;NO- BRANCH
007040  104002                 ERROR                   ;WRITING LOCATION WITH WRITE WRONG PARITY
                                                       ;CLEAR DIDN'T CLEAR BAD PARITY
                                                       ;R5 CONTAINS ADDRESS OF THE TEST
                                                       ;LOCATION. "TREG" CONTAINS THE ADDRESS
                                                       ;OF THE PARITY REGISTER DETECTING
                                                       ;THE ERROR
007042  005167  171512         CN16:   COM     ODDFLG          ;TOGGLE BYTE INDICATOR
007046  100401                 BMI     .+4             ;BRANCH IF READY TO TEST HIGH BYTE
007050  005725                 TST     (5)+            ;UPDATE ADDRESS POINTER
007052  020567  171452         CMP     R5,HIADR        ;THIS 4K DONE?
007056  103401                 BLO     1$              ;NO, TEST NEXT LOCATION
007060  000207                 RTS     %7              ;RETURN TO TEST NEXT BANK
007062  000167  177324         1$:     JMP     WWP16A
                        ;*****************************************************************
                        ;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY ABOVE 28K
                        ;WRITE WRONG PARITY AND READ IT WITH ACTION ENABLE SET, MAKING SURE
                        ;THAT A TRAP OCCURS, THEN WRITE AND READ THE SAME LOCATION WITH GOOD PARITY
                        ;(USING SAME DATA) TO SHOW THAT THE PARITY BIT TOGGLES. MAKE SURE THAT
                        ;THE ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11) ARE CORRECT.
                        ;*****************************************************************
007066  104001         TEST17: SCOPE
007070  012737  000017  177570         MOV     #17,@#DISPLY    ;LOAD TEST NUMBER INTO THE DISPLAY
007076  005767  171450         TST     NOKT            ;KT11 PRESENT?
007102  001402                 BEQ     .+6             ;YES, BRANCH
007104  000167  000636         JMP     XFR1            ;NO, SKIP TO NEXT TEST
007110  004767  004250         JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
007114  012737  007456  000114         MOV     #TRP17,@#PARVEC ;SETUP FOR PARITY TRAP
007122  012767  000200  171370         MOV     #200,BITPT      ;INITIALIZE 4K BIT POINTER
007130  004767  004130         JSR     %7,NRALL        ;INITIALIZE ALL PAGES TO NON-RESIDENT
007134  004767  004274         JSR     %7,MAP1         ;MAP KERNEL 0 TO BANK 0,RW; KERNEL 7
```



```
                                                       ;TO THE EXTERNAL BANK, RW, TURN ON
                                                       ;THE KT11 AND MAKE KERNEL 1 RW
007140  012777  001600  172074         MOV     #1600,@KPAR1    ;INITIALIZE KERNEL PAGE 1 TO 28K
007146  005067  171404         CLR     LOWFLG          ;CLEAR FLAG TO INDICATE TESTING LOWER 64K
007152  016767  171714  171716         MOV     PMEML,PMEMX
007160  012767  000002  000366         MOV     #2,INDX17       ;SETUP OFFSET TO CHECK LOWER 64K OF MAPS
007166  036767  171326  171702  1$:     BIT     BITPT,PMEMX     ;DOES THIS 4K HAVE PARITY?
007174  001025                 BNE     3$              ;YES, BRANCH TO TEST IT
007176  062777  000200  172036  2$:     ADD     #200,@KPAR1     ;NO, MAP TO NEXT 4K
007204  006367  171310         ASL     BITPT           ;UPDATE BIT POINTER
007210  103366                 BCC     1$              ;GO CHECK TO SEE IF THIS 4K HAS PARITY
007212  005767  171340         TST     LOWFLG          ;END OF 128K?
007216  001025                 BNE     DONE17          ;YES, EXIT
007220  005267  171332         INC     LOWFLG          ;NO, SET FLAG TO INDICATE SHIFT TO
007224  016767  171644  171644         MOV     PMEMH,PMEMX     ;HIGH 64K AND CHANGE MAPS
007232  012767  000001  171620         MOV     #1,BITPT
007240  012767  000004  000306         MOV     #4,INDX17       ;SETUP OFFSET TO CHECK UPPER 64K OF MAPS
007246  000747                 BR      1$
007250  012705  020000         3$:     MOV     #20000,R5
007254  005025         5$:     CLR     (5)+            ;CLEAR BANK UNDER TEST
007256  020527  040000         CMP     R5,#40000
007262  103774                 BLO     5$
007264  004767  000020         6$:     JSR     %7,WWP17        ;GO WRITE WRONG PARITY AND CHECK IT
007270  000742                 BR      2$              ;UPDATE AND CHECK NEXT 4K
007272  005037  177572         DONE17: CLR     @#SR0           ;TURN OFF KT11 WHEN DONE
007276  012737  000116  000114         MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAP CATCHER
007304  000167  000436         JMP     XFR1            ;GO TO SETUP FOR NEXT TEST
                        ;WRITE WRONG PARITY TEST ROUTINE TO TEST MEMORY ABOVE 28K
007310  012705  020000         WWP17:  MOV     #20000,R5       ;SET TEST ADDRESS POINTER
007314  005067  171240         CLR     ODDFLG          ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
007320  012767  125253  171206         MOV     #125253,SHDBE   ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
007326  012715  125253         WWP17A: MOV     #125253,@R5     ;INITIALIZE LOCATION
007332  012701  000570         MOV     #MPR0,R1        ;INITIALIZE REGISTER ADDRESS POINTER
007336  032711  000001         1$:     BIT     #1,(1)          ;DOES THIS CONTROL EXIST?
007342  001003                 BNE     .+10            ;NO, GET NEXT
007344  012771  000005  000000         MOV     #WWP+AE,@(R1)   ;YES- SET WRITE WRONG PARITY
                                                       ;AND ACTION ENABLE
007352  062701  000010         ADD     #10,R1
007356  020127  000770         CMP     R1,#TREG        ;ALL REGISTERS SETUP?
007362  103765                 BLO     1$              ;NO- LOOP
007364  005767  171170         TST     ODDFLG          ;YES- TESTING HIGH BYTE?
007370  100405                 BMI     2$              ;YES, BRANCH
007372  112715  000253         MOVB    #253,@R5        ;NO, WRITE WRONG PARITY IN LOW BYTE
007376  142715  000377         BICB    #377,@R5        ;DETECT WRONG PARITY WITH DATIP-
                                                       ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
007402  000406                 BR      3$
007404  112765  000252  000001  2$:     MOVB    #252,1(R5)      ;WRITE WRONG PARITY IN HIGH BYTE
007412  142765  000377  000001         BICB    #377,1(R5)      ;DETECT WRONG PARITY WITH DATIP
                                                       ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
007420  012701  000570         3$:     MOV     #MPR0,R1        ;IF NO TRAP, CLEAR AE AND WWP IN ALL
007424  032711  000001         4$:     BIT     #1,@R1          ;PARITY REGISTERS
007430  001003                 BNE     .+10
007432  042771  000005  000000         BIC     #AE+WWP,@(R1)
```

```
2084  007440  062701  000010                  ADD     #10,R1
2085  007444  020127  000770                  CMP     R1,#TREG
2086  007450  103765                          BLO     4$
2087  007452  104002                          ERROR                   ;NO TRAP AFTER WRITING AND READING
2088                                                                  ;WRONG PARITY WITH AE SET- VIRTUAL
2089                                                                  ;ADDRESS OF LOCATION IS IN R5
2090                                                                  ;(MAPPED THRU KERNEL PAGE 1)
2091                                                                  ;WROTE LOW BYTE IF ODDFLG IS POSITIVE
2092                                                                  ;WROTE HIGH BYTE IF IT IS NEGATIVE
2093                                                                  ;NOTE THAT WWP AND AE WERE CLEARED
2094                                                                  ;BEFORE ERROR PRINTOUT
2095  007454  000512                          BR      CNT17
2096
2097                                  ;WHEN WRONG PARITY DATA IS READ BACK, SHOULD ENTER HERE VIA TRAP TO 114
2098  007456  012701  000570          TRP17:  MOV     #MPR0,R1        ;PARITY TRAP OCCURRED- BEFORE CHECKING
2099  007462  032711  000001          TRP17A: BIT     #1,@R1          ;IT, CLEAR WWP AND AE IN ALL REGISTERS
2100  007466  001003                          BNE     .+10
2101  007470  042771  000005  000000        BIC     #AE+WWP,@(R1)
2102  007476  062701  000010                  ADD     #10,R1
2103  007502  020127  000770                  CMP     R1,#TREG
2104  007506  103765                          BLO     TRP17A
2105  007510  012701  000570                  MOV     #MPR0,R1        ;FIND REGISTER THAT SENSED THE ERROR
2106  007514  012703  000772                  MOV     #INDC0,R3               ;SETUP PTR TO INDICATOR
2107  007520  005067  171244                  CLR     TREG
2108  007524  032711  000001          1$:     BIT     #1,@R1          ;DOES THIS CONTROL EXIST?
2109  007530  001017                          BNE     2$              ;NO, BRANCH
2110  007532  005771  000000                  TST     @(R1)           ;YES- IS ERROR SET?
2111  007536  100014                          BPL     2$              ;NO, BRANCH
2112  007540  005767  171224                  TST     TREG            ;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
2113  007544  001401                          BEQ     .+4             ;NO- BRANCH
2114  007546  104002                          ERROR                   ;ERROR SET IN MORE THAN ONE PARITY REGISTER
2115                                                                  ;AFTER READING WRONG PARITY IN LOCATION
2116                                                                  ;WHOSE VIRTUAL ADDRESS IS IN R5
2117  007550  036761  170744  000002        BIT     BITPT,2(R1)     ;DOES MAP INDICATE THAT THIS REGISTER
2118                                                                  ;CONTROLS THIS MEMORY?
2119          007554                          INDX17=.-2
2120  007556  001001                          BNE     .+4             ;YES- BRANCH
2121  007560  104002                          ERROR                   ;PARITY REGISTER RESPONDED TO MEMORY
2122                                                                  ;NOT INCLUDED IN ITS MAP, R1 POINTS TO
2123                                                                  ;THE PARITY REGISTER'S ADDRESS, R5
2124                                                                  ;CONTAINS VIRTUAL ADDRESS OF THE LOCATION
2125                                                                  ;BEING TESTED (MAPPED THRU KERNEL
2126                                                                  ;PAGE 1)
2127  007562  011167  171202                  MOV     @R1,TREG        ;STORE REGISTER ADDRESS
2128  007566  010304                          MOV     R3,R4           ;STORE PTR TO INDICATOR
2129  007570  062701  000010          2$:     ADD     #10,R1
2130  007574  005723                          TST     (R3)+           ;INCREMENT PTR TO INDICATOR
2131  007576  020127  000770                  CMP     R1,#TREG
2132  007602  103750                          BLO     1$              ;LOOP UNTIL ALL THE PARITY REGISTERS
2133                                                                  ;HAVE BEEN CHECKED
2134  007604  011567  170726                  MOV     (5),WAS         ;SAVE CONTENTS OF LOCATION UNDER TEST
2135  007610  022715  125253                  CMP     #125253,@R5     ;DID BICB CHANGE DATA?
2136  007614  001401                          BEQ     .+4             ;NO, CONTINUE
2137  007616  104002                          ERROR                   ;DATA WAS MODIFIED BY THE BICB WHICH
```



```
2138                                                                  ;GOT A PARITY ERROR TRAP- SINCE PARITY
2139                                                                  ;TRAP OCCURRED, CONTENTS SHOULD NOT
2140                                                                  ;HAVE BEEN MODIFIED, R5 CONTAINS TEST
2141                                                                  ;LOCATION ADDRESS (VIRTUAL, MAPPED THRU
2142                                                                  ;KERNEL PAGE 1), "TREG" CONTAINS ADDRESS OF
2143                                                                  ;PARITY REGISTER DETECTING PARITY ERROR
2144  007620  005767  171144                  TST     TREG            ;WAS PARITY ERROR SET IN ANY REGISTER?
2145  007624  001003                          BNE     3$              ;YES- BRANCH
2146  007626  104002                          ERROR                   ;PARITY TRAP OCCURRED ON READING
2147                                                                  ;WRONG PARITY WITH AE SET BUT NO
2148                                                                  ;REGISTER HAS THE PARITY ERROR BIT
2149                                                                  ;SET. R5 CONTAINS VIRTUAL ADDRESS
2150                                                                  ;OF THE TEST LOCATION (MAPPED THRU
2151                                                                  ;KERNEL PAGE 1)
2152  007630  000423                          BR      4$
2153  007632  001022                          BNE     4$              ;NO, BRANCH, DO NOT CHECK THE
2154  007634  022714  000001          3$:     CMP     #1,(R4)         ;IS THIS A CORE PARITY REGISTER?
2155  007640  001017                          BNE     4$              ;NO, BRANCH, DO NOT CHECK THE
2156                                                                  ;ADDRESS BITS OF THE LAST PARITY
2157                                                                  ;ERROR
2158                                                                  ;IF CORE REG, CHECK IF ADDRESS
2159                                                                  ;BITS OF LAST PAR ERR WERE STORED
2160  007642  017701  171122                  MOV     @TREG,R1        ;GET CONTENTS OF REGISTER DETECTING PARITY ERROR
2161  007646  042701  170037                  BIC     #170037,R1      ;MASK OFF ALL BUT ADDRESS BITS
2162  007652  010502                          MOV     R5,R2           ;CALCULATE TOP 7 BITS OF ERROR ADDRESS
2163  007654  042702  163777                  BIC     #163777,R2
2164  007660  000302                          SWAB    R2
2165  007662  006302                          ASL     R2
2166  007664  006302                          ASL     R2
2167  007666  067702  171350                  ADD     @KPAR1,R2
2168  007672  020102                          CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
2169  007674  001401                          BEQ     .+4
2170  007676  104004                          ERROR5                  ;PARITY ERROR ADDRESS BITS (PARITY
2171                                                                  ;REGISTER BITS 11-5) INCORRECT
2172                                                                  ;"TREG" CONTAINS ADDRESS OF PARITY
2173                                                                  ;REGISTER, R5 CONTAINS THE VIRTUAL
2174                                                                  ;ADDRESS OF THE TEST LOCATION
2175                                                                  ;(MAPPED THRU KERNEL PAGE 1)
2176  007700  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
2177  007702  011515                  CNT17:  MOV     (5),(5)         ;RESTORE TEST LOCATION TO FIX BAD PARITY
2178  007704  005077  171060                  CLR     @TREG           ;CLEAR ERROR BIT IN THE PARITY REGISTER
2179  007710  005715                          TST     @R5             ;READ LOCATION TO SEE IF PARITY IS GOOD
2180  007712  005777  171052                  TST     @TREG           ;IS PARITY ERROR SET?
2181  007716  100001                          BPL     .+4             ;NO, BRANCH
2182  007720  104002                          ERROR                   ;WRITING LOCATION WITH WRITE WRONG
2183                                                                  ;PARITY CLEAR DIDN'T CLEAR BAD PARITY
2184                                                                  ;"TREG" CONTAINS THE ADDRESS OF THE
2185                                                                  ;PARITY REGISTER, R5 CONTAINS THE
2186                                                                  ;VIRTUAL ADDRESS OF THE TEST LOCATION
2187                                                                  ;(MAPPED THRU KERNEL PAGE 1)
2188  007722  005167  170632                  COM     ODDFLG          ;TOGGLE BYTE INDICATOR
2189  007726  100401                          BMI     .+4             ;BRANCH IF HIGH BYTE NOT YET TESTED
2190  007730  005725                          TST     (5)+            ;UPDATE ADDRESS POINTER
2191  007732  020527  040000                  CMP     R5,#40000       ;THIS 4K DONE?
```

```
2192  007736  103401                          BLO     1$                      ;NO,TEST NEXT LOCATION
2193  007740  000207                          RTS     %7                      ;YES, RETURN TO CHECK FOR NEXT
2194                                                                          ;BANK TO BE TESTED
2195  007742  000167  177360          1$:     JMP     WWP17A                  ;GO AND TEST NEXT LOCATION
2196  007746  104001                  XFR1:   SCOPE
2197                                  ;IF THE FIRST BANK (BANK 0) IS PARITY MEMORY, SETUP TO TEST IT
2198                                  ;COPY THE FIRST 4K TO THE SECOND 4K, MOVE THE STACK POINTER TO BANK 1,
2199                                  ;AND THEN JUMP TO THE COPY OF TEST20 IN BANK 1
2200  007750  032767  000001  171114          BIT     #1,PMEML                ;IS FIRST 4K PARITY?
2201  007756  001002                          BNE     .+6                     ;BRANCH IF YES
2202  007760  000167  001254                  JMP     DONE                    ;NO- DONE WITH TEST
2203  007764  032767  000002  171074          BIT     #2,MEML                 ;IS THERE A SECOND 4K(BANK 1)?
2204  007772  001002                          BNE     .+6                     ;YES, BRANCH
2205  007774  000167  001240                  JMP     DONE                    ;NO, EXIT
2206  010000  005037  177776                  CLR     @#PS                    ;CLEAR STATUS REGISTER
2207  010004  004767  003354                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2208  010010  012700  010000                  MOV     #10000,R0               ;R0 IS COUNTER TO MOVE 4K
2209  010014  005001                          CLR     R1                      ;R1 POINTS TO LOCATION IN BANK 0
2210  010016  011161  020000          1$:     MOV     @R1,20000(R1)           ;COPY FROM BANK 0 TO BANK 1
2211  010022  005721                          TST     (R1)+                   ;MOVE POINTER
2212  010024  005300                          DEC     R0                      ;DONE WITH 4K?
2213  010026  001373                          BNE     1$                      ;NO- BRANCH
2214  010030  062706  020000                  ADD     #20000,SP               ;YES, MOVE STACK POINTER TO POINT TO BANK 1
2215  010034  012767  030114  025764          MOV     #TEST20+20010,RETURN+20000      ;UPDATE SCOPE RETURN IN BANK 1
2216  010042  062767  020000  023144          ADD     #20000,ERRA1+20000      ;UPDATE ADDRESSES USED IN ERROR TYPEOUT
2217  010050  062767  020000  023140          ADD     #20000,ERRA2+20000
2218  010056  062767  020000  023142          ADD     #20000,ERRA3+20000
2219  010064  062767  020000  023136          ADD     #20000,ERRA4+20000
2220  010072  062767  020000  023140          ADD     #20000,ERRA5+20000
2221  010100  000167  020010                  JMP     TEST20+20010            ;GO TO TEST 20 IN BANK 1
2222
2223
2224
2225                                  ;*********************************************************************
2226                                  ;IF FIRST 4K IS PARITY MEMORY, CHECK IT WITH A SERIES OF PATTERNS
2227                                  ;THIS SUBTEST IS RUN IN BANK 1 (20000 ABOVE THE ADDRESSES IN THE LISTING)
2228                                  ;*********************************************************************
2229  010104  013746  177776  TEST20: MOV     @#PS,-(SP)              ;THESE 2 LINES DO THE SAME AS A SCOPE WITHOUT
2230  010110  004767  005614                  JSR     PC,SCOPEC               ;USING AN EMT
2231  010114  012737  000020  177570          MOV     #20,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
2232  010122  004767  003236                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2233  010126  012704  021040                  MOV     #PARPAT+20000,R4        ;INITIALIZE PATTERN POINTER
2234  010132  005005                          CLR     R5
2235  010134  005025          1$:     CLR     (5)+                    ;INITIALLY CLEAR BANK 0
2236  010136  020527  020000                  CMP     R5,#20000
2237  010142  103774                          BLO     1$
2238  010144  012737  030352  000114          MOV     #TRP20+20000,@#PARVEC   ;SETUP TRAP RETURN
2239  010152  012701  020570                  MOV     #MPR0+20000,R1          ;SETUP TO SET ACTION ENABLE IN ALL
2240                                                                          ;PARITY REGISTERS PRESENT
2241  010156  032711  000001          2$:     BIT     #1,@R1
2242  010162  001003                          BNE     .+10
2243  010164  012771  000001  000000          MOV     #AE,@(R1)               ;SET ACTION ENABLE IF REGISTER IS PRESENT
2244  010172  062701  000010                  ADD     #10,R1
2245  010176  020127  020770                  CMP     R1,#TREG+20000
```



```
2246  010202  103765                          BLO     2$                      ;BRANCH UNTIL ALL REGISTER ADDRESSES
2247                                                                          ;HAVE BEEN CHECKED
2248  010204  004767  000022          3$:     JSR     %7,CKBK0                ;EXERCISE THIS 4K
2249  010210  005726                          TST     (6)+                    ;UPDATE PATTERN
2250  010212  005714                          TST     (4)                     ;LAST PATTERN?
2251  010214  001373                          BNE     3$                      ;NO, LOOP
2252  010216  004767  003142          DONE20: JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2253  010222  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
2254  010230  000526                          BR      TEST21                  ;GO TO NEXT TEST
2255
2256                                  ;PARITY MEMORY TEST ROUTINE
2257                                  ;WRITES AND CHECKS EACH LOCATION IN BANK 0 (EXCEPT 114 AND 116)
2258                                  ;WITH VALUE POINTED TO BY R4
2259  010232  005005          CKBK0:  CLR     R5                      ;SET ADDRESS POINTER
2260  010234  011415          1$:     MOV     (4),(5)                 ;WRITE PATTERN
2261  010236  021415                  CMP     (4),(5)                 ;DATA OK?
2262  010240  001404                  BEQ     .+12                    ;YES- BRANCH
2263  010242  013746  177776                  MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
2264  010246  004767  002672                  JSR     PC,ERR                  ;ERROR- DATA INCORRECT IN LOCATION
2265                                                                          ;WHOSE ADDRESS IS IN R5. R4 POINTS
2266                                                                          ;TO THE VALUE WRITTEN.
2267  010252  005725                          TST     (5)+                    ;UPDATE ADDRESS POINTER
2268  010254  020527  000114                  CMP     R5,#114                 ;DON'T CHANGE CONTENTS OF 114 AND 116
2269  010260  001002                          BNE     .+6
2270  010262  062705  000004                  ADD     #4,R5
2271  010266  020527  020000                  CMP     R5,#20000               ;HAS THE WHOLE BANK BEEN TESTED WITH
2272                                                                          ;THIS PATTERN?
2273  010272  103760                          BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
2274  010274  005067  010470                  CLR     TREG+20000                      ;YES, DID ANY PARITY ERROR BITS SET?
2275  010300  012701  020570                  MOV     #MPR0+20000,R1
2276  010304  032711  000001          2$:     BIT     #1,(1)
2277  010310  001003                          BNE     .+10
2278  010312  005771  000000                  TST     @(R1)
2279  010316  100406                          BMI     3$                      ;YES- BRANCH
2280  010320  062701  000010                  ADD     #10,R1
2281  010324  020127  020770                  CMP     R1,#TREG+20000
2282  010330  103765                          BLO     2$
2283  010332  000207                          RTS     %7                      ;NO- RETURN
2284  010334  013746  177776          3$:     MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
2285  010340  004767  002600                  JSR     PC,ERR                  ;ERROR- PARITY ERROR BIT SET AND NO
2286                                                                          ;PARITY TRAP OCCURRED
2287                                                                          ;(AE WAS SET) - R1 POINTS TO ADDRESS
2288                                                                          ;OF PARITY REGISTER
2289  010344  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
2290  010346  000167  177644                  JMP     DONE20
2291
2292                                  ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
2293  010352  005067  010412  TRP20:  CLR     TREG+20000
2294  010356  012701  020570                  MOV     #MPR0+20000,R1          ;FIND THE REGISTER RECORDING A PARITY ERROR
2295  010362  032711  000001          1$:     BIT     #1,(R1)
2296  010366  001003                          BNE     .+10
2297  010370  005771  000000                  TST     @(R1)
2298  010374  100412                          BMI     2$                      ;BRANCH IF ERROR IS SET
2299  010376  062701  000010                  ADD     #10,R1
```

```
 2300  010402  020127  020770                CMP     R1,#TREG+20000
 2301  010406  103765                      BLO     1$
 2302  010410  013746  177776                MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA A JSR
 2303  010414  004767  002524                JSR     PC,ERR                  ;ERROR= PARITY TRAP TO 114 OCCURRED DURING
 2304                                                                        ;TEST 20 BUT NO REGISTERS HAVE PARITY
 2305  010420  000430                      BR      5$                      ;ERROR SET
 2306  010422  017102  000000          2$:   MOV     @(R1),R2                ;SAVE CONTENTS OF PARITY REGISTER
 2307  010426  005003                      CLR     R3                      ;INITIALIZE ADDRESS POINTER
 2308  010430  005071  000000          3$:   CLR     @(R1)                   ;CLEAR PARITY REGISTER
 2309  010434  005713                      TST     @R3                     ;READ LOCATION
 2310  010436  005771  000000                TST     @(R1)                   ;PARITY ERROR SET?
 2311  010442  100413                      BMI     4$                      ;YES= BRANCH
 2312  010444  005723                      TST     (R3)+                   ;MOVE POINTER
 2313  010446  020327  020000                CMP     R3,#20000
 2314  010452  103766                      BLO     3$
 2315  010454  010271  000000                MOV     R2,@(R1)                ;RESTORE PARITY REGISTER CONTENTS
 2316  010460  013746  177776                MOV     @#PS,-(SP)              ;SETUP TO CALL ERROR VIA JSR
 2317  010464  004767  002454                JSR     PC,ERR                  ;PARITY ERROR OCCURRED WHILE TESTING
 2318                                                                        ;BANK 0 BUT NO BAD PARITY WAS FOUND
 2319                                                                        ;DURING SCAN OF BANK 0, R1 POINTS
 2320                                                                        ;TO THE ADDRESS OF THE PARITY REGISTER
 2321                                                                        ;DETECTING THE ERROR
 2322  010470  000404                      BR      5$
 2323  010472  013746  177776          4$:   MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
 2324  010476  004767  002442                JSR     PC,ERR                  ;PARITY ERROR WHILE EXERCISING MEMORY
 2325                                                                        ;BANK 0= R1 POINTS TO ADDRESS OF PARITY
 2326                                                                        ;REGISTER DETECTING THE ERROR, R3 CONTAINS
 2327                                                                        ;THE ADDRESS OF THE LOCATION HAVING
 2328                                                                        ;BAD PARITY
 2329  010502  022626                5$:   CMP     (SP)+,(SP)+
 2330  010504  000207                      RTS     %7                      ;RETURN TO CHECK USING THE NEXT PATTERN
 2331
 2332
 2333
 2334                                ;*********************************************************************
 2335                                ;FORCE WRONG PARITY IN EACH LOCATION IN BANK 0
 2336                                ;NOTE THAT THIS SUBTEST IS EXECUTED IN BANK 1 (20000 ABOVE ADDRESSES
 2337                                ;IN THE LISTING), MAKE SURE THAT WRONG PARITY IN EACH BYTE CAN BE DETECTED,
 2338                                ;AND THAT WHEN GOOD PARITY IS WRITTEN AND READ NO PARITY ERROR IS DETECTED.
 2339                                ;CHECK ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11)
 2340                                ;*********************************************************************
 2341  010506  013746  177776          TEST21: MOV     @#PS,-(SP)              ;SAME AS SCOPE WITHOUT DOING EMT
 2342  010512  004767  005212                JSR     PC,SCOPEC
 2343  010516  012737  000021  177570        MOV     #21,@#DISPLY            ;LOAD TEST NUMBER INTO THE DISPLAY
 2344  010524  004767  002634                JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
 2345  010530  005005                      CLR     R5
 2346  010532  005025                1$:   CLR     (R5)+                   ;CLEAR PARITY ERRORS IN BANK 0
 2347  010534  020527  020000                CMP     R5,#20000
 2348  010540  103774                      BLO     1$
 2349
 2350                                ;WRITE WRONG PARITY TEST ROUTINE
 2351                                ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
 2352                                ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
 2353  010542  005005                WWP21:  CLR     R5                      ;SET TEST ADDRESS POINTER
```



```
 2354  010544  005067  170010                CLR     ODDFLG                  ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
 2355  010550  012767  125253  167756        MOV     #125253,SHDBE           ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
 2356  010556  012715  125253          WWP21A: MOV     #125253,@R5             ;INITIALIZE LOCATION
 2357  010562  012701  020570                MOV     #MPR0+20000,R1          ;SETUP TO SET WRITE WRONG PARITY
 2358                                                                        ;IN ALL REGISTERS
 2359  010566  032711  000001          1$:   BIT     #1,(1)
 2360  010572  001003                      BNE     .+10
 2361  010574  012771  000004  000000        MOV     #WWP,@(R1)              ;SET WRITE WRONG PARITY
 2362  010602  062701  000010                ADD     #10,R1
 2363  010606  020127  020770                CMP     R1,#TREG+20000
 2364  010612  103765                      BLO     1$
 2365  010614  005767  167740                TST     ODDFLG                  ;TESTING HIGH BYTE?
 2366  010620  100404                      BMI     2$                      ;YES, BRANCH
 2367  010622  112715  000253                MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
 2368  010626  005715                      TST     (R5)                    ;DETECT WRONG PARITY WITH DATI
 2369  010630  000405                      BR      3$
 2370  010632  112765  000252  000001  2$:   MOVB    #252,1(R5)              ;WRITE WRONG PARITY IN HIGH BYTE
 2371  010640  105765  000001                TSTB    1(R5)                   ;DETECT WRONG PARITY WITH DATI
 2372  010644  012701  020570          3$:   MOV     #MPR0+20000,R1          ;SETUP TO CLEAR WWP IN ALL PARITY REGISTERS
 2373  010650  012703  000772                MOV     #INDCO,R3               ;POINTER TO INDICATOR
 2374  010654  032711  000001          4$:   BIT     #1,@R1
 2375  010660  001003                      BNE     .+10
 2376  010662  042771  000004  000000        BIC     #WWP,@(R1)              ;CLEAR WWP IN ALL PARITY REGISTERS
 2377  010670  062701  000010                ADD     #10,R1
 2378  010674  020127  020770                CMP     R1,#TREG+20000
 2379  010700  103765                      BLO     4$
 2380  010702  012701  020570                MOV     #MPR0+20000,R1          ;CHECK TO SEE IF ANY REGISTER DETECTED
 2381  010706  005067  170056                CLR     TREG                    ;THE PARITY ERROR
 2382  010712  032711  000001          LOOP21: BIT     #1,@R1
 2383  010716  001024                      BNE     5$
 2384  010720  005771  000000                TST     @(R1)                   ;CHECK PARITY REGISTER
 2385  010724  100021                      BPL     5$                      ;BRANCH IF PARITY ERROR IS NOT SET
 2386  010726  005767  170036                TST     TREG                    ;WAS PARITY ERROR SET IN ANY OTHER REGISTER?
 2387  010732  001404                      BEQ     .+12                    ;NO= BRANCH
 2388  010734  013746  177776                MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA A JSR
 2389  010740  004767  002200                JSR     PC,ERR                  ;ERROR= PARITY ERROR SET IN MORE THAN
 2390                                                                        ;ONE REGISTER AFTER WRITING WRONG PARITY
 2391                                                                        ;IN LOCATION WHOSE ADDRESS IS IN R5
 2392  010744  032761  000001  000002        BIT     #1,2(R1)                ;DOES MAP INDICATE THAT THIS REGISTER
 2393                                                                        ;CONTROLS THIS MEMORY?
 2394  010752  001004                      BNE     .+12                    ;YES, BRANCH
 2395  010754  013746  177776                MOV     @#PS,-(SP)              ;NO= SETUP TO DO ERROR CALL VIA A JSR
 2396  010760  004767  002160                JSR     PC,ERR                  ;ERROR= PARITY REGISTER RESPONDED
 2397                                                                        ;TO MEMORY WHICH IS NOT IN ITS MAP.
 2398                                                                        ;R1 POINTS TO PARITY REGISTER'S ADDRESS,
 2399                                                                        ;R5 CONTAINS ADDRESS OF LOCATION BEING
 2400                                                                        ;TESTED
 2401  010764  011167  170000                MOV     @R1,TREG                ;STORE REGISTER ADDRESS
 2402  010770  062701  000010          5$:   ADD     #10,R1
 2403  010774  005723                      TST     (R3)+
 2404  010776  020127  020770                CMP     R1,#TREG+20000
 2405  011002  103743                      BLO     LOOP21                  ;BRANCH UNTIL ALL REGISTERS HAVE BEEN
 2406                                                                        ;CHECKED
 2407  011004  005767  167760                TST     TREG                    ;WAS PARITY ERROR SET IN ANY REGISTER?
```

```
2408  011010  001005                        BNE     6$                      ;YES- BRANCH
2409  011012  013746  177776                MOV     @#PS,-(SP)              ;NO- SETUP TO DO ERROR CALL VIA A JSR
2410  011016  004767  002122                JSR     PC,ERR                  ;ERROR- NO REGISTER HAS PARITY ERROR
2411                                                                          ;SET AFTER READING WRONG PARITY IN LOCATION
2412                                                                          ;WHOSE ADDRESS IS IN R5
2413  011022  000423                        BR      7$
2414  011024  022713  000001          6$:   CMP     #1,(R3)                 ;IS THIS A CORE PAR REG?
2415  011030  001020                        BNE     7$                      ;NO, BRANCH(MOS PAR REG DOES NOT
2416                                                                          ;STORE ADDR BITS OF PAR ERROR)
2417  011032  017701  167732                MOV     @TREG,R1                ;GET PARITY REGISTER CONTENTS
2418  011036  042701  170037                BIC     #170037,R1              ;MASK ALL BUT ERROR ADDRESS BITS
2419  011042  010502                        MOV     R5,R2                   ;GET ADDRESS OF LOCATION UNDER TEST
2420  011044  042702  003777                BIC     #3777,R2                ;POSITION BITS IN R2
2421  011050  000302                        SWAB    R2
```



```
2422  011052  006302                        ASL     R2
2423  011054  006302                        ASL     R2
2424  011056  020102                        CMP     R1,R2                   ;PARITY ERROR ADDRESS BITS CORRECT?
2425  011060  001404                        BEQ     7$                      ;BRANCH IF YES
2426  011062  013746  177776                MOV     @#PS,-(SP)              ;NO- SETUP TO DO ERROR CALL VIA A JSR
2427  011066  004767  002052                JSR     PC,ERR                  ;ERROR- ADDRESS BITS (PARITY REGISTER
2428                                                                          ;BITS 5-11) INCORRECT- ADDRESS OF PARITY
2429                                                                          ;REGISTER IS CONTAINED IN LOCATION "TREG"
2430                                                                          ;ADDRESS OF TEST LOCATION IS IN R5
2431  011072  011515                  7$:   MOV     @R5,@R5                 ;RESTORE TEST LOCATION TO FIX BAD PARITY
2432  011074  005077  167670                CLR     @TREG                   ;CLEAR ERROR BIT IN PARITY REGISTER
2433  011100  005715                        TST     @R5                     ;READ LOCATION TO SET IF PARITY IS GOOD
2434  011102  005777  167662                TST     @TREG                   ;CHECK PARITY ERROR BIT
2435  011106  100004                        BPL     .+12                    ;BRANCH IF NOT SET
2436  011110  013746  177776                MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA A JSR
2437  011114  004767  002024                JSR     PC,ERR                  ;ERROR- WRITING LOCATION WITH WRITE
2438                                                                          ;WRONG PARITY CLEAR DIDN'T CLEAR BAD
2439                                                                          ;PARITY (ADDRESS OF LOCATION IS IN R5)
2440                                                                          ;"TREG" +20000 CONTAINS THE ADDRESS
2441                                                                          ;OF THE PARITY REGISTER
2442  011120  005167  167434                COM     ODDFLG                  ;TOGGLE BYTE INDICATOR
2443  011124  100401                        BMI     .+4                     ;BRANCH IF HIGH BYTE NOT YET TESTED
2444  011126  005725                        TST     (R5)+                   ;UPDATE ADDRESS POINTER
2445  011130  020527  020000                CMP     R5,#20000               ;THIS 4K DONE?
2446  011134  103610                        BLO     WWP21A                  ;LOOP TILL ALL 4K HAS BEEN TESTED
2447  011136  004767  002222                JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2448  011142  013746  177776                MOV     @#PS,-(SP)              ;SETUP TO CALL SCOPE VIA JSR
2449  011146  004767  004556                JSR     PC,SCOPEC               ;SCOPE
2450
2451                                  ;COPY SECOND 4K BANK BACK TO FIRST 4K AND RETURN TO FIRST 4K BANK
2452  011152  012700  010000          XFR2: MOV     #10000,R0               ;R0 IS USED AS A COUNTER
2453  011156  005001                        CLR     R1                      ;R1 POINTS TO THE CURRENT LOCATION
2454  011160  016111  020000          1$:   MOV     20000(R1),@R1           ;COPY BANK 1 TO BANK 0
2455  011164  005721                        TST     (R1)+
2456  011166  005300                        DEC     R0
2457  011170  001373                        BNE     1$
2458  011172  162706  020000                SUB     #20000,SP               ;RESTORE STACK POINTER
2459  011176  162737  020000  013214        SUB     #20000,@#ERRA1
2460  011204  162737  020000  013216        SUB     #20000,@#ERRA2
2461  011212  162737  020000  013226        SUB     #20000,@#ERRA3
2462  011220  162737  020000  013230        SUB     #20000,@#ERRA4
2463  011226  162737  020000  013240        SUB     #20000,@#ERRA5
2464  011234  000137  011240                JMP     @#DONE                  ;RETURN TO BANK 0
2465
2466
2467
2468                                  ;*************************************************************************
2469                                  ;AT THIS POINT EXECUTION RETURNS TO BANK 0
2470                                  ;*************************************************************************
2471  011240  012737  000116  000114  DONE: MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
2472  011246  012706  000510                MOV     #STKPT,SP               ;REINITIALIZE STACK POINTER
2473  011252  004767  002106                JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2474  011256  005267  167262                INC     PASCNT                  ;KEEP TRACK OF PASSES COMPLETED
2475  011262  004567  003502                JSR     R5,DACNV
```

```
2476  011266  000544                        PASCNT
2477  011270  016733                        MPCNT
2478  011272  000006                        6
2479  011274  104000                        TYPE                            ;TYPE BELL, "END PASS#" AND PASS COUNT
2480  011276  016715                        MPGEND
2481  011300  013705  000042        LOGICAL:        MOV     @#42,R5         ;LOADED BY MONITOR?
2482  011304  001405                        BEQ     CONT                    ;BRANCH IF NO
2483  011306  000005                        RESET                           ;SETUP FOR MONITOR EXIT
2484  011310  004715                        JSR     7,(5)
2485  011312  000240                        NOP
2486  011314  000240                        NOP
2487  011316  000240                        NOP
2488  011320  032737  000400  177570  CONT: BIT     #BIT8,@#SR              ;SWITCH 8 SET?
2489  011326  001401                        BEQ     .+4
2490  011330  000000                        HALT                            ;HALT AT END OF PASS SET
2491  011332  105767  167550                TSTB    $TPFLG
2492  011336  001006                        BNE     1$                      ;IF NO TERMINAL, SKIP
2493  011340  105777  167534                TSTB    @TPS                    ;WAIT FOR TTY TO FINISH SO THAT RESET
2494  011344  100375                        BPL     .-4                     ;WON'T CLOBBER THE BELL
```



```
2495  011346  112777  000000  167526        MOVB    #0,@TPB                 ;OUTPUT A NULL
2496  011354  000167  170404          1$:   JMP     BEGIN
2497
2498
2499
2500
2501                                  ;*********************************************************************
2502                                  ;CREATE MAP INDICATING WHERE 4K BLOCKS OF MEMORY ARE PRESENT
2503                                  ;*********************************************************************
2504  011360  012737  011564  000004  MAPMEM: MOV     #MAPMB,@#4              ;SET NO MEM MANAGEMENT TRAP
2505  011366  005737  177572                TST     @#SR0                   ;IS KT PRESENT? (TIMEOUT IF NO)
2506
2507                                  ;MAP MEMORY USING KT11 = MAX OF 124K POSSIBLE
2508  011372  005067  167154        MAPMA: CLR     NOKT                    ;INDICATE KT11 PRESENT
2509  011376  004767  001662                JSR     %7,NRALL                ;INITIALLY SET ALL PAGES NONRESIDENT, BANK 0
2510  011402  004767  002026                JSR     %7,MAP1                 ;MAP KERNEL 0 TO BANK 0, RW
2511                                                                        ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
2512                                                                        ;MAP KERNEL 1 RW, AND TURN ON KT11
2513  011406  005067  167134                CLR     TBANK
2514  011412  012767  177777  167446        MOV     #177777,MEML            ;SET UP CORE MAPS
2515  011420  012767  077777  167442        MOV     #77777,MEMH
2516  011426  012767  000001  167064        MOV     #1,BITPT                ;SET UP 4K POINTER
2517  011434  012767  001066  167106        MOV     #MEML,MEMUT
2518  011442  012737  011552  000004        MOV     #5$,@#4                 ;SET UP FOR TIME OUTS
2519  011450  016777  167072  167564  2$:   MOV     TBANK,@KPAR1            ;MAP KERNEL PAGE 1 TO BANK BEING TESTED
2520  011456  005737  021000                TST     @#21000                 ;1ST K PRESENT?
2521  011462  005737  025000                TST     @#25000                 ;2ND K PRESENT?
2522  011466  005737  031000                TST     @#31000                 ;3RD K PRESENT
2523  011472  005737  035000                TST     @#35000                 ;4TH K PRESENT?
2524  011476  062767  000200  167042  3$:   ADD     #200,TBANK              ;UPDATE TEST ADDRESS
2525  011504  006367  167010                ASL     BITPT                   ;UPDATE BANK POINTER
2526  011510  103006                        BCC     4$                      ;BRANCH IF NOT DONE WITH 64K SECTION
2527  011512  012767  000001  167000        MOV     #1,BITPT                ;YES, DO MEMH(64-124K)
2528  011520  012767  001070  167022        MOV     #MEMH,MEMUT
2529  011526  022767  007600  167012  4$:   CMP     #7600,TBANK             ;EXTERNAL BANK YET?
2530  011534  003345                        BGT     2$                      ;NO, NOT YET
2531  011536  005037  177572                CLR     @#SR0                   ;YES- DISABLE KT11
2532  011542  012737  000006  000004        MOV     #6,@#4                  ;RESTORE TRAPCATCHER
2533  011550  000207                        RTS     %7                      ;RETURN
2534  011552  046777  166742  166770  5$:   BIC     BITPT,@MEMUT            ;TIMEOUT OCCURRED-CLEAR BIT TO INDICATE
2535                                                                        ;4K BLOCK NOT PRESENT
2536  011560  022626                        CMP     (SP)+,(SP)+             ;ADJUST STACK
2537  011562  000745                        BR      3$                      ;CHECK NEXT BLOCK
2538
2539                                  ;NO KT PRESENT - MAP MAX OF 28K IN 4K CONTIGUOUS BLOCKS
2540  011564  012767  000001  166760  MAPMB: MOV     #1,NOKT                 ;SET FLAG TO INDICATE KT11 NOT PRESENT
2541  011572  022626                        CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
2542  011574  012737  011674  000004        MOV     #3$,@#4                 ;SET UP TIMEOUT RETURN
2543  011602  012767  000177  167256        MOV     #177,MEML               ;INITIALIZE MAP
2544  011610  005067  167254                CLR     MEMH
2545  011614  012767  000001  166676        MOV     #1,BITPT                ;SETUP 4K POINTER
2546  011622  005001                        CLR     R1                      ;INITIALIZE BANK ADDRESS
2547  011624  005761  001000          1$:   TST     1000(1)                 ;1ST K PRESENT
2548  011630  005761  005000                TST     5000(1)                 ;2ND K PRESENT
```

```
2549  011634  005761  011000                  TST     11000(1)                ;3RD K PRESENT
2550  011640  005761  015000                  TST     15000(1)                ;4TH K PRESENT
2551  011644  062701  020000          2$:     ADD     #20000,R1               ;UPDATE TEST ADDRESS
2552  011650  006367  166644                  ASL     BITPT                   ;UPDATE POINTER TO NEXT 4K
2553  011654  022767  000200  166636          CMP     #200,BITPT              ;28K CHECKED YET?
2554  011662  003360                          BGT     1$                      ;NO, CHECK NEXT 4K BLOCK
2555  011664  012737  000006  000004          MOV     #6,@#4
2556  011672  000207                          RTS     %7
2557  011674  046767  166620  167164  3$:     BIC     BITPT,MEML              ;TIMEOUT OCCURRED- CLEAR BIT TO
2558                                                                          ;INDICATE 4K BLOCK NOT PRESENT
2559  011702  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK POINTER
2560  011704  000757                          BR      2$
2561
2562
2563
2564                                  ;*****************************************************************
2565                                  ;MAP PARITY CORE AND CORRESPONDENCE TO ASSOCIATED REGISTERS
2566                                  ;*****************************************************************
2567  011706  013767  001066  166634  MAPREG: MOV     @#MEML,MEMUT            ;LOAD MAP OF MEMORY PRESENT IN LOWER 64K
2568  011714  012767  000001  166576          MOV     #1,BITPT                ;INITIALIZE 4K POINTER
2569  011722  012767  000001  167266          MOV     #1,KTSTART              ;INDICATE KT11 NOT IN USE
2570  011730  005067  166632                  CLR     RELOC
2571  011734  005067  166606                  CLR     TBANK                   ;INITIALIZE ADDRESS OF TEST BANK
2572  011740  005067  166610                  CLR     HIWORD                  ;CLEAR FLAG TO INDICATE FIRST 64K
2573                                                                          ;BEING CHECKED
2574  011744  005067  167250                  CLR     ADRTYP
2575
2576                                  ;SET WRITE WRONG PARITY IN ALL REGISTERS PRESENT
2577                                  ;THEN WRITE TEST LOCATION VIA DATO AND READ TEST LOCATION VIA DATI
2578                                  ;THEN CLEAR WRITE WRONG PARITY IN ALL REGISTERS
2579  011750  012702  000570          MAPRB:  MOV     #MPR0,R2                ;LOAD ADDRESS OF TABLE
2580  011754  032712  000001          1$:     BIT     #1,(2)                  ;IS THIS REGISTER PRESENT?
2581  011760  001003                          BNE     .+10                    ;NO, GET NEXT ONE
2582  011762  012772  000004  000000          MOV     #WWP,@(2)               ;YES, SET WRITE WRONG PARITY AND CLEAR REST
2583  011770  062702  000010                  ADD     #10,R2
2584  011774  020227  000770                  CMP     R2,#TREG                ;DONE WITH TABLE?
2585  012000  103765                          BLO     1$                      ;BRANCH IF NOT
2586  012002  016703  166540                  MOV     TBANK,R3                ;LOAD ADDRESS OF 4K BANK UNDER TEST
2587  012006  066703  167206                  ADD     ADRTYP,R3               ;ADD ADDRESS OFFSET (EITHER 0,2,4,OR 6)
2588  012012  011313                          MOV     (3),(3)                 ;WRITE WRONG PARITY
2589  012014  005713                          TST     (3)                     ;READ WRONG PARITY
2590  012016  012702  000570                  MOV     #MPR0,R2
2591  012022  032712  000001          2$:     BIT     #1,(2)
2592  012026  001003                          BNE     .+10
2593  012030  042772  000004  000000          BIC     #WWP,@(2)               ;CLEAR WRITE WRONG PARITY IN ALL
2594                                                                          ;PARITY REGISTERS
2595  012036  062702  000010                  ADD     #10,R2
2596  012042  020227  000770                  CMP     R2,#TREG
2597  012046  103765                          BLO     2$
2598  012050  012702  000560          MAPRC:  MOV     #MPR0-10,R2             ;INIT FOR ERROR CHECKS
2599  012054  062702  000010          1$:     ADD     #10,R2
2600  012060  020227  000770                  CMP     R2,#TREG
2601  012064  002031                          BGE     MAPRD                   ;BRANCH IF DONE WITH TABLE
2602  012066  032712  000001                  BIT     #1,(2)                  ;IS THIS REGISTER PRESENT?
```



```
2603  012072  001370                          BNE     1$                      ;NO, GET NEXT ADDRESS
2604  012074  005772  000000                  TST     @(R2)                   ;YES, DID THIS CONTROLLER GET A
2605                                                                          ;PARITY ERROR?
2606  012100  100365                          BPL     1$                      ;NO, CHECK NEXT
2607  012102  005767  166446                  TST     HIWORD                  ;YES, WHICH 64K IS UNDER TEST?
2608  012106  001004                          BNE     2$                      ;BRANCH IF UPPER 64K
2609  012110  056762  166404  000002          BIS     BITPT,2(2)              ;SET BIT IN MAP FOR THIS PARITY REGISTER
2610  012116  000403                          BR      3$
2611  012120  056762  166374  000004  2$:     BIS     BITPT,4(2)              ;SET BIT IN MAP FOR THIS PARITY REGISTER
2612  012126  105762  000007          3$:     TSTB    7(2)                    ;IS THIS THE FIRST ADDRESS FOUND FOR
2613                                                                          ;THIS PARITY REGISTER?
2614  012132  001005                          BNE     4$                      ;NO, BRANCH
2615  012134  116762  167060  000006          MOVB    ADRTYP,6(2)             ;YES, RECORD LOW BYTE OF ADDRESS (0,2,4,OR 6)
2616  012142  105262  000007                  INCB    7(2)                    ;INDICATE AN ADDRESS HAS BEEN FOUND FOR
2617                                                                          ;THIS REGISTER
2618  012146  000742                  4$:     BR      1$
2619  012150  011313          MAPRD:  MOV     @R3,@R3                 ;CLEAR BAD PARITY
2620  012152  062767  000002  167040          ADD     #2,ADRTYP               ;CHECK FIRST 4 ADDRESSES IN EACH 4K
2621  012160  026727  167034  000010          CMP     ADRTYP,#10
2622  012166  001402                          BEQ     1$                      ;BRANCH IF FIRST 4 ADDRESSES TESTED
2623  012170  000167  177554                  JMP     MAPRB                   ;IF NOT, GO TEST NEXT ONE
2624  012174  005767  166354          1$:     TST     HIWORD                  ;IS LOWER MEMORY DONE?
2625  012200  001021                          BNE     MAPRE                   ;YES, BRANCH
2626  012202  026727  166312  000100          CMP     BITPT,#100              ;DONE WITH 1ST 28K?
2627  012210  103015                          BHIS    MAPRE                   ;YES, BRANCH
2628  012212  062767  020000  166326          ADD     #20000,TBANK            ;NO- ADD 4K TO ADDRESS
2629  012220  006367  166274                  ASL     BITPT                   ;SHIFT BIT POINTER
2630  012224  005067  166770                  CLR     ADRTYP                  ;START WITH FIRST ADDRESS IN BANK
2631  012230  036767  166264  166312          BIT     BITPT,MEMUT             ;DOES THIS 4K BLOCK EXIST?
2632  012236  001756                          BEQ     1$                      ;NO- BRANCH
2633  012240  000167  177504                  JMP     MAPRB                   ;YES, TEST IT
2634  012244  005767  166302          MAPRE:  TST     NOKT                    ;KT11 PRESENT?
2635  012250  001401                          BEQ     .+4                     ;YES, BRANCH
2636  012252  000207                          RTS     %7                      ;NO, DONE
2637  012254  005767  166736                  TST     KTSTART                 ;KT11 ALREADY ON?
2638  012260  001417                          BEQ     1$                      ;YES, BRANCH
2639  012262  012767  020000  166256          MOV     #20000,TBANK            ;NO, INIT TBANK TO SELECT KERNEL PAGE 1
2640  012270  012767  001400  166270          MOV     #1400,RELOC             ;SET UP FOR ACCESS TO 28K BANK
2641  012276  004767  000762                  JSR     %7,NRALL                ;INITIALLY MAP ALL PAGES NR, BANK 0
2642  012302  004767  001126                  JSR     %7,MAP1                 ;MAP KERNEL 0 TO BANK 0, RW
2643                                                                          ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
2644                                                                          ;SET KERNEL 1 RW AND TURN ON KT11
2645  012306  005067  166704                  CLR     KTSTART                 ;INDICATE KT11 NOW IN USE
2646  012312  012737  000001  177572          MOV     #1,@#SR0                        ;TURN ON KT11
2647  012320  006367  166174          1$:     ASL     BITPT                   ;SHIFT BANK INDICATOR
2648  012324  103011                          BCC     2$                      ;BRANCH IF FIRST 64K NOT DONE
2649  012326  012767  000001  166164          MOV     #1,BITPT                ;IF FIRST 64K DONE, SETUP FOR
2650  012334  013767  001070  166206          MOV     @#MEMM, MEMUT           ;SECOND 64K
2651  012342  012767  000001  166204          MOV     #1,HIWORD               ;INDICATE NOW TESTING HIGH 64K
2652  012350  062767  000200  166210  2$:     ADD     #200,RELOC
2653  012356  022767  007600  166202          CMP     #7600,RELOC             ;UP TO EXTERNAL BANK YET?
2654  012364  003003                          BGT     3$                      ;NO, CONTINUE
2655  012366  005037  177572                  CLR     @#SR0                   ;YES, TURN OFF KT11 AND EXIT
2656  012372  000207                          RTS     %7
```

```
2657  012374  036767  166120  166146  38:     BIT     BITPT,MEMUT             ;IS THIS 4K PRESENT?
2658  012402  001746                          BEQ     1$                      ;NO- BRANCH
2659  012404  016777  166156  166630          MOV     RELOC,@KPAR1            ;YES, MAP PAGE 1 TO THIS BANK
2660  012412  005067  166602                  CLR     ADRTYP
2661  012416  000167  177326                  JMP     MAPRB                   ;GO TEST FOR PARITY MEMORY
2662
2663
2664
2665                                  ;*************************************************************************
2666                                  ;ROUTINE TO TYPE MAP OF WHERE PARITY MEMORY IS PRESENT
2667                                  ;AND WHICH CONTROL REGISTERS CONTROL WHICH MEMORY
2668                                  ;*************************************************************************
2669  012422  004767  000736          TMAP:   JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS PRESENT
2670  012426  104000                          TYPE                            ;TYPE "THE PARITY REGISTERS CONTROL MEMORY
2671  012430  016440                          MTMAP                           ;AS FOLLOWS:"
2672  012432  012701  000560                  MOV     #MPR0-10,R1             ;SET UP POINTER
2673  012436  062701  000010          TMAPA:  ADD     #10,R1
2674  012442  020127  000770                  CMP     R1,#TREG                ;DONE WITH MAP?
2675  012446  002136                          BGE     TMAPEX                  ;YES, BRANCH
2676  012450  005067  166100                  CLR     HIWORD
2677  012454  005067  166042                  CLR     TRFLG                   ;INITIALIZE TRANSITION FLAG (USED TO
2678                                                                          ;FIGURE MEMORY LIMITS)
2679  012460  005067  166040                  CLR     TYFLG                   ;INITIALIZE TO INDICATE NOTHING TYPED FOR
2680                                                                          ;THIS REGISTER YET
2681  012464  012767  177774  166034          MOV     #-4,TYCOR
2682  012472  016167  000002  166016          MOV     2(1),ADRPT              ;GET LOW 64K MEMORY MAP WORD
2683  012500  012767  000001  166012          MOV     #1,BITPT                ;INITIALIZE 4K POINTER
2684  012506  032711  000001                  BIT     #1,(1)                  ;DOES THIS CONTROL EXIST?
2685  012512  001351                          BNE     TMAPA                   ;NO, GET ADDRESS OF NEXT ONE
2686  012514  011167  165774                  MOV     (1),TEMPX               ;YES, PRINT ITS ADDRESS
2687  012520  004567  002244                  JSR     R5,OACNV
2688  012524  000514                          TEMPX
2689  012526  016607                          MPRAD
2690  012530  000006                          6
2691  012532  104000                          TYPE
2692  012534  017001                          MX1
2693  012536  104000                          TYPE
2694  012540  016607                          MPRAD
2695  012542  062767  000004  165756  TMAPB:  ADD     #4,TYCOR                ;KEEP TRACK OF # OF K OF CORE
2696  012550  036767  165744  165740          BIT     BITPT,ADRPT             ;DOES THIS PARITY REGISTER CONTROL THIS 4K?
2697  012556  001424                          BEQ     TMAPC                   ;NO- BRANCH
2698  012560  005767  165736                  TST     TRFLG                   ;YES, DOES IT CONTROL PREVIOUS 4K?
2699  012564  001043                          BNE     TMAPD                   ;YES- DON'T TYPE IT
2700  012566  012767  000001  165726          MOV     #1,TRFLG                ;NO- SET FLAG INDICATING TRANSITION
2701  012574  004567  002276                  JSR     R5,BDCNV                ;CONVERT K CORE TO ASCII
2702  012600  000526                          TYCOR
2703  012602  016507                          MTYCOR
2704  012604  000003                          3
2705  012606  104000                          TYPE                            ;TYPE "CONTROLS", AND ADDRESS OF CORE
2706  012610  017020                          MX2
2707  012612  104000                          TYPE
2708  012614  016507                          MTYCOR
2709  012616  104000                          TYPE
2710  012620  016514                          MDASH
```



```
2711  012622  005267  165676                  INC     TYFLG                   ;INDICATE TYPED
2712  012626  000422                          BR      TMAPD
2713  012630  005767  165666          TMAPC:  TST     TRFLG                   ;DID THIS PARITY REGISTER CONTROL PREVIOUS 4K?
2714  012634  001417                          BEQ     TMAPD                   ;NO, SKIP PRINTING
2715  012636  005067  165660                  CLR     TRFLG                   ;YES, TRANSITION OCCURRED- CLEAR FLAG
2716  012642  004567  002230                  JSR     R5,BDCNV                ;CONVERT K CORE TO ASCII
2717  012646  000526                          TYCOR
2718  012650  016507                          MTYCOR
2719  012652  000003                          3
2720  012654  104000                          TYPE
2721  012656  016507                          MTYCOR                          ;TYPE RIGHT AND RETURN
2722  012660  104000                          TYPE
2723  012662  016522                          MK
2724  012664  104000                          TYPE
2725  012666  016517                          MCR
2726  012670  005267  165630                  INC     TYFLG                   ;INDICATE TYPED
2727  012674  006367  165620          TMAPD:  ASL     BITPT                   ;UPDATE BIT POINTER TO NEXT 4K
2728  012700  103320                          BCC     TMAPB                   ;TEST NEXT 4K IF NOT DONE WITH 1ST 64K
2729  012702  005767  165646                  TST     HIWORD                  ;64-124K DONE?
2730  012706  001405                          BEQ     1$                      ;NO, BRANCH
2731  012710  005767  165610                  TST     TYFLG                   ;YES, WAS ANY PARITY MEMORY
2732                                                                          ;FOUND FOR THIS REGISTER?
2733
2734  012714  001250                          BNE     TMAPA
2735  012716  104002                          ERROR                           ;NO PARITY MEMORY WAS FOUND FOR THIS
2736                                                                          ;REGISTER- EITHER WRITE WRONG PARITY
2737                                                                          ;FAILED, PARITY ERROR GENERATE OR
2738                                                                          ;DETECT FAILED, OR THE PARITY ERROR
2739                                                                          ;BIT FAILED TO SET
2740  012720  000646                          BR      TMAPA
2741  012722  016167  000004  165566  1$:     MOV     4(1),ADRPT              ;UPDATE TO MAP WORD FOR THE UPPER 64K
2742  012730  012767  000001  165562          MOV     #1,BITPT                ;RESET BIT POINTER
2743  012736  005267  165612                  INC     HIWORD                  ;INDICATE LOW 64K DONE
2744  012742  000677                          BR      TMAPB                   ;LOOP
2745  012744  000207          TMAPEX: RTS     %7                              ;RETURN WHEN DONE
2746
2747
2748
2749                                  ;*************************************************************************
2750                                  ;ERROR HANDLER
2751                                  ;*************************************************************************
2752                                  ;ERRORS CALL ENTERS HERE
2753                                  ;TYPES PC, ICNT, MPR ADDRESS, AND MPR CONTENTS
2754                                  ;TREG SHOULD CONTAIN ADDRESS OF PARITY REGISTER
2755  012746  012767  016063  000266  ERRST:  MOV     #MSTR,ERRB              ;SETUP TO TYPE MPR ADDRESS AND CONTENTS
2756  012754  012767  177777  000262          MOV     #-1,ERRBX
2757  012762  012767  000240  000256          MOV     #240,ERRBX+2            ;NOP LOCATION AFTER MESSAGE
2758  012770  017767  165774  165542          MOV     @TREG,TRDATA            ;SETUP DATA
2759  012776  004567  001766                  JSR     R5,OACNV                ;CONVERT TO ASCII
2760  013002  000770                          TREG
2761  013004  016072                          MTREG
2762  013006  000006                          6
2763  013010  004567  001754                  JSR     R5,OACNV                ;CONVERT TO ASCII
2764  013014  000540                          TRDATA
```

```
 2765  013016  016114                          MDATA
 2766  013020  000006                          6
 2767  013022  000461                          BR      ERRA
 2768
 2769
 2770                                  ;ERRORP CALL ENTERS HERE
 2771                                  ;TYPES PC, ICNT, MPR ADDRESS, MPR CONTENTS
 2772                                  ;TEST LOCATION ADDRESS, VALUE EXPECTED, VALUE FOUND
 2773                                  ;R5 MUST CONTAIN ADDRESS OF TEST LOCATION
 2774                                  ;SHDBE MUST CONTAIN EXPECTED VALUE
 2775                                  ;WAS MUST CONTAIN ACTUAL DATA
 2776                                  ;TREG MUST CONTAIN ADDRESS OF PARITY REGISTER
 2777  013024  012767  016063  000210  ERRP:  MOV     #MSTR,ERRB              ;IN ADDITION TO BASIC PRINTOUT, TYPE
 2778                                                                          ;MPR ADDRESS AND CONTENTS
 2779  013032  012767  016125  000204          MOV     #MSTRX,ERRBX            ;ALSO OUTPUT DATA EXPECTED AND ACTUAL
 2780  013040  012767  177777  000200          MOV     #-1,ERRBX+2             ;NOP LOCATION AFTER MESSAGE
 2781  013046  010567  165460                  MOV     R5,TSTLOC               ;STORE ADDRESS BEING TESTED
 2782  013052  017767  165712  165460          MOV     @TREG,TRDATA
 2783  013060  004567  001704                  JSR     R5,OACNV
 2784  013064  000770                          TREG
 2785  013066  016072                          MTREG
 2786  013070  000006                          6
 2787  013072  004567  001672                  JSR     R5,OACNV
 2788  013076  000540                          TRDATA
 2789  013100  016114                          MDATA
 2790  013102  000006                          6
 2791  013104  004567  001660                  JSR     R5,OACNV
 2792  013110  000532                          TSTLOC
 2793  013112  016151                          MSTRX1
 2794  013114  000006                          6
 2795  013116  004567  001646                  JSR     R5,OACNV
 2796  013122  000534                          SHDBE
 2797  013124  016165                          MSTRX3
 2798  013126  000006                          6
 2799  013130  004567  001634                  JSR     R5,OACNV
 2800  013134  000536                          WAS
 2801  013136  016201                          MSTRX5
 2802  013140  000006                          6
 2803  013142  000411                          BR      ERRA
 2804
 2805
 2806
 2807                                  ;ERROR CALL ENTERS HERE
 2808                                  ;TYPE PC AND ICNT ONLY
 2809  013144  012767  177777  000070  ERR:   MOV     #-1,ERRB                ;SET UP ONE MESSAGE CALL
 2810  013152  012767  000240  000064          MOV     #240,ERRBX
 2811  013160  012767  000240  000060          MOV     #240,ERRBX+2
 2812  013166  032737  020000  177570  ERRA:  BIT     #BIT13,@#SR             ;INHIBIT ERROR PRINT?
 2813  013174  001025                          BNE     ERRC                    ;YES- BRANCH
 2814  013176  011667  000060                  MOV     (SP),ERRD               ;NO- DEVELOP CALLING ADDRESS
 2815  013202  162767  000002  000052          SUB     #2,ERRD
 2816  013210  004567  001554                  JSR     R5,OACNV                ;GO TO OCTAL TO ASCII CONVERT
 2817  013214  013262                  ERRA1:  ERRD                            ;SOURCE ADDRESS
 2818  013216  016036                  ERRA2:  MPC                             ;DESTINATION ADDRESS
```



```
 2819  013220  000006                          6                               ;#OF DIGITS TO CONVERT
 2820  013222  004567  001542                  JSR     R5,OACNV                ;CONVERT ICNT TO ASCII
 2821  013226  016024                  ERRA3:  ICNT
 2822  013230  016054                  ERRA4:  MICNT
 2823  013232  000006                          6
 2824  013234  004567  001476                  JSR     R5,TYPSX                ;TYPE MESSAGE
 2825  013240  016030                  ERRA5:  MEO                             ;ERROR HEADER
 2826  013242  000000                  ERRB:   OPEN                            ;ADDITIONAL ERROR MESSAGES IF ANY
 2827  013244  000000                  ERRBX:  OPEN
 2828  013246  177777                          -1
 2829  013250  005737  177570          ERRC:   TST     @#SR                    ;HALT ON ERROR SET?
 2830  013254  100001                          BPL     .+4                     ;NO- BRANCH
 2831  013256  000000                          HALT                            ;YES- ERROR OCCURRED SO HALT
 2832  013260  000002                          RTI
 2833  013262  000000                  ERRD:   OPEN
 2834
 2835
 2836
 2837                                  ;MAP ALL PAGES NON-RESIDENT, BANK 0
 2838  013264  013746  000004          NRALL:  MOV     @#4,-(SP)
 2839  013270  013746  000006                  MOV     @#6,-(SP)
 2840  013274  012737  000006  000004          MOV     #6,@#4
 2841  013302  012737  000002  000006          MOV     #RTI,@#6
 2842  013310  010146                          MOV     R1,-(SP)
 2843  013312  010246                          MOV     R2,-(SP)
 2844  013314  010346                          MOV     R3,-(SP)
 2845  013316  012701  001222                  MOV     #PDRTAB,R1
 2846  013322  012703  000040          1$:     MOV     #32.,R3
 2847  013326  012102                          MOV     (R1)+,R2
 2848  013330  005022                  2$:     CLR     (R2)+
 2849  013332  005303                          DEC     R3
 2850  013334  001375                          BNE     2$
 2851  013336  020127  001226                  CMP     R1,#PDREND
 2852  013342  003767                          BLE     1$
 2853  013344  012603                          MOV     (SP)+,R3
 2854  013346  012602                          MOV     (SP)+,R2
 2855  013350  012601                          MOV     (SP)+,R1
 2856  013352  012637  000006                  MOV     (SP)+,@#6
 2857  013356  012637  000004                  MOV     (SP)+,@#4
 2858  013362  000207                          RTS     %7
 2859
 2860
 2861
 2862                                  ;ROUTINE TO CLEAR ALL PARITY REGISTERS PRESENT
 2863  013364  010146          CLRPAR: MOV     R1,-(SP)
 2864  013366  010246                  MOV     R2,-(SP)
 2865  013370  010701                  MOV     PC,R1
 2866  013372  062701  165176          ADD     #MPR0-.,R1
 2867  013376  010702                  MOV     PC,R2
 2868  013400  062702  165370          ADD     #TREG-.,R2
 2869  013404  032711  000001  1$:     BIT     #1,@R1                  ;IS THIS REGISTER PRESENT?
 2870  013410  001002                  BNE     .+6
 2871  013412  005071  000000          CLR     @(R1)                   ;CLEAR ALL PARITY REGISTERS
 2872  013416  062701  000010          ADD     #10,R1
```

```
 2873  013422  020102                        CMP     R1,R2
 2874  013424  103767                        BLO     1$
 2875  013426  012602                        MOV     (SP)+,R2
 2876  013430  012601                        MOV     (SP)+,R1
 2877  013432  000207                        RTS     %7
 2878
 2879
 2880
 2881                                  ;ROUTINE TO MAP KERNEL 0 TO BANK 0 READ/WRITE,
 2882                                  ;KERNEL 1 READ/WRITE BUT BANK MAPPED BY CALLING ROUTINE,
 2883                                  ;AND KERNEL 7 TO EXTERNAL BANK, READ/WRITE
 2884  013434  012777  077406  165566  MAP1:   MOV     #77406,@KPDR0
 2885  013442  012777  077406  165562          MOV     #77406,@KPDR1
 2886  013450  012777  077406  165560          MOV     #77406,@KPDR7
 2887  013456  012777  007600  165562          MOV     #7600,@KPAR7
 2888  013464  005077  165550                  CLR     @KPAR0
 2889  013470  012737  000001  177572          MOV     #1,@#SR0
 2890  013476  000207                          RTS     %7
 2891
 2892
 2893
 2894                                  ;ROUTINE TO LOCATE THE FIRST PARITY MEMORY ADDRESS (ABOVE BANK 0)
 2895                                  ;CORRESPONDING TO A GIVEN PARITY REGISTER-REQUIRES THAT THE ROUTINES
 2896                                  ;MAPMEM AND MAPREG HAVE ALREADY BEEN RUN
 2897                                  ;TO USE, PUT THE ADDRESS OF THE ADDRESS OF THE REGISTER IN R0 (I.E. POINT TO
 2898                                  ;TO MAP TABLE)-THE DESIRED ADDRESS IS RETURNED IN R1, USING KERNEL PAGE 1 IF
 2899                                  ;KT11 IS PRESENT
 2900  013500  010246                  LOCATM: MOV     R2,-(SP)
 2901  013502  012702  000200                  MOV     #200,R2
 2902  013506  012701  000002                  MOV     #2,R1                   ;SKIP USE OF BANK 0
 2903  013512  005767  165034          1$:     TST     NOKT                    ;KT11 PRESENT?
 2904  013516  001403                          BEQ     3$                      ;YES, BRANCH
 2905  013520  022701  000200                  CMP     #200,R1                 ;NO, CHECK ONLY FOR MEMORY IN FIRST 28K
 2906  013524  101420                          BLOS    4$                      ;IF NO MEMORY FOUND IN 1ST 28K, GIVE
 2907                                                                          ;ERROR RETURN
 2908  013526  030160  000002          3$:     BIT     R1,2(R0)                ;DOES THIS 4K CORRESPOND TO THIS REGISTER?
 2909  013532  001021                          BNE     LOCAT1                  ;YES, BRANCH
 2910  013534  062702  000200                  ADD     #200,R2                 ;NO, CHECK TO SEE IF NEXT 4K CORRESPONDS
 2911  013540  006301                          ASL     R1
 2912  013542  103363                          BCC     1$
 2913  013544  012701  000001                  MOV     #1,R1
 2914  013550  030160  000004          2$:     BIT     R1,4(R0)                ;CHECK HIGH 64K
 2915  013554  001010                          BNE     LOCAT1
 2916  013556  062702  000200                  ADD     #200,R2
 2917  013562  006301                          ASL     R1
 2918  013564  103371                          BCC     2$
 2919  013566  012701  000001          4$:     MOV     #1,R1                   ;NO PARITY MEMORY CORRESPONDS TO
 2920  013572  012602                          MOV     (SP)+,R2                ;THIS REGISTER- RETURN WITH ERROR
 2921  013574  000207                          RTS     %7                      ;INDICATION
 2922  013576  005767  164750          LOCAT1: TST     NOKT                    ;KT11 PRESENT?
 2923  013602  001005                          BNE     1$                      ;NO- BRANCH OVER
 2924  013604  010277  165432                  MOV     R2,@KPAR1               ;YES- SETUP R1 TO REFERENCE 4K
 2925  013610  012701  020000                  MOV     #20000,R1               ;BANK USING KERNEL PAGE 1
 2926  013614  000407                          BR      2$
```



```
 2927  013616  010201                  1$:     MOV     R2,R1                   ;SETUP R1 TO REFERENCE 4K BANK
 2928  013620  006301                          ASL     R1                      ;WITHOUT KT11
 2929  013622  006301                          ASL     R1
 2930  013624  006301                          ASL     R1
 2931  013626  006301                          ASL     R1
 2932  013630  006301                          ASL     R1
 2933  013632  006301                          ASL     R1
 2934  013634  116067  000006  000010  2$:     MOVB    6(R0),LSAV
 2935  013642  066701  000004                  ADD     LSAV,R1
 2936  013646  012602                          MOV     (SP)+,R2                ;RESTORE R2
 2937  013650  000207                          RTS     %7                      ;AND RETURN
 2938  013652  000000                  LSAV:   0
 2939
 2940
 2941
 2942                                  ;SAVE LOADER IN TOP OF FIRST 4K
 2943  013654  013746  000004          SAVLDR: MOV     @#4,-(SP)               ;SAVE CONTENTS OF TIMEOUT VECTOR
 2944  013660  013746  000006                  MOV     @#6,-(SP)
 2945  013664  012737  000006  000004          MOV     #6,@#4
 2946  013672  012737  000002  000006          MOV     #RTI,@#6                ;SETUP TO RTI ON TIMEOUT
 2947  013700  010146                          MOV     R1,-(SP)                ;SAVE REGISTERS
 2948  013702  010246                          MOV     R2,-(SP)
 2949  013704  012701  157500                  MOV     #157500,R1
 2950  013710  000261                  1$:     SEC
 2951  013712  005711                          TST     @R1                     ;IF TIMEOUT, C BIT WILL STILL BE SET
 2952  013714  103006                          BCC     2$                      ;IF NO TIMEOUT, C BIT WILL BE CLEAR
 2953  013716  162701  020000                  SUB     #20000,R1               ;TIMEOUT OCCURRED, CHECK FOR NEXT LOWER BANK
 2954  013722  020127  020000                  CMP     R1,#20000
 2955  013726  101370                          BHI     1$
 2956  013730  000410                          BR      SAVLDX                  ;ONLY 4K OF 1ST 28K PRESENT- EXIT
 2957  013732  012702  017500          2$:     MOV     #17500,R2               ;THIS BANK IS HIGHEST OF 1ST 28K- COPY
 2958  013736  012122                  3$:     MOV     (R1)+,(R2)+             ;LOADER AREA TO BANK 0
 2959  013740  020227  020000                  CMP     R2,#20000
 2960  013744  103774                          BLO     3$
 2961  013746  005267  000016                  INC     LDRSVD                  ;INDICATE LOADER HAS BEEN MOVED TO
 2962                                                                          ;THE TOP OF THE FIRST 4K
 2963  013752  012602                  SAVLDX: MOV     (SP)+,R2
 2964  013754  012601                          MOV     (SP)+,R1
 2965  013756  012637  000006                  MOV     (SP)+,@#6
 2966  013762  012637  000004                  MOV     (SP)+,@#4
 2967  013766  000207                          RTS     %7
 2968  013770  000000                  LDRSVD: 0
 2969
 2970
 2971
 2972                                  ;ROUTINE TO RESTORE THE LOADER FROM BANK 0 (WHERE IT WAS SAVED) TO THE
 2973                                  ;HIGHEST BANK IN THE FIRST 28K OF MEMORY
 2974  013772  005767  177772          RSTLDR: TST     LDRSVD
 2975  013776  001431                          BEQ     RSTLDX
 2976  014000  012737  000006  000004          MOV     #6,@#4
 2977  014006  012737  000002  000006          MOV     #RTI,@#6
 2978  014014  012701  157500                  MOV     #157500,R1
 2979  014020  000261                  1$:     SEC
 2980  014022  005711                          TST     @R1                     ;IF TIMEOUT, C BIT WILL BE SET
```

```
 2981  014024  103006                          BCC     2$                      ;IF NO TIMEOUT, C BIT WILL BE CLEAR
 2982  014026  162701  020000                  SUB     #20000,R1               ;TIMEOUT OCCURRED, CHECK FOR NEXT
 2983  014032  020127  020000                  CMP     R1,#20000               ;LOWER BANK
 2984  014036  101370                          BHI     1$
 2985  014040  000410                          BR      RSTLDX
 2986  014042  012702  017500          2$:     MOV     #17500,R2
 2987  014046  012221                  3$:     MOV     (R2)+,(R1)+
 2988  014050  020227  020000                  CMP     R2,#20000
 2989  014054  103774                          BLO     3$
 2990  014056  104000                          TYPE                            ;TYPE MESSAGE "LOADER RESTORED"
 2991  014060  017032                          LDRMSG
 2992  014062  000000                  RSTLDX: HALT                            ;LOADER HAS BEEN RESTORED
 2993  014064  000776                          BR      .-2                     ;TO HIGHEST BANK IN FIRST 28K
 2994
 2995
 2996
 2997                                  ;SCAN ALL MEMORY FOR BAD PARITY, TYPE 18 BIT ADDRESSES OF
 2998                                  ;LOCATIONS FOUND TO BE BAD, AND WRITE INTO LOCATIONS WITH GOOD PARITY
 2999  014066  010146                  PSCAN:  MOV     R1,-(SP)                ;STORE REGISTERS AND LOCATIONS TO BE
 3000  014070  010246                          MOV     R2,-(SP)                ;ALTERRED
 3001  014072  010346                          MOV     R3,-(SP)
 3002  014074  010446                          MOV     R4,-(SP)
 3003  014076  013746  000004                  MOV     @#4,-(SP)
 3004  014102  013746  000006                  MOV     @#6,-(SP)
 3005  014106  013746  000114                  MOV     @#114,-(SP)
 3006  014112  013746  000116                  MOV     @#116,-(SP)
 3007  014116  012737  000006  000004          MOV     #6,@#4                  ;SETUP TIMEOUT TRAPCATCHER
 3008  014124  005037  000006                  CLR     @#6
 3009  014130  012737  000116  000114          MOV     #116,@#114              ;SETUP PARITY TRAP TRAPCATCHER
 3010  014136  005037  000116                  CLR     @#116
 3011  014142  005767  164404                  TST     NOKT                    ;KT11 PRESENT?
 3012  014146  001513                          BEQ     PSCAN1                  ;YES, BRANCH
 3013  014150  005002                          CLR     R2                      ;R2 CONTAINS TEST ADDRESS
 3014  014152  012703  000001                  MOV     #1,R3                   ;R3 USED AS A BIT POINTER
 3015  014156  004767  177202          1$:     JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
 3016  014162  005712                          TST     @R2                     ;READ LOCATION TO CHECK FOR BAD PARITY
 3017  014164  000240                          NOP
 3018  014166  012701  000570                  MOV     #MPR0,R1                ;SETUP TO SCAN REGISTERS FOR PARITY
 3019                                                                          ;ERROR SET
 3020  014172  032711  000001          3$:     BIT     #1,@R1
 3021  014176  001003                          BNE     .+10
 3022  014200  005771  000000                  TST     @(R1)                   ;PARITY ERROR SET?
 3023  014204  100424                          BMI     6$                      ;YES- BRANCH
 3024  014206  062701  000010                  ADD     #10,R1                  ;NO- CHECK NEXT REGISTER
 3025  014212  020127  000770                  CMP     R1,#TREG
 3026  014216  103765                          BLO     3$                      ;LOOP UNTIL ALL REGISTERS HAVE BEEN CHECKED
 3027  014220  062702  000002          4$:     ADD     #2,R2                   ;MOVE ADDRESS POINTER
 3028  014224  032702  017777                  BIT     #17777,R2               ;DONE WITH 4K?
 3029  014230  001352                          BNE     1$                      ;NO, CONTINUE
 3030  014232  006303                  5$:     ASL     R3                      ;YES, CHECK FOR TESTING NEXT 4K
 3031  014234  020327  000200                  CMP     R3,#200
 3032  014240  103035                          BHIS    PSCANX                  ;EXIT IF DONE WITH 28K
 3033  014242  030367  164620                  BIT     R3,MEML                 ;IS THIS MEMORY PRESENT?
 3034  014246  001343                          BNE     1$                      ;YES, GO TEST IT
```



```
 3035  014250  062702  020000                  ADD     #20000,R2               ;NO, UPDATE ADDRESS
 3036  014254  000766                          BR      5$                      ;LOOP
 3037  014256  010267  000110          6$:     MOV     R2,PSADR$               ;PARITY ERROR OCCURRED
 3038  014262  004567  000502                  JSR     R5,OACNV                ;GET ASCII OF ADDRESS CONTAINING BAD PARITY
 3039  014266  014372                          PSADRS
 3040  014270  016772                          MPSER1
 3041  014272  000006                          6
 3042  014274  104000                          TYPE                            ;TYPE MESSAGE "BAD PARITY FOUND IN LOCATION"
 3043  014276  016742                          MPSER                           ;AND ADDRESS OF FAILING LOCATION
 3044  014300  032737  002000  177570          BIT     #2000,@#SR              ;SWITCH 10 SET?
 3045  014306  001401                          BEQ     .+4                     ;NO- CONTINUE
 3046  014310  000000                          HALT                            ;HALT ON BAD PARITY SET
 3047  014312  011212                          MOV     @R2,@R2                 ;WRITE INTO LOCATION- SHOULD
 3048                                                                          ;CLEAR BAD PARITY
 3049  014314  005071  000000                  CLR     @(R1)                   ;CLEAR CORRESPONDING PARITY REGISTER
 3050  014320  005712                          TST     @R2                     ;READ LOCATION TO SEE IF BAD PARITY WAS
 3051  014322  005771  000000                  TST     @(R1)                   ;CLEARED
 3052  014326  100001                          BPL     .+4                     ;OK- BRANCH
 3053  014330  104002                          ERROR                           ;BAD PARITY DIDN'T CLEAR WHEN LOCATION
 3054                                                                          ;WAS REWRITTEN
 3055  014332  000732                          BR      4$                      ;GO CHECK NEXT LOCATION
 3056  014334  004767  177024          PSCANX: JSR     PC,CLRPAR               ;DONE- CLEAR ALL PARITY REGISTERS
 3057  014340  012637  000116                  MOV     (SP)+,@#116             ;RESTORE LOCATIONS ALTERRED
 3058  014344  012637  000114                  MOV     (SP)+,@#114
 3059  014350  012637  000006                  MOV     (SP)+,@#6
 3060  014354  012637  000004                  MOV     (SP)+,@#4
 3061  014360  012604                          MOV     (SP)+,R4
 3062  014362  012603                          MOV     (SP)+,R3
 3063  014364  012602                          MOV     (SP)+,R2
 3064  014366  012601                          MOV     (SP)+,R1
 3065  014370  000207                          RTS     %7                      ;RETURN
 3066  014372  000000          PSADRS: 0
 3067  014374  000000          PSCANH: 0
 3068
 3069
 3070
 3071                                  ;SCAN ALL MEMORY FOR BAD PARITY USING KT11
 3072                                  ;TYPE 18 BIT ADDRESSES OF LOCATIONS FOUND BAD, AND WRITE GOOD PARITY BACK IN
 3073  014376  017746  164640          PSCAN1: MOV     @KPAR1,-(SP)            ;SAVE CONTENTS OF KERNEL PAR1
 3074  014402  032737  000001  177572          BIT     #1,@#SR0                        ;SKIP IF KT11 IS ALREADY ON
 3075  014410  001004                          BNE     1$
 3076  014412  004767  176646                  JSR     PC,NRALL                ;MAP KERNEL 0 TO BANK 0,RW
 3077  014416  004767  177012                  JSR     PC,MAP1                 ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
 3078                                                                          ;MAP KERNEL 1 RW, AND TURN ON KT11
 3079  014422  005067  177746          1$:     CLR     PSCANH                  ;CLEAR FLAG TO INDICATE CHECKING FIRST 64K
 3080  014426  005077  164610                  CLR     @KPAR1                  ;INITIALIZE TO BANK 0
 3081  014432  012703  000001          PSLOOP: MOV     #1,R3                   ;R3 IS USED AS A BIT POINTER
 3082  014436  005767  177732          PSLUP:  TST     PSCANH                  ;TESTING TOP 64K?
 3083  014442  001004                          BNE     2$                      ;YES, BRANCH
 3084  014444  030367  164416                  BIT     R3,MEML                 ;NO, IS PARITY MEMORY PRESENT IN THIS 4K?
 3085  014450  001022                          BNE     PSXTST                  ;YES- GO TEST IT
 3086  014452  000403                          BR      PSNXT                   ;NO- CHECK FOR NEXT 4K
 3087  014454  030367  164410          2$:     BIT     R3,MEMH                 ;IS PARITY MEMORY PRESENT IN THIS 4K?
 3088  014460  001016                          BNE     PSXTST                  ;YES- GO TEST IT
```

```
3089  014462  062777  000200  164552  PSNXT:  ADD     #200,@KPAR1             ;NO= MAP TO NEXT 4K
3090  014470  006303                  1$:     ASL     R3
3091  014472  103361                          BCC     PSLUP                   ;BRANCH IF NOT END OF 64K
3092  014474  005767  177674                  TST     PSCANH                  ;END OF TOP 64K?
3093  014500  001003                          BNE     PSCX1                   ;YES, GET READY TO EXIT
3094  014502  005267  177666                  INC     PSCANH                  ;NO, SET FLAG INDICATING DONE WITH
3095                                                                          ;LOWER 64K
3096  014506  000751                          BR      PSLOOP
3097  014510  012677  164526          PSCX1:  MOV     (SP)+,@KPAR1
3098  014514  000707                          BR      PSCANX
3099  014516  012702  020000          PSXTST: MOV     #20000,R2               ;R2 USED AS ADDRESS POINTER
3100  014522  004767  176636          1$:     JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
3101  014526  005712                          TST     @R2                     ;READ LOCATION
3102  014530  012701  000570                  MOV     #MPR0,R1                ;SETUP TO SCAN REGISTERS FOR PARITY ERROR SET
3103  014534  032711  000001          2$:     BIT     #1,@R1
3104  014540  001003                          BNE     .+10
3105  014542  005771  000000                  TST     @(R1)                   ;PARITY ERROR SET?
3106  014546  100413                          BMI     4$                      ;YES, BRANCH
3107  014550  062701  000010                  ADD     #10,R1                  ;NO, CHECK NEXT
3108  014554  020127  000770                  CMP     R1,#TREG
3109  014560  103765                          BLO     2$                      ;LOOP UNTIL ALL REGISTERS HAVE BEEN CHECKED
3110  014562  062702  000002          3$:     ADD     #2,R2                   ;UPDATE TEST ADDRESS POINTER
3111  014566  020227  040000                  CMP     R2,#40000               ;DONE WITH BANK?
3112  014572  103753                          BLO     1$                      ;NO= LOOP
3113  014574  000732                          BR      PSNXT                   ;YES= GO CHECK FOR ANOTHER BANK
3114  014576  010267  177570          4$:     MOV     R2,PSADRS               ;PARITY ERROR OCCURRED= GET 18 BIT
3115  014602  042767  160000  177562          BIC     #160000,PSADRS          ;OCTAL ADDRESS OF BAD LOCATION
3116  014610  005046                          CLR     -(SP)
3117  014612  017746  164424                  MOV     @KPAR1,-(SP)
3118  014616  006316                          ASL     @SP
3119  014620  006316                          ASL     @SP
3120  014622  006316                          ASL     @SP
3121  014624  006316                          ASL     @SP
3122  014626  006316                          ASL     @SP
3123  014630  006166  000002                  ROL     2(SP)
3124  014634  006316                          ASL     @SP
3125  014636  006166  000002                  ROL     2(SP)
3126  014642  006366  000002                  ASL     2(SP)
3127  014646  062667  177520                  ADD     (SP)+,PSADRS
3128  014652  004567  000112                  JSR     R5,OACNV                ;CONVERT LOW 16 OCTAL BITS TO ASCII
3129  014656  014372                          PSADRS
3130  014660  016772                          MPSER1
3131  014662  000006                          6
3132  014664  116704  002102                  MOVB    MPSER1,R4
3133  014670  062604                          ADD     (SP)+,R4                ;CHANGE TO ASCII FOR 18 BITS
3134  014672  110467  002074                  MOVB    R4,MPSER1
3135  014676  104000                          TYPE                            ;TYPE ADDRESS OF LOCATION WITH BAD PARITY
3136  014700  016742                          MPSER
3137  014702  032737  002000  177570          BIT     #2000,@#SR              ;SWITCH 10 SET?
3138  014710  001401                          BEQ     .+4                     ;NO= BRANCH
3139  014712  000000                          HALT                            ;HALT ON BAD PARITY SET
3140  014714  011212                          MOV     @R2,@R2                 ;REWRITE LOCATION CONTAINING BAD PARITY
3141  014716  005071  000000                  CLR     @(R1)                   ;CLEAR PARITY ERROR BIT
3142  014722  005712                          TST     @R2                     ;READ LOCATION TO SEE IF PARITY IS NOW GOOD
```



```
3143  014724  005771  000000                  TST     @(R1)                   ;CHECK PARITY ERROR BIT
3144  014730  100001                          BPL     .+4
3145  014732  104002                          ERROR                           ;REWRITING LOCATION DID NOT CLEAR BAD PARITY
3146  014734  000712                          BR      3$                      ;GO TEST NEXT LOCATION
3147
3148
3149
3150
3151                                  ;PIC ROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES (CALLED VIA JSR R5)
3152  014736  012567  000022          TYPSX:  MOV     (R5)+,TYPSBX            ;GET ADDRESS OF MESSAGE
3153  014742  022767  177777  000014          CMP     #-1,TYPSBX              ;TERMINATOR?
3154  014750  001001                          BNE     TYPSAX                  ;NO, BRANCH
3155  014752  000205                          RTS     %5                      ;YES, RETURN
3156  014754  013746  177776          TYPSAX: MOV     @#PS,-(SP)              ;SETUP TO CALL TYPE ROUTINE VIA JSR
3157  014760  004767  164124                  JSR     PC,@#TYPE               ;TYPE ASCII MESSAGE
3158  014764  000000                  TYPSBX: OPEN
3159  014766  000763                          BR      TYPSX
3160
3161
3162
3163                                  ;SUBROUTINE FOR OCTAL TO ASCII CONVERSION
3164  014770  013567  000074          OACNV:  MOV     @(5)+,OACNVX            ;GET OCTAL VALUE
3165  014774  012567  000072                  MOV     (5)+,OACDST             ;GET DESTINATION ADDRESS
3166  015000  012567  000070                  MOV     (5)+,OACNT              ;GET CONVERT COUNT
3167  015004  066767  000064  000060          ADD     OACNT,OACDST            ;DEVELOP ADDRESS TO STORE 1ST CHAR.
3168  015012  016746  000052          OACNVA: MOV     OACNVX,-(SP)
3169  015016  042716  177770                  BIC     #177770,@SP             ;ISOLATE LEAST SIGNIFICANT DIGIT
3170  015022  062716  000060                  ADD     #60,@SP                 ;CONVERT DIGIT TO ASCII
3171  015026  005367  000040                  DEC     OACDST
3172  015032  112677  000034                  MOVB    (SP)+,@OACDST           ;STORE ASCII CHARACTER
3173  015036  042767  000007  000024          BIC     #7,OACNVX
3174  015044  006067  000020                  ROR     OACNVX
3175  015050  006067  000014                  ROR     OACNVX
3176  015054  006067  000010                  ROR     OACNVX
3177  015060  005367  000010                  DEC     OACNT                   ;DONE ALL DIGITS?
3178  015064  001352                          BNE     OACNVA                  ;BRANCH IF NOT DONE
3179  015066  000205                          RTS     R5                      ;DONE, EXIT
3180  015070  000000                  OACNVX: OPEN
3181  015072  000000                  OACDST: 0
3182  015074  000000                  OACNT:  0
3183
3184
3185
3186                                  ;SUBROUTINE FOR BINARY TO DECIMAL ASCII CONVERSION
3187  015076  104005                  BDCNV:  SAV04                           ;SAVE REGS
3188  015100  012700  015254                  MOV     #DECVAL,%0              ;SET UP ADDR TO STORE DECIMAL ASCII
3189  015104  013501                          MOV     @(5)+,R1                ;BINARY VALUE TO R1
3190  015106  012567  000052                  MOV     (5)+,BDCNVC             ;DESTINATION ADDR TO BDCNVC
3191  015112  012567  000050                  MOV     (5)+,BDCNVD             ;CHARACTER COUNT TO BDCNVD
3192  015116  012702  015242                  MOV     #ADTENP,R2              ;ADDR OF TEN POWER STRING
3193  015122  012767  000005  000104          MOV     #5,CNVCTR               ;SET UP FOR 5 POWER CONVERSIONS
3194  015130  012267  000104          BDCNVA: MOV     (2)+,TENPWR             ;MOVE POWER OF TEN VALUE
3195  015134  004767  000034                  JSR     PC,SUBTEN               ;PERFORM CONVERSION
3196  015140  005367  000070                  DEC     CNVCTR                  ;DONE 5 CONVERSIONS?
```

```
3197  015144  001371                          BNE     BDCNVA                  ;BRANCH IF NOT YET 5.
3198  015146  166700  000014                  SUB     BDCNVD,#0
3199  015152  010067  000004                  MOV     #0,BDCNVB
3200  015156  004567  000100                  JSR     R5,BMOVE
3201  015162  000000              BDCNVB: OPEN
3202  015164  000000              BDCNVC: OPEN
3203  015166  000000              BDCNVD: OPEN
3204  015170  104006                          RST04                           ;RESTORE REGS AND EXIT
3205  015172  000205                          RTS     R5
3206  015174  005067  000036      SUBTEN: CLR     DIGIT
3207  015200  166701  000034      SUBTNA: SUB     TENPWR,R1               ;SUBTRACT TEN POWER FROM BINARY VALUE
3208  015204  103403                          BCS     SUBTNB                  ;BRANCH IF UNSUCESSFUL SUBTRACTION
3209  015206  005267  000024              INC     DIGIT
3210  015212  000772                          BR      SUBTNA
3211  015214  066701  000020      SUBTNB: ADD     TENPWR,R1               ;RESTORE SUBTRACTED VALUE.
3212  015220  062767  000060  000010          ADD     #60,DIGIT               ;CONVERT (DIGIT) TO ASCII
3213  015226  116720  000004                  MOVB    DIGIT,(0)+              ;MOVE ASCII CHAR TO DECVAL FIELD
3214  015232  000207                          RTS     PC                      ;EXIT
3215  015234  000000              CNVCTR: OPEN
3216  015236  000000              DIGIT:  OPEN
3217  015240  000000              TENPWR: OPEN
3218  015242  023420              ADTENP: 10000.
3219  015244  001750                      1000.
3220  015246  000144                      100.
3221  015250  000012                      10.
3222  015252  000001                      1.
3223  015254      040     040     040 DECVAL: .BYTE   040,040,040,040,040,040
3224  015257      040     040     040
3225
3226
3227
3228                                  ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES
3229  015262  104005                  BMOVE:  SAV04                           ;SAVE REGS
3230  015264  012501                          MOV     (5)+,R1                 ;GET FROM ADDRESS
3231  015266  012502                          MOV     (5)+,R2                 ;GET TO ADDRESS
3232  015270  012503                          MOV     (5)+,R3                 ;GET COUNT
3233  015272  112122                  BMOVA:  MOVB    (1)+,(2)+               ;MOVE BYTE
3234  015274  005303                          DEC     R3                      ;DECREMENT COUNT
3235  015276  001375                          BNE     BMOVA                   ;BRANCH IF NOT DONE
3236  015300  104006                          RST04                           ;RESTORE REGS AND EXIT
3237  015302  000205                          RTS     R5
3238
3239
3240
3241                                  ;UNEXPECTED POWER FAIL SERVICE
3242                                  ;BECAUSE WWP MAY BE SET IN MPR'S AND ALL PROCESSOR REGISTERS
3243                                  ;MAY BE IN USE, CONTINUATION AFTER POWER FAIL IS NOT ATTEMPTED.
3244                                  ;INSTEAD, THE PROGRAM RESTARTS AFTER A POWER FAILURE
3245  015304  012737  015350  000024  PWRDN:  MOV     #PWRUP,@#24             ;SET UP FOR POWER UP
3246  015312  012701  000570                  MOV     #MPR0,R1
3247  015316  032711  000001          1$:     BIT     #1,@R1
3248  015322  001002                          BNE     .+6
3249  015324  005071  000000                  CLR     @(R1)                   ;CLEAR PARITY REGISTERS IN CASE
3250  015330  062701  000010                  ADD     #10,R1                  ;WWP IS SET
```



```
3251  015334  020127  000770                  CMP     R1,#TREG
3252  015340  103766                          BLO     1$
3253  015342  010667  163702                  MOV     SP,SPSAV
3254  015346  000000                          HALT                            ;POWER DOWN HALT
3255  015350  012737  015304  000024  PWRUP:  MOV     #PWRDN,@#24             ;SET UP FOR POWER DOWN
3256  015356  016706  163666                  MOV     SPSAV,SP
3257  015362  005027  000000                  CLR     #0                      ;STALL SO OUTPUT WON'T BE GARBLED
3258  015366  005367  177772                  DEC     .-2
3259  015372  001375                          BNE     .-4
3260  015374  104000                          TYPE                            ;TYPE RECOVERY MESSAGE
3261  015376  016676                          MPWRF
3262  015400  000167  163662                  JMP     RSTART                  ;RESTART
3263
3264
3265
3266                                  ;EMT HANDLER
3267  015404  011646              EMTINT: MOV     (SP),-(SP)              ;GET SAVED PC
3268  015406  162716  000002                  SUB     #2,(SP)                 ;DECREMENT PC BY 2
3269  015412  017616  000000                  MOV     @(SP),(SP)              ;GET CALL
3270  015416  121667  000050                  CMPB    (SP),EMTLIM             ;CHECK IF CALL WITHIN LIMITS
3271  015422  101402                          BLOS    EMTA
3272  015424  000000                          HALT                            ;CALL IS NOT WITHIN LIMITS
3273  015426  000776                          BR      .-2
3274  015430  006116              EMTA:   ROL     (SP)                    ;EMT ARG X 2
3275  015432  042716  177001                  BIC     #177001,(SP)            ;REMOVE 7 MSB
3276  015436  062716  015450                  ADD     #EMTTAB,(SP)            ;FORM EMT RTN ADDRESS
3277  015442  017616  000000                  MOV     @(SP),(SP)
3278  015446  000136                          JMP     @(SP)+                  ;GO TO EMT RETURN
3279
3280
3281                                  ;EMT DEFINITIONS AND ASSIGNMENTS
3282  015450                          EMTTAB:
3283          104000                          TYPE=EMT+EMTX
3284  015450  001110                          $TYPE
3285          104001                          SCOPE=EMT+EMTX
3286  015452  015730                          $SCOPEC
3287          104002                          ERROR=EMT+EMTX
3288  015454  013144                          ERR
3289          104003                          ERRJRP=EMT+EMTX
3290  015456  013024                          ERRP
3291          104004                          ERRORS=EMT+EMTX
3292  015460  012746                          ERRST
3293          104005                          SAV04=EMT+EMTX
3294  015462  015474                          SV04
3295          104006                          RST04=EMT+EMTX
3296  015464  015562                          RS04
3297          104007                          RST05S=EMT+EMTX
3298  015466  015610                          RS05S
3299          104010                          SAV05S=EMT+EMTX
3300  015470  015514                          SV05S
3301  015472  000010                  EMTLIM: EMTX=1
3302
3303
3304                                  ;SUBROUTINE TO SAVE REGS 0-4
```

```
3305  015474  012666  177764          SV04:   MOV     (SP)+,-12.(SP)          ;MOVE PC+PS UP STACK
3306  015500  012666  177764                  MOV     (SP)+,-12.(SP)
3307  015504  012767  000002  000040          MOV     #RTI,SV05C
3308  015512  000411                          BR      SV05B
3309
3310
3311
3312                                  ;SUBROUTINE TO SAVE REGS 0-5 + PLACE EMT PC IN R5
3313  015514  012767  000240  000030  SV05B:  MOV     #NOP,SV05C
3314  015522  000400                          BR      SV05A
3315
3316                                  ;SUBROUTINE TO SAVE REGS 0-5
3317  015524  012666  177762          SV05A:  MOV     (SP)+,-14.(SP)
3318  015530  012666  177762                  MOV     (SP)+,-14.(SP)
3319  015534  010546                          MOV     R5,-(SP)
3320  015536  010446                  SV05B:  MOV     R4,-(SP)
3321  015540  010346                          MOV     R3,-(SP)
3322  015542  010246                          MOV     R2,-(SP)
3323  015544  010146                          MOV     R1,-(SP)
3324  015546  010046                          MOV     %0,-(SP)
3325  015550  024646                          CMP     -(SP),-(SP)
3326  015552  000002                  SV05C:  RTI                             ;RTI OR NOP
3327  015554  016605  000020                  MOV     16.(SP),R5              ;EMT PC TO R5
3328  015560  000002                          RTI
3329
3330
3331
3332                                  ;SUBROUTINE TO RESTORE REGS 0-4
3333  015562  022626                  RS04:   CMP     (SP)+,(SP)+
3334  015564  012600                          MOV     (SP)+,%0
3335  015566  012601                          MOV     (SP)+,R1
3336  015570  012602                          MOV     (SP)+,R2
3337  015572  012603                          MOV     (SP)+,R3
3338  015574  012604                          MOV     (SP)+,R4
3339  015576  016646  177764                  MOV     -12.(SP),-(SP)          ;MOVE PC+PS DOWN STACK
3340  015602  016646  177764                  MOV     -12.(SP),-(SP)
3341  015606  000002                          RTI
3342
3343
3344
3345                                  ;SUBROUTINE TO RESTORE REGS 0-5
3346  015610  010566  000020          RS05B:  MOV     R5,16.(SP)              ;SET EMT PC TO R5
3347  015614  022626                          CMP     (SP)+,(SP)+
3348  015616  012600                          MOV     (SP)+,%0
3349  015620  012601                          MOV     (SP)+,R1
3350  015622  012602                          MOV     (SP)+,R2
3351  015624  012603                          MOV     (SP)+,R3
3352  015626  012604                          MOV     (SP)+,R4
3353  015630  012605                          MOV     (SP)+,R5
3354  015632  016646  177762                  MOV     -14.(SP),-(SP)
3355  015636  016646  177762                  MOV     -14.(SP),-(SP)
3356  015642  000002                          RTI
3357
3358
```



```
3359
3360                                  ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
3361                                  ;LOAD THE STARTING ADDRESS OF THE TEST
3362                                  ;YOU WISH TO RUN (THE ADDRESS OF THE TESTXX
3363                                  ;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
3364                                  ;OPTIONS AT THE 2ND HALT.
3365                                  ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
3366  015644  005037  177776          TESTX:  CLR     @#PS
3367  015650  000000                          HALT                            ;WAIT FOR STARTING ADDRESS
3368  015652  013767  177570  000146          MOV     @#SR,RETURN             ;LOAD STARTING ADDRESS IN RETURN
3369  015660  062767  000002  000140          ADD     #2,RETURN               ;ADD 2 TO POINT TO INSTRUCTION AFTER
3370  015666  000000                          HALT                            ;SET SR OPTIONS
3371  015670  012767  177777  162612          MOV     #-1,TSTX                ;SET FLAG
3372  015676  032737  010000  177570          BIT     #10000,@#SR             ;CHECK SW12
3373  015704  001404                          BEQ     .+12                    ;BRANCH IF NOT SET
3374  015706  042737  000020  177776          BIC     #20,@#PS                ;CLEAR TRACE BIT
3375  015714  000403                          BR      .+10                    ;SKIP NEXT INSTRUCTION
3376  015716  052737  000020  177776          BIS     #20,@#PS                ;SET TRACE BIT
3377  015724  000177  000076                  JMP     @RETURN                 ;JUMP TO TEST
3378
3379
3380
3381                                  ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 64 TIMES
3382                                  ;A SETUP ROUTINE SHOULD INITIALIZE RETURN AND IMAX
3383  015730  032737  040000  177570  SCOPEC: BIT     #40000,@#SR             ;TEST SR FOR SCOPE
3384  015736  001015                          BNE     SCOPEB                  ;YES, SCOPE
3385  015740  032737  004000  177570          BIT     #4000,@#SR              ;NO-TEST FOR ITERATION
3386  015746  001016                          BNE     SCOPEG                  ;INHIBIT ITERATION
3387  015750  005767  162534                  TST     TSTX                    ;USING SINGLE SUBTEST STARTUP?
3388  015754  001006                          BNE     SCOPEB                  ;YES, LOOP
3389  015756  026767  000042  000036          CMP     ICNT,IMAX               ;COMPARE CURRENT COUNT TO MAX NUMBER
3390  015764  100007                          BPL     SCOPEG                  ;EXIT-DONE
3391  015766  005267  000032                  INC     ICNT                    ;INCREMENT COUNT
3392  015772  022606                  SCOPEB: CMP     (6)+,%6                 ;REPOSITION STACK
3393  015774  012677  161776                  MOV     (6)+,@PS                ;RESTORE PREVIOUS PROCESSOR STATUS
3394  016000  000177  000022                  JMP     @RETURN                 ;REPEAT TEST
3395  016004  005067  162500          SCOPEG: CLR     TSTX                    ;IF USING TESTX STARTUP, RETURN TO NORMAL FLOW
3396  016010  005067  000010                  CLR     ICNT                    ;CLEAR COUNT
3397  016014  011667  000006                  MOV     @%6,RETURN              ;SAVE SCOPE RETURN POINTER
3398  016020  000002                          RTI                             ;RETURN INLINE-NEXT TEST
3399  016022  000100                  IMAX:   100                             ;ITERATION COUNT
3400  016024  000000                  ICNT:   0                               ;COUNT LOCATION FOR ITERATION LOOP
3401  016026  000000                  RETURN: 0                               ;ADDRESS OF LAST TEST
3402
3403
3404
3405
3406                                  ;ASCII MESSAGES
3407  016030                          MES:
3408  016030  005015  041520  020075  MTNUM:  .ASCII  <15><12>'PC= '
3409  016036  020040  020040  020040  MPC:    .ASCII  '        ICNT= '
3410  016044  020040  041511  052116
3411  016052  020075
3412  016054  020040  020040  020040  MICNT:  .ASCIZ  '       '
```

```
3413  016062     000
3414  016063     040  046440  051120  MSTR:   .ASCII  '  MPR= '
3415  016070  020075
3416  016072  020040  020040  020040  MTREG:  .ASCII  '        MPR DATA= '
3417  016100  020040  050115  020122
3418  016106  040504  040524  020075
3419  016114  020040  020040  020040  MDATA:  .ASCIZ  '        '
3420  016122  020040     000
3421  016125     015  020012  020040  MSTRX:  .ASCII  <15><12>'        TEST LOC= '
3422  016132  020040  020040  052040
3423  016140  051505  020124  047514
3424  016146  036503     040
3425  016151     040  020040  020040  MSTRX1: .ASCII  '      '
3426  016156     040
3427  016157     040  027523  035102          .ASCII  ' S/B: '
3428  016164     040
3429  016165     040  020040  020040  MSTRX3: .ASCII  '      '
3430  016172     040
3431  016173     040  040527  035123          .ASCII  ' WAS: '
3432  016200     040
3433  016201     040  020040  020040  MSTRX5: .ASCIZ  '       '
3434  016206  020040     000
```



```
3435  016211     015  051412  052105  MSETSR: .ASCIZ  <15><12>'SET SR OPTIONS'
3436  016216  051440  020122  050117
3437  016224  044524  047117  000123
3438  016232  020054  051120  051505  MCON:   .ASCIZ  ', PRESS CONTNUE'
3439  016240  020123  047503  052116
3440  016246  052516  000105
3441  016252  005015  042523  020124  MMDEV:  .ASCIZ  <15><12>'SET DEVICE ADDRESS IN SR'
3442  016260  042504  044526  042503
3443  016266  040440  042104  042522
3444  016274  051523  044440  020116
3445  016302  051123     000
3446  016305     015  051412  052105  MMADR:  .ASCIZ  <15><12>'SET MEMORY TEST LOC IN SR'
3447  016312  046440  046505  051117
3448  016320  020131  042524  052123
3449  016326  046040  041517  044440
3450  016334  020116  051123     000
3451  016341     015  051412  052105  MMPAT:  .ASCIZ  <15><12>'SET TEST PATTERN IN SR'
3452  016346  052040  051505  020124
3453  016354  040520  052124  051105
3454  016362  020116  047111  051440
3455  016370  000122
3456  016372  005015  046412  046505  MMPRS:  .ASCIZ  <15><12><12>'MEMORY PARITY REGISTERS PRESENT:'<15><12>
3457  016400  051117  020131  040520
3458  016406  044522  054524  051040
3459  016414  043505  051511  042524
3460  016422  051522  050040  042522
3461  016430  042523  052116  006472
3462  016436  000012
3463  016440  005015  040520  052122  MTMAP:  .ASCIZ  <15><12>'PARTY REGISTERS CONTROL MEMORY AS:'<15><12>
3464  016446  020131  042522  044507
3465  016454  052123  051105  020123
3466  016462  047503  052116  047522
3467  016470  020114  042515  047515
3468  016476  054522  040440  035123
3469  016504  005015     000
3470  016507     040  020040  000040  MTYCOR: .ASCIZ  '    '
3471  016514  020055     000          MDASH:  .ASCIZ  '- '
3472  016517     015  000012          MCR:    .ASCIZ  <15><12>
3473  016522  000113                  MK:     .ASCIZ  'K'
3474  016524  047516  050040  051101  MT:     .ASCIZ  'NO PARITY MEMORY FOUND'<15><12>
3475  016532  052111  020131  042515
3476  016540  047515  054522  043040
3477  016546  052517  042116  005015
3478  016554     000
3479  016555     116  020117  040520  MTR:    .ASCIZ  'NO PARITY REGSTER FOUND'<15><12>
3480  016562  044522  054524  051040
3481  016570  043505  052123  051105
3482  016576  043040  052517  042116
3483  016604  005015     000
3484  016607     040  020040  020040  MPRAD:  .ASCIZ  '        '<15><12>
3485  016614  020040  006440  000012
3486  016622  005015  005015  042515  MTIT:   .ASCIZ  <15><12><15><12>'MEMORY PARITY TEST - MAINDEC-11-DCMFA-B'
3487  016630  047515  054522  050040
3488  016636  051101  052111  020131
```

```
 3489  016644  042524  052123  026440
 3490  016652  046440  044501  042116
 3491  016660  041505  030455  026461
 3492  016666  041504  043115  026501
 3493  016674  000102
 3494  016676  005015  047520  042527  MPWRF:  .ASCIZ  <15><12>'POWER FAILED'
 3495  016704  020122  040506  046111
 3496  016712  042105     000
 3497  016715     007             MPGEND: .BYTE   007
 3498  016716  005015  047105  020104          .ASCII  <15><12>'END PASS # '
 3499  016724  040520  051523  036440
 3500  016732     040
 3501  016733     040  020040  020040  MPCNT:  .ASCIZ  '      '
 3502  016740  000040
 3503  016742  006415  041012  042101  MPSER:  .ASCII  <15><15><12>'BAD PAR FOUND IN LOC '
 3504  016750  050040  051101  043040
 3505  016756  052517  042116  044440
 3506  016764  020116  047514  020103
 3507  016772  020040  020040  020040  MPSER1: .ASCIZ  '      '
 3508  017000     000
 3509  017001     015  051012  043505  MX1:    .ASCIZ  <15><12>'REGISTER AT '
 3510  017006  051511  042524  020122
 3511  017014  052101  000040
 3512  017020  047503  052116  047522  MX2:    .ASCIZ  'CONTROLS '
 3513  017026  051514  000040
 3514  017032  005015  047514  042101  LDRMSG: .ASCIZ  <15><12>'LOADERS RESTORED'
 3515  017040  051105  020123  042522
 3516  017046  052123  051117  042105
 3517  017054     000
 3518  017055     015  041012  042101  PSMSG:  .ASCIZ  <15><12>'BAD PARITY SCAN COMPLETE'
 3519  017062  050040  051101  052111
 3520  017070  020131  041523  047101
 3521  017076  041440  046517  046120
 3522  017104  052105  000105
 3523  017110  005015  047514  042101  MLDRSV: .ASCII  <15><12>'LOADERS SAVED IN BANK 0'
 3524  017116  051105  020123  040523
 3525  017124  042526  020104  047111
 3526  017132  041040  047101  020113
 3527  017140     060
 3528  017141     015  052012  020117          .ASCIZ  <15><12>'TO RESTORE LOADERS USE SA 210,'
 3529  017146  042522  052123  051117
 3530  017154  020105  047514  042101
 3531  017162  051105  020123  051525
 3532  017170  020105  040523  031040
 3533  017176  030061  000056
 3534  017202  020040  020040  020040  MPRCOR: .ASCIZ  '        = CORE PARITY REGISTER'<15><12>
 3535  017210  020040  020055  047503
 3536  017216  042522  050040  051101
 3537  017224  052111  020131  042522
 3538  017232  044507  052123  051105
 3539  017240  005015     000
 3540  017243     040  020040  020040  MPRMOS: .ASCIZ  '        = MOS PARITY REGISTER'<15><12>
 3541  017250  020040  026440  046440
 3542  017256  051517  050040  051101
```



```
 3543  017264  052111  020131  042522
 3544  017272  044507  052123  051105
 3545  017300  005015     000
 3546          017304                          .EVEN
 3547          000001                          .END
```

```
ADRPT    000516          ADRS   = 077400          ADRTYP   001220          ADTENP   015242
AE     = 000001          BDCNV    015076          BDCNVA   015130          BDCNVB   015162
BDCNVC   015164          BDCNVD   015166          BEGIN    001764          BITPT    000520
BIT0   = 000001          BIT1   = 000002          BIT10  = 002000          BIT11  = 004000
BIT12  = 010000          BIT13  = 020000          BIT14  = 040000          BIT15  = 100000
BIT2   = 000004          BIT3   = 000010          BIT4   = 000020          BIT5   = 000040
BIT6   = 000100          BIT7   = 000200          BIT8   = 000400          BIT9   = 001000
BMOVA    015272          BMOVE    015262          CKBK0    010232          CLRPAR   013364
CNT16    007022          CNT17    007702          CNVCTR   015234          CN16     007042
CONT     011320          CONT10   004112          CONT12   004642          CONT13   005122
CONT3    002434          CONT4    002712          COUNT    004154          DECVAL   015254
DIGIT    015236          DISPLA = 177570          DISPLY = 177570          DONE     011240
DONE10   004140          DONE11   004360          DONE12   004656          DONE13   005136
DONE14   005362          DONE15   006026          DONE16   006356          DONE17   007272
DONE20   010216          DONE4    002756          DONE5    003140          DONE6    003500
DONE7    003612          EMTA     015430          EMTINT   015404          EMTLIM   015472
EMTTAB   015450          EMTX   = 000011          ERR      013144          ERRA     013166
ERRA1    013214          ERRA2    013216          ERRA3    013226          ERRA4    013230
ERRA5    013240          ERRB     013242          ERRBX    013244          ERRC     013250
ERRD     013262          ERROR  = 104002          ERRORP = 104003          ERRORS = 104004
ERRP     013024          ERRST    012746          FTITLE   000512          GMPRA    001542
GMPRB    001670          GMPRC    001676          GMPRD    001732          HIADR    000530
HIWORD   000554          ICNT     016024          IMAX     016022          INDC0    000772
INDC1    000774          INDC10   001016          INDC11   001020          INDC12   001022
INDC13   001024          INDC14   001026          INDC15   001030          INDC2    000776
INDC3    001000          INDC4    001002          INDC5    001004          INDC6    001006
INDC7    001010          INDC8    001012          INDC9    001014          INDX17 = 007554
INST1    004002          INST2    004010          KPAR0    001240          KPAR1    001242
KPAR2    001244          KPAR7    001246          KPDR0    001230          KPDR1    001232
KPDR2    001234          KPDR7    001236          KTSTAR   001216          LDRMSG   017032
LDRSVD   013770          LOCATM   013500          LOCAT1   013576          LOGICA   011300
LOOP10   003704          LOOP11   004234          LOOP12   004450          LOOP13   004742
LOOP14   005220          LOOP15   005652          LOOP21   010712          LOOP4    002600
LOOP5    003034          LOOP6    003210          LOOP7    003552          LOP15    005662
LOP4     002606          LOP5     003042          LOP6     003216          LOWFLG   000556
LSAV     013652          LUP10    003676          LUP11    004220          LUP12    004434
LUP13    004734          LUP14    005230          LUP7     003544          MAPMA    011372
MAPMB    011564          MAPMEM   011360          MAPRB    011750          MAPRC    012050
MAPRD    012150          MAPRE    012244          MAPREG   011706          MAP1     013434
MCON     016232          MCR      016517          MDASH    016514          MDATA    016114
MEMH     001070          MEML     001066          MEMUT    000550          MEQ      016030
MICNT    016054          MK       016522          MLDRSV   017110          MMADR    016305
MMDEV    016252          MMPAT    016341          MMPRS    016372          MPC      016036
MPCNT    016733          MPGEND   016715          MPRAD    016607          MPRCOR   017202
MPRMOS   017243          MPROK    000542          MPR0     000570          MPR1     000600
MPR10    000710          MPR11    000720          MPR12    000730          MPR13    000740
MPR14    000750          MPR15    000760          MPR2     000610          MPR3     000620
MPR4     000630          MPR5     000640          MPR6     000650          MPR7     000660
MPR8     000670          MPP9     000700          MPSER    016742          MPSER1   016772
MPWRF    016676          MSETSR   016211          MSTR     016063          MSTRX    016125
MSTRX1   016151          MSTRX3   016165          MSTRX5   016201          MT       016524
MTIT     016622          MTMAP    016440          MTNUM    016030          MTR      016555
MTREG    016072          MTYCOR   016507          MTYFG    000564          MX1      017001
MX2      017020          NOKT     000552          NOP    = 000240          NOREG    001740
```



```
NRALL    013264          OACDST   015072          OACNT    015074          OACNV    014770
OACNVA   015012          OACNVX   015070          ODDFLG   000560          OPEN   = 000000
PARPAT   001040          PARVEC = 000114          PASCNT   000544          PC     =%000007
PDREND   001226          PDRTAB   001222          PERR   = 100000          PMEMH    001074
PMEML    001072          PMEMX    001076          PS     = 177776          PSADRS   014372
PSCAN    014066          PSCANH   014374          PSCANX   014334          PSCAN1   014376
PSCX1    014510          PSLOOP   014432          PSLUP    014436          PSMSG    017055
PSNXT    014462          PSXTST   014516          PWRDN    015304          PWRUP    015350
RELOC    000586          RESRVD   001032          RESVC    001034          RESVM    001036
RETURN   016026          RSTART   001266          RSTLDR   013772          RSTLDX   014062
RST04  = 104006          RST05S = 104007          RS04     015562          R505S    015610
R0     =%000000          R0SAV    001252          R1     =%000001          R1SAV    001254
R2     =%000002          R2SAV    001256          R3     =%000003          R3SAV    001260
R4     =%000004          R4SAV    001262          R5     =%000005          R5SAV    001264
R6     =%000006          SAVLDR   013654          SAVLDX   013752          SAV04  = 104005
SAV05S = 104010          SCAN     001346          SCANA    001370          SCANB    001374
SCNFLG   001214          SCOPE  = 104001          SCOPEB   015772          SCOPEC   015730
SCOPEG   016004          SHDBE    000534          SP     =%000006          SPSAV    001250
SR     = 177570          SR0    = 177572          START    001426          START1   001510
STKPT  = 000510          SURTEN   015174          SUBTNA   015200          SUBTNB   015214
SV04     015474          SV05A    015524          SV05B    015536          SV05C    015552
SV05S    015514          TBANK    000546          TEMP     000562          TEMPX    000514
TENPWR   015240          TESTX    015644          TEST1    002006          TEST10   003634
TEST11   004156          TEST12   004372          TEST13   004676          TEST14   005156
TEST15   005560          TEST16   006226          TEST17   007066          TEST2    002160
TEST20   010104          TEST21   010506          TEST3    002402          TEST4    002530
TEST5    002776          TEST6    003152          TEST7    003512          TMAP     012422
TMAPA    012436          TMAPB    012542          TMAPC    012630          TMAPD    012674
TMAPEX   012744          TNUM   = 000022          TPB      001102          TPCORE   005372
TPCORX   006042          TPS      001100          TRDATA   000540          TREG     000770
TRFLG    000522          TRP10    004020          TRP10A   004070          TRP12    004542
TRP13    005032          TRP14    005500          TRP15    006146          TRP16    006604
TRP16A   006610          TRP17    007456          TRP17A   007462          TRP20    010352
TRP4A    002726          TRP4B    002734          TRP4C    002742          TRP7     003622
TSTLOC   000532          TSTX     000510          TST10    003722          TST11    004252
TST12    004466          TST13    004756          TST14    005254          TST15    005726
TST4     002620          TST5     003056          TST6     003232          TST7     003566
TYCOR    000526          TYFLG    000524          TYPE   = 104000          TYPSAX   014754
TYPSBX   014764          TYPSX    014736          WAS      000536          WWP    = 000004
WWP16    006374          WWP16A   006412          WWP17    007310          WWP17A   007326
WWP21    010542          WWP21A   010556          XFR1     007746          XFR2     011152
$FILLS   001105          $NULL    001104          $TPFLG   001106          $TYPE    001110
.      = 017304
```

ERRORS DETECTED: 0