MS,MF,MA11-P

MS11, MF11, MA11-P MEMORY
CCMFAF0

AH-7885F-MC

FICHE 1 OF 1

MAR 1978
digital
MADE IN USA

```
.REM    *
```

IDENTIFICATION

PRODUCT CODE: AC-7883F-MC

PRODUCT NAME: CCMFAF0 MS11, MF11, MA11-P MEM

DATE RELEASED: FEB 1978

MAINTAINER: DIAGNOSTIC GROUP

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 TO LOAD AND 8K TO RUN.

3.0 LOADING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER OR XXDP.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

200= NORMAL (WORST CASE) TESTING
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).

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

THE DIAGNOSTIC IS DESIGNED TO USE HARDWARE SWR FOR SYSTEMS HAVING THIS REGISTER, HOWEVER FOR SYSTEMS NOT HAVING HARDWARE SWITCH REGISTER IT WILL USE LOCATION 176 TO GIVE THE FOLLOWING OPTIONS:

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

THE PROGRAM REQUIRES A MINIMUM OF 8K MEMORY TO RUN.
XXDP CHAINING IS POSSIBLE FOR SYSTEMS GREATER THAN 8K.

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 MS-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-D
                        ;COPYRIGHT 1973, 1977, 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                     ;BIT DEFINITIONS
000001                          BIT0=1
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                          DSWR=177570                     ;HARDWARE SWITCH REGISTER
177570                          DDISP=177570                    ;HARDWARE DISPLAY REGISTER
177776                          PS=177776
000007                          PC=%7
000006                          SP=%6
000240                          NOP=240
000000                          OPEN=0
000510                          STKPT=TSTX
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  016032                                      EMTINT
000032  000340                                      340
                    000046                          .=46
000046  011702                                      $ENDAD
                    000052                          .=52
000052  040000                                      BIT14

                                            ;LOAD STARTING ADDRESS AREA
                    000174                          .=174
000174  000000                                      DISPREG:.WORD   0               ;THIS IS THE SOFTWARE DISPLAY REGISTER
000176  000000                                      SWREG:  .WORD   0               ;THIS IS THE SOFTWARE SWITCH REGISTER
000200  000167  001232                              JMP     START                   ;GO TO START OF PROGRAM
                    000210                          .=210
000210  000167  014204                              JMP     RSTLDR                  ;GO RESTORE THE LOADERS
                    000220                          .=220
000220  000167  001132                              JMP     SCAN                    ;SCAN FOR BAD PARITY
                    000230                          .=230
000230  000167  001042                              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          TBANK:  0
000546  000000          MEMUT:  0
000550  000000          NOKT:   0               ;SET TO INDICATE NO KT11 PRESENT
000552  000000          HIWORD: 0
000554  000000          LOWFLG: 0
000556  000000          ODDFLG: 0               ;IF SET INDICATES TESTING HIGH BYTE
                                                ;OF MEMORY LOCATION
000560  000000          TEMP:   0
000562  000000          MTYFG:  0               ;SET TO INDICATE MAP OF PARITY MEMORY
                                                ;ALREADY TYPED
000564  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

000566  172101  MPR0:   172100+1               ;PARITY STATUS REGISTERS
000570  000000          0                      ;0-64K PARITY MEM UNDER THIS CONTROL
000572  000000          0                      ;64-124K PARITY MEM UNDER THIS CONTROL
000574  000000          0                      ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000576  172103  MPR1:   172102+1
000600  000000          0                      ;0-64K PARITY MEM UNDER THIS CONTROL
000602  000000          0                      ;64-124K PARITY MEM UNDER THIS CONTROL
000604  000000          0                      ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000606  172105  MPR2:   172104+1
000610  000000          0                      ;0-64K PARITY MEM UNDER THIS CONTROL
000612  000000          0                      ;64-124K PARITY MEM UNDER THIS CONTROL
```

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

000766  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

000770  000000                  INDC0:  0               ;CORE-MOS PARITY INDICATOR FOR MPR
000772  000000                  INDC1:  0               ;CORE-MOS PARITY IDCICATOR FOR MPR1
000774  000000                  INDC2:  0               ;CORE-MOS PARITY INDICATOR FOR MPR2
000776  000000                  INDC3:  0               ;CORE-MOS PARITY INDICATOR FOR MPR3
001000  000000                  INDC4:  0               ;FOR MPR4
001002  000000                  INDC5:  0               ;FOR MPR5
001004  000000                  INDC6:  0               ;FOR MPR6
001006  000000                  INDC7:  0               ;FOR MPR7
001010  000000                  INDC8:  0               ;FOR MPR8
001012  000000                  INDC9:  0               ;FOR MPR9
001014  000000                  INDC10: 0               ;FOR MPR10
001016  000000                  INDC11: 0               ;FOR MPR11
001020  000000                  INDC12: 0               ;FOR MPR12
001022  000000                  INDC13: 0               ;FOR MPR13
001024  000000                  INDC14: 0               ;FOR MPR14
001026  000000                  INDC15: 0               ;FOR MPR15
001030  000000                  RESRVD: 0

                                                        ;BIT POSITIONS WHICH ARE RESERVED
                                                        ;FOR FUTURE USE IN PARITY REGISTERS
001032  070032                  RESVC:  70032           ;CORE PARITY
001034  077772                  RESVM:  77772           ;MOS PARITY


                                ;PARITY PATTERNS
001036  125325                  PARPAT: 125325          ;EVEN,ODD BYTES
001040  152652                          152652          ;ODD,EVEN
001042  052452                          052452          ;EVEN,ODD
001044  025125                          025125          ;ODD,EVEN
001046  102070                          102070          ;EVEN,EVEN
001050  072527                          072527          ;ODD,ODD
001052  177777                          177777          ;EVEN,EVEN
001054  107030                          107030          ;ODD,ODD
001056  152525                          152525          ;ODD,EVEN
001060  000000                          0               ;EXTRA PATTERN AREA
001062  000000                          0               ;TERMINATOR, DO NOT USE THIS LOC


                                ;THIS IS A MAP OF THE TOTAL MEMORY PRESENT IN THE SYSTEM.
001064  000000                  MEML:   0               ;0-64K MEM PRESENT IN 4K CONTIGUOUS BLOCKS
001066  000000                  MEMH:   0               ;64-124 MEM PRESENT IN 4K CONTIGOUS BLOCKS

                                ;THIS IS A MAP OF THE TOTAL PARITY MEMORY PRESENT IN THE SYSTEM.
001070  000000                  PMEML:  0               ;0-64K PARITY MEMORY PRESENT
```

```
                                                        ;(IN 4K CONTIGUOUS BLOCKS)
001072  000000          PMEMH:  0                     ;64-124K PARITY MEMORY PRESENT
                                                        ;(IN 4K CONTIGUOUS BLOCKS)
001074  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  177570          SWR:    .WORD   DSWR          ;ADDRESS OF THE SWITCH REGISTER
001102  177570          DISPLAY:.WORD   DDISP         ;ADDRESS OF THE DISPLAY REGISTER
001104  177564          TPS:    177564                ;PRINTER STATUS REGISTER ADDRESS
001106  177566          TPB:    177566                ;PRINTER BUFFER REGISTER ADDRESS
001110  000             $NULL:  .BYTE   0             ;CONTAINS NULL CHARACTER FOR FILLS
001111  002             $FILLS: .BYTE   2             ;CONTAINS # OF FILLER CHARACTERS REQUIRED
001112  000             $TPFLG: .BYTE   0             ;"TERMINAL AVAILABLE" FLAG (0=YES)
001113  000                     .BYTE   0             ;RESERVED

001114  105767  177772  $TYPE:  TSTB    $TPFLG        ;IS THERE A TERMINAL?
001120  001401                  BEQ     6$            ;BR IF YES
001122  000000                  HALT                  ;HALT HERE IF NO TERMINAL
001124  010046          6$:     MOV     R0,-(SP)      ;SAVE R0
001126  017600  000002          MOV     @2(SP),R0     ;GET ADDRESS OF ASCIZ STRING
001132  112046          1$:     MOVB    (R0)+,-(SP)   ;PUSH CHARACTER TO BE TYPED ONTO STACK
001134  001005                  BNE     2$            ;BR IF IT ISN'T THE TERMINATOR
001136  005726                  TST     (SP)+         ;IF TERMINATOR POP IT OFF THE STACK
001140  012600                  MOV     (SP)+,R0      ;RESTORE R0
001142  062716  000002  7$:     ADD     #2,(SP)       ;ADJUST RETURN PC
001146  000002                  RTI                   ;RETURN
001150  004767  000026  2$:     JSR     PC,5$         ;GO TYPE THIS CHARACTER
001154  122726  000012  3$:     CMPB    #12,(SP)+     ;CHECK IF THE CHARACTER TYPED
                                                      ;WAS A LINE FEED
001160  001364                  BNE     1$            ;GO GET NEXT CHARACTER IF NOT LINE FEED
001162  016746  177722          MOV     $NULL,-(SP)   ;GET # OF FILLER CHARACTERS NEEDED
                                                      ;AND THE NULL CHARACTER
001166  105366  000001  4$:     DECB    1(SP)         ;DOES A NULL NEED TO BE TYPED?
001172  002770                  BLT     3$            ;BR IF NO--GO POP THE NULL OF THE STACK
001174  004767  000002          JSR     PC,5$         ;GO TYPE A NULL
001200  000772                  BR      4$            ;LOOP
001202  105777  177676  5$:     TSTB    @TPS          ;WAIT UNTIL PRINTER IS READY
001206  100375                  BPL     5$
001210  116677  000002  177670  MOVB    2(SP),@TPB    ;LOAD CHARACTER TO BE
                                                      ;TYPED INTO DATA REGISTER
001216  000207                  RTS     PC


                        ;GENERAL DATA AREA

001220  000000          SCNFLG: 0                     ;SCNFLG GETS SET IF USING
                                                      ;SCAN ROUTINE (SA=220)
```

```
001222  000000                  CACHFL: 0                               ;CACHE FLAG
001224  177746                  CACHE:  177746                          ;CACHE CSR
001226  000000                  KTSTART:        0
001230  000000                  ADRTYP: 0
001232  177600                  PDRTAB: 177600
001234  172200                          172200
001236  172300                  PDREND: 172300
001240  172300                  KPDR0:  172300                          ;KERNEL PAGE DESCRIPTOR REGISTER ADDRESSES
001242  172302                  KPDR1:  172302
001244  172304                  KPDR2:  172304
001246  172316                  KPDR7:  172316
001250  172340                  KPAR0:  172340                          ;KERNEL PAGE ADDRESS REGISTER ADDRESSES
001252  172342                  KPAR1:  172342
001254  172344                  KPAR2:  172344
001256  172356                  KPAR7:  172356
001260  000000                  SPSAV:  0                               ;REGISTER SAVE LOCATIONS
001262  000000                  R0SAV:  0
001264  000000                  R1SAV:  0
001266  000000                  R2SAV:  0
001270  000000                  R3SAV:  0
001272  000000                  R4SAV:  0
001274  000000                  R5SAV:  0


                        ;ROUTINE TO RESTART WITHOUT RETYPING MAP AFTER TEST HAS BEEN RUNNING
001276  012706  000510  RSTART: MOV     #STKPT,SP                       ;SET UP STACK POINTER
001302  012767  000001  177252         MOV     #1,MTYFG                ;SET FLAG TO INDICATE MAP HAS BEEN TYPED
001310  005067  010436               CLR     PASCNT                  ;INITIALIZE PASS COUNT
001314  012737  015732  000024       MOV     #PWRDN,@#24
001322  012737  000340  000026       MOV     #340,@#26
001330  005067  177154               CLR     TSTX
001334  005037  177776               CLR     @#PS                    ;CLEAR PROCESSOR STATUS REGISTER
001340  012737  000006  000004       MOV     #6,@#4
001346  005037  000006               CLR     @#6
001352  000167  000530               JMP     BEGIN


                        ;ROUTINE TO SCAN ALL MEMORY FOR BAD PARITY AND TYPE 18 BIT ADDRESSES OF BAD
                        ;LOCATIONS
001356  012706  000510  SCAN:   MOV     #STKPT,SP                       ;SETUP STACK POINTER
001362  005767  177124          TST     FTITLE                          ;IF TITLE HAS BEEN PRINTED, REGISTERS
                                                                        ;HAVE ALREADY BEEN LOCATED- GO
                                                                        ;AND LOCATE IF NOT ALREADY DONE
001366  001006                  BNE     SCANB                           ;BRANCH, REGISTERS HAVE ALREADY BEEN LOCATED
001370  005267  177624          INC     SCNFLG                          ;INCREMENT SCNFLG
001374  000167  000232          JMP     START1                          ;GO TO LOCATE THE REGISTERS
001400  005067  177614  SCANA:  CLR     SCNFLG                          ;RETURN HERE AFTER LOCATING THE REGISTERS
001404  004767  010346  SCANB:  JSR     PC,MAPMEM                       ;SETUP MEMORY MAP
001410  004767  013100          JSR     PC,PSCAN                        ;SCAN FOR BAD PARITY
001414  104000                  TYPE                                    ;TYPE MESSAGE "BAD PARITY SCAN COMPLETE"
001416  017603                  PSMSG
001420  005767  177124          TST     NOKT
001424  001002                  BNE     .+6
```

```
                                                CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
                                                HALT                    ;END OF PARITY SCAN
                                                BR      SCAN


                                        ;NORMAL STARTUP
                                        START:  MOV     #STKPT,SP       ;SET UP STACK POINTER
                                                CLR     MTYFG           ;CLEAR FLAG WHICH INDICATES MAP TYPED
                                                CLR     PASCNT          ;INITIALIZE PASS COUNT
                                                MOV     #PWRDN,@#24     ;SETUP POWER FAIL RETURN
                                                MOV     #340,@#26
                                                MOV     @#4,-(SP)       ;SAVE THE ERROR VECTOR
                                                MOV     @#6,-(SP)       ;
                                                MOV     #2$,@#4         ;SET UP TIME OUT VECTOR
                                                TST     @SWR            ;TRY TO REFERENCE THE SWITCH REGISTER
                                                BR      4$              ;BRANCH IF NO TIME OUT
                                        2$:     MOV     #SWREG,SWR      ;POINT TO THE SOFTWARE SWITCH REGISTER
                                                MOV     #DISPREG,DISPLAY ;POINT TO THE SOFTWARE DISPLAY REG.
                                                CMP     (SP)+,(SP)+     ;RESTORE THE STACK POINTER
                                          4$:   MOV     #6$,@#4
                                                MOV     #340,@#6
                                                BIS     #14,@#177746    ;DISABLE CACHE
                                                BIS     #200,CACHFL     ;SET FLAG FOR CACHE
                                                BR      5$
                                          6$:   CMP     (SP)+,(SP)+
                                          5$:   MOV     (SP)+,@#6       ;RESTORE ERROR VECTOR
                                                MOV     (SP)+,@#4       ;
                                                CLR     TSTX
                                        1$:     TST     FTITLE          ;IS TITLE PRINTED YET?
                                                BNE     START1          ;YES, SKIP OVER
                                                INC     FTITLE          ;SET FLAG
                                                CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
                                                BEQ     START1          ;YES, SKIP TITLE
                                                TYPE                    ;TYPE TITLE "MEMORY PARITY TEST
                                                MTIT                    ;MAINDEC-11-DCMFA"
                                                TYPE
                                                MLDRSV                  ;TYPE "LOADERS SAVED IN BANK 1.


                                        ;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
                                        START1: TYPE                    ;TYPE "MEMORY PARITY REGISTERS PRESENT ARE:"
                                                MMPRS
                                                CLR     MPROK           ;CLEAR MPR FLAG
                                                MOV     #MPR0,R2        ;SET UP POINTERS
                                                MOV     #INDC0,R3       ;POINTER TO CORE-MOS
                                                MOV     #GMPRB,@#4      ;SET UP TIMEOUT TRAP RETURN
                                                CLR     @#6
                                        GMPRA:  BIC     #1,(2)          ;CLEAR FLAG BIT IN TABLE
                                                CLR     2(R2)           ;INITIALIZE LOCATIONS IN THE TABLE
                                                CLR     4(R2)
```

```
                                        CLR     6(R2)
                                        TST     @(2)                    ;DOES THIS MPR EXIST? (IF NO, TIMES OUT)
                                        BIS     #340,@(2)               ;YES- IS IT AN MFII-LP OR MA11-P CORE PARITY REG
                                        BIT     #340,@(2)
                                        BEQ     1$                      ;NO, IS IT A MOS-11 PARITY REGISTER?  BRANCH
                                        MOV     (2),TEMPX               ;YES- PRINT REGISTER ADDRESS
                                        JSR     R5,@ACNV                ;(GET ASCII)
                                        TEMPX
                                        MPRCOR
                                        6
                                        TYPE                            ;(TYPE ADDRESS)
                                        MPRCOR
                                        MOV     #1,(R3)                 ;SET INDICATOR FOR CORE PARITY
                                        BR      2$
                                1$:     MOV     (2),TEMPX               ;IT IS A MOS REGISTER, PRINT ADDRESS
                                        JSR     R5,@ACNV                ;(GET ASCII)
                                        TEMPX
                                        MPRMOS
                                        6
                                        TYPE                            ;(TYPE ADDRESS)
                                        MPRMOS
                                        MOV     #-1,(R3)                ;SET INDICATOR FOR MOS PARITY
                                2$:     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"
                                        MTR
                                        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
                                        TSTB    CACHFL                  ;CACHE ?
                                        BEQ     .+10                    ;BRANCH IF NO
                                        MOV     #14,@CACHE              ;DISABLE CACHE
                                        JSR     PC,$AVLDR               ;SAVE MONITOR
                                        MOV     #100,IMAX               ;MAXIMUM ITERATION COUNT
                                        MOV     #6,@#4                  ;RESTORE TRAPCATCHER IN TIMEOUT VECTOR
                                        CLR     BITPT
```

```
                                ;*****************************************************************
                                ;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
                                ;*****************************************************************
002154  104001                  TEST1:  SCOPE
002156  012777  000001  176716          MOV     #1,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
002164  012700  000566                  MOV     #MPRO,R0                ;LOAD ADDRESS OF TABLE INTO R0
002170  012704  000770                  MOV     #INDCO,R4               ;LOAD ADDRESS OF INDICATOR IN R4
002174  032710  000001          1$:     BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
002200  001042                          BNE     4$                      ;NO- BRANCH TO GET NEXT ADDRESS
002202  011001                          MOV     @R0,R1                  ;YES- LOAD R1 WITH ADDRESS OF
                                                                        ;PARITY REGISTER
002204  022714  000001                  CMP     #1,(R4)                 ;IS THIS REGISTER CORE?
002210  001004                          BNE     5$                      ;NO
002212  016767  176614  176610          MOV     RESVC,RESRVD            ;YES, CORE. STORE RESERVED BITS
002220  000403                          BR      .+10
002222  016767  176606  176600  5$:     MOV     RESVM,RESRVD            ;MOS. STORE RESERVED BITS
002230  012702  000001                  MOV     #1,R2                   ;LOAD R2 WITH VALUE OF FIRST BIT
                                                                        ;TO BE TESTED
002234  005011                          CLR     @R1                     ;INITIALIZE PARITY REGISTER
002236  011167  176524                  MOV     @R1,TREG                ;READ CONTENTS OF PARITY REGISTER
002242  046767  176562  176516          BIC     RESRVD,TREG             ;CLEAR BITS WHICH ARE RESERVED
002250  001401                          BEQ     .+4                     ;CHECK OTHER BITS- BRANCH IF OK
002252  104002                          ERROR                           ;CLEAR INSTRUCTION DID NOT INITIALIZE
                                                                        ;ALL USED BITS IN PARITY REGISTER
                                                                        ;TO ZERO (R1 CONTAINS ADDRESS OF
                                                                        ;FAILING REGISTER)
002254  030267  176550          2$:     BIT     R2,RESRVD               ;IS THIS BIT RESERVED?
002260  001010                          BNE     3$                      ;YES- DON'T TEST IT SINCE IT
                                                                        ;MAY BE ZERO OR ONE
002262  010211                          MOV     R2,@R1                  ;NO- SET THIS BIT IN THE PARITY REGISTER
002264  011103                          MOV     @R1,R3                  ;READ AND SAVE CONTENTS OF PARITY REGISTER
002266  005011                          CLR     @R1                     ;CLEAR PARITY REGISTER
002270  046703  176534                  BIC     RESRVD,R3               ;CLEAR BIT LOCATIONS THAT ARE
                                                                        ;RESERVED
002274  020203                          CMP     R2,R3                   ;CHECK REST
002276  001401                          BEQ     .+4                     ;BRANCH IF OK
002300  104002                          ERROR                           ;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
002302  006302                  3$:     ASL     R2                      ;ROTATE BIT TO BE TESTED
002304  103363                          BCC     2$                      ;IF NOT DONE WITH ALL BIT POSITIONS
                                                                        ;GO TEST THIS ONE
002306  062700  000010          4$:     ADD     #10,R0                  ;MOVE R0 TO POINT TO NEXT POSSIBLE ADDRESS
002312  005724                          TST     (R4)+                   ;OF A PARITY REGISTER
002314  020027  000766                  CMP     R0,#TREG                ;AT END OF TABLE?
002320  002725                          BLT     1$                      ;NO, BRANCH
002322  005067  176440                  CLR     TREG
                                ;*****************************************************************
                                ;SHOW THAT RESET CLEARS BITS 0,2, AND 15 OF EACH PARITY REGISTER
```

```
                                        ;PRESENT.
                                        ;*************************************************************************
                                TEST2:  SCOPE
002326  104001                          MOV     #2,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
002330  012777  000002  176544          CLR     IMAX                    ;DON'T ITERATE TEST
002336  005067  014114                  MOV     #MPR0,R0                ;LOAD POINTER
002342  012700  000566                  MOV     #INDC0,R3               ;POINTER TO INDICATOR
002346  012703  000770                  BIT     #1,@R0                  ;IS THIS PARITY REGISTER PRESENT?
002352  032710  000001          1$:     BNE     5$                      ;NO BRANCH
002356  001012                          CMP     #1,(R3)                 ;IS THIS CORE OR MOS PARITY REGISTER?
002360  022713  000001                  BEQ     6$                      ;NO- BRANCH
002364  001404                          MOV     #100015,@(R0)           ;MOS-SET ALL DEFINED BITS TO 1
002366  012770  100015  000000          BR      .+10
002374  000403
002376  012770  107745  000000  6$:     MOV     #107745,@(R0)           ;CORE- SET ALL DEFINED BITS TO 1
002404  062700  000010          5$:     ADD     #10,R0                  ;MOVE POINTER TO POINT TO NEXT MPR ADDRESS
002410  005723                          TST     (R3)+                   ;INCREMENT POINTER TO INDICATOR
                                                                        ;OF A PARITY REGISTER
002412  020027  000766                  CMP     R0,#TREG                ;AT END OF TABLE?

002416  002755                          BLT     1$                      ;NO- CONTINUE
002420  105767  176466                  TSTB    $TPFLG                  ;YES- TERMINAL AVAILABLE?
002424  001003                          BNE     4$                      ;NO- BRANCH
002426  105777  176452                  TSTB    @TPS                    ;YES- WAIT FOR TERMINAL TO FINISH
002432  100375                          BPL     .-4
002434  000005          4$:             RESET                           ;ISSUE INIT
002436  012700  000566                  MOV     #MPR0,R0                ;LOAD ADDRESS OF THE TABLE
002442  012703  000770                  MOV     #INDC0,R3               ;POINTER TO INDICATOR
002446  105767  176550                  TSTB    CACHFL                  ;CACHE ?
002452  001403                          BEQ     2$                      ;BRANCH IF NO
002454  012777  000014  176542          MOV     #14,@CACHE              ;DISABLE CACHE
002462  032710  000001          2$:     BIT     #1,@R0                  ;IS THIS PARITY REGISTER PRESENT?
002466  001030                          BNE     3$                      ;NO- BRANCH
002470  022713  000001                  CMP     #1,(R3)                 ;IS THIS A CORE PAR REGISTER?
002474  001012                          BNE     7$                      ;NO, BRANCH
002476  017002  000000                  MOV     @(R0),R2                ;YES, GET CONTENTS OF REGISTER
002502  005070  000000                  CLR     @(R0)                   ;MAKE SURE THAT WWP AND AE ARE CLEAR
002506  042702  077772                  BIC     #77772,R2               ;MASK RESERVED BITS FOR CORE PAR
                                                                        ;-ITY REGISTER. BITS 5-11(ADDRS
                                                                        ; BITS) ARE ALSO MASKED
002512  005702                          TST     R2                      ;CHECK, IF REST WERE CLEARED
002514  001401                          BEQ     .+4
002516  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

002520  000411                          BR      3$-4
002522  017002  000000          .7$:    MOV     @(R0),R2                ;MOS, GET CONTENTS OF REGISTER
002526  005070  000000                  CLR     @(R0)                   ;MAKE SURE THAT WWP BAE ARE CLEAR
002532  046702  176276                  BIC     RESVM,R2                ;MASK RESERVED BITS FOR MOS PAR REG
002536  005702                          TST     R2                      ;CHECK REST
002540  001401                          BEQ     .+4                     ;RESET DID CLEAR ALL BITS
002542  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)
```

```
                        002544  005070  000000          3$:     CLR     @(R0)                   ;REINITIALIZE PARITY REGISTER
                        002550  062700  000010                  ADD     #10,R0                  ;MOVE POINTER TO POINT TO ADDRESS
                                                                                                ;OF NEXT REGISTER
                        002554  005723                          TST     (R3)+                   ;INCREMENT POINTER TO INDICATOR
                        002556  020027  000766                  CMP     R0,#TREG                ;DONE?
                        002562  002737                          BLT     2$                      ;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 "MPRO" 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.
                                                ;*************************************************************************
                        002564  104001          TEST3:  SCOPE
                        002566  012777  000003  176306          MOV     #3,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
                        002574  005767  175762                  TST     MTYFG                   ;IF MAPPING HAS ALREADY BEEN DONE
                        002600  001044                          BNE     TEST4                   ;SKIP SUBTEST
                        002602  004767  011204                  JSR     %7,CLRPAR
                        002606  004767  007144                  JSR     %7,MAPMEM               ;MAP MEMORY
                        002612  004767  007466                  JSR     %7,MAPREG               ;FIND PARITY MEMORY AND CORRESPONDING
                                                                                                ;REGISTERS USING WRITE WRONG PARITY
                                                                                                ;WITHOUT ACTION ENABLE SET
                        002616  005067  176246          CONT3:  CLR     PMEML                   ;INITIALIZE LOCATIONS INDICATING
                        002622  005067  176244                  CLR     PMEMH                   ;TOTAL PARITY MEMORY PRESENT
                        002626  012701  000566                  MOV     #MPRO,R1
                        002632  032711  000001          1$:     BIT     #1,@R1
                        002636  001006                          BNE     2$
                        002640  056167  000002  176222          BIS     2(R1),PMEML             ;FLAG EXISTING PARITY MEMORY (LOW 64K)
                        002646  056167  000004  176216          BIS     4(R1),PMEMH             ;FLAG EXISTING PARITY MEMORY (HIGH 64K)
                        002654  062701  000010          2$:     ADD     #10,R1
                        002660  020127  000766                  CMP     R1,#TREG
                        002664  103762                          BLO     1$
                        002666  004767  010126                  JSR     %7,TMAP                 ;TYPE MAP
                        002672  005267  175664                  INC     MTYFG                   ;INDICATE MAPPING DONE
                        002676  032777  001000  176174          BIT     #BIT9,@SWR              ;SWITCH 9 SET?
                        002704  001402                          BEQ     .+6                     ;NO- BRANCH
                        002706  000000                          HALT                            ;YES- SWITCH 9 SET INDICATING HALT
                                                                                                ;AFTER TYPING PARITY MEMORY MAP
                        002710  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
                                        ;*************************************************************************
002712  104001                  TEST4:  SCOPE
002714  012777  000004  176160          MOV     #4,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
002722  012767  000100  013526          MOV     #100,IMAX
002730  004767  011056                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
002734  005767  175610                  TST     NOKT                    ;KT11 PRESENT?
002740  001004                          BNE     1$                      ;NO-BRANCH
002742  004767  010744                  JSR     %7,NRALL                ;YES-MAP ALL PAGES NON RESIDENT
002746  004767  011110                  JSR     %7,MAP1                 ;THEN MAP KERNEL 0 TO BANK 0, KERNEL
                                                                        ;7 TO EXTERNAL BANK, SET KERNEL
                                                                        ;0,1, AND 7 RW, AND TURN ON KT11
002752  012700  000566          1$:     MOV     #MPR0,R0                ;SETUP TO FIND REGISTERS PRESENT
002756  012702  00077C                  MOV     #INDC0,R2
002762  032710  000001          LOOP4:  BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
002766  001405                          BEQ     TST4                    ;YES-BRANCH TO TEST IT
002770  062700  000010          LOP4:   ADD     #10,R0                  ;NO-CHECK FOR ANOTHER ONE
002774  005722                          TST     (R2)+                   ;INCREMENT PTR9LT
002776  103771                          BLO     LOOP4
003000  000457                          BR      DONE4                   ;BRANCH WHEN ALL REGISTERS HAVE BEEN TESTED
003002  004767  011120          TST4:   JSR     %7,LOCATM               ;LOCATE MEMORY CORRESPONDING TO
                                                                        ;THIS REGISTER- IF NO KT11, R1 SHOULD
                                                                        ;BE RETURNED CONTAINING THE ADDRESS
                                                                        ;OF THE 1ST LOCATION CONTROLLED BY THIS
                                                                        ;REGISTER (INCLUDING EXTERNAL INTERLEAVE
                                                                        ;OFFSET IF NEEDED)
                                                                        ;IF KT11 IS PRESENT, R1 SHOULD BE RETURNED
                                                                        ;POINTING TO THE 1ST LOCATION CONTROLLED
                                                                        ;BY THIS REGISTER, MAPPED THRU KERNEL
                                                                        ;PAGE 1. KERNEL PAGE 1 SHOULD BE
                                                                        ;MAPPED TO THE CORRECT BANK
003006  032701  000001                  BIT     #1,R1                   ;IS ERROR RETURN INDICATED?
003012  001403                          BEQ     .+10                    ;NO- BRANCH
003014  104002                          ERROR                           ;MAP INDICATES NO PARITY MEMORY
                                                                        ;IS CONTROLLED BY THIS REGISTER
                                                                        ;R0 POINTS TO THE ADDRESS OF THE
                                                                        ;PARITY REGISTER
003016  000167  177746                  JMP     LOP4
003022  012737  003110  000114          MOV     #TRP4A,@#114            ;SETUP PARITY TRAP RETURN
003030  012770  000004  000000          MOV     #WWP,@(R0)              ;SET WRITE WRONG PARITY
003036  011111                          MOV     @R1,@R1                 ;WRITE CONTENTS OF LOCATION WITH
                                                                        ;WRONG PARITY
003040  005070  000000                  CLR     @(R0)                   ;CLEAR PARITY REGISTER
003044  005711                          TST     @R1                     ;READ BAD PARITY WITH ACTION ENABLE
                                                                        ;CLEARED- NO TRAP EXPECTED
003046  012737  003116  000114          MOV     #TRP4B,@#PARVEC         ;CHANGE PARITY TRAP RETURN
003054  052770  000001  000000          BIS     #AE,@(R0)               ;SET ACTION ENABLE WITH PARITY ERROR
                                                                        ;ALREADY SET- SHOULDN'T TRAP YET
003062  012737  003124  000114          MOV     #TRP4C,@#PARVEC         ;CHANGE PARITY TRAP RETURN
003070  005711                          TST     @R1                     ;READ LOCATION AGAIN- SHOULD GET
                                                                        ;A PARITY TRAP DUE TO READING BAD
                                                                        ;PARITY WITH ACTION ENABLE SET
003072  104002                          ERROR                           ;NO PARITY TRAP AFTER READING LOCATION
                                                                        ;WHICH SHOULD CONTAIN BAD PARITY-
                                                                        ;R1 CONTAINS ADDRESS OF MEMORY LOCATION
```

```
1049
1050
1051
1052  003074  005070  000000          CONT4:  CLR     @(R0)                   ;(VIRTUAL, THRU KERNEL PAGE 1, IF KT11
1053  003100  005511                          ADC     @R1                     ;PRESENT). R0 POINTS TO THE ADDRESS OF
1054  003102  005070  000000                  CLR     @(R0)                   ;THE PARITY REGISTER IN WHICH AE WAS SET
1055  003106  000730                          BR      LOP4                    ;CLEAR PARITY REGISTER
1056  003110  104002                  TRP4A:  ERROR                           ;CLEAR BAD PARITY
1057                                                                          ;CLEAR PARITY ERROR BIT
1058                                                                          ;GO TO TEST NEXT REGISTER
1059                                                                          ;PARITY TRAP OCCURRED WITH ACTION
1060                                                                          ;ENABLE CLEAR- R0 POINTS TO THE ADDRESS
1061  003112  022626                          CMP     (SP)+,(SP)+             ;OF THE PARITY REGISTER UNDER TEST
1062  003114  000767                          BR      CONT4                   ;R1 CONTAINS THE ADDRESS OF THE MEMORY
1063  003116  104002                  TRP4B:  ERROR                           ;UNDER TEST (VIRTUAL IF KT11 IS PRESENT)
1064                                                                          ;RESTORE STACK POINTER
1065
1066                                                                          ;PARITY TRAP OCCURRED WHEN ACTION
1067                                                                          ;ENABLE WAS SET WITH PARITY ERROR
1068                                                                          ;ALREADY SET
1069                                                                          ;R0 POINTS TO THE ADDRESS OF THE
1070  003120  022626                          CMP     (SP)+,(SP)+             ;PARITY REGISTER UNDER TEST
1071  003122  000764                          BR      CONT4                   ;R1 CONTAINS THE MEMORY ADDRESS UNDER
1072  003124  005770  000000          TRP4C:  TST     @(R0)                   ;TEST (VIRTUAL IF KT11 IS PRESENT)
1073  003130  100401                          BMI     .+4                     ;RESTORE STACK POINTER
1074  003132  104002                          ERROR                           ;ERROR BIT SET AFTER PARITY TRAP?
1075                                                                          ;YES- BRANCH
1076                                                                          ;ERROR BIT NOT SET AFTER PARITY
1077  003134  022626                          CMP     (SP)+,(SP)+             ;TRAP- R0 POINTS TO THE ADDRESS
1078  003136  000756                          BR      CONT4                   ;OF THE PARITY REGISTER UNDER TEST
1079  003140  012737  000116  000114  DONE4:  MOV     #PARVEC+2,@#PARVEC      ;RESTORE STACK POINTER
1080  003146  005767  175376                  TST     NOKT                    ;RESTORE TRAP CATCHER
1081  003152  001002                          BNE     .+6
1082  003154  005037  177572                  CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1083
1084
1085                                  ;*****************************************************************************
1086                                  ;SHOW THAT READING GOOD PARITY AFTER BAD PARITY DOESN'T CLEAR PARITY ERROR BIT
1087                                  ;*****************************************************************************
1088  003160  104001                  TEST5:  SCOPE
1089  003162  012777  000005  175712          MOV     #5,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
1090  003170  004767  010616                  JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1091  003174  005767  175350                  TST     NOKT                    ;KT11 PRESENT?
1092  003200  001004                          BNE     1$                      ;NO, BRANCH
1093  003202  004767  010504                  JSR     PC,NRALL                ;YES, MAP ALL PAGES NON-RESIDENT
1094  003206  004767  010650                  JSR     PC,MAP1                 ;THEN MAP KERNEL 0 TO BANK 0. MAP KERNEL
1095                                                                          ;7 TO THE EXTERNAL BANK. SET KERNEL
1096                                                                          ;0,1, AND 7 RW, AND TURN ON KT11
1097  003212  012700  000566          1$:     MOV     #MPR0,R0                ;SETUP TO FIND REGISTERS PRESENT
1098  003216  032710  000001          LOOP5:  BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
1099  003222  001406                          BEQ     TST5                    ;YES- BRANCH TO TEST IT
1100  003224  062700  000010          LOP5:   ADD     #10,R0                  ;NO- CHECK FOR ANOTHER ONE
1101  003230  020027  000766                  CMP     R0,#TREG
1102  003234  103770                          BLO     LOOP5
1103  003236  000431                          BR      DONE5                   ;EXIT WHEN ALL REGISTERS HAVE BEEN TESTED
1104  003240  004767  010662          TST5:   JSR     PC,LOCATM               ;LOCATE MEMORY CORRESPONDING TO  THIS
```

```
1105
1106
1107
1108
1109  003244  032701  000001            BIT     #1,R1           ;REGISTER- R1 WILL BE RETURNED CONTAINING
1110  003250  001403                    BEQ     .+10            ;THE ADDRESS OF THE FIRST LOCATION
1111  003252  104002                    ERROR                   ;CONTROLLED BY THIS REGISTER (MAPPED
1112                                                            ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1113                                                            ;IS ERROR RETURN INDICATED?
1114                                                            ;NO- BRANCH
1115  003254  000167  177744            JMP     LOP5            ;MAP INDICATES NO PARITY MEMORY IS
1116  003260  012770  000004  000000    MOV     #WWP,@(R0)      ;CONTROLLED BY THIS REGISTER. R0
1117  003266  011111                    MOV     @R1,@R1         ;POINTS TO THE ADDRESS OF THE
1118                                                            ;PARITY REGISTER
1119  003270  005711                    TST     @R1             ;SET WWP IN THIS REGISTER
1120  003272  042770  000004  000000    BIC     #WWP,@(R0)      ;WRITE CONTENTS OF LOCATION WITH
1121  003300  011111                    MOV     @R1,@R1         ;WRONG PARITY
1122  003302  005711                    TST     @R1             ;DETECT WRONG PARITY
1123  003304  005770  000000            TST     @(R0)           ;CLEAR WWP IN PARITY REGISTER
1124  003310  100401                    BMI     .+4             ;RESTORE GOOD PARITY
1125  003312  104002                    ERROR                   ;REREAD LOCATION
1126                                                            ;READ CONTENTS OF PARITY REGISTER
1127                                                            ;BRANCH IF PARITY ERROR IS STILL SET
1128                                                            ;PARITY ERROR BIT CLEARED BY READING
1129  003314  005070  000000            CLR     @(R0)           ;GOOD PARITY (OR POSSIBLY WHILE DOING
1130  003320  000741                    BR      LOP5            ;THE BIC TO CLEAR WWP). R0 POINTS TO
1131  003322  005767  175222  DONE5:   TST     NOKT            ;THE PARITY REGISTER ADDRESS.
1132  003326  001002                    BNE     .+6             ;CLEAR THE PARITY REGISTER
1133  003330  005037  177572            CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1134
1135
1136                                    ;*************************************************************************
1137                                    ;SHOW THAT PARITY GENERATE AND DETECT LOGIC WORKS CORRECTLY FOR EACH BYTE
1138                                    ;SHOW THAT WRITE WRONG PARITY WORKS FOR HIGH AND LOW BYTES
1139                                    ;SHOW THAT WRITING INTO LOCATION WHEN WRITE WRONG PARITY IS NOT SET
1140                                    ;RESTORES GOOD PARITY
1141                                    ;*************************************************************************
1142  003334  104001            TEST6:  SCOPE
1143  003336  012777  000006  175536    MOV     #6,@DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY
1144  003344  004767  010442            JSR     %7,CLRPAR       ;CLEAR ALL PARITY REGISTERS
1145  003350  005767  175174            TST     NOKT            ;KT11 PRESENT?
1146  003354  001004                    BNE     1$              ;NO, BRANCH
1147  003356  004767  010330            JSR     %7,NRALL        ;YES- MAP IT (KERNEL 0 TO BANK 0, RW;
1148  003362  004767  010474            JSR     %7,MAP1         ;KERNEL 7 TO EXTERNAL BANK, RW; KERNEL 1 RW)
1149                                                            ;AND TURN IT ON
1150  003366  012700  000566    1$:     MOV     #MPR0,R0        ;SETUP TO FIND REGISTERS PRESENT
1151  003372  032710  000001    LOOP6:  BIT     #1,@R0          ;IS THIS REGISTER PRESENT?
1152  003376  001406                    BEQ     TST6            ;YES- BRANCH TO TEST IT
1153  003400  062700  000010    LOP6:   ADD     #10,R0          ;NO- CHECK FOR ANOTHER ONE
1154  003404  020027  000766            CMP     R0,#TREG
1155  003410  103770                    BLO     LOOP6
1156  003412  000523                    BR      DONE6           ;BRANCH TO DONE IF ALL REGISTERS
1157                                                            ;HAVE BEEN TESTED
1158  003414  004767  010506    TST6:   JSR     %7,LOCATM       ;LOCATE MEMORY CORRESPONDING TO
1159                                                            ;THIS REGISTER- R1 SHOULD BE RETURNED
1160                                                            ;CONTAINING THE ADDRESS OF THE FIRST
```

```
                                                        ;LOCATION CONTROLLED BY THIS REGISTER
                                                        ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 PRESENT)
003420  032701  000001          BIT     #1,R1           ;IF NO MEMORY WAS FOUND TO CORRESPOND
003424  001403                  BEQ     .+10            ;ODD ADDRESS IS RETURNED-BRANCH IF OK
003426  104002                  ERROR                   ;MAP INDICATES NO PARITY MEMORY IS
                                                        ;CONTROLLED BY THIS REGISTER
                                                        ;R0 POINTS TO THE ADDRESS OF THE
                                                        ;PARITY REGISTER
003430  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
003434  005011                  CLR     @R1             ;INITIALIZE LOCATION UNDER TEST
003436  005070  000000          CLR     @(R0)           ;INITIALLY CLEAR PARITY REGISTER
003442  005002                  CLR     R2              ;R2 CONTAINS VALUE TO BE LOADED
                                                        ;INTO MEMORY
003444  110211          1$:     MOVB    R2,@R1          ;WRITE VALUE INTO LOW BYTE
003446  005711                  TST     @R1             ;READ WORD TO CHECK PARITY
003450  005770  000000          TST     @(R0)           ;CHECK PARITY REGISTER
003454  100006                  BPL     5$              ;BRANCH IF ERROR NOT SET
003456  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)
003460  005070  000000          CLR     @(R0)           ;CLEAR ERROR BIT
003464  005011                  CLR     @R1             ;REINITIALIZE TEST LOCATION
003466  005002                  CLR     R2              ;REINITIALIZE VALUE TO BE USED
003470  000402                  BR      2$
003472  105202          5$:     INCB    R2              ;INCREMENT VALUE TO BE LOADED
003474  001363                  BNE     1$              ;LOOP UNTIL ALL VALUES HAVE BEEN USED
003476  110261  000001  2$:     MOVB    R2,1(R1)        ;WRITE ALL VALUES INTO HIGH BYTE
003502  005711                  TST     @R1             ;READ WORD TO CHECK PARITY
003504  005770  000000          TST     @(R0)           ;CHECK PARITY REGISTER
003510  100002                  BPL     .+6             ;BRANCH IF ERROR NOT SET
003512  104002                  ERROR                   ;PARITY ERROR SET WHEN VALUE IN R2
                                                        ;WAS WRITTEN AND READ BACK FROM HIGH BYTE
                                                        ;OF LOCATION WHOSE ADDRESS IS CONTAINED
                                                        ;IN R1 (WWP WAS NOT SET)
003514  000402                  BR      6$              ;INCREMENT VALUE TO BE LOADED
003516  105202                  INCB    R2              ;LOOP UNTIL ALL VALUES HAVE BEEN USED
003520  001366                  BNE     2$

                        ;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
003522  005011          6$:     CLR     @R1             ;INITIALIZE LOCATION UNDER TEST
003524  005002                  CLR     R2              ;INITIALIZE VALUE TO BE WRITTEN
003526  012770  000004  000000  3$:     MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
003534  110211                  MOVB    R2,@R1          ;WRITE WRONG PARITY IN LOW BYTE
003536  005070  000000          CLR     @(R0)           ;CLEAR WRITE WRONG PARITY, AND CLEAR
                                                        ;PARITY ERROR IF SET
003542  005711                  TST     @R1             ;READ BACK WRONG PARITY
003544  005770  000000          TST     @(R0)           ;PARITY ERROR SET?
003550  100402                  BMI     .+6             ;YES-BRANCH
003552  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
```

```
1217
1218
1219
1220
1221  003554  000402                          BR      .+6
1222  003556  105202                          INCB    R2
1223  003560  001362                          BNE     3$
1224  003562  005011                          CLR     @R1
1225  003564  005070  000000                  CLR     @(R0)
1226  003570  005711                          TST     @R1
1227
1228  003572  005770  000000                  TST     @(R0)
1229  003576  100001                          BPL     .+4
1230  003600  104002                          ERROR
1231
1232
1233  003602  012770  000004  000000  4$:     MOV     #WWP,@(R0)
1234  003610  110261  000001                  MOVB    R2,1(R1)
1235  003614  005070  000000                  CLR     @(R0)
1236
1237  003620  005711                          TST     @R1
1238  003622  005770  000000                  TST     @(R0)
1239  003626  100402                          BMI     .+6
1240  003630  104002                          ERROR
1241
1242
1243
1244
1245
1246
1247
1248
1249  003632  000402                          BR      .+6
1250  003634  105202                          INCB    R2
1251  003636  001361                          BNE     4$
1252  003640  005011                          CLR     @R1
1253  003642  005070  000000                  CLR     @(R0)
1254  003646  005711                          TST     @R1
1255
1256  003650  005770  000000                  TST     @(R0)
1257  003654  100001                          BPL     .+4
1258  003656  104002                          ERROR
1259
1260
1261  003660  000647                          BR      LOP6
1262  003662  005767  174662          DONE6:  TST     NOKT
1263  003666  001002                          BNE     .+6
1264  003670  005037  177572                  CLR     @#SR0
1265
1266
1267
1268                                  ;*****************************************************************
1269                                  ;SHOW THAT SETTING PARITY ERROR AFTER SETTING ACTION ENABLE WON'T CAUSE A TRAP
1270                                  ;*****************************************************************
1271  003674  104001                  TEST7:  SCOPE
1272  003676  012777  000007  175176          MOV     #7,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
```

```
;REGISTER. R1 CONTAINS ADDRESS OF LOCATION
;BEING TESTED (VIRTUAL, THRU KERNEL
;PAGE 1 IF KT11 IS PRESENT). R2
;CONTAINS THE VALUE WRITTEN
;EXIT LOOP AFTER ERROR
;INCREMENT DATA
;LOOP TILL DONE WITH ALL VALUES
;REINITIALIZE LOCATION TO CLEAR BAD PARITY
;CLEAR ERROR IF SET
;READ LOCATION, WHICH SHOULD NOW HAVE
;GOOD PARITY
;PARITY ERROR SET?
;NO, BRANCH
;GOOD PARITY WAS NOT RESTORED BY
;WRITING INTO THE LOCATION WITH
;WRITE WRONG PARITY CLEARED
;SET WRITE WRONG PARITY
;WRITE WRONG PARITY IN HIGH BYTE
;CLEAR WRITE WRONG PARITY AND PARITY
;ERROR IF SET
;READ BACK WRONG PARITY
;PARITY ERROR SET?
;YES-BRANCH
;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
;EXIT LOOP AFTER ERROR
;INCREMENT DATA
;LOOP TILL DONE WITH ALL VALUES
;CLEAR BAD PARITY IN TEST LOCATION
;CLEAR PARITY ERROR BIT IF SET
;READ LOCATION, WHICH SHOULD NOW
;HAVE GOOD PARITY
;CHECK PARITY ERROR BIT
;BRANCH IF CLEAR
;WRITING INTO LOCATION WHEN WRITE
;WRONG PARITY WAS NOT SET DID NOT
;WRITE GOOD PARITY
;GO CHECK FOR ANOTHER PARITY REGISTER

;TURN OFF KT11 IF PRESENT
```

```
1273  003704  004767  010102                      JSR     %7,CLRPAR       ;INITIALLY CLEAR ALL PARITY REGISTERS
1274  003710  012737  004004  000114              MOV     #TRP7,@#PARVEC  ;SETUP PARITY TRAP RETURN
1275  003716  005037  000116                      CLR     @#PARVEC+2
1276  003722  012700  000566                      MOV     #MPR0,R0        ;SETUP TO GET ADDRESS OF REGISTER PRESENT
1277  003726  032710  000001          LUP7:       BIT     #1,@R0
1278  003732  001406                              BEQ     TST7            ;BRANCH TO TEST REGISTER
1279  003734  062700  000010          LOOP7:      ADD     #10,R0
1280  003740  020027  000766                      CMP     R0,#TREG
1281  003744  103770                              BLO     LUP7
1282  003746  000412                              BR      DONE7           ;BRANCH IF DONE TESTING ALL REGISTERS
1283  003750  012770  000001  000000  TST7:       MOV     #AE,@(R0)       ;SET ACTION ENABLE
1284  003756  052770  100000  000000              BIS     #PERR,@(R0)     ;SET PARITY ERROR
1285  003764  000240                              NOP                     ;SHOULD NOT TRAP
1286  003766  005070  000000                      CLR     @(R0)           ;CLEAR PARITY REGISTER
1287  003772  000760                              BR      LOOP7           ;GO CHECK NEXT REGISTER
1288  003774  012737  000116  000114  DONE7:     MOV     #PARVEC+2,@#PARVEC
1289  004002  000405                              BR      TEST10
1290  004004  104002                  TRP7:       ERROR                   ;TRAP OCCURRED WHEN PARITY ERROR BIT
1291                                                                      ;WAS SET VIA A BIS INSTRUCTION
1292                                                                      ;WITH ACTION ENABLE ALREADY SET.
1293                                                                      ;R0 POINTS TO THE ADDRESS OF THE
1294                                                                      ;PARITY REGISTER
1295  004006  022626                              CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1296  004010  005070  000000                      CLR     @(R0)           ;CLEAR PARITY REGISTER
1297  004014  000747                              BR      LOOP7
1298
1299
1300
1301                                      ;*****************************************************************************
1302                                      ;SHOW THAT REPEATED PARITY ERRORS WILL CAUSE REPEATED TRAPS IF ACTION
1303                                      ;ENABLE IS SET AND PARITY ERROR IS LEFT SET.
1304                                      ;SHOW THAT THE ERROR ADDRESS BITS (11-5) TRACK (ONLY FOR CORE PARITY REGISTERS)
1305                                      ;*****************************************************************************
```

```
                                TEST10: SCOPE
004016  104001                          
004020  012777  000010  175054          MOV     #10,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
004026  004767  007760                  JSR     %7,CLRPAR       ;INITIALLY CLEAR ALL PARITY REGISTERS
004032  005767  174512                  TST     NOKT            ;KT11 PRESENT?
004036  001004                          BNE     1$              ;NO- BRANCH
004040  004767  007646                  JSR     %7,NRALL        ;YES- INITIALLY MAP ALL PAGES NR
004044  004767  010012                  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
004050  012700  000566          1$:     MOV     #MPR0,R0
004054  012702  000770                  MOV     #INDC0,R2
004060  032710  000001          LUP10:  BIT     #1,@R0
004064  001407                          BEQ     TST10           ;BRANCH TO TEST REGISTER IF PRESENT
004066  062700  000010          LOOP10: ADD     #10,R0
004072  005722                          TST     (R2)+
004074  020027  000766                  CMP     R0,#TREG
004100  103767                          BLO     LUP10
004102  000507                          BR      DONE10          ;BRANCH IF ALL REGISTERS HAVE BEEN TESTED
004104  004767  010016          TST10:  JSR     %7,LOCATM       ;ERROR RETURN INDICATED?
004110  032701  000001                  BIT     #1,R1           ;BRANCH IF NO
004114  001403                          BEQ     .+10            ;MAP INDICATES THERE IS NO MEMORY
004116  104002                          ERROR                   ;CORRESPONDING TO THIS REGISTER
                                                                ;R0 POINTS TO THE ADDRESS OF
                                                                ;THE PARITY REGISTER

004120  000167  177742                  JMP     LOOP10
004124  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
004132  011111                          MOV     @R1,@R1         ;WRITE WRONG PARITY IN FIRST LOCATION
004134  016161  004000  004000          MOV     4000(R1),4000(R1) ;WRITE WRONG PARITY IN SECOND LOCATION
004142  012770  000001  000000          MOV     #AE,@(R0)       ;SET ACTION ENABLE AND CLEAR REST
004150  012737  004202  000114          MOV     #TRP10,@#PARVEC ;SETUP PARITY TRAP RETURN
004156  012767  000010  000152          MOV     #10,COUNT       ;SETUP COUNTER TO EXECUTE INSTRUCTION 1
                                                                ;(INST1) TEN TIMES
004164  005711                  INST1:  TST     @R1             ;READ WRONG PARITY WITH AE SET- SHOULD
                                                                ;TRAP TO TRP10
004166  104002                          ERROR                   ;NO PARITY TRAP OCCURRED. R0 POINTS TO
                                                                ;ADDRESS OF THE PARITY REGISTER BEING
                                                                ;TESTED.

004170  000441                          BR      CONT10
004172  005761  004000          INST2:  TST     4000(R1)        ;READ WRONG PARITY FROM SECOND ADDRESS
                                                                ;WITH AE SET- SHOULD TRAP TO TRP10A
004176  104002                          ERROR                   ;NO PARITY TRAP OCCURRED. R0 POINTS TO
                                                                ;THE ADDRESS OF THE PARITY REGISTER
                                                                ;BEING TESTED

004200  000435                          BR      CONT10
004202  005367  000130          TRP10:  DEC     COUNT           ;HAS PARITY TRAP OCCURRED TEN TIMES?
004206  001413                          BEQ     1$              ;YES- BRANCH
004210  022712  000001                  CMP     #1,(R2)         ;IS THIS A CORE PAR REG?
004214  001005                          BNE     2$              ;NO, BRANCH (NO ERROR
                                                                ;ADDRESS BITS FOR MOS PAR
004216  032770  000040  000000          BIT     #BIT5,@(R0)     ;IF ERROR ADDRESS BITS ARE TRACKING,
                                                                ;BIT 5 SHOULD BE CLEAR (ONLY FOR CORE PARITY)
004224  001401                          BEQ     .+4
004226  104002                          ERROR                   ;PARITY ERROR ADDRESS BITS INCORRECT
                                                                ;R0 POINTS TO THE ADDRESS OF THE PARITY
                                                                ;REGISTER. R1 CONTAINS THE ADDRESS
```

```
1362
1363
1364  004230  012716  004164          2$:     MOV     #INST1,@SP              ;REFERENCED TO CAUSE A PARITY TRAP
1365  004234  000002                  RTI                                     ;(VIRTUAL IF KT11 IS PRESENT)
1366  004236  012737  004252  000114  1$:     MOV     #TRP10A,@#PARVEC        ;GO EXECUTE INSTRUCTION 1 AGAIN
1367  004244  012716  004172                  MOV     #INST2,@SP
1368  004250  000002                  RTI                             ;CHANGE PARITY TRAP RETURN
1369  004252  022712  000001          TRP10A: CMP     #1,(R2)                 ;GO EXECUTE INSTRUCTION 2
1370  004256  001005                  BNE     1$
1371  004260  032770  000040  000000          BIT     #BIT5,@(R0)             ;IS THIS A CORE REG?
1372                                                                          ;NO, BRANCH
1373  004266  001001                  BNE     .+4                             ;PARITY TRAP OCCURRED- CHECK PARITY
1374                                                                          ;ERROR ADDRESS BITS
1375  004270  104002                  ERROR                                   ;BRANCH IF OK (IF THE PARITY ERROR
1376                                                                          ;ADDRESS BITS TRACKED, BIT 5 WILL BE SET)
1377                                                                          ;PARITY ERROR ADDRESS BITS INCORRECT
1378                                                                          ;R0 POINTS TO THE ADDRESS OF THE
1379                                                                          ;PARITY REGISTER. THE ADDRESS REFERENCED
1380  004272  022626                  1$:     CMP     (SP)+,(SP)+             ;TO CAUSE THE ERROR WAS THAT IN
1381  004274  012737  000116  000114  CONT10: MOV     #PARVEC+2,@#PARVEC      ;R1 PLUS 4000 (OCTAL).
1382  004302  005070  000000                  CLR     @(R0)                           ;RESTORE TRAPCATCHER
1383  004306  005511                  ADC     @R1                     ;CLEAR PARITY REGISTER
1384  004310  005561  004000                  ADC     4000(R1)                ;CLEAR BAD PARITY
1385  004314  005070  000000                  CLR     @(R0)
1386  004320  000662                  BR      LOOP10                  ;CLEAR PARITY ERROR BIT IF SET
1387  004322  005767  174222          DONE10: TST     NOKT
1388  004326  001002                  BNE     .+6
1389  004330  005037  177572                  CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1390  004334  000401                  BR      TEST11
1391  004336  000000          COUNT:  0
1392
1393
1394
1395                                  ;*************************************************************************
1396                                  ;IF MULTIPLE PARITY ERRORS OCCUR DURING ONE INSTRUCTION (WITH ACTION ENABLE
1397                                  ;NOT SET) THE ERROR ADDRESS BITS WILL RECORD THE LAST ERROR (ONLY FOR CORE PARITY
1398                                  ;REGISTERS)
1399                                  ;*************************************************************************
1400  004340  104001          TEST11: SCOPE
1401  004342  012777  000011  174532          MOV     #11,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
1402  004350  004767  007436                  JSR     %7,CLRPAR               ;INITIALLY CLEAR ALL PARITY REGISTERS
1403  004354  005767  174170                  TST     NOKT                    ;KT11 PRESENT?
1404  004360  001004                  BNE     1$                      ;NO- BRANCH
1405  004362  004767  007324                  JSR     %7,NRALL                ;YES, MAP KERNEL PAGE 0 TO BANK 0,RW
1406  004366  004767  007470                  JSR     %7,MAP1                 ;MAP KERNEL PAGE 7 TO THE EXTERNAL BANK, RW
1407                                                                          ;SET KERNEL PAGE 1 RW AND TURN ON KT11
1408  004372  012700  000566          1$:     MOV     #MPR0,R0                ;SETUP TO GET ADDRESSES OF REGISTERS.PRESENT
1409  004376  012703  000770                  MOV     #INDC0,R3
1410  004402  032710  000001          LUP11:  BIT     #1,@R0
1411  004406  001003                  BNE     LOOP11                  ;IF THIS REG NOT PRESENT, SKIP
1412  004410  022713  000001                  CMP     #1,(R3)                 ;IS THIS A CORE PAR REG?
1413  004414  001407                  BEQ     TST11                   ;YES, THEN TEST IT
1414                                                                          ;IF NOT CORE, SKIP THIS REGISTER
1415  004416  062700  000010          LOOP11: ADD     #10,R0
1416  004422  005723                  TST     (R3)+
1417  004424  020027  000766                  CMP     R0,#TREG
```

```
1418  004430  103764                        BLO     LUP11
1419  004432  000443                        BR      DONE11
1420  004434  004767  007466        TST11:  JSR     %7,LOCATM       ;BRANCH OUT IF ALL REGISTERS HAVE BEEN TESTED
1421                                                                ;GET THE ADDRESS OF A MEMORY LOCATION
1422  004440  032701  000001                BIT     #1,R1           ;CORRESPONDING TO THIS PARITY REGISTER
1423  004444  001403                        BEQ     .+10            ;ERROR RETURN INDICATED?
1424  004446  104002                        ERROR                   ;BRANCH IF NOT
1425                                                                ;NO MEMORY IN MAP CORRESPONDING TO
1426                                                                ;THIS PARITY REGISTER. R0 POINTS
1427  004450  000167  177742                JMP     LOOP11          ;TO THE ADDRESS OF THE PARITY REGISTER
1428  004454  010102                        MOV     R1,R2           ;SETUP SECOND TEST ADDRESS LOCATION
1429  004456  062702  010000                ADD     #10000,R2
1430  004462  012770  000004  000000        MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
1431  004470  011111                        MOV     @R1,@R1         ;WRITE WRONG PARITY IN FIRST TEST LOCATION
1432  004472  011212                        MOV     @R2,@R2         ;WRITE WRONG PARITY IN SECOND TEST LOCATION
1433  004474  005070  000000                CLR     @(R0)           ;CLEAR PARITY REGISTER
1434  004500  021112                        CMP     @R1,@R2         ;READ FIRST TEST LOCATION, AND
1435                                                                ;THEN READ SECOND TEST LOCATION
1436  004502  005770  000000                TST     @(R0)           ;MAKE SURE PARITY ERROR SET
1437  004506  100401                        BMI     .+4
1438  004510  104002                        ERROR                   ;PARITY ERROR NOT SET AFTER
1439                                                                ;READING TWO LOCATIONS WHICH
1440                                                                ;SHOULD HAVE BAD PARITY
1441  004512  032770  000100  000000        BIT     #BIT6,@(R0)     ;CHECK ERROR ADDRESS- IF THE LAST
1442                                                                ;ADDRESS WAS RECORDED, BIT 6 WILL
1443                                                                ;BE SET
1444  004520  001001                        BNE     .+4
1445  004522  104002                        ERROR                   ;PARITY ERROR ADDRESS BITS INCORRECT
1446                                                                ;R0 POINTS TO ADDRESS OF PARITY REGISTER
1447                                                                ;R2 CONTAINS ADDRESS OF LAST BAD PARITY
1448                                                                ;LOCATION REFERENCED (IF KT11 PRESENT,
1449                                                                ;ADDRESS IS VIRTUAL THRU KERNEL PAGE 1)
1450  004524  005070  000000                CLR     @(R0)           ;CLEAR PARITY REGISTER
1451  004530  005511                        ADC     @R1             ;CLEAR BAD PARITY
1452  004532  005512                        ADC     @R2
1453  004534  005070  000000                CLR     @(R0)           ;CLEAR PARITY ERROR BIT
1454  004540  000726                        BR      LOOP11
1455  004542  005767  174002        DONE11: TST     NOKT
1456  004546  001002                        BNE     .+6
1457  004550  005037  177572                CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1458
1459
1460                                ;*****************************************************************************
1461                                ;SHOW THAT IF AN INSTRUCTION DOING A DATIP GETS A PARITY ERROR,
1462                                ;THE ORIGINAL DATA IS REWRITTEN IF ACTION ENABLE IS SET, AND IS
1463                                ;ALTERRED IF ACTION ENABLE IS CLEAR
1464                                ;*****************************************************************************
1465
1466  004554  104001                TEST12: SCOPE
1467  004556  012777  000012  174316        MOV     #12,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
1468  004564  004767  007222                JSR     %7,CLRPAR
1469  004570  005767  173754                TST     NOKT            ;KT11 PRESENT?
1470  004574  001004                        BNE     1$              ;NO- BRANCH
1471  004576  004767  007110                JSR     %7,NRALL        ;YES, MAP KERNEL 0 TO BANK 0, RW
1472  004602  004767  007254                JSR     %7,MAP1         ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
1473                                                                ;SET KERNEL 1 RW AND TURN ON KT11
```

```
1474  004606  012700  000566          1$:     MOV     #MPR0,R0        ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1475  004612  012702  000770                  MOV     #INDC0,R2
1476  004616  032710  000001          LUP12:  BIT     #1,@R0          ;SKIP, IF THIS REG NOT PRESENT
1477  004622  001003                          BNE     LOOP12          ;REG PRESENT, IS IT CORE?
1478  004624  022712  000001                  CMP     #1,(R2)         ;YES, DO* THIS TEST
1479  004630  001407                          BEQ     TST12           ;SKIP, IF THIS REG IS NOT CORE
1480
1481  004632  062700  000010          LOOP12: ADD     #10,R0          ;BRANCH TO DONE IF ALL REGISTERS
1482  004636  005722                          TST     (R2)+           ;HAVE BEEN TESTED
1483  004640  020027  000766                  CMP     R0,#TREG        ;LOCATE MEMORY CORRESPONDING TO THIS
1484  004644  103764                          BLO     LUP12           ;REGISTER
1485  004646  000474                          BR      DONE12          ;ERROR RETURN INDICATED?
1486                                                                  ;NO- BRANCH
1487  004650  004767  007252          TST12:  JSR     %7,LOCATM       ;NO MEMORY IN MAP CORRESPONDING TO
1488                                                                  ;THIS REGISTER. R0 POINTS TO
1489  004654  032701  000001                  BIT     #1,R1           ;THE ADDRESS OF THE PARITY REGISTER
1490  004660  001403                          BEQ     .+10
1491  004662  104002                          ERROR
1492
1493
1494  004664  000167  177742                  JMP     LOOP12
1495  004670  012737  004724  000114          MOV     #TRP12,@#PARVEC ;SET UP PARITY TRAP RETURN
1496  004676  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
1497  004704  012711  125252                  MOV     #125252,@R1     ;WRITE WRONG PARITY IN TEST LOCATION
1498  004710  012770  000001  000000          MOV     #AE,@(R0)       ;SET ACTION ENABLE AND CLEAR
1499                                                                  ;WRITE WRONG PARITY
1500  004716  005211                          INC     @R1             ;DO DATIP,DATO WITH ACTION ENABLE
1501                                                                  ;SET-  SHOULD ABORT ON DATIP AND
1502                                                                  ;RESTORE ORIGINAL DATA
1503  004720  104002                          ERROR                   ;NO ABORT OCCURRED ON READING LOCATION
1504                                                                  ;WHICH  SHOULD CONTAIN BAD PARITY
1505                                                                  ;(WITH AE SET).
1506                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER.
1507                                                                  ;R1 CONTAINS ADDRESS OF TEST LOCATION
1508                                                                  ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 IS
1509                                                                  ;PRESENT)
1510  004722  000440                          BR      CONT12
1511  004724  005070  000000          TRP12:  CLR     @(R0)           ;PARITY TRAP OCCURRED- CLEAR PARITY REGISTER
1512  004730  021127  125252                  CMP     @R1,#125252     ;ORIGINAL DATA RESTORED?
1513  004734  001401                          BEQ     .+4             ;YES, BRANCH
1514  004736  104002                          ERROR                   ;NO- DATIP WHICH GOT A PARITY ERROR
1515                                                                  ;TRAP ALTERRED CONTENTS OF LOCATION
1516                                                                  ;READ. ADDRESS OF TEST LOCATION IS IN R1
1517                                                                  ;(IF KT11 IS PRESENT, ADDRESS IN R1
1518                                                                  ;IS VIRTUAL THRU KERNEL PAGE 1)
1519                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER
1520  004740  005770  000000                  TST     @(R0)           ;MAKE SURE PARITY ERROR SET WHEN
1521  004744  100401                          BMI     .+4             ;DATA WAS REREAD IN THE ABOVE CMP
1522  004746  104002                          ERROR                   ;DATIP WHICH GOT A PARITY ERROR TRAP
1523                                                                  ;ALTERRED THE PARITY OF THE LOCATION READ
1524                                                                  ;R1 CONTAINS ADDRESS OF TEST LOCATION
1525                                                                  ;(VIRTUAL THRU KERNEL 1 IF KT11 PRESENT)
1526  004750  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1527  004752  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY AND CLEAR
1528                                                                  ;PARITY ERROR
1529  004760  012711  125252                  MOV     #125252,@R1     ;REWRITE DATA WITH WRONG PARITY
```

```
1530  004764  005070  000000                  CLR     @(R0)                   ;CLEAR PARITY REGISTER
1531  004770  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
1532  004776  005211                          INC     @R1                     ;SINCE AE IS CLEAR, INSTRUCTION SHOULD
1533                                                                          ;COMPLETE AND SHOULD CLEAR BAD PARITY
1534  005000  005070  000000                  CLR     @(R0)                   ;CLEAR PARITY ERROR BIT
1535  005004  022711  125253                  CMP     #125253,@R1             ;CHECK DATA
1536  005010  001401                          BEQ     .+4
1537  005012  104002                          ERROR

1538                                                                          ;DATIP, DATO TO A LOCATION CONTAINING BAD
1539                                                                          ;PARITY WITHOUT AE SET LEFT INCORRECT
1540                                                                          ;DATA. R0 POINTS TO THE ADDRESS OF
1541                                                                          ;THE PARITY REGISTER. R1 CONTAINS THE
1542                                                                          ;ADDRESS OF THE TEST LOCATION (VIRTUAL
1543  005014  005770  000000                  TST     @(R0)                   ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1544  005020  100001                          BPL     .+4                     ;CHECK PARITY ERROR BIT
1545  005022  104002                          ERROR

1546                                                                          ;DATIP, DATO WITH AE CLEAR DID
1547                                                                          ;NOT CLEAR BAD PARITY IN LOCATION
1548                                                                          ;ADDRESSED. R0 POINTS TO THE ADDRESS
1549                                                                          ;OF THE PARITY REGISTER. R1 CONTAINS
1550                                                                          ;THE ADDRESS OF THE TEST LOCATION
1551                                                                          ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11
1552  005024  005070  000000          CONT12: CLR     @(R0)                   ;IS PRESENT)
1553  005030  005011                          CLR     @R1                     ;CLEAR PARITY REGISTER
1554  005032  005070  000000                  CLR     @(R0)                   ;CLEAR LOCATION TO RESTORE GOOD PARITY
1555  005036  000675                          BR      LOOP12                  ;CLEAR PARITY ERROR IF SET
1556                                                                          ;GO CHECK FOR ANOTHER PARITY
1557  005040  012737  000116  000114  DONE12: MOV     #PARVEC+2,@#PARVEC      ;REGISTER
1558  005046  005767  173476                  TST     NOKT                            ;RESTORE TRAPCATCHER
1559  005052  001002                          BNE     .+6
1560  005054  005037  177572                  CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1561
1562
1563                                  ;*****************************************************************************
1564                                  ;SHOW THAT IF AN INSTRUCTION DOING A DATI (BUT NO DATO TO THE SAME LOCATION)
1565                                  ;GETS A PARITY ERROR, THE ORIGINAL DATA IS UNALTERRED, WHETHER OR NOT ACTION
1566                                  ;ENABLE IS SET
1567                                  ;*****************************************************************************
1568
1569  005060  104001          TEST13: SCOPE
1570  005062  012777  000013  174012          MOV     #13,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
1571  005070  004767  006716                  JSR     %7,CLRPAR
1572  005074  005767  173450                  TST     NOKT                    ;KT11 PRESENT?
1573  005100  001004                          BNE     1$                      ;NO- BRANCH
1574  005102  004767  006604                  JSR     %7,NRALL                ;YES, MAP KERNEL 0 TO BANK 0,KERNEL
1575  005106  004767  006750                  JSR     %7,MAP1                 ;7 TO THE EXTERNAL BANK, AND KERNEL
1576                                                                          ;0,1,AND 7 RW
1577  005112  012700  000566          1$:     MOV     #MPR0,R0                ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1578  005116  032710  000001          LUP13:  BIT     #1,@R0
1579  005122  001406                          BEQ     TST13                   ;IF THIS REGISTER IS PRESENT, GO
1580                                                                          ;TEST IT
1581  005124  062700  000010          LOOP13: ADD     #10,R0
1582  005130  020027  000766                  CMP     R0,#TREG
1583  005134  103770                          BLO     LUP13
1584  005136  000470                          BR      DONE13                  ;BRANCH IF ALL REGISTERS HAVE BEEN
1585                                                                          ;TESTED
```

```
1586  005140  004767  006762          TST13:  JSR     %7,LOCATM       ;LOCATE MEMORY CORRESPONDING TO THIS
1587                                          ;REGISTER, AND IF KT11 IS PRESENT
1588                                          ;MAP KERNEL 1 TO THAT MEMORY
1589  005144  032701  000001                  BIT     #1,R1           ;ERROR RETURN INDICATED?
1590  005150  001403                          BEQ     .+10            ;NO- BRANCH
1591  005152  104002                          ERROR                   ;MAP INDICATES NO MEMORY WAS FOUND
1592                                          ;CORRESPONDING TO THIS REGISTER
1593                                          ;R0 POINTS TO THE ADDRESS OF THE
1594                                          ;PARITY REGISTER
1595  005154  000167  177744                  JMP     LOOP13
1596  005160  012737  005214  000114          MOV     #TRP13,@#PARVEC ;SETUP PARITY TRAP RETURN
1597  005166  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
1598  005174  012711  125252                  MOV     #125252,@R1     ;WRITE WRONG PARITY
1599  005200  012770  000001  000000          MOV     #AE,@(R0)       ;SET ACTION ENABLE AND CLEAR
1600                                          ;WRITE WRONG PARITY
1601  005206  005711                          TST     @R1             ;DATI WITH ACTION ENABLE SET SHOULD
1602                                          ;ABORT LEAVING DATA UNCHANGED
1603  005210  104002                          ERROR                   ;NO ABORT ON READING BAD PARITY
1604                                          ;WITH ACTION ENABLE SET. R0 POINTS
1605                                          ;TO THE ADDRESS OF THE PARITY REGISTER.
1606                                          ;R1 CONTAINS THE ADDRESS OF THE TEST
1607                                          ;LOCATION (VIRTUAL THRU KERNEL PAGE
1608                                          ;1 IF KT11 IS PRESENT).
1609  005212  000434                          BR      CONT13
1610  005214  005070  000000          TRP13:  CLR     @(R0)           ;ABORT OCCURRED AS EXPECTED- CLEAR REGISTER
1611  005220  021127  125252                  CMP     @R1,#125252     ;ORIGINAL DATA RESTORED?
1612  005224  001401                          BEQ     .+4             ;YES, BRANCH
1613  005226  104002                          ERROR                   ;DATI WHICH GOT A PARITY ERROR ALTERRED
1614                                          ;THE CONTENTS OF THE LOCATION ADDRESSED.
1615                                          ;(R1 CONTAINS THE ADDRESS OF
1616                                          ;MEMORY BEING TESTED- IF KT11 IS
1617                                          ;PRESENT, ADDRESS IN R1 IS VIRTUAL)
1618                                          ;R0 POINTS TO THE ADDRESS OF THE
1619                                          ;PARITY REGISTER
1620  005230  005770  000000                  TST     @(R0)           ;CHECK PARITY REGISTER
1621  005234  100401                          BMI     .+4             ;BRANCH IF PARITY ERROR SET
1622  005236  104002                          ERROR                   ;PARITY ERROR NOT SET AFTER READING
1623                                          ;DATA WITH BAD PARITY
1624                                          ;R0 POINTS TO THE ADDRESS OF THE PARITY
1625                                          ;REGISTER. R1 CONTAINS THE ADDRESS
1626                                          ;OF THE TEST LOCATION (VIRTUAL THRU
1627                                          ;KERNEL PAGE 1 IF KT11 PRESENT)
1628  005240  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1629  005242  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY, CLEAR PARITY ERROR
1630  005250  012711  125252                  MOV     #125252,@R1     ;REWRITE DATA WITH WRONG PARITY
1631  005254  005070  000000                  CLR     @(R0)           ;CLEAR PARITY REGISTER
1632  005260  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAPCATCHER
1633  005266  005711                          TST     @R1             ;DATI TO LOCATION WITH BAD PARITY
1634                                          ;AE NOT SET-INSTRUCTION SHOULD COMPLETE
1635  005270  005070  000000                  CLR     @(R0)           ;CLEAR PARITY ERROR BIT
1636  005274  022711  125252                  CMP     #125252,@R1     ;CHECK DATA
1637  005300  001401                          BEQ     .+4
1638  005302  104002                          ERROR
1639                                          ;DATI TO LOCATION WITH BAD PARITY
1640                                          ;WITHOUT ACTION ENABLE SET LEFT INCORRECT DATA
1641                                          ;R1 CONTAINS THE ADDRESS OF THE TEST
                                              ;LOCATION (VIRTUAL THRU KERNEL PAGE 1
```

```
                                                        ;IF KT11 IS PRESENT), R0 POINTS TO THE
                                                        ;ADDRESS OF THE PARTIY REGISTER.
005304 005070 000000          CONT13: CLR     @(R0)     ;CLEAR PARITY REGISTER
005310 005011                         CLR     @R1       ;CLEAR LOCATION
005312 005070 000000                  CLR     @(R0)     ;CLEAR PARITY ERROR IF SET
005316 000702                         BR      LOOP13    ;GO CHECK FOR ANOTHER PARITY
                                                        ;REGISTER
005320 012737 000116 000114   DONE13: MOV     #PARVEC+2,@#PARVEC
                                                                ;RESTORE TRAPCATCHER
005326 005767 173216                  TST     NOKT
005332 001002                         BNE     .+6
005334 005037 177572                  CLR     @#SR0     ;TURN OFF KT11 IF PRESENT


                        ;*****************************************************************************
                        ;CHECK PARITY MEMORY WITH SERIES OF PATTERNS FROM 4K TO 28K
                        ;ENABLE PARITY TRAP
                        ;*****************************************************************************
005340 104001           TEST14: SCOPE
005342 012777 000014 173532           MOV     #14,@DISPLAY      ;LOAD THE TEST NUMBER INTO THE DISPLAY
005350 005067 011102                  CLR     IMAX              ;DON'T ITERATE THE REST OF THE SUBTESTS
005354 004767 006432                  JSR     PC,CLRPAR         ;CLEAR ALL PARITY REGISTERS
005360 012737 005734 000114           MOV     #TRP14,@#PARVEC   ;SETUP PARITY TRAP RETURN
005366 022767 000002 173474           CMP     #2,PMEML          ;TEST FOR AN 8K SYSTEM
005374 103402                         BLO     .+6               ;SYST > 8K
005376 004767 005742                  JSR     PC,SMLSYS         ;SYST < OR = 8K
005402 032767 000002 173110           BIT     #2,BITPT          ;TEST BANK INDICATOR FOR BANK 1
005410 001007                         BNE     1$
005412 012767 000004 173100           MOV     #4,BITPT          ;INITIALIZE BANK INDICATOR TO BANK 2
005420 012767 040000 173070           MOV     #40000,ADRPT      ;INITIALIZE MEMORY STARTING ADDRESS
005426 000403                         BR      LOOP14
005430 012767 020000 173060     1$:   MOV     #20000,ADRPT      ;INITIALIZE BANK INDICATOR TO BANK 1
005436 036767 173056 173424 LOOP14:   BIT     BITPT,PMEML       ;DOES THIS 4K HAVE PARITY?
005444 001021                         BNE     TST14             ;YES, TEST IT
005446 032767 000002 173044  LUP14:   BIT     #2,BITPT          ;TEST FOR BANK 1 INDICATOR IF SET GO TO TEST 16
005454 001001                         BNE     EX
005456 000402                         BR      .+6
005460 000167 000776           EX:    JMP     TEST16
005464 062767 020000 173024           ADD     #20000,ADRPT      ;NO- UPDATE MEMORY ADDRESS
005472 006367 173022                  ASL     BITPT             ;UPDATE BIT POINTER
005476 022767 000200 173014           CMP     #200,BITPT        ;THIS 28K DONE?
005504 003354                         BGT     LOOP14            ;NO, BRANCH TO SEE IF NEXT 4K
                                                                ;SHOULD BE TESTED
005506 000443                         BR      DONE14            ;YES, EXIT
005510 012704 001036 TST14:   MOV     #PARPAT,R4        ;INITIALIZE PATTERN POINTER
005514 016767 172776 173006           MOV     ADRPT,HIADR       ;SET UPPER LIMIT FOR THIS 4K
005522 062767 020000 173000           ADD     #20000,HIADR
005530 016705 172762                  MOV     ADRPT,R5
005534 005025                 2$:     CLR     (5)+              ;INITIALLY CLEAR CORE BLOCK UNDER TEST
005536 020567 172766                  CMP     R5,HIADR
005542 103774                         BLO     2$
005544 012701 000566                  MOV     #MPR0,R1          ;INITIALIZE TO SET AE IN ALL REGISTERS
005550 032711 000001          3$:     BIT     #1,@R1
005554 001003                         BNE     .+10
005556 012771 000001 000000           MOV     #AE,@(R1)         ;SET ACTION ENABLE IF REGISTER IS PRESENT
005564 062701 000010                  ADD     #10,R1
```

```
1698  005570  020127  000766                  CMP     R1,#TREG
1699  005574  103765                          BLO     3$
1700  005576  004767  000024          4$:     JSR     %7,TPCOPE               ;GO TO ROUTINE TO EXERCISE THIS 4K
1701                                                                          ;WITH THE CURRENT PATTERN
1702  005602  005724                          TST     (4)+                    ;UPDATE PATTERN
1703  005604  005714                          TST     (4)                     ;LAST PATTERN?
1704  005606  001373                          BNE     4$                      ;NO, LOOP
1705  005610  004767  006176                  JSR     PC,CLRPAR               ;YES, CLEAR ALL PARITY REGISTERS
1706  005614  000714                          BR      LUP14                   ;UPDATE AND CHECK NEXT 4K
1707  005616  012737  000116  000114  DONE14: MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
1708  005624  000473                          BR      TEST15                  ;GO TO NEXT TEST
1709
1710                                  ;ROUTINE TO WRITE AND CHECK EACH LOCATION IN 4K (STARTING AT ADDRESS
1711                                  ;IN ADRPT) WITH VALUE POINTED TO BY R4
1712  005626  016705  172664          TPCORE: MOV     ADRPT,R5                ;SETUP R5 TO ADDRESS MEMORY
1713                                                                          ;LOCATION BEING CHECKED
1714  005632  011415                  1$:     MOV     (4),(5)                 ;WRITE PATTERN INTO MEMORY
1715  005634  011567  172676                  MOV     (5),WAS                 ;READ TEST LOCATION
1716  005640  021467  172672                  CMP     (4),WAS                 ;DATA OK?
1717  005644  001401                          BEQ     .+4                     ;YES- BRANCH
1718  005646  104002                          ERROR                           ;DATA INCORRECT IN LOCATION WHOSE
1719                                                                          ;ADDRESS IS IN R5. R4 POINTS TO THE
1720                                                                          ;DATA WRITTEN.
1721  005650  005725                          TST     (5)+                    ;UPDATE ADDRESS POINTER
1722  005652  020567  172652                  CMP     R5,HIADR                ;THIS 4K DONE?
1723  005656  103765                          BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
1724  005660  005067  173102                  CLR     TREG                    ;YES, DID ANY PARITY ERRORS OCCUR
1725                                                                          ;WITHOUT TRAPPING?
1726  005664  012701  000566                  MOV     #MPRO,R1
1727  005670  032711  000001          2$:     BIT     #1,(1)
1728  005674  001003                          BNE     .+10
1729  005676  005771  000000                  TST     @(R1)
1730  005702  100406                          BMI     3$                      ;YES- BRANCH
1731  005704  062701  000010                  ADD     #10,R1
1732  005710  020127  000766                  CMP     R1,#TREG
1733  005714  103765                          BLO     2$
1734  005716  000207                          RTS     %7                      ;NO, RETURN
1735  005720  011167  173042          3$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1736  005724  104004                          ERRORS                          ;PARITY ERROR SET (WITH AE SET) AND
1737                                                                          ;NO TRAP OCCURRED. TREG CONTAINS
1738                                                                          ;ADDRESS OF PARITY REGISTER WHICH
1739                                                                          ;HAS ERROR BIT SET.
1740  005726  004767  006562                  JSR     %7,PSCAN                ;SCAN FOR PARITY ERRORS AND PRINT
1741                                                                          ;18 BIT ADDRESSES OF THOSE FOUND.
1742                                                                          ;AFTER REPORTING EACH ERROR CLEAR IT
1743  005732  000207                          RTS     %7
1744
1745                                  ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
1746  005734  005067  173026          TRP14:  CLR     TREG
1747  005740  012701  000566                  MOV     #MPRO,R1                ;FIND PARITY REGISTER INDICATING PARITY ERROR
1748  005744  032711  000001          1$:     BIT     #1,(R1)
1749  005750  001003                          BNE     .+10
1750  005752  005771  000000                  TST     @(R1)
1751  005756  100407                          BMI     2$                      ;BRANCH IF PARITY ERROR SET
1752  005760  062701  000010                  ADD     #10,R1
1753  005764  020127  000766                  CMP     R1,#TREG
```

```
1754  005770  103765                        BLO     1$
1755  005772  104002                        ERROR
1756  005774  000405                        BR      3$
1757
1758  005776  011167  172764          2$:   MOV     @R1,TREG
1759  006002  104004                        ERRORS
1760
1761
1762
1763
1764  006004  004767  006504                JSR     %7,PSCAN
1765
1766  006010  022626                  3$:   CMP     (SP)+,(SP)+
1767  006012  000207                        RTS     %7
1768
1769
1770
1771
1772                                  ;*****************************************************************************
1773                                  ;CHECK PARITY MEMORY WITH SERIES OF PATTERNS ABOVE 28K
1774                                  ;ENABLE PARITY ERROR TRAPPING
1775                                  ;*****************************************************************************
1776  006014  104001                  TEST15: SCOPE
1777  006016  012777  000015  173056          MOV     #15,@DISPLAY
1778  006024  005767  172520                 TST     NOKT
1779  006030  001402                         BEQ     .+6
1780  006032  000167  000424                 JMP     TEST16
1781  006036  004767  005750                JSR     PC,CLRPAR
1782  006042  004767  005644                JSR     %7,NRALL
1783  006046  004767  006010                JSR     %7,MAP1
1784
1785  006052  012777  001600  173172          MOV     #1600,@KPAR1
1786  006060  005067  172470                 CLR     LOWFLG
1787  006064  016767  173000  173002          MOV     PMEML,PMEMX
1788  006072  012737  006402  000114          MOV     #TRP15,@#PARVEC
1789  006100  012767  000200  172412          MOV     #200,BITPT
1790  006106  036767  172406  172760  LOOP15: BIT     BITPT,PMEMX
1791  006114  001032                         BNE     TST15
1792  006116  062777  000200  173126  LOP15:  ADD     #200,@KPAR1
1793  006124  006367  172370                 ASL     BITPT
1794  006130  103366                         BCC     LOOP15
1795  006132  005767  172416                 TST     LOWFLG
1796  006136  001051                         BNE     DONE15
1797  006140  005267  172410                 INC     LOWFLG
1798  006144  016767  172722  172722          MOV     PMEMH,PMEMX
1799  006152  012767  000001  172340          MOV     #1,BITPT
1800  006160  000752                         BR      LOOP15
1801  006162  012704  001036          TST15:  MOV     #PARPAT,R4
1802  006166  012767  020000  172322          MOV     #20000,ADRPT
1803
1804  006174  012705  020000                 MOV     #20000,R5
1805  006200  005025                  2$:    CLR     (5)+
1806  006202  020527  040000                 CMP     R5,#40000
1807  006206  103774                         BLO     2$
1808  006210  012701  000566                 MOV     #MPR0,R1
1809
```

;PARITY TRAP TO 114 OCCURRED DURING
;TEST 14 BUT NO REGISTERS HAVE
;PARITY ERROR SET
;STORE ADDRESS OF REGISTER GETTING ERROR
;PARITY TRAP TO 114 OCCURRED DUE TO
;PARITY ERROR WHILE EXERCISING MEMORY
;R1 POINTS TO THE ADDRESS OF THE
;PARITY REGISTER HAVING PARITY ERROR
;BIT SET
;SCAN FOR BAD PARITY AND TYPE 18 BIT ADDRESSES
;OF LOCATIONS FOUND BAD
;RESTORE STACK POINTER
;RETURN (FROM JSR TO TPCORE) TO
;CHECK NEXT PATTERN

;LOAD THE TEST NUMBER INTO THE DISPLAY
;KT11 PRESENT?
;YES- BRANCH
;NO, SKIP TEST
;CLEAR ALL PARITY REGISTERS
;MAP KERNEL 0 TO BANK 0,RW
;MAP KERNEL 7 TO THE EXTERNAL BANK
;SET KERNEL 1 RW AND TURN ON KT11
;MAP KERNEL PAGE 1 TO BEGINNING OF 28-32K
;CLEAR FLAG TO INDICATE CHECKING LOWER 64K

;SETUP PARITY TRAP RETURN
;INITIALIZE BIT POINTER
;DOES THIS 4K HAVE PARITY?
;YES, BRANCH TO TEST IT
;NO- MAP TO NEXT 4K
;UPDATE BIT POINTER
;BRANCH IF NOT DONE WITH 64K
;DONE WITH 128K?
;YES, BRANCH
;NO, SET FLAG INDICATING UPPER 64K
;SETUP PARITY MAP WORD
;SETUP BIT POINTER FOR UPPER 64K
;CONTINUE
;INITIALIZE PATTERN POINTER
;INITIALIZE VIRTUAL ADDRESS OF MEMORY
;BEING TESTED

;INITIALLY CLEAR CORE BLOCK UNDER TEST

;INITIALIZE TO SET ACTION ENABLE IN ALL
;PARITY REGISTERS

```
1810  006214  032711  000001          3$:     BIT     #1,@R1
1811  006220  001003                          BNE     .+10
1812  006222  012771  000001  000000          MOV     #AE,@(R1)               ;SET ACTION ENABLE IF THIS REGISTER
1813                                                                          ;IS PRESENT
1814  006230  062701  000010                  ADD     #10,R1
1815  006234  020127  000766                  CMP     R1,#TREG
1816  006240  103765                          BLO     3$
1817  006242  004767  000030          4$:     JSR     %7,TPCORX               ;EXERCISE THIS 4K
1818  006246  005724                          TST     (4)+                    ;UPDATE PATTERN
1819  006250  005714                          TST     (4)                     ;LAST PATTERN?
1820  006252  001373                          BNE     4$                      ;NO, LOOP
1821  006254  004767  005532                  JSR     PC,CLRPAR               ;YES, CLEAR ALL PARITY REGISTERS
1822  006260  000716                          BR      LOP15                   ;UPDATE AND CHECK NEXT 4K
1823  006262  005037  177572          DONE15: CLR     @#SR0                   ;TURN OFF KT11 WHEN DONE
1824  006266  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
1825  006274  000472                          BR      TEST16                  ;GO TO NEXT TEST
1826
1827                                  ;PARITY MEMORY TEST ROUTINE USING KT11 AND TESTING MEMORY ABOVE 28K
1828                                  ;WRITES AND CHECKS EACH LOCATION IN 4K USING KERNEL PAGE 1 MAPPED TO CURRENT BANK
1829  006276  000240                  TPCORX: NOP
1830  006300  012705  020000                  MOV     #20000,R5               ;SETUP R5 TO POINT TO THE LOCATION
1831                                                                          ;UNDER TEST (VIRTUAL ADDRESS)
1832  006304  011415                  1$:     MOV     (4),(5)                 ;WRITE PATTERN
1833  006306  011567  172224                  MOV     (5),WAS                 ;READ TEST LOCATION
1834  006312  021467  172220                  CMP     (4),WAS                 ;DATA OK?
1835  006316  001401                          BEQ     .+4                     ;YES- BRANCH
1836  006320  104002                          ERROR                           ;NO- DATA INCORRECT IN LOCATION WHOSE
1837                                                                          ;VIRTUAL ADDRESS IS IN R1 (GOES THRU
1838                                                                          ;KERNEL PAGE 1), R4 POINTS TO
1839                                                                          ;THE VALUE WRITTEN.
1840  006322  005725                          TST     (5)+                    ;UPDATE ADDRESS POINTER
1841  006324  020527  040000                  CMP     R5,#40000               ;THIS 4K DONE?
1842  006330  103765                          BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
1843  006332  005067  172430                  CLR     TREG                    ;YES, CHECK TO SEE IF ANY PARITY
1844                                                                          ;ERRORS OCCURRED WITHOUT TRAPPING
1845  006336  012701  000566                  MOV     #MPR0,R1
1846  006342  032711  000001          2$:     BIT     #1,(1)                  ;IS THIS PARITY REGISTER PRESENT?
1847  006346  001003                          BNE     .+10                    ;NO, GET NEXT ONE
1848  006350  005771  000000                  TST     @(R1)                   ;YES- DID ERROR SET?
1849  006354  100406                          BMI     3$                      ;YES- BRANCH
1850  006356  062701  000010                  ADD     #10,R1                  ;NO- GET NEXT REGISTER
1851  006362  020127  000766                  CMP     R1,#TREG
1852  006366  103765                          BLO     2$
1853  006370  000207                          RTS     %7                      ;NO ERRORS- EXIT
1854  006372  011167  172370          3$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1855  006376  104004                          ERRORS                          ;PARITY ERROR SET (AE ALREADY SET)
1856                                                                          ;AND NO TRAP OR TIMEOUT OCCURRED
1857                                                                          ;R1 POINTS TO THE ADDRESS OF THE
1858                                                                          ;PARITY REGISTER
1859  006400  000207                          RTS     %7
1860
1861                                  ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
1862  006402  005067  172360          TRP15:  CLR     TREG
1863  006406  012701  000566                  MOV     #MPR0,R1                ;LOCATE PARITY REGISTER INDICATING ERROR
1864  006412  032711  000001          1$:     BIT     #1,(R1)
1865  006416  001003                          BNE     .+10
```

```
1866  006420  005771  000000                  TST     @(R1)
1867  006424  100407                          BMI     2$                      ;BRANCH IF PARITY ERROR IS SET
1868  006426  062701  000010                  ADD     #10,R1
1869  006432  020127  000766                  CMP     R1,$TREG
1870  006436  103765                          BLO     1$
1871  006440  104002                          ERROR                           ;TRAP TO 114 OCCURRED DURING TEST 15 BUT
                                                                              ;NO PARITY REGISTERS HAVE PARITY ERROR SET
1872
1873  006442  000405                          BR      3$
1874  006444  011167  172316          2$:     MOV     @R1,TREG                ;STORE ADDRESS OF REGISTER GETTING ERROR
1875  006450  104004                          ERRORS                          ;PARITY TRAP TO 114 OCCURRED DUE TO
                                                                              ;PARITY ERROR WHILE EXERCISING MEMORY
1876                                                                          ;"TREG" CONTAINS ADDRESS OF PARITY REGISTER
1877                                                                          ;HAVING PARITY ERROR BIT SET
1878
1879  006452  004767  006036                  JSR     %7,PSCAN                ;SCAN MEMORY FOR BAD PARITY AND PRINT 18
1880                                                                          ;BIT ADDRESSES OF LOCATIONS FOUND
1881                                                                          ;CLEAR BAD PARITY IN EACH AFTER
1882                                                                          ;REPORTING IT
1883  006456  022626          3$:     CMP     (SP)+,(SP)+                     ;RESTORE STACK POINTER
1884  006460  000207                  RTS     %7                              ;RETURN (FROM JSR TO TPCORX) TO
1885                                                                          ;TEST NEXT PATTERN
1886
1887
1888                                  ;*************************************************************************
1889                                  ;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY FROM 4K TO 28K
1890                                  ;WRITE WRONG PARITY AND READ IT BACK WITH ACTION ENABLE SET, MAKING
1891                                  ;SURE THAT A TRAP OCCURS. THEN WRITE GOOD PARITY AND MAKE SURE THAT
1892                                  ;NO TRAP OCCURS WHEN IT IS READ. MAKE SURE THAT THE ERROR ADDRESS BITS
1893                                  ;(PARITY REGISTER BITS 5-11) ARE CORRECT.
1894                                  ;*************************************************************************
1895
1896  006462  104001          TEST16: SCOPE
1897  006464  012777  000016  172410          MOV     #16,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
1898  006472  012737  007076  000114          MOV     #TRP16,@#PARVEC         ;SET UP TRAP RETURN
1899  006500  032767  000002  172012          BIT     #2,BITPT                ;TEST BANK INDICATOR FOR BANK 1
1900  006506  001007                          BNE     LUP16
1901  006510  012767  000004  172002          MOV     #4,BITPT                ;INIT POINTER TO BANK 2
1902  006516  012767  040000  171772          MOV     #40000,ADRPT
1903  006524  000403                          BR      .+10
1904  006526  012767  020000  171762  LUP16:  MOV     #20000,ADRPT
1905  006534  036767  171760  172326  1$:     BIT     BITPT,PMEML             ;DOES THIS 4K HAVE PARITY?
1906  006542  001021                          BNE     3$                      ;YES, BRANCH TO TEST IT
1907  006544  032767  000002  171746  2$:     BIT     #2,BITPT                ;TEST FOR BANK 1 INDICATOR IF SET GO OUT
1908  006552  001001                          BNE     .+4
1909  006554  000402                          BR      .+6
1910  006556  000167  001546                  JMP     OUT
1911  006562  062767  020000  171726          ADD     #20000,ADRPT            ;NO, UPDATE MEMORY ADDRESS BY 4K
1912  006570  006367  171724                  ASL     BITPT                   ;UPDATE BIT POINTER
1913  006574  022767  000200  171716          CMP     #200,BITPT              ;THIS 28K DONE?
1914  006602  003354                          BGT     1$                      ;NO, CHECK NEXT 4K
1915  006604  000421                          BR      DONE16                  ;YES, EXIT
1916  006606  016767  171704  171714  3$:     MOV     ADRPT,HIADR             ;SET UPPER LIMIT THIS 4K
1917  006614  062767  020000  171706          ADD     #20000,HIADR
1918  006622  004767  005164                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1919  006626  016705  171664                  MOV     ADRPT,R5
1920  006632  005025          5$:     CLR     (5)+                            ;CLEAR BANK UNDER TEST
1921  006634  020567  171670                  CMP     R5,HIADR
```

```
1922  006640  103774                        BLO     5$
1923  006642  004767  000020        6$:     JSR     %7,WWP16                ;GO WRITE WRONG PARITY IN EACH BYTE
1924  006646  000736                        BR      2$                      ;UPDATE AND CHECK NEXT 4K
1925  006650  004767  005136        DONE16: JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS IF DONE
1926  006654  012737  000116 000114         MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
1927  006662  000167  000472                JMP     TEST17                  ;GO TO NEXT TEST
1928
1929                                        ;WRITE WRONG PARITY TEST ROUTINE - TESTS EACH BYTE IN 4K
1930                                        ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
1931                                        ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
1932  006666  016705  171624        WWP16:  MOV     ADRPT,R5                ;SET TEST ADDRESS POINTER
1933  006672  005067  171660                CLR     ODDFLG                  ;INDICATE TESTING LOW BYTE
1934  006676  012767  125253 171630         MOV     #125253,SHDBE           ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
1935  006704  012715  125253        WWP16A: MOV     #125253,@R5             ;INITIALIZE TEST LOCATION
1936  006710  012701  000566                MOV     #MPR0,R1                ;SETUP TO LOAD PARITY REGISTERS
1937  006714  032711  000001        1$:     BIT     #1,(1)
1938  006720  001003                        BNE     .+10
1939  006722  012771  000005 000000         MOV     #WWP+AE,@(R1)           ;SET WRITE WRONG PARITY AND ACTION
1940                                                                        ;ENABLE IF THIS PARITY REGISTER
1941                                                                        ;IS PRESENT
1942  006730  062701  000010                ADD     #10,R1
1943  006734  020127  000766                CMP     R1,#TREG
1944  006740  103765                        BLO     1$
1945  006742  005767  171610                TST     ODDFLG                  ;WRITING HIGH BYTE?
1946  006746  100425                        BMI     2$                      ;YES, BRANCH
1947  006750  112715  000253                MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
1948  006754  012701  000566        5$:     MOV     #MPR0,R1                ;THIS CODE CLEARS WWP BIT IN ALL
1949  006760  032711  000001                BIT     #1,(1)                  ;PARITY REGISTERS
1950  006764  001003                        BNE     .+10                    ;
1951  006766  042771  000004 000000         BIC     #WWP,@(R1)
1952  006774  062701  000010                ADD     #10,R1
1953  007000  020127  000766                CMP     R1,#TREG
1954  007004  103765                        BLO     5$+4
1955  007006  005767  171544                TST     ODDFLG                  ;TESTING HIGH OR LOW BYTE?
1956  007012  100407                        BMI     6$                      ;BRANCH,IF HIGH BYTE
1957  007014  142715  000377                BICB    #377,@R5                ;DETECT WRONG PARITY WITH DATIP-
1958                                                                        ;SHOULD TRAP TO TRP16 BEFORE DOING THE DATOB
1959  007020  000407                        BR      3$
1960  007022  112765  000252 000001  2$:     MOVB    #252,1(R5)              ;WRITE WRONG PARITY IN HIGH BYTE
1961  007030  000751                        BR      5$
1962  007032  142765  000377 000001  6$:     BICB    #377,1(R5)              ;DETECT WRONG PARITY WITH DATIP-
1963                                                                        ;SHOULD TRAP TO TRP16 BEFORE DOING THE DATOB
1964  007040  012701  000566          3$:     MOV     #MPR0,R1                ;NO TRAP OCCURRED- SETUP TO CLEAR
1965                                                                        ;AE AND WWP
1966  007044  032711  000001          4$:     BIT     #1,@R1
1967  007050  001003                        BNE     .+10
1968  007052  042771  000005 000000         BIC     #WWP+AE,@(R1)           ;CLEAR AE AND WWP IN ALL PARITY REGISTERS
1969  007060  062701  000010                ADD     #10,R1
1970  007064  020127  000766                CMP     R1,#TREG
1971  007070  103765                        BLO     4$
1972  007072  104002                        ERROR                           ;ERROR, NO TRAP AFTER WRITING AND
1973                                                                        ;READING WRONG PARITY IN LOCATION
1974                                                                        ;WHOSE ADDRESS IS IN R5 (AE AND WWP
1975                                                                        ;WERE SET IN ALL PARITY REGISTERS)
1976                                                                        ;TESTING LOW BYTE IF ODDFLG IS POSITIVE
1977                                                                        ;TESTING HIGH BYTE IF ODDFLG IS NEGATIVE
```

```
                                                                ;NOTE THAT AE AND WWP WERE CLEARED
                                                                ;BEFORE TYPING THE ERROR PRINTOUT
007074  000517                        BR      CN16

                        ;WHEN WRONG PARITY DATA IS READ BACK SHOULD ENTER HERE VIA TRAP TO 114
007076  012701  000566  TRP16:  MOV     #MPR0,R1                ;PARITY TRAP OCCURRED- FIRST CLEAR
007102  032711  000001  TRP16A: BIT     #1,@R1                  ;WWP AND AE IN ALL REGISTERS
007106  001003                  BNE     .+10
007110  042771  000005  000000  BIC     #AE+WWP,@(R1)
007116  062701  000010          ADD     #10,R1
007122  020127  000766          CMP     R1,#TREG
007126  103765                  BLO     TRP16A
007130  012701  000566          MOV     #MPR0,R1                ;FIND THE REGISTER THAT SENSED THE ERROR
007134  012703  000770          MOV     #INDC0,R3
007140  005067  171622          CLR     TREG
007144  032711  000001  1$:     BIT     #1,@R1                  ;DOES THIS CONTROL EXIST?
007150  001017                  BNE     2$                      ;NO, BRANCH
007152  005771  000000          TST     @(R1)                   ;YES- IS ERROR SET?
007156  100014                  BPL     2$                      ;NO, BRANCH
007160  005767  171602          TST     TREG                    ;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
007164  001401                  BEQ     .+4                     ;NO- BRANCH
007166  104002                  ERROR                           ;ERROR SET IN MORE THAN ONE PARITY REGISTER
                                                                ;AFTER WRITING WRONG PARITY IN LOCATION
                                                                ;WHOSE ADDRESS IS IN R5
007170  036761  171324  000002  BIT     BITPT,2(R1)             ;DOES MAP INDICATE THIS PARITY REGISTER
                                                                ;CONTROLS THIS MEMORY?
007176  001001                  BNE     .+4                     ;YES, BRANCH
007200  104002                  ERROR                           ;PARITY REGISTER RESPONDED TO MEMORY
                                                                ;NOT INCLUDED IN ITS MAP
                                                                ;PARITY REGISTER'S ADDRESS IS POINTED
                                                                ;TO BY R1. ADDRESS OF LOCATION CAUSING
                                                                ;PARITY ERROR IS IN R5

007202  010304                  MOV     R3,R4
007204  011167  171556          MOV     @R1,TREG                ;STORE REGISTER ADDRESS
007210  062701  000010  2$:     ADD     #10,R1
007214  005723                  TST     (R3)+
007216  020127  000766          CMP     R1,#TREG                ;BRANCH UNTIL ALL THE PARITY
007222  103750                  BLO     1$                      ;REGISTERS HAVE BEEN CHECKED

007224  011567  171306          MOV     (5),WAS                 ;SAVE DATA FROM LOCATION UNDER TEST
007230  022715  125253          CMP     #125253,@R5             ;DID BICB CHANGE DATA?
007234  001401                  BEQ     .+4                     ;NO, CONTINUE
007236  104003                  ERRORP                          ;DATA WAS MODIFIED BY THE BICB WHICH
                                                                ;GOT A PARITY ERROR TRAP- SINCE PARITY ERROR
                                                                ;TRAP OCCURRED, CONTENTS SHOULD NOT HAVE
                                                                ;BEEN MODIFIED. R5 CONTAINS ADDRESS
                                                                ;OF TEST LOCATION. "TREG" CONTAINS
                                                                ;ADDRESS OF PARITY REGISTER SENSING
                                                                ;ERROR
007240  005767  171522          TST     TREG                    ;WAS PARITY ERROR SET IN ANY REGISTERS?
007244  001002                  BNE     3$                      ;YES- BRANCH
007246  104002                  ERROR                           ;PARITY TRAP OCCURRED ON READING
                                                                ;WRONG PARITY (WITH AE SET) BUT NO
                                                                ;REGISTERS HAD PARITY ERROR BIT SET.
                                                                ;R5 CONTAINS THE ADDRESS OF THE
                                                                ;TEST LOCATION.
```

```
2034  007250  000420                    BR      4$
2035  007252  022714  000001    3$:     CMP     #1,(R4)
2036  007256  001015                    BNE     4$
2037  007260  017701  171502            MOV     @TREG,R1        ;GET PARITY REGISTER CONTENTS
2038  007264  042701  170037            BIC     #170037,R1      ;MASK OFF ALL BUT ERROR ADDRESS BITS
2039  007270  010502                    MOV     R5,R2           ;GET ADDRESS OF LOCATION UNDER TEST
2040  007272  042702  003777            BIC     #3777,R2        ;POSITION BITS IN R2
2041  007276  000302                    SWAB    R2
2042  007300  006302                    ASL     R2
2043  007302  006302                    ASL     R2
2044  007304  020102                    CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
2045  007306  001401                    BEQ     .+4
2046  007310  104003                    ERRORP                  ;ERROR ADDRESS BITS (PARITY REGISTER
2047                                                            ;BITS 11-5) ARE INCORRECT. R5 CONTAINS
2048                                                            ;THE ADDRESS OF THE TEST LOCATION.
2049                                                            ;"TREG" CONTAINS THE ADDRESS OF THE
2050                                                            ;PARITY REGISTER DETECTING THE ERROR
2051  007312  022626            4$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
2052  007314  011515            CNT16:  MOV     (5),(5)         ;RESTORE TEST LOCATION TO FIX BAD PARITY
2053  007316  005077  171444            CLR     @TREG           ;CLEAR ERROR BIT IN PARITY REGISTER
2054  007322  005715                    TST     @R5             ;READ LOCATION TO SEE IF PARITY IS GOOD
2055  007324  005777  171436            TST     @TREG           ;IS PARITY ERROR SET?
2056  007330  100001                    BPL     .+4             ;NO- BRANCH
2057  007332  104002                    ERROR                   ;WRITING LOCATION WITH WRITE WRONG PARITY
2058                                                            ;CLEAR DIDN'T CLEAR BAD PARITY
2059                                                            ;R5 CONTAINS ADDRESS OF THE TEST
2060                                                            ;LOCATION. "TREG" CONTAINS THE ADDRESS
2061                                                            ;OF THE PARITY REGISTER DETECTING
2062                                                            ;THE ERROR
2063  007334  005167  171216    CN16:   COM     ODDFLG          ;TOGGLE BYTE INDICATOR
2064  007340  100401                    BMI     .+4             ;BRANCH IF READY TO TEST HIGH BYTE
2065  007342  005725                    TST     (5)+            ;UPDATE ADDRESS POINTER
2066  007344  020567  171160            CMP     R5,HIADR        ;THIS 4K DONE?
2067  007350  103401                    BLO     1$              ;NO, TEST NEXT LOCATION
2068  007352  000207                    RTS     %7              ;RETURN TO TEST NEXT BANK
2069  007354  000167  177324    1$:     JMP     WWP16A
2070
2071
2072
2073                                    ;*****************************************************************
2074                                    ;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY ABOVE 28K
2075                                    ;WRITE WRONG PARITY AND READ IT WITH ACTION ENABLE SET, MAKING SURE
2076                                    ;THAT A TRAP OCCURS. THEN WRITE AND READ THE SAME LOCATION WITH GOOD PARITY
2077                                    ;(USING SAME DATA) TO SHOW THAT THE PARITY BIT TOGGLES. MAKE SURE THAT
2078                                    ;THE ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11) ARE CORRECT.
2079                                    ;*****************************************************************
2080  007360  104001            TEST17: SCOPE
2081  007362  004767  005032            JSR     PC,RSTLDR
2082  007366  012777  000017  171506    MOV     #17,@DISPLAY
2083  007374  012777  000017  171500    MOV     #17,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
2084  007402  005767  171142            TST     NOKT            ;KT11 PRESENT?
2085  007406  001402                    BEQ     .+6             ;YES, BRANCH
2086  007410  000167  000636            JMP     XFR1            ;NO, SKIP TO NEXT TEST
2087  007414  004767  004372            JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2088  007420  012737  007762  000114    MOV     #TRP17,@#PARVEC ;SETUP FOR PARITY TRAP
2089  007426  012767  000200  171064    MOV     #200,BITPT      ;INITIALIZE 4K BIT POINTER
```

```
2090  007434  004767  004252                    JSR     %7,NRALL        ;INITIALIZE ALL PAGES TO NON-RESIDENT
2091  007440  004767  004416                    JSR     %7,MAP1         ;MAP KERNEL 0 TO BANK 0,RW; KERNEL 7
2092                                                                      ;TO THE EXTERNAL BANK, RW. TURN ON
2093                                                                      ;THE KT11 AND MAKE KERNEL 1 RW
2094  007444  012777  001600  171600              MOV     #1600,@KPAR1    ;INITIALIZE KERNEL PAGE 1 TO 28K
2095  007452  005067  171076                    CLR     LOWFLG          ;CLEAR FLAG TO INDICATE TESTING LOWER 64K
2096  007456  016767  171406  171410              MOV     PMEML,PMEMX
2097  007464  012767  000002  000366              MOV     #2,INDX17       ;SETUP OFFSET TO CHECK LOWER 64K OF MAPS
2098  007472  036767  171022  171374  1$:         BIT     BITPT,PMEMX     ;DOES THIS 4K HAVE PARITY?
2099  007500  001025                            BNE     3$              ;YES, BRANCH TO TEST IT
2100  007502  062777  000200  171542  2$:         ADD     #200,@KPAR1     ;NO, MAP TO NEXT 4K
2101  007510  006367  171004                    ASL     BITPT           ;UPDATE BIT POINTER
2102  007514  103366                            BCC     1$              ;GO CHECK TO SEE IF THIS 4K HAS PARITY
2103  007516  005767  171032                    TST     LOWFLG          ;END OF 128K?
2104  007522  001025                            BNE     DONE17          ;YES, EXIT
2105  007524  005267  171024                    INC     LOWFLG          ;NO, SET FLAG TO INDICATE SHIFT TO
2106  007530  016767  171336  171336              MOV     PMEMH,PMEMX     ;HIGH 64K AND CHANGE MAPS
2107  007536  012767  000001  170754              MOV     #1,BITPT
2108  007544  012767  000004  000306              MOV     #4,INDX17       ;SETUP OFFSET TO CHECK UPPER 64K OF MAPS
2109  007552  000747                            BR      1$
2110  007554  012705  020000          3$:         MOV     #20000,R5
2111  007560  005025                  5$:         CLR     (5)+            ;CLEAR BANK UNDER TEST
2112  007562  020527  040000                    CMP     R5,#40000
2113  007566  103774                            BLO     5$
2114  007570  004767  000020          6$:         JSR     %7,WWP17        ;GO WRITE WRONG PARITY AND CHECK IT
2115  007574  000742                            BR      2$              ;UPDATE AND CHECK NEXT 4K
2116  007576  005037  177572          DONE17:     CLR     @#SR0           ;TURN OFF KT11 WHEN DONE
2117  007602  012737  000116  000114              MOV     #PARVEC+2,@#PARVEC
                                                                        ;RESTORE TRAP CATCHER
2118  007610  000167  000436                    JMP     XFR1            ;GO TO SETUP FOR NEXT TEST
2119
2120                          ;WRITE WRONG PARITY TEST ROUTINE TO TEST MEMORY ABOVE 28K
2121  007614  012705  020000  WWP17:             MOV     #20000,R5       ;SET TEST ADDRESS POINTER
2122  007620  005067  170732                    CLR     ODDFLG          ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
2123  007624  012767  125253  170702              MOV     #125253,SHDBE   ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
2124  007632  012715  125253  WWP17A:             MOV     #125253,@R5     ;INITIALIZE LOCATION
2125  007636  012701  000566                    MOV     #MPR0,R1        ;INITIALIZE REGISTER ADDRESS POINTER
2126  007642  032711  000001  1$:               BIT     #1,(1)          ;DOES THIS CONTROL EXIST?
2127  007646  001003                            BNE     .+10            ;NO, GET NEXT
2128  007650  012771  000005  000000              MOV     #WWP+AE,@(R1)   ;YES- SET WRITE WRONG PARITY
2129                                                                      ;AND ACTION ENABLE
2130  007656  062701  000010                    ADD     #10,R1
2131  007662  020127  000766                    CMP     R1,#TREG        ;ALL REGISTERS SETUP?
2132  007666  103765                            BLO     1$              ;NO- LOOP
2133  007670  005767  170662                    TST     ODDFLG          ;YES- TESTING HIGH BYTE?
2134  007674  100405                            BMI     2$              ;YES, BRANCH
2135  007676  112715  000253                    MOVB    #253,@R5        ;NO, WRITE WRONG PARITY IN LOW BYTE
2136  007702  142715  000377                    BICB    #377,@R5        ;DETECT WRONG PARITY WITH DATIP-
2137                                                                      ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
2138  007706  000406                            BR      3$
2139  007710  112765  000252  000001  2$:         MOVB    #252,1(R5)      ;WRITE WRONG PARITY IN HIGH BYTE
2140  007716  142765  000377  000001              BICB    #377,1(R5)      ;DETECT WRONG PARITY WITH DATIP
2141                                                                      ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
2142  007724  012701  000566          3$:         MOV     #MPR0,R1        ;IF NO TRAP, CLEAR AE AND WWP IN ALL
2143  007730  032711  000001          4$:         BIT     #1,@R1          ;PARITY REGISTERS
2144  007734  001003                            BNE     .+10
2145  007736  042771  000005  000000              BIC     #AE+WWP,@(R1)
```

```
2146  007744  062701  000010                ADD     #10,R1
2147  007750  020127  000766                CMP     R1,#TREG
2148  007754  103765                        BLO     4$
2149  007756  104002                        ERROR                   ;NO TRAP AFTER WRITING AND READING
2150                                                                  ;WRONG PARITY WITH AE SET- VIRTUAL
2151                                                                  ;ADDRESS OF LOCATION IS IN R5
2152                                                                  ;(MAPPED THRU KERNEL PAGE 1)
2153                                                                  ;WROTE LOW BYTE IF ODDFLG IS POSITIVE
2154                                                                  ;WROTE HIGH BYTE IF IT IS NEGATIVE
2155                                                                  ;NOTE THAT WWP AND AE WERE CLEARED
2156                                                                  ;BEFORE ERROR PRINTOUT
2157  007760  000512                        BR      CNT17
2158
2159                                ;WHEN WRONG PARITY DATA IS READ BACK, SHOULD ENTER HERE VIA TRAP TO 114
2160  007762  012701  000566        TRP17:  MOV     #MPR0,R1        ;PARITY TRAP OCCURRED- BEFORE CHECKING
2161  007766  032711  000001        TRP17A: BIT     #1,@R1          ;IT, CLEAR WWP AND AE IN ALL REGISTERS
2162  007772  001003                        BNE     .+10
2163  007774  042771  000005  000000      BIC     #AE+WWP,@(R1)
2164  010002  062701  000010              ADD     #10,R1
2165  010006  020127  000766              CMP     R1,#TREG
2166  010012  103765                      BLO     TRP17A
2167  010014  012701  000566              MOV     #MPR0,R1                ;FIND REGISTER THAT SENSED THE ERROR
2168  010020  012703  000770              MOV     #INDC0,R3                       ;SETUP PTR TO INDICATOR
2169  010024  005067  170736              CLR     TREG
2170  010030  032711  000001        1$:   BIT     #1,@R1          ;DOES THIS CONTROL EXIST?
2171  010034  001017                      BNE     2$              ;NO, BRANCH
2172  010036  005771  000000              TST     @(R1)           ;YES- IS ERROR SET?
2173  010042  100014                      BPL     2$              ;NO, BRANCH
2174  010044  005767  170716              TST     TREG            ;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
2175  010050  001401                      BEQ     .+4             ;NO- BRANCH
2176  010052  104002                      ERROR                   ;ERROR SET IN MORE THAN ONE PARITY REGISTER
2177                                                                ;AFTER READING WRONG PARITY IN LOCATION
2178                                                                ;WHOSE VIRTUAL ADDRESS IS IN R5
2179  010054  036761  170440  000002      BIT     BITPT,2(R1)     ;DOES MAP INDICATE THAT THIS REGISTER
2180                                                                ;CONTROLS THIS MEMORY?
2181          010060                      INDX17=.-2
2182  010062  001001                      BNE     .+4             ;YES- BRANCH
2183  010064  104002                      ERROR                   ;PARITY REGISTER RESPONDED TO MEMORY
2184                                                                ;NOT INCLUDED IN ITS MAP. R1 POINTS TO
2185                                                                ;THE PARITY REGISTER'S ADDRESS. R5
2186                                                                ;CONTAINS VIRTUAL ADDRESS OF THE LOCATION
2187                                                                ;BEING TESTED (MAPPED THRU KERNEL
2188                                                                ;PAGE 1)
2189  010066  011167  170674              MOV     @R1,TREG        ;STORE REGISTER ADDRESS
2190  010072  010304                      MOV     R3,R4           ;STORE PTR TO INDICATOR
2191  010074  062701  000010        2$:   ADD     #10,R1
2192  010100  005723                      TST     (R3)+           ;INCREMENT PTR TO INDICATOR
2193  010102  020127  000766              CMP     R1,#TREG
2194  010106  103750                      BLO     1$              ;LOOP UNTIL ALL THE PARITY REGISTERS
2195                                                                ;HAVE BEEN CHECKED
2196  010110  011567  170422              MOV     (5),WAS         ;SAVE CONTENTS OF LOCATION UNDER TEST
2197  010114  022715  125253              CMP     #125253,@R5     ;DID BICB CHANGE DATA?
2198  010120  001401                      BEQ     .+4             ;NO, CONTINUE
2199  010122  104002                      ERROR                   ;DATA WAS MODIFIED BY THE BICB WHICH
2200                                                                ;GOT A PARITY ERROR TRAP- SINCE PARITY
2201                                                                ;TRAP OCCURRED, CONTENTS SHOULD NOT
```

```
2202
2203
2204
2205
2206  010124  005767  170636          TST     TREG
2207  010130  001003                  BNE     3$
2208  010132  104002                  ERROR
2209
2210
2211
2212
2213
2214  010134  000423                  BR      4$
2215  010136  001022                  BNE     4$
2216  010140  022714  000001    3$:   CMP     #1,(R4)
2217  010144  001017                  BNE     4$
2218
2219
2220
2221
2222  010146  017701  170614          MOV     @TREG,R1
2223  010152  042701  170037          BIC     #170037,R1
2224  010156  010502                  MOV     R5,R2
2225  010160  042702  163777          BIC     #163777,R2
2226  010164  000302                  SWAB    R2
2227  010166  006302                  ASL     R2
2228  010170  006302                  ASL     R2
2229  010172  067702  171054          ADD     @KPAR1,R2
2230  010176  020102                  CMP     R1,R2
2231  010200  001401                  BEQ     .+4
2232  010202  104004                  ERRORS
2233
2234
2235
2236
2237
2238  010204  022626            4$:   CMP     (SP)+,(SP)+
2239  010206  011515          CNT17:  MOV     (5),(5)
2240  010210  005077  170552          CLR     @TREG
2241  010214  005715                  TST     @R5
2242  010216  005777  170544          TST     @TREG
2243  010222  100001                  BPL     .+4
2244  010224  104002                  ERROR
2245
2246
2247
2248
2249
2250  010226  005167  170324          COM     ODDFLG
2251  010232  100401                  BMI     .+4
2252  010234  005725                  TST     (5)+
2253  010236  020527  040000          CMP     R5,#40000
2254  010242  103401                  BLO     1$
2255  010244  000207                  RTS     %7
2256
2257  010246  000167  177360    1$:   JMP     WWP17A

;HAVE BEEN MODIFIED. R5 CONTAINS TEST
;LOCATION ADDRESS (VIRTUAL, MAPPED THRU
;KERNEL PAGE 1). "TREG" CONTAINS ADDRESS OF
;PARITY REGISTER DETECTING PARITY ERROR
;WAS PARITY ERROR SET IN ANY REGISTER?
;YES- BRANCH
;PARITY TRAP OCCURRED ON READING
;WRONG PARITY WITH AE SET BUT NO
;REGISTER HAS THE PARITY ERROR BIT
;SET. R5 CONTAINS VIRTUAL ADDRESS
;OF THE TEST LOCATION (MAPPED THRU
;KERNEL PAGE 1)

;NO, BRANCH. DO NOT CHECK THE
;IS THIS A CORE PARITY REGISTER?
;NO, BRANCH. DO NOT CHECK THE
;ADDRESS BITS OF THE LAST PARITY
;ERROR
;IF CORE REG, CHECK IF ADDRESS
;BITS OF LAST PAR ERR WERE STORED
;GET CONTENTS OF REGISTER DETECTING PARITY ERROR
;MASK OFF ALL BUT ADDRESS BITS
;CALCULATE TOP 7 BITS OF ERROR ADDRESS

;PARITY ERROR ADDRESS BITS CORRECT?

;PARITY ERROR ADDRESS BITS (PARITY
;REGISTER BITS 11-5) INCORRECT
;"TREG" CONTAINS ADDRESS OF PARITY
;REGISTER. R5 CONTAINS THE VIRTUAL
;ADDRESS OF THE TEST LOCATION
;(MAPPED THRU KERNEL PAGE 1)
;RESTORE STACK POINTER
;RESTORE TEST LOCATION TO FIX BAD PARITY
;CLEAR ERROR BIT IN THE PARITY REGISTER
;READ LOCATION TO SEE IF PARITY IS GOOD
;IS PARITY ERROR SET?
;NO, BRANCH
;WRITING LOCATION WITH WRITE WRONG
;PARITY CLEAR DION'T CLEAR BAD PARITY
;"TREG" CONTAINS THE ADDRESS OF THE
;PARITY REGISTER. R5 CONTAINS THE
;VIRTUAL ADDRESS OF THE TEST LOCATION
;(MAPPED THRU KERNEL PAGE 1)
;TOGGLE BYTE INDICATOR
;BRANCH IF HIGH BYTE NOT YET TESTED
;UPDATE ADDRESS POINTER
;THIS 4K DONE?
;NO, TEST NEXT LOCATION
;YES, RETURN TO CHECK FOR NEXT
;BANK TO BE TESTED
;GO AND TEST NEXT LOCATION
```

```
2258  010252  104001                    XFR1:   SCOPE
2259                                    ;IF THE FIRST BANK (BANK 0) IS PARITY MEMORY, SETUP TO TEST IT
2260                                    ;COPY THE FIRST 4K TO THE SECOND 4K, MOVE THE STACK POINTER TO BANK 1,
2261                                    ;AND THEN JUMP TO THE COPY OF TEST20 IN BANK 1
2262  010254  012767  000002  170236            MOV     #2,BITPT                ;XXXXX
2263  010262  000167  175052                   JMP     TEST14                  ;XXXX
2264  010266  005737  000042                     TST   @#42
2265  010272  001402                           BEQ     .+6
2266  010274  000167  001324                   JMP     DONE                    ;IF THE PROGRAM IS RUNNING UNDER ACT THEN
2267                                                                            ;GO TO DONE
2268  010300  032767  000001  170562            BIT     #1,PMEML
2269  010306  001002                           BNE     .+6                     ;BRANCH IF YES
2270  010310  000167  001310                   JMP     DONE                    ;NO- DONE WITH TEST
2271  010314  032767  000002  170542            BIT     #2,MEML                 ;IS THERE A SECOND 4K(BANK 1)?
2272  010322  001002                           BNE     .+6                     ;YES, BRANCH
2273  010324  000167  001274                   JMP     DONE                    ;NO, EXIT
2274  010330  005037  177776        OUT:      CLR     @#PS                    ;CLEAR STATUS REGISTER
2275  010334  004767  003452                   JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2276  010340  012700  017770                   MOV     #17770,R0               ;R0 IS COUNTER TO MOVE 4K
2277  010344  005001                           CLR     R1                      ;R1 POINTS TO LOCATION IN BANK 0
2278  010346  011161  020000        1$:       MOV     @R1,20000(R1)           ;COPY FROM BANK 0 TO BANK 1
2279  010352  005721                           TST     (R1)+                   ;MOVE POINTER
2280  010354  005300                           DEC     R0                      ;DONE WITH 4K?
2281  010356  001373                           BNE     1$                      ;NO- BRANCH
2282  010360  062706  020000                   ADD     #20000,SP               ;YES, MOVE STACK POINTER TO POINT TO BANK 1
2283  010364  012767  030462  026070          MOV     #TEST20+20010,RETURN+20000      ;UPDATE SCOPE RETURN IN BANK 1
2284  010372  062767  020000  023232          ADD     #20000,ERRA1+20000      ;UPDATE ADDRESSES USED IN ERROR TYPEOUT
2285  010400  062767  020000  023226          ADD     #20000,ERRA2+20000
2286  010406  062767  020000  023230          ADD     #20000,ERRA3+20000
2287  010414  062767  020000  023224          ADD     #20000,ERRA4+20000
2288  010422  062767  020000  023226          ADD     #20000,ERRA5+20000
2289  010430  022767  177570  170442          CMP     #177570,SWR             ;DOES THE CONSOLE HAVE A SWITCH REG. ?
2290  010436  001403                           BEQ     2$                      ;IF SO THEN GO TO 2$
2291  010440  062767  020000  010432          ADD     #20000,SWR+20000        ;OTHERWISE THERE IS A SOFTWARE SWITCH LOCATION
2292                                                                            ;AND ITS ADDRESS SHOULD BE UPDATED
2293  010446  000167  020010        2$:       JMP     TEST20+20010            ;GO TO TEST 20 IN BANK 1
2294
2295
2296
2297                                    ;*************************************************************************
2298                                    ;IF FIRST 4K IS PARITY MEMORY, CHECK IT WITH A SERIES OF PATTERNS
2299                                    ;THIS SUBTEST IS RUN IN BANK 1 (20000 ABOVE THE ADDRESSES IN THE LISTING)
2300                                    ;*************************************************************************
2301  010452  013746  177776        TEST20: MOV     @#PS,-(SP)              ;THESE 2 LINES DO THE SAME AS A SCOPE WITHOUT
2302  010456  004767  005674                JSR     PC,SCOPEC               ;USING AN EMT
2303  010462  012777  000020  170412        MOV     #20,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
2304  010470  004767  003316                JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2305  010474  012704  021036                MOV     #PARPAT+20000,R4        ;INITIALIZE PATTERN POINTER
2306  010500  005005                        CLR     R5
2307  010502  005025                1$:     CLR     (5)+                    ;INITIALLY CLEAR BANK 0
2308  010504  020527  020000                CMP     R5,#20000
2309  010510  103774                        BLO     1$
2310  010512  012737  030720  000114        MOV     #TRP20+20000,@#PARVEC   ;SETUP TRAP RETURN
2311  010520  012701  020566                MOV     #MPR0+20000,R1          ;SETUP TO SET ACTION ENABLE IN ALL
2312                                                                            ;PARITY REGISTERS PRESENT
2313  010524  032711  000001        2$:     BIT     #1,@R1
```

```
2314  010530  001003                        BNE     .+10
2315  010532  012771  000001  000000        MOV     #AE,@(R1)       ;SET ACTION ENABLE IF REGISTER IS PRESENT
2316  010540  062701  000010                ADD     #10,R1
2317  010544  020127  020766                CMP     R1,#TREG+20000
2318  010550  103765                        BLO     2$              ;BRANCH UNTIL ALL REGISTER ADDRESSES
2319                                                                 ;HAVE BEEN CHECKED
2320  010552  004767  000022          3$:   JSR     %7,CKBK0        ;EXERCISE THIS 4K
2321  010556  005724                        TST     (4)+            ;UPDATE PATTERN
2322  010560  005714                        TST     (4)             ;LAST PATTERN?
2323  010562  001373                        BNE     3$              ;NO, LOOP
2324  010564  004767  003222      DONE20:   JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2325  010570  012737  000116  000114        MOV     #PARVEC+2,@#PARVEC
                                                                        ;RESTORE TRAP CATCHER
2326  010576  000526                        BR      TEST21          ;GO TO NEXT TEST
2327
```

```
                        ;PARITY MEMORY TEST ROUTINE
                        ;WRITES AND CHECKS EACH LOCATION IN BANK 0 (EXCEPT 114 AND 116)
                        ;WITH VALUE POINTED TO BY R4
010600  005005          CKBK0:  CLR     R5                      ;SET ADDRESS POINTER
010602  011415          1$:     MOV     (4),(5)                 ;WRITE PATTERN
010604  021415                  CMP     (4),(5)                 ;DATA OK?
010606  001404                  BEQ     .+6                     ;YES- BRANCH
010610  013746  177776          MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
010614  004767  002742          JSR     PC,ERR                  ;ERROR- DATA INCORRECT IN LOCATION
                                                                ;WHOSE ADDRESS IS IN R5. R4 POINTS
                                                                ;TO THE VALUE WRITTEN.
010620  005725                  TST     (5)+                    ;UPDATE ADDRESS POINTER
010622  020527  000114          CMP     R5,#114                 ;DON'T CHANGE CONTENTS OF 114 AND 116
010626  001002                  BNE     .+6
010630  062705  000004          ADD     #4,R5
010634  020527  020000          CMP     R5,#20000               ;HAS THE WHOLE BANK BEEN TESTED WITH
                                                                ;THIS PATTERN?
010640  103760                  BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
010642  005067  170120          CLR     TREG                    ;YES, DID ANY PARITY ERROR BITS SET?
010646  012701  020566          MOV     #MPR0+20000,R1
010652  032711  000001  2$:     BIT     #1,(1)
010656  001003                  BNE     .+10
010660  005771  000000          TST     @(R1)
010664  100406                  BMI     3$                      ;YES- BRANCH
010666  062701  000010          ADD     #10,R1
010672  020127  020766          CMP     R1,#TREG+20000
010676  103765                  BLO     2$
010700  000207                  RTS     %7                      ;NO- RETURN
010702  013746  177776  3$:     MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
010706  004767  002650          JSR     PC,ERR                  ;ERROR- PARITY ERROR BIT SET AND NO
                                                                ;PARITY TRAP OCCURRED
                                                                ;(AE WAS SET) - R1 POINTS TO ADDRESS
                                                                ;OF PARITY REGISTER
010712  022626                  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
010714  000167  177644          JMP     DONE20

                        ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
010720  005067  170042  TRP20:  CLR     TREG
010724  012701  020566          MOV     #MPR0+20000,R1          ;FIND THE REGISTER RECORDING A PARITY ERROR
010730  032711  000001  1$:     BIT     #1,(R1)
010734  001003                  BNE     .+10
010736  005771  000000          TST     @(R1)
010742  100412                  BMI     2$                      ;BRANCH IF ERROR IS SET
010744  062701  000010          ADD     #10,R1
010750  020127  020766          CMP     R1,#TREG+20000
010754  103765                  BLO     1$
010756  013746  177776          MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA A JSR
010762  004767  002574          JSR     PC,ERR                  ;ERROR- PARITY TRAP TO 114 OCCURRED DURING
                                                                ;TEST 20 BUT NO REGISTERS HAVE PARITY
010766  000430                  BR      5$                      ;ERROR SET
010770  017102  000000  2$:     MOV     @(R1),R2                ;SAVE CONTENTS OF PARITY REGISTER
010774  005003                  CLR     R3                      ;INITIALIZE ADDRESS POINTER
010776  005071  000000  3$:     CLR     @(R1)                   ;CLEAR PARITY REGISTER
011002  005713                  TST     @R3                     ;READ LOCATION
011004  005771  000000          TST     @(R1)                   ;PARITY ERROR SET?
011010  100413                  BMI     4$                      ;YES- BRANCH
```

```
2384  011012  005723                  TST     (R3)+                   ;MOVE POINTER
2385  011014  020327  020000          CMP     R3,#20000
2386  011020  103766                  BLO     3$
2387  011022  010271  000000          MOV     R2,@(R1)                ;RESTORE PARITY REGISTER CONTENTS
2388  011026  013746  177776          MOV     @#PS,-(SP)              ;SETUP TO CALL ERROR VIA JSR
2389  011032  004767  002524          JSR     PC,ERR                  ;PARITY ERROR OCCURRED WHILE TESTING
2390                                                                  ;BANK 0 BUT NO BAD PARITY WAS FOUND
2391                                                                  ;DURING SCAN OF BANK 0. R1 POINTS
2392                                                                  ;TO THE ADDRESS OF THE PARITY REGISTER
2393                                                                  ;DETECTING THE ERROR
2394  011036  000404                  BR      5$
2395  011040  013746  177776  4$:     MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
2396  011044  004767  002512          JSR     PC,ERR                  ;PARITY ERROR WHILE EXERCISING MEMORY
2397                                                                  ;BANK 0- R1 POINTS TO ADDRESS OF PARITY
2398                                                                  ;REGISTER DETECTING THE ERROR. R3 CONTAINS
2399                                                                  ;THE ADDRESS OF THE LOCATION HAVING
2400                                                                  ;BAD PARITY
2401  011050  022626          5$:     CMP     (SP)+,(SP)+
2402  011052  000207                  RTS     %7                      ;RETURN TO CHECK USING THE NEXT PATTERN
2403
2404
2405
2406                                  ;*************************************************************************
2407                                  ;FORCE WRONG PARITY IN EACH LOCATION IN BANK 0
2408                                  ;NOTE THAT THIS SUBTEST IS EXECUTED IN BANK 1 (20000 ABOVE ADDRESSES
2409                                  ;IN THE LISTING). MAKE SURE THAT WRONG PARITY IN EACH BYTE CAN BE DETECTED,
2410                                  ;AND THAT WHEN GOOD PARITY IS WRITTEN AND READ NO PARITY ERROR IS DETECTED.
2411                                  ;CHECK ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11)
2412                                  ;*************************************************************************
2413  011054  013746  177776  TEST21: MOV     @#PS,-(SP)              ;SAME AS SCOPE WITHOUT DOING EMT
2414  011060  004767  005272          JSR     PC,SCOPEC
2415  011064  012777  000021  170010  MOV     #21,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
2416  011072  004767  002714          JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2417  011076  005005                  CLR     R5
2418  011100  005025          1$:     CLR     (R5)+                   ;CLEAR PARITY ERRORS IN BANK 0
2419  011102  020527  020000          CMP     R5,#20000
2420  011106  103774                  BLO     1$
2421
2422                                  ;WRITE WRONG PARITY TEST ROUTINE
2423                                  ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
2424                                  ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
2425  011110  005005          WWP21:  CLR     R5                      ;SET TEST ADDRESS POINTER
2426  011112  005067  167440          CLR     ODDFLG                  ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
2427  011116  012767  125253  167410  MOV     #125253,SHDBE           ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
2428  011124  012715  125253  WWP21A: MOV     #125253,@R5             ;INITIALIZE LOCATION
2429  011130  012701  020566          MOV     #MPR0+20000,R1          ;SETUP TO SET WRITE WRONG PARITY
2430                                                                  ;IN ALL REGISTERS
2431  011134  032711  000001  1$:     BIT     #1,(1)
2432  011140  001003                  BNE     .+10
2433  011142  012771  000004  000000  MOV     #WWP,@(R1)              ;SET WRITE WRONG PARITY
2434  011150  062701  000010          ADD     #10,R1
2435  011154  020127  020766          CMP     R1,#TREG+20000
2436  011160  103765                  BLO     1$
2437  011162  005767  167370          TST     ODDFLG                  ;TESTING HIGH BYTE?
2438  011166  100404                  BMI     2$                      ;YES, BRANCH
2439  011170  112715  000253          MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
```

```
2440  011174  005715                          TST     (R5)            ;DETECT WRONG PARITY WITH DATI
2441  011176  000405                          BR      3$
2442  011200  112765  000252  000001  2$:     MOVB    #252,1(R5)      ;WRITE WRONG PARITY IN HIGH BYTE
2443  011206  105765  000001                  TSTB    1(R5)           ;DETECT WRONG PARITY WITH DATI
2444  011212  012701  020566          3$:     MOV     #MPR0+20000,R1  ;SETUP TO CLEAR WWP IN ALL PARITY REGISTERS
2445  011216  012703  000770                  MOV     #INDCO,R3       ;POINTER TO INDICATOR
2446  011222  032711  000001          4$:     BIT     #1,@R1
2447  011226  001003                          BNE     .+10
2448  011230  042771  000004  000000          BIC     #WWP,@(R1)      ;CLEAR WWP IN ALL PARITY REGISTERS
2449  011236  062701  000010                  ADD     #10,R1
2450  011242  020127  020766                  CMP     R1,#TREG+20000
2451  011246  103765                          BLO     4$
2452  011250  012701  020566                  MOV     #MPR0+20000,R1  ;CHECK TO SEE IF ANY REGISTER DETECTED
2453  011254  005067  167506                  CLR     TREG            ;THE PARITY ERROR
2454  011260  032711  000001          LOOP21: BIT     #1,@R1
2455  011264  001024                          BNE     5$
2456  011266  005771  000000                  TST     @(R1)           ;CHECK PARITY REGISTER
2457  011272  100021                          BPL     5$              ;BRANCH IF PARITY ERROR IS NOT SET
2458  011274  005767  167466                  TST     TREG            ;WAS PARITY ERROR SET IN ANY OTHER REGISTER?
2459  011300  001404                          BEQ     .+12            ;NO- BRANCH
2460  011302  013746  177776                  MOV     @#PS,-(SP)      ;SETUP TO DO ERROR CALL VIA A JSR
2461  011306  004767  002250                  JSR     PC,ERR          ;ERROR- PARITY ERROR SET IN MORE THAN
2462                                                                  ;ONE REGISTER AFTER WRITING WRONG PARITY
2463                                                                  ;IN LOCATION WHOSE ADDRESS IS IN R5
2464  011312  032761  000001  000002          BIT     #1,2(R1)        ;DOES MAP INDICATE THAT THIS REGISTER
2465                                                                  ;CONTROLS THIS MEMORY?
2466  011320  001004                          BNE     .+12            ;YES, BRANCH
2467  011322  013746  177776                  MOV     @#PS,-(SP)      ;NO- SETUP TO DO ERROR CALL VIA A JSR
2468  011326  004767  002230                  JSR     PC,ERR          ;ERROR- PARITY REGISTER RESPONDED
2469                                                                  ;TO MEMORY WHICH IS NOT IN ITS MAP.
2470                                                                  ;R1 POINTS TO PARITY REGISTER'S ADDRESS.
2471                                                                  ;R5 CONTAINS ADDRESS OF LOCATION BEING
2472                                                                  ;TESTED
2473  011332  011167  167430                  MOV     @R1,TREG        ;STORE REGISTER ADDRESS
2474  011336  062701  000010          5$:     ADD     #10,R1
2475  011342  005723                          TST     (R3)+
2476  011344  020127  020766                  CMP     R1,#TREG+20000
2477  011350  103743                          BLO     LOOP21          ;BRANCH UNTIL ALL REGISTERS HAVE BEEN
2478                                                                  ;CHECKED
2479  011352  005767  167410                  TST     TREG            ;WAS PARITY ERROR SET IN ANY REGISTER?
2480  011356  001005                          BNE     6$              ;YES- BRANCH
2481  011360  013746  177776                  MOV     @#PS,-(SP)      ;NO- SETUP TO DO ERROR CALL VIA A JSR
2482  011364  004767  002172                  JSR     PC,ERR          ;ERROR- NO REGISTER HAS PARITY ERROR
2483                                                                  ;SET AFTER READING WRONG PARITY IN LOCATION
2484                                                                  ;WHOSE ADDRESS IS IN R5
2485  011370  000423                          BR      7$
2486  011372  022713  000001          6$:     CMP     #1,(R3)         ;IS THIS A CORE PAR REG?
2487  011376  001020                          BNE     7$              ;NO, BRANCH(MOS PAR REG DOES NOT
2488                                                                  ;STORE ADDR BITS OF PAR ERROR)
2489  011400  017701  167362                  MOV     @TREG,R1        ;GET PARITY REGISTER CONTENTS
2490  011404  042701  170037                  BIC     #170037,R1      ;MASK ALL BUT ERROR ADDRESS BITS
2491  011410  010502                          MOV     R5,R2           ;GET ADDRESS OF LOCATION UNDER TEST
2492  011412  042702  003777                  BIC     #3777,R2        ;POSITION BITS IN R2
2493  011416  000302                          SWAB    R2
2494  011420  006302                          ASL     R2
2495  011422  006302                          ASL     R2
```

```
2496  011424  020102                    CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
2497  011426  001404                    BEQ     7$              ;BRANCH IF YES
2498  011430  013746  177776            MOV     @#PS,-(SP)      ;NO- SETUP TO DO ERROR CALL VIA A JSR
2499  011434  004767  002122            JSR     PC,ERR          ;ERROR- ADDRESS BITS (PARITY REGISTER
2500                                                            ;BITS 5-11) INCORRECT- ADDRESS OF PARITY
2501                                                            ;REGISTER IS CONTAINED IN LOCATION "TREG"
2502                                                            ;ADDRESS OF TEST LOCATION IS IN R5
2503  011440  011515            7$:     MOV     @R5,@R5         ;RESTORE TEST LOCATION TO FIX BAD PARITY
2504  011442  005077  167320            CLR     @TREG           ;CLEAR ERROR BIT IN PARITY REGISTER
2505  011446  005715                    TST     @R5             ;READ LOCATION TO SET IF PARITY IS GOOD
2506  011450  005777  167312            TST     @TREG           ;CHECK PARITY ERROR BIT
2507  011454  100004                    BPL     .+12            ;BRANCH IF NOT SET
2508  011456  013746  177776            MOV     @#PS,-(SP)      ;SETUP TO DO ERROR CALL VIA A JSR
2509  011462  004767  002074            JSR     PC,ERR          ;ERROR- WRITING LOCATION WITH WRITE
2510                                                            ;WRONG PARITY CLEAR DIDN'T CLEAR BAD
2511                                                            ;PARITY (ADDRESS OF LOCATION IS IN R5)
2512                                                            ;"TREG" +20000 CONTAINS THE ADDRESS
2513                                                            ;OF THE PARITY REGISTER
2514  011466  005167  167064            COM     ODDFLG          ;TOGGLE BYTE INDICATOR
2515  011472  100401                    BMI     .+4             ;BRANCH IF HIGH BYTE NOT YET TESTED
2516  011474  005725                    TST     (R5)+           ;UPDATE ADDRESS POINTER
2517  011476  020527  020000            CMP     R5,#20000       ;THIS 4K DONE?
2518  011502  103610                    BLO     WWP21A          ;LOOP TILL ALL 4K HAS BEEN TESTED
2519  011504  004767  002302            JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2520  011510  013746  177776            MOV     @#PS,-(SP)      ;SETUP TO CALL SCOPE VIA JSR
2521  011514  004767  004636            JSR     PC,SCOPEC       ;SCOPE
2522
2523                                    ;COPY SECOND 4K BANK BACK TO FIRST 4K AND RETURN TO FIRST 4K BANK
2524  011520  012700  017770    XFR2:   MOV     #17770,R0       ;R0 IS USED AS A COUNTER
2525  011524  005001                    CLR     R1              ;R1 POINTS TO THE CURRENT LOCATION
2526  011526  016111  020000    1$:     MOV     20000(R1),@R1   ;COPY BANK 1 TO BANK 0
2527  011532  005721                    TST     (R1)+
2528  011534  005300                    DEC     R0
2529  011536  001373                    BNE     1$
2530  011540  162706  020000            SUB     #20000,SP       ;RESTORE STACK POINTER
2531  011544  162737  020000  013632    SUB     #20000,@#ERRA1
2532  011552  162737  020000  013634    SUB     #20000,@#ERRA2
2533  011560  162737  020000  013644    SUB     #20000,@#ERRA3
2534  011566  162737  020000  013646    SUB     #20000,@#ERRA4
2535  011574  162737  020000  013656    SUB     #20000,@#ERRA5
2536  011602  022737  177570  001100    CMP     #177570,@#SWR   ;DOES THE CONSOLE HAVE A SWITCH REG. ?
2537  011610  001403                    BEQ     2$              ;IF SO THEN GO TO 2$
2538  011612  162737  020000  001100    SUB     #20000,@#SWR    ;OTHERWISE RESTORE THE ADDRESS OF SOFTWARE
2539                                                            ;SWITCH REGISTER
2540  011620  000167  160000    2$:     JMP     DONE-20000              ;RETURN TO BANK 0
2541
2542
2543                                    ;*************************************************************************
2544                                    ;AT THIS POINT EXECUTION RETURNS TO BANK 0
2545                                    ;*************************************************************************
2546
2547  011624  012737  000116  000114 DONE:   MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
2548  011632  012706  000510            MOV     #STKPT,SP       ;REINITIALIZE STACK POINTER
2549  011636  004767  002150            JSR     %7,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2550  011642  005237  011752            INC     @#PASCNT                ;KEEP TRACK OF PASSES COMPLETED
2551  011646  004567  003544            JSR     R5,OACNV
```

```
                                        PASCNT
                                        MPCNT
                                        6
                                        TYPE                            ;TYPE BELL, "END PASS=" AND PASS COUNT
                                        MPGEND
                                        CLR     R0
                                1$:     INC     R0                      ;ALLOW TIME FOR END PASS MESSAGE TO PRINT
                                        BNE     1$
                                LOGICAL:        MOV     @#42,R5         ;LOADED BY MONITOR?
                                        BEQ     CONT                    ;BRANCH IF NO
                                        RESET
                                $ENDAD: JSR     7,(5)                   ;GO TO MONITOR
                                        NOP
                                        NOP
                                        NOP
                                CONT:   BIT     #BIT8,@SWR              ;SWITCH 8 SET?
                                        BEQ     .+4
                                        HALT                            ;HALT AT END OF PASS SET
                                        TSTB    $TPFLG
                                        BNE     1$                      ;IF NO TERMINAL, SKIP
                                        TSTB    @TPS                    ;WAIT FOR TTY TO FINISH SO THAT RESET
                                        BPL     .-4                     ;WON'T CLOBBER THE BELL
                                        MOVB    #0,@TPB                 ;OUTPUT A NULL
                                1$:     JMP     BEGIN
                                PASCNT: 0                       ;PASS COUNT
                                MONCNT: 0


                                ;*************************************************************************
                                ;CREATE MAP INDICATING WHERE 4K BLOCKS OF MEMORY ARE PRESENT
                                ;*************************************************************************
                                MAPMEM: MOV     #MAPMB,@#4              ;SET NO MEM MANAGEMENT TRAP
                                        TST     @#SRD                   ;IS KT PRESENT? (TIMEOUT IF NO)

                                ;MAP MEMORY USING KT11 - MAX OF 124K POSSIBLE
                                MAPMA:  CLR     NOKT                    ;INDICATE KT11 PRESENT
                                        JSR     %7,NRALL                ;INITIALLY SET ALL PAGES NONRESIDENT, BANK 0
                                        JSR     %7,MAP1                 ;MAP KERNEL 0 TO BANK 0, RW
                                                                        ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
                                                                        ;MAP KERNEL 1 RW, AND TURN ON KT11

                                        CLR     TBANK
                                        MOV     #177777,MEML            ;SET UP CORE MAPS
                                        MOV     #77777,MEMH
                                        MOV     #1,BITPT                ;SET UP 4K POINTER
                                        MOV     #MEML,MEMUT
                                        MOV     #5$,@#4                 ;SET UP FOR TIME CUTS
                                2$:     MOV     TBANK,@KPAR1            ;MAP KERNEL PAGE 1 TO BANK BEING TESTED
                                        TST     @#21000                 ;1ST K PRESENT?
                                        TST     @#25000                 ;2ND K PRESENT?
                                        TST     @#31000                 ;3RD K PRESENT
                                        TST     @#35000                 ;4TH K PRESENT?
                                3$:     ADD     #200,TBANK              ;UPDATE TEST ADDRESS
                                        ASL     BITPT                   ;UPDATE BANK POINTER
                                        BCC     4$                      ;BRANCH IF NOT DONE WITH 64K SECTION
```

```
2608  012110  012767  000001  166402          MOV     #1,BITPT                ;YES, DO MEMH(64-124K)
2609  012116  012767  001066  166422          MOV     #MEMH,MEMUT
2610  012124  022767  007600  166412  4$:     CMP     #7600,TBANK             ;EXTERNAL BANK YET?
2611  012132  003345                          BGT     2$                      ;NO, NOT YET
2612  012134  005037  177572                  CLR     @#SR0                   ;YES- DISABLE KT11
2613  012140  012737  000006  000004          MOV     #6,@#4                  ;RESTORE TRAPCATCHER
2614  012146  000207                          RTS     %7                      ;RETURN
2615  012150  046777  166344  166370  5$:     BIC     BITPT,@MEMUT            ;TIMEOUT OCCURRED-CLEAR BIT TO INDICATE
2616                                                                          ;4K BLOCK NOT PRESENT
2617  012156  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK
2618  012160  000745                          BR      3$                      ;CHECK NEXT BLOCK
2619
2620                          ;NO KT PRESENT - MAP MAX OF 28K IN 4K CONTIGUOUS BLOCKS
2621  012162  012767  000001  166360  MAPMB:  MOV     #1,NOKT                 ;SET FLAG TO INDICATE KT11 NOT PRESENT
2622  012170  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
2623  012172  012737  012272  000004          MOV     #3$,@#4                 ;SET UP TIMEOUT RETURN
2624  012200  012767  000177  166656          MOV     #177,MEML               ;INITIALIZE MAP
2625  012206  005067  166654                  CLR     MEMH
2626  012212  012767  000001  166300          MOV     #1,BITPT                ;SETUP 4K POINTER
2627  012220  005001                          CLR     R1                      ;INITIALIZE BANK ADDRESS
2628  012222  005761  001000          1$:     TST     1000(1)                 ;1ST K PRESENT
2629  012226  005761  005000                  TST     5000(1)                 ;2ND K PRESENT
2630  012232  005761  011000                  TST     11000(1)                ;3RD K PRESENT
2631  012236  005761  015000                  TST     15000(1)                ;4TH K PRESENT
2632  012242  062701  020000          2$:     ADD     #20000,R1               ;UPDATE TEST ADDRESS
2633  012246  006367  166246                  ASL     BITPT                   ;UPDATE POINTER TO NEXT 4K
2634  012252  022767  000200  166240          CMP     #200,BITPT              ;28K CHECKED YET?
2635  012260  003360                          BGT     1$                      ;NO, CHECK NEXT 4K BLOCK
2636  012262  012737  000006  000004          MOV     #6,@#4
2637  012270  000207                          RTS     %7
2638  012272  046767  166222  166564  3$:     BIC     BITPT,MEML              ;TIMEOUT OCCURRED- CLEAR BIT TO
2639                                                                          ;INDICATE 4K BLOCK NOT PRESENT
2640  012300  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK POINTER
2641  012302  000757                          BR      2$
2642
2643
2644
2645                          ;*************************************************************************
2646                          ;MAP PARITY CORE AND CORRESPONDENCE TO ASSOCIATED REGISTERS
2647                          ;*************************************************************************
2648  012304  013767  001064  166234  MAPREG: MOV     @#MEML,MEMUT            ;LOAD MAP OF MEMORY PRESENT IN LOWER 64K
2649  012312  012767  000001  166200          MOV     #1,BITPT                ;INITIALIZE 4K POINTER
2650  012320  012767  000001  166700          MOV     #1,KTSTART              ;INDICATE KT11 NOT IN USE
2651  012326  005067  166232                  CLR     RELOC
2652  012332  005067  166206                  CLR     TBANK                   ;INITIALIZE ADDRESS OF TEST BANK
2653  012336  005067  166210                  CLR     HIWORD                  ;CLEAR FLAG TO INDICATE FIRST 64K
2654                                                                          ;BEING CHECKED
2655  012342  005067  166662                  CLR     ADRTYP
2656
2657                          ;SET WRITE WRONG PARITY IN ALL REGISTERS PRESENT
2658                          ;THEN WRITE TEST LOCATION VIA DATO AND READ TEST LOCATION VIA DATI
2659                          ;THEN CLEAR WRITE WRONG PARITY IN ALL REGISTERS
2660  012346  012702  000566  MAPRB:  MOV     #MPR0,R2                ;LOAD ADDRESS OF TABLE
2661  012352  032712  000001  1$:     BIT     #1,(2)                  ;IS THIS REGISTER PRESENT?
2662  012356  001003                          BNE     .+10                    ;NO, GET NEXT ONE
2663  012360  012772  000004  000000          MOV     #WWP,@(2)               ;YES, SET WRITE WRONG PARITY AND CLEAR REST
```

```
2664  012366  062702  000010                  ADD     #10,R2
2665  012372  020227  000766                  CMP     R2,#TREG        ;DONE WITH TABLE?
2666  012376  103765                          BLO     1$              ;BRANCH IF NOT
2667  012400  016703  166140                  MOV     TBANK,R3        ;LOAD ADDRESS OF 4K BANK UNDER TEST
2668  012404  066703  166620                  ADD     ADRTYP,R3       ;ADD ADDRESS OFFSET (EITHER 0,2,4,OR 6)
2669  012410  011313                          MOV     (3),(3)         ;WRITE WRONG PARITY
2670  012412  005713                          TST     (3)             ;READ WRONG PARITY
2671  012414  012702  000566                  MOV     #MPR0,R2
2672  012420  032712  000001          2$:     BIT     #1,(2)
2673  012424  001003                          BNE     .+10
2674  012426  042772  000004  000000          BIC     #WWP,@(2)       ;CLEAR WRITE WRONG PARITY IN ALL
2675                                                                  ;PARITY REGISTERS
2676  012434  062702  000010                  ADD     #10,R2
2677  012440  020227  000766                  CMP     R2,#TREG
2678  012444  103765                          BLO     2$
2679  012446  012702  000556          MAPRC:  MOV     #MPR0-10,R2     ;INIT FOR ERROR CHECKS
2680  012452  062702  000010          1$:     ADD     #10,R2
2681  012456  020227  000766                  CMP     R2,#TREG
2682  012462  002031                          BGE     MAPRD           ;BRANCH IF DONE WITH TABLE
2683  012464  032712  000001                  BIT     #1,(2)          ;IS THIS REGISTER PRESENT?
2684  012470  001370                          BNE     1$              ;NO, GET NEXT ADDRESS
2685  012472  005772  000000                  TST     @(R2)           ;YES, DID THIS CONTROLLER GET A
2686                                                                  ;PARITY ERROR?
2687  012476  100365                          BPL     1$              ;NO, CHECK NEXT
2688  012500  005767  166046                  TST     HIWORD          ;YES, WHICH 64K IS UNDER TEST?
2689  012504  001004                          BNE     2$              ;BRANCH IF UPPER 64K
2690  012506  056762  166006  000002          BIS     BITPT,2(2)      ;SET BIT IN MAP FOR THIS PARITY REGISTER
2691  012514  000403                          BR      3$
2692  012516  056762  165776  000004  2$:     BIS     BITPT,4(2)      ;SET BIT IN MAP FOR THIS PARITY REGISTER
2693  012524  105762  000007          3$:     TSTB    7(2)            ;IS THIS THE FIRST ADDRESS FOUND FOR
2694                                                                  ;THIS PARITY REGISTER?
2695  012530  001005                          BNE     4$              ;NO, BRANCH
2696  012532  116762  166472  000006          MOVB    ADRTYP,6(2)     ;YES, RECORD LOW BYTE OF ADDRESS (0,2,4,OR 6)
2697  012540  105262  000007                  INCB    7(2)            ;INDICATE AN ADDRESS HAS BEEN FOUND FOR
2698                                                                  ;THIS REGISTER
2699  012544  000742                  4$:     BR      1$
2700  012546  011313                  MAPRD:  MOV     @R3,@R3         ;CLEAR BAD PARITY
2701  012550  062767  000002  166452          ADD     #2,ADRTYP       ;CHECK FIRST 4 ADDRESSES IN EACH 4K
2702  012556  026727  166446  000010          CMP     ADRTYP,#10
2703  012564  001402                          BEQ     1$              ;BRANCH IF FIRST 4 ADDRESSES TESTED
2704  012566  000167  177554                  JMP     MAPRB           ;IF NOT, GO TEST NEXT ONE
2705  012572  005767  165754          1$:     TST     HIWORD          ;IS LOWER MEMORY DONE?
2706  012576  001021                          BNE     MAPRE           ;YES, BRANCH
2707  012600  026727  165714  000100          CMP     BITPT,#100      ;DONE WITH 1ST 28K?
2708  012606  103015                          BHIS    MAPRE           ;YES, BRANCH
2709  012610  062767  020000  165726          ADD     #20000,TBANK    ;NO- ADD 4K TO ADDRESS
2710  012616  006367  165676                  ASL     BITPT           ;SHIFT BIT POINTER
2711  012622  005067  166402                  CLR     ADRTYP          ;START WITH FIRST ADDRESS IN BANK
2712  012626  036767  165666  165712          BIT     BITPT,MEMUT     ;DOES THIS 4K BLOCK EXIST?
2713  012634  001756                          BEQ     1$              ;NO- BRANCH
2714  012636  000167  177504                  JMP     MAPRB           ;YES, TEST IT
2715  012642  005767  165702          MAPRE:  TST     NOKT            ;KT11 PRESENT?
2716  012646  001401                          BEQ     .+4             ;YES, BRANCH
2717  012650  000207                          RTS     %7              ;NO, DONE
2718  012652  005767  166350                  TST     KTSTART         ;KT11 ALREADY ON?
2719  012656  001417                          BEQ     1$              ;YES, BRANCH
```

```
2720  012660  012767  020000  165656          MOV     #20000,TBANK            ;NO, INIT TBANK TO SELECT KERNEL PAGE 1
2721  012666  012767  001400  165670          MOV     #1400,RELOC             ;SET UP FOR ACCESS TO 28K BANK
2722  012674  004767  001012                  JSR     %7,NRALL                ;INITIALLY MAP ALL PAGES NR, BANK 0
2723  012700  004767  001156                  JSR     %7,MAP1                 ;MAP KERNEL 0 TO BANK 0, RW
2724                                                                          ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
2725                                                                          ;SET KERNEL 1 RW AND TURN ON KT11
2726  012704  005067  166316                  CLR     KTSTART                 ;INDICATE KT11 NOW IN USE
2727  012710  012737  000001  177572          MOV     #1,@#SR0                ;TURN ON KT11
2728  012716  006367  165576          1$:     ASL     BITPT                   ;SHIFT BANK INDICATOR
2729  012722  103011                          BCC     2$                      ;BRANCH IF FIRST 64K NOT DONE
2730  012724  012767  000001  165566          MOV     #1,BITPT                ;IF FIRST 64K DONE, SETUP FOR
2731  012732  013767  001066  165606          MOV     @#MEMH, MEMUT           ;SECOND 64K
2732  012740  012767  000001  165604          MOV     #1,HIWORD               ;INDICATE NOW TESTING HIGH 64K
2733  012746  062767  000200  165610  2$:     ADD     #200,RELOC
2734  012754  022767  007600  165602          CMP     #7600,RELOC             ;UP TO EXTERNAL BANK YET?
2735  012762  003003                          BGT     3$                      ;NO, CONTINUE
2736  012764  005037  177572                  CLR     @#SR0                   ;YES, TURN OFF KT11 AND EXIT
2737  012770  000207                          RTS     %7
2738  012772  036767  165522  165546  3$:     BIT     BITPT,MEMUT             ;IS THIS 4K PRESENT?
2739  013000  001746                          BEQ     1$                      ;NO- BRANCH
2740  013002  016777  165556  166242          MOV     RELOC,@KPAR1            ;YES, MAP PAGE 1 TO THIS BANK
2741  013010  005067  166214                  CLR     ADRTYP
2742  013014  000167  177326                  JMP     MAPRB                   ;GO TEST FOR PARITY MEMORY
2743
2744
2745                                  ;*************************************************************************
2746                                  ;ROUTINE TO TYPE MAP OF WHERE PARITY MEMORY IS PRESENT
2747                                  ;AND WHICH CONTROL REGISTERS CONTROL WHICH MEMORY
2748                                  ;*************************************************************************
2749  013020  004767  000766  TMAP:   JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS PRESENT
2750  013024  104000                  TYPE                            ;TYPE "THE PARITY REGISTERS CONTROL MEMORY
2751  013026  017074                  MTMAP                           ;AS FOLLOWS:"
2752  013030  012701  000556          MOV     #MPR0-10,R1             ;SET UP POINTER
2753  013034  062701  000010  TMAPA:  ADD     #10,R1
2754  013040  020127  000766          CMP     R1,#TREG                ;DONE WITH MAP?
2755  013044  002136                  BGE     TMAPEX                  ;YES, BRANCH
2756  013046  005067  165500          CLR     HIWORD
2757  013052  005067  165444          CLR     TRFLG                   ;INITIALIZE TRANSITION FLAG (USED TO
2758                                                                  ;FIGURE MEMORY LIMITS)
2759  013056  005067  165442          CLR     TYFLG                   ;INITIALIZE TO INDICATE NOTHING TYPED FOR
2760                                                                  ;THIS REGISTER YET
2761
2762  013062  012767  177774  165436  MOV     #-4,TYCOR
2763  013070  016167  000002  165420  MOV     2(1),ADRPT              ;GET LOW 64K MEMORY MAP WORD
2764  013076  012767  000001  165414  MOV     #1,BITPT                ;INITIALIZE 4K POINTER
2765  013104  032711  000001          BIT     #1,(1)                  ;DOES THIS CONTROL EXIST?
2766  013110  001351                  BNE     TMAPA                   ;NO, GET ADDRESS OF NEXT ONE
2767  013112  011167  165376          MOV     (1),TEMPX               ;YES, PRINT ITS ADDRESS
2768  013116  004567  002274          JSR     R5,0ACNV
2769  013122  000514                  TEMPX
2770  013124  017244                  MPRAD
2771  013126  000006                  6
2772  013130  104000                  TYPE
2773  013132  017447                  MX1
2774  013134  104000                  TYPE
2775  013136  017244                  MPRAD
```

```
2776  013140  062767  000004  165360  TMAPB:  ADD     #4,TYCOR                ;KEEP TRACK OF # OF K OF CORE
2777  013146  036767  165346  165342          BIT     BITPT,ADRPT             ;DOES THIS PARITY REGISTER CONTROL THIS 4K?
2778  013154  001424                          BEQ     TMAPC                   ;NO- BRANCH
2779  013156  005767  165340                  TST     TRFLG                   ;YES, DOES IT CONTROL PREVIOUS 4K?
2780  013162  001043                          BNE     TMAPD                   ;YES- DON'T TYPE IT
2781  013164  012767  000001  165330          MOV     #1,TRFLG                ;NO- SET FLAG INDICATING TRANSITION
2782  013172  004567  002326                  JSR     R5,BDCNV                ;CONVERT K CORE TO ASCII
2783  013176  000526                          TYCOR
2784  013200  017144                          MTYCOR
2785  013202  000003                          3
2786  013204  104000                          TYPE                            ;TYPE "CONTROLS", AND ADDRESS OF CORE
2787  013206  017546                          MX2
2788  013210  104000                          TYPE
2789  013212  017144                          MTYCOR
2790  013214  104000                          TYPE
2791  013216  017151                          MDASH
2792  013220  005267  165300                  INC     TYFLG                   ;INDICATE TYPED
2793  013224  000422                          BR      TMAPD
2794  013226  005767  165270          TMAPC:  TST     TRFLG                   ;DID THIS PARITY REGISTER CONTROL PREVIOUS 4K?
2795  013232  001417                          BEQ     TMAPD                   ;NO, SKIP PRINTING
2796  013234  005067  165262                  CLR     TRFLG                   ;YES, TRANSITION OCCURRED- CLEAR FLAG
2797  013240  004567  002260                  JSR     R5,BDCNV                ;CONVERT K CORE TO ASCII
2798  013244  000526                          TYCOR
2799  013246  017144                          MTYCOR
2800  013250  000003                          3
2801  013252  104000                          TYPE
2802  013254  017144                          MTYCOR                          ;TYPE RIGHT AND RETURN
2803  013256  104000                          TYPE
2804  013260  017157                          MK
2805  013262  104000                          TYPE
2806  013264  017154                          MCR
2807  013266  005267  165232                  INC     TYFLG                   ;INDICATE TYPED
2808  013272  006367  165222          TMAPD:  ASL     BITPT                   ;UPDATE BIT POINTER TO NEXT 4K
2809  013276  103320                          BCC     TMAPB                   ;TEST NEXT 4K IF NOT DONE WITH 1ST 64K
2810  013300  005767  165246                  TST     HIWORD                  ;64-124K DONE?
2811  013304  001405                          BEQ     1$                      ;NO, BRANCH
2812  013306  005767  165212                  TST     TYFLG                   ;YES, WAS ANY PARITY MEMORY
2813                                                                          ;FOUND FOR THIS REGISTER?
2814
2815  013312  001250                          BNE     TMAPA
2816  013314  104002                          ERROR                           ;NO PARITY MEMORY WAS FOUND FOR THIS
2817                                                                          ;REGISTER- EITHER WRITE WRONG PARITY
2818                                                                          ;FAILED, PARITY ERROR GENERATE OR
2819                                                                          ;DETECT FAILED, OR THE PARITY ERROR
2820                                                                          ;BIT FAILED TO SET
2821  013316  000646                          BR      TMAPA
2822  013320  016167  000004  165170  1$:     MOV     4(1),ADRPT              ;UPDATE TO MAP WORD FOR THE UPPER 64K
2823  013326  012767  000001  165164          MOV     #1,BITPT                ;RESET BIT POINTER
2824  013334  005267  165212                  INC     HIWORD                  ;INDICATE LOW 64K DONE
2825  013340  000677                          BR      TMAPB                   ;LOOP
2826  013342  000207          TMAPEX: RTS     %7                              ;RETURN WHEN DONE
2827
2828                           ;ROUTINE TO HANDLE 8K SYSTEMS
2829                           ;
2830
2831  013344  012767  000002  165146  SMLSYS: MOV     #2,BITPT                ;SET UP POINTER FOR BANK 1
```

```
2832  013352  005037  000042                  CLR     @#42                    ;MAKE EXIT STANDALONE
2833  013356  104000                          TYPE
2834  013360  017466                          NOMON
2835  013362  000207                          RTS     PC
2836
2837
2838                                  ;*************************************************************************
2839                                  ;ERROR HANDLER
2840                                  ;*************************************************************************
2841                                  ;ERRORS CALL ENTERS HERE
2842                                  ;TYPES PC, ICNT, MPR ADDRESS, AND MPR CONTENTS
2843                                  ;TREG SHOULD CONTAIN ADDRESS OF PARITY REGISTER
2844  013364  012767  016517  000266  ERRST:  MOV     #MSTR,ERRB              ;SETUP TO TYPE MPR ADDRESS AND CONTENTS
2845  013372  012767  177777  000262          MOV     #-1,ERRBX
2846  013400  012767  000240  000256          MOV     #240,ERRBX+2            ;NOP LOCATION AFTER MESSAGE
2847  013406  017767  165354  165124          MOV     @TREG,TRDATA            ;SETUP DATA
2848  013414  004567  001776                  JSR     R5,OACNV                ;CONVERT TO ASCII
2849  013420  000766                          TREG
2850  013422  016526                          MTREG
2851  013424  000006                          6
2852  013426  004567  001764                  JSR     R5,OACNV                ;CONVERT TO ASCII
2853  013432  000540                          TRDATA
2854  013434  016550                          MDATA
2855  013436  000006                          6
2856  013440  000461                          BR      ERRA
2857
2858
2859                                  ;ERRORP CALL ENTERS HERE
2860                                  ;TYPES PC, ICNT, MPR ADDRESS, MPR CONTENTS
2861                                  ;TEST LOCATION ADDRESS, VALUE EXPECTED, VALUE FOUND
2862                                  ;R5 MUST CONTAIN ADDRESS OF TEST LOCATION
2863                                  ;SHDBE MUST CONTAIN EXPECTED VALUE
2864                                  ;WAS MUST CONTAIN ACTUAL DATA
2865                                  ;TREG MUST CONTAIN ADDRESS OF PARITY REGISTER
2866  013442  012767  016517  000210  ERRP:  MOV     #MSTR,ERRB              ;IN ADDITION TO BASIC PRINTOUT, TYPE
2867                                                                          ;MPR ADDRESS AND CONTENTS
2868  013450  012767  016561  000204          MOV     #MSTRX,ERRBX            ;ALSO OUTPUT DATA EXPECTED AND ACTUAL
2869  013456  012767  177777  000200          MOV     #-1,ERRBX+2             ;NOP LOCATION AFTER MESSAGE
2870  013464  010567  165042                  MOV     R5,TSTLOC               ;STORE ADDRESS BEING TESTED
2871  013470  017767  165272  165042          MOV     @TREG,TRDATA
2872  013476  004567  001714                  JSR     R5,OACNV
2873  013502  000766                          TREG
2874  013504  016526                          MTREG
2875  013506  000006                          6
2876  013510  004567  001702                  JSR     R5,OACNV
2877  013514  000540                          TRDATA
2878  013516  016550                          MDATA
2879  013520  000006                          6
2880  013522  004567  001670                  JSR     R5,OACNV
2881  013526  000532                          TSTLOC
2882  013530  016605                          MSTRX1
2883  013532  000006                          6
2884  013534  004567  001656                  JSR     R5,OACNV
2885  013540  000534                          SHDBE
2886  013542  016621                          MSTRX3
2887  013544  000006                          6
```

```
                                        JSR     R5,OACNV
                                        WAS
                                        MSTRX5
                                        6
                                        BR      ERRA


                        ;ERROR CALL ENTERS HERE
                        ;TYPE PC AND ICNT ONLY
                ERR:    MOV     #-1,ERRB                ;SET UP ONE MESSAGE CALL
                        MOV     #240,ERRBX
                        MOV     #240,ERRBX+2
                ERRA:   BIT     #BIT13,@SWR             ;INHIBIT ERROR PRINT?
                        BNE     ERRC                    ;YES- BRANCH
                        MOV     (SP),ERRD               ;NO- DEVELOP CALLING ADDRESS
                        SUB     #2,ERRD
                        JSR     R5,OACNV                ;GO TO OCTAL TO ASCII CONVERT
                ERRA1:  ERRD                            ;SOURCE ADDRESS
                ERRA2:  MPC                             ;DESTINATION ADDRESS
                        6                               ;#OF DIGITS TO CONVERT
                        JSR     R5,OACNV                ;CONVERT ICNT TO ASCII
                ERRA3:  ICNT
                ERRA4:  MICNT
                        6
                        JSR     R5,TYPSX                ;TYPE MESSAGE
                ERRA5:  MEO                             ;ERROR HEADER
                ERRB:   OPEN                            ;ADDITIONAL ERROR MESSAGES IF ANY
                ERRBX:  OPEN
                        -1
                ERRC:   CMP     @#42,@#46               ;ARE WE IN ACT11 AUTOMATIC MODE?
                        BEQ     .+10                    ;YES, HALT ON ERROR
                        TST     @SWR                    ;HALT ON ERROR SET?
                        BPL     .+4                     ;NO- BRANCH
                        HALT                            ;YES- ERROR OCCURRED SO HALT
                        RTI
                ERRD:   OPEN


                        ;MAP ALL PAGES NON-RESIDENT, BANK 0
                NRALL:  MOV     @#4,-(SP)
                        MOV     @#6,-(SP)
                        MOV     #6,@#4
                        MOV     #RTI,@#6
                        MOV     R1,-(SP)
                        MOV     R2,-(SP)
                        MOV     R3,-(SP)
                        MOV     #PDRTAB,R1
                1$:     MOV     #32.,R3
                        MOV     (R1)+,R2
                2$:     CLR     (R2)+
                        DEC     R3
                        BNE     2$
                        CMP     R1,#PDREND
                        BLE     1$
```

```
2944  013772  012603                        MOV     (SP)+,R3
2945  013774  012602                        MOV     (SP)+,R2
2946  013776  012601                        MOV     (SP)+,R1
2947  014000  012637  000006                MOV     (SP)+,@#6
2948  014004  012637  000004                MOV     (SP)+,@#4
2949  014010  000207                        RTS     %7
2950
2951
2952
2953                                ;ROUTINE TO CLEAR ALL PARITY REGISTERS PRESENT
2954  014012  010146                CLRPAR: MOV     R1,-(SP)
2955  014014  010246                        MOV     R2,-(SP)
2956  014016  010701                        MOV     PC,R1
2957  014020  062701  164546                ADD     #MPR0-.,R1
2958  014024  010702                        MOV     PC,R2
2959  014026  062702  164740                ADD     #TREG-.,R2
2960  014032  032711  000001        1$:     BIT     #1,@R1                  ;IS THIS REGISTER PRESENT?
2961  014036  001002                        BNE     .+6
2962  014040  005071  000000                CLR     @(R1)                   ;CLEAR ALL PARITY REGISTERS
2963  014044  062701  000010                ADD     #10,R1
2964  014050  020102                        CMP     R1,R2
2965  014052  103767                        BLO     1$
2966  014054  012602                        MOV     (SP)+,R2
2967  014056  012601                        MOV     (SP)+,R1
2968  014060  000207                        RTS     %7
2969
2970
2971
2972                                ;ROUTINE TO MAP KERNEL 0 TO BANK 0 READ/WRITE.
2973                                ;KERNEL 1 READ/WRITE BUT BANK MAPPED BY CALLING ROUTINE,
2974                                ;AND KERNEL 7 TO EXTERNAL BANK, READ/WRITE
2975  014062  012777  077406  165150 MAP1:   MOV     #77406,@KPDR0
2976  014070  012777  077406  165144        MOV     #77406,@KPDR1
2977  014076  012777  077406  165142        MOV     #77406,@KPDR7
2978  014104  012777  007600  165144        MOV     #7600,@KPAR7
2979  014112  005077  165132                CLR     @KPAR0
2980  014116  012737  000001  177572        MOV     #1,@#SR0
2981  014124  000207                        RTS     %7
2982
2983
2984
2985                                ;ROUTINE TO LOCATE THE FIRST PARITY MEMORY ADDRESS (ABOVE BANK 0)
2986                                ;CORRESPONDING TO A GIVEN PARITY REGISTER-REQUIRES THAT THE ROUTINES
2987                                ;MAPMEM AND MAPREG HAVE ALREADY BEEN RUN
2988                                ;TO USE, PUT THE ADDRESS OF THE ADDRESS OF THE REGISTER IN R0 (I.E. POINT TO
2989                                ;TO MAP'TABLE)-THE DESIRED ADDRESS IS RETURNED IN R1, USING KERNEL PAGE 1 IF
2990                                ;KT11 IS PRESENT
2991  014126  010246                LOCATM: MOV     R2,-(SP)
2992  014130  012702  000200                MOV     #200,R2
2993  014134  012701  000002                MOV     #2,R1                   ;SKIP USE OF BANK 0
2994  014140  005767  164404        1$:     TST     NOKT                    ;KT11 PRESENT?
2995  014144  001403                        BEQ     3$                      ;YES, BRANCH
2996  014146  022701  000200                CMP     #200,R1                 ;NO, CHECK ONLY FOR MEMORY IN FIRST 28K
2997  014152  101420                        BLOS    4$                      ;IF NO MEMORY FOUND IN 1ST 28K, GIVE
2998                                                                        ;ERROR RETURN
2999  014154  030160  000002        3$:     BIT     R1,2(R0)                ;DOES THIS 4K CORRESPOND TO THIS REGISTER?
```

```
3000  014160  001021                        BNE     LOCAT1                  ;YES, BRANCH
3001  014162  062702  000200                ADD     #200,R2                 ;NO, CHECK TO SEE IF NEXT 4K CORRESPONDS
3002  014166  006301                        ASL     R1
3003  014170  103363                        BCC     1$
3004  014172  012701  000001                MOV     #1,R1
3005  014176  030160  000004        2$:     BIT     R1,4(R0)                ;CHECK HIGH 64K
3006  014202  001010                        BNE     LOCAT1
3007  014204  062702  000200                ADD     #200,R2
3008  014210  006301                        ASL     R1
3009  014212  103371                        BCC     2$
3010  014214  012701  000001        4$:     MOV     #1,R1                   ;NO PARITY MEMORY CORRESPONDS TO
3011  014220  012602                        MOV     (SP)+,R2                ;THIS REGISTER- RETURN WITH ERROR
3012  014222  000207                        RTS     %7                      ;INDICATION
3013  014224  005767  164320        LOCAT1: TST     NOKT                    ;KT11 PRESENT?
3014  014230  001005                        BNE     1$                      ;NO- BRANCH OVER
3015  014232  010277  165014                MOV     R2,@KPAR1               ;YES- SETUP R1 TO REFERENCE 4K
3016  014236  012701  020000                MOV     #20000,R1               ;BANK USING KERNEL PAGE 1
3017  014242  000407                        BR      2$
3018  014244  010201                1$:     MOV     R2,R1                   ;SETUP R1 TO REFERENCE 4K BANK
3019  014246  006301                        ASL     R1                      ;WITHOUT KT11
3020  014250  006301                        ASL     R1
3021  014252  006301                        ASL     R1
3022  014254  006301                        ASL     R1
3023  014256  006301                        ASL     R1
3024  014260  006301                        ASL     R1
3025  014262  116067  000006  000010  2$:     MOVB    6(R0),LSAV
3026  014270  066701  000004                ADD     LSAV,R1
3027  014274  012602                        MOV     (SP)+,R2                ;RESTORE R2
3028  014276  000207                        RTS     %7                      ;AND RETURN
3029  014300  000000                LSAV:   0
3030
3031
3032
3033                                  ;SAVE LOADER IN TOP OF FIRST 4K
3034  014302  013746  000004        SAVLDR: MOV     @#4,-(SP)               ;SAVE CONTENTS OF TIMEOUT VECTOR
3035  014306  013746  000006                MOV     @#6,-(SP)
3036  014312  012737  000006  000004        MOV     #6,@#4
3037  014320  012737  000002  000006        MOV     #RTI,@#6                ;SETUP TO RTI ON TIMEOUT
3038  014326  010146                        MOV     R1,-(SP)                ;SAVE REGISTERS
3039  014330  010246                        MOV     R2,-(SP)
3040  014332  012701  152234                MOV     #152234,R1
3041  014336  000261                1$:     SEC
3042  014340  005711                        TST     @R1                     ;IF TIMEOUT, C BIT WILL STILL BE SET
3043  014342  103006                        BCC     2$                      ;IF NO TIMEOUT, C BIT WILL BE CLEAR
3044  014344  162701  020000                SUB     #20000,R1               ;TIMEOUT OCCURRED, CHECK FOR NEXT LOWER BANK
3045  014350  020127  020000                CMP     R1,#20000
3046  014354  101370                        BHI     1$
3047  014356  000410                        BR      SAVLDX                  ;ONLY 4K OF 1ST 28K PRESENT- EXIT
3048  014360  012702  032234        2$:     MOV     #32234,R2               ;THIS BANK IS HIGHEST OF 1ST 28K- COPY
3049  014364  012122                3$:     MOV     (R1)+,(R2)+             ;LOADER AREA TO BANK 0
3050  014366  020227  040000                CMP     R2,#40000
3051  014372  103774                        BLO     3$
3052  014374  005267  000016                INC     LDRSVD                  ;INDICATE LOADER HAS BEEN MOVED TO
3053                                                                        ;THE TOP OF THE FIRST 4K
3054  014400  012602                SAVLDX: MOV     (SP)+,R2
3055  014402  012601                        MOV     (SP)+,R1
```

```
3056  014404  012637  000006                  MOV     (SP)+,@#6
3057  014410  012637  000004                  MOV     (SP)+,@#4
3058  014414  000207                          RTS     %7
3059  014416  000000                LDRSVD:   0
3060
3061
3062
3063                                ;ROUTINE TO RESTORE THE LOADER FROM BANK 0 (WHERE IT WAS SAVED) TO THE
3064                                ;HIGHEST BANK IN THE FIRST 28K OF MEMORY
3065  014420  005767  177772        RSTLDR:   TST     LDRSVD
3066  014424  001431                          BEQ     RSTLDX
3067  014426  012737  000006  000004        MOV     #6,@#4
3068  014434  012737  000002  000006        MOV     #RTI,@#6
3069  014442  012701  152234                MOV     #152234,R1
3070  014446  000261                1$:       SEC
3071  014450  005711                          TST     @R1                     ;IF TIMEOUT, C BIT WILL BE SET
3072  014452  103006                          BCC     2$                      ;IF NO TIMEOUT, C BIT WILL BE CLEAR
3073  014454  162701  020000                  SUB     #20000,R1               ;TIMEOUT OCCURRED, CHECK FOR NEXT
3074  014460  020127  020000                  CMP     R1,#20000               ;LOWER BANK
3075  014464  101370                          BHI     1$
3076  014466  000410                          BR      RSTLDX
3077  014470  012702  032234        2$:       MOV     #32234,R2
3078  014474  012221                3$:       MOV     (R2)+,(R1)+
3079  014476  020227  040000                  CMP     R2,#40000
3080  014502  103774                          BLO     3$
3081  014504  104000                          TYPE                            ;TYPE MESSAGE "LOADER RESTORED"
3082  014506  017560                          LDRMSG
3083  014510  000207                RSTLDX:   RTS     PC                      ;LOADER HAS BEEN RESTORED
3084  014512  000776                          BR      .-2                     ;TO HIGHEST BANK IN FIRST 28K
3085
3086
3087
3088                                ;SCAN ALL MEMORY FOR BAD PARITY, TYPE 18 BIT ADDRESSES OF
3089                                ;LOCATIONS FOUND TO BE BAD, AND WRITE INTO LOCATIONS WITH GOOD PARITY
3090  014514  010146                PSCAN:    MOV     R1,-(SP)                ;STORE REGISTERS AND LOCATIONS TO BE
3091  014516  010246                          MOV     R2,-(SP)                ;ALTERRED
3092  014520  010346                          MOV     R3,-(SP)
3093  014522  010446                          MOV     R4,-(SP)
3094  014524  013746  000004                  MOV     @#4,-(SP)
3095  014530  013746  000006                  MOV     @#6,-(SP)
3096  014534  013746  000114                  MOV     @#114,-(SP)
3097  014540  013746  000116                  MOV     @#116,-(SP)
3098  014544  012737  000006  000004          MOV     #6,@#4                  ;SETUP TIMEOUT TRAPCATCHER
3099  014552  005037  000006                  CLR     @#6
3100  014556  012737  000116  000114          MOV     #116,@#114              ;SETUP PARITY TRAP TRAPCATCHER
3101  014564  005037  000116                  CLR     @#116
3102  014570  005767  163754                  TST     NOKT                    ;KT11 PRESENT?
3103  014574  001513                          BEQ     PSCAN1                  ;YES, BRANCH
3104  014576  005002                          CLR     R2                      ;R2 CONTAINS TEST ADDRESS
3105  014600  012703  000001                  MOV     #1,R3                   ;R3 USED AS A BIT POINTER
3106  014604  004767  177202        1$:       JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
3107  014610  005712                          TST     @R2                     ;READ LOCATION TO CHECK FOR BAD PARITY
3108  014612  000240                          NOP
3109  014614  012701  000566                  MOV     #MPR0,R1                ;SETUP TO SCAN REGISTERS FOR PARITY
3110                                                                          ;ERROR SET
3111  014620  032711  000001        3$:       BIT     #1,@R1
```

```
3112  014624  001003                        BNE     .+10
3113  014626  005771  000000                TST     @(R1)                   ;PARITY ERROR SET?
3114  014632  100424                        BMI     6$                      ;YES- BRANCH
3115  014634  062701  000010                ADD     #10,R1                  ;NO- CHECK NEXT REGISTER
3116  014640  020127  000766                CMP     R1,#TREG
3117  014644  103765                        BLO     3$                      ;LOOP UNTIL ALL REGISTERS HAVE BEEN CHECKED
3118  014646  062702  000002        4$:     ADD     #2,R2                   ;MOVE ADDRESS POINTER
3119  014652  032702  017777                BIT     #17777,R2               ;DONE WITH 4K?
3120  014656  001352                        BNE     1$                      ;NO, CONTINUE
3121  014660  006303                5$:     ASL     R3                      ;YES, CHECK FOR TESTING NEXT 4K
3122  014662  020327  000200                CMP     R3,#200
3123  014666  103035                        BHIS    PSCANX                  ;EXIT IF DONE WITH 28K
3124  014670  030367  164170                BIT     R3,MEML                 ;IS THIS MEMORY PRESENT?
3125  014674  001343                        BNE     1$                      ;YES, GO TEST IT
3126  014676  062702  020000                ADD     #20000,R2               ;NO, UPDATE ADDRESS
3127  014702  000766                        BR      5$                      ;LOOP
3128  014704  010267  000110        6$:     MOV     R2,PSADRS               ;PARITY ERROR OCCURRED
3129  014710  004567  000502                JSR     R5,OACNV                ;GET ASCII OF ADDRESS CONTAINING BAD PARITY
3130  014714  015020                        PSADRS
3131  014716  017440                        MPSER1
3132  014720  000006                        6
3133  014722  104000                        TYPE                            ;TYPE MESSAGE "BAD PARITY FOUND IN LOCATION"
3134  014724  017410                        MPSER                           ;AND ADDRESS OF FAILING LOCATION
3135  014726  032777  002000  164144        BIT     #2000,@SWR              ;SWITCH 10 SET?
3136  014734  001401                        BEQ     .+4                     ;NO- CONTINUE
3137  014736  000000                        HALT                            ;HALT ON BAD PARITY SET
3138  014740  011212                        MOV     @R2,@R2                 ;WRITE INTO LOCATION- SHOULD
3139                                                                        ;CLEAR BAD PARITY
3140  014742  005071  000000                CLR     @(R1)                   ;CLEAR CORRESPONDING PARITY REGISTER
3141  014746  005712                        TST     @R2                     ;READ LOCATION TO SEE IF BAD PARITY WAS
3142  014750  005771  000000                TST     @(R1)                   ;CLEARED
3143  014754  100001                        BPL     .+4                     ;OK- BRANCH
3144  014756  104002                        ERROR                           ;BAD PARITY DIDN'T CLEAR WHEN LOCATION
3145                                                                        ;WAS REWRITTEN
3146  014760  000732                        BR      4$                      ;GO CHECK NEXT LOCATION
3147  014762  004767  177064        PSCANX: JSR     PC,CLRPAR               ;DONE- CLEAR ALL PARITY REGISTERS
3148  014766  012637  000116                MOV     (SP)+,@#116             ;RESTORE LOCATIONS ALTERRED
3149  014772  012637  000114                MOV     (SP)+,@#114
3150  014776  012637  000006                MOV     (SP)+,@#6
3151  015002  012637  000004                MOV     (SP)+,@#4
3152  015006  012604                        MOV     (SP)+,R4
3153  015010  012603                        MOV     (SP)+,R3
3154  015012  012602                        MOV     (SP)+,R2
3155  015014  012601                        MOV     (SP)+,R1
3156  015016  000207                        RTS     %7                      ;RETURN
3157  015020  000000        PSADRS: 0
3158  015022  000000        PSCANH: 0
3159
3160
3161
3162                                        ;SCAN ALL MEMORY FOR BAD PARITY USING KT11
3163                                        ;TYPE 18 BIT ADDRESSES OF LOCATIONS FOUND BAD, AND WRITE GOOD PARITY BACK IN
3164  015024  017746  164222        PSCAN1: MOV     @KPAR1,-(SP)            ;SAVE CONTENTS OF KERNEL PAR1
3165  015030  032737  000001  177572        BIT     #1,@#SR0                ;SKIP IF KT11 IS ALREADY ON
3166  015036  001004                        BNE     1$
3167  015040  004767  176646                JSR     PC,NRALL                ;MAP KERNEL 0 TO BANK 0,RW
```

```
3168  015044  004767  177012                JSR     PC,MAP1          ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
3169                                                                   ;MAP KERNEL 1 RW, AND TURN ON KT11
3170  015050  005067  177746          1$:   CLR     PSCANH           ;CLEAR FLAG TO INDICATE CHECKING FIRST 64K
3171  015054  005077  164172                CLR     @KPAR1           ;INITIALIZE TO BANK 0
3172  015060  012703  000001          PSLOOP: MOV     #1,R3            ;R3 IS USED AS A BIT POINTER
3173  015064  005767  177732          PSLUP:  TST     PSCANH           ;TESTING TOP 64K?
3174  015070  001004                        BNE     2$               ;YES, BRANCH
3175  015072  030367  163766                BIT     R3,MEML          ;NO, IS PARITY MEMORY PRESENT IN THIS 4K?
3176  015076  001022                        BNE     PSXTST           ;YES- GO TEST IT
3177  015100  000403                        BR      PSNXT            ;NO- CHECK FOR NEXT 4K
3178  015102  030367  163760          2$:   BIT     R3,MEMH          ;IS PARITY MEMORY PRESENT IN THIS 4K?
3179  015106  001016                        BNE     PSXTST           ;YES- GO TEST IT
3180  015110  062777  000200  164134  PSNXT:  ADD     #200,@KPAR1      ;NO- MAP TO NEXT 4K
3181  015116  006303                        ASL     R3
3182  015120  103361                        BCC     PSLUP            ;BRANCH IF NOT END OF 64K
3183  015122  005767  177674                TST     PSCANH           ;END OF TOP 64K?
3184  015126  001003                        BNE     PSCX1            ;YES, GET READY TO EXIT
3185  015130  005267  177666                INC     PSCANH           ;NO, SET FLAG INDICATING DONE WITH
3186                                                                   ;LOWER 64K
3187  015134  000751                        BR      PSLOOP
3188  015136  012677  164110          PSCX1:  MOV     (SP)+,@KPAR1
3189  015142  000707                        BR      PSCANX
3190  015144  012702  020000          PSXTST: MOV     #20000,R2        ;R2 USED AS ADDRESS POINTER
3191  015150  004767  176636          1$:   JSR     %7,CLRPAR        ;CLEAR ALL PARITY REGISTERS
3192  015154  005712                        TST     @R2              ;READ LOCATION
3193  015156  012701  000566                MOV     #MPRO,R1         ;SETUP TO SCAN REGISTERS FOR PARITY ERROR SET
3194  015162  032711  000001          2$:   BIT     #1,@R1
3195  015166  001003                        BNE     .+10
3196  015170  005771  000000                TST     @(R1)            ;PARITY ERROR SET?
3197  015174  100413                        BMI     4$               ;YES, BRANCH
3198  015176  062701  000010                ADD     #10,R1           ;NO, CHECK NEXT
3199  015202  020127  000766                CMP     R1,#TREG
3200  015206  103765                        BLO     2$               ;LOOP UNTIL ALL REGISTERS HAVE BEEN CHECKED
3201  015210  062702  000002          3$:   ADD     #2,R2            ;UPDATE TEST ADDRESS POINTER
3202  015214  020227  040000                CMP     R2,#40000        ;DONE WITH BANK?
3203  015220  103753                        BLO     1$               ;NO- LOOP
3204  015222  000732                        BR      PSNXT            ;YES- GO CHECK FOR ANOTHER BANK
3205  015224  010267  177570          4$:   MOV     R2,PSADRS        ;PARITY ERROR OCCURRED- GET 18 BIT
3206  015230  042767  160000  177562        BIC     #160000,PSADRS   ;OCTAL ADDRESS OF BAD LOCATION
3207  015236  005046                        CLR     -(SP)
3208  015240  017746  164006                MOV     @KPAR1,-(SP)
3209  015244  006316                        ASL     @SP
3210  015246  006316                        ASL     @SP
3211  015250  006316                        ASL     @SP
3212  015252  006316                        ASL     @SP
3213  015254  006316                        ASL     @SP
3214  015256  006166  000002                ROL     2(SP)
3215  015262  006316                        ASL     @SP
3216  015264  006166  000002                ROL     2(SP)
3217  015270  006366  000002                ASL     2(SP)
3218  015274  062667  177520                ADD     (SP)+,PSADRS
3219  015300  004567  000112                JSR     R5,OACNV         ;CONVERT LOW 16 OCTAL BITS TO ASCII
3220  015304  015020                        PSADRS
3221  015306  017440                        MPSER1
3222  015310  000006                        6
3223  015312  116704  002122                MOVB    MPSER1,R4
```

```
3224  015316  062604                        ADD     (SP)+,R4            ;CHANGE TO ASCII FOR 18 BITS
3225  015320  110467  002114                 MOVB    R4,MPSER1
3226  015324  104000                        TYPE                        ;TYPE ADDRESS OF LOCATION WITH BAD PARITY
3227  015326  017410                        MPSER
3228  015330  032777  002000  163542        BIT     #2000,@SWR          ;SWITCH 10 SET?
3229  015336  001401                        BEQ     .+4                 ;NO- BRANCH
3230  015340  000000                        HALT                        ;HALT ON BAD PARITY SET
3231  015342  011212                        MOV     @R2,@R2             ;REWRITE LOCATION CONTAINING BAD PARITY
3232  015344  005071  000000                CLR     @(R1)               ;CLEAR PARITY ERROR BIT
3233  015350  005712                        TST     @R2                 ;READ LOCATION TO SEE IF PARITY IS NOW GOOD
3234  015352  005771  000000                TST     @(R1)               ;CHECK PARITY ERROR BIT
3235  015356  100001                        BPL     .+4
3236  015360  104002                        ERROR                       ;REWRITING LOCATION DID NOT CLEAR BAD PARITY
3237  015362  000712                        BR      3$                  ;GO TEST NEXT LOCATION
3238
3239
3240
3241
3242                                  ;PIC ROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES (CALLED VIA JSR R5)
3243  015364  012567  000022        TYPSX:  MOV     (R5)+,TYPSBX        ;GET ADDRESS OF MESSAGE
3244  015370  022767  177777  000014        CMP     #-1,TYPSBX          ;TERMINATOR?
3245  015376  001001                        BNE     TYPSAX              ;NO, BRANCH
3246  015400  000205                        RTS     %5                  ;YES, RETURN
3247  015402  013746  177776        TYPSAX: MOV     @#PS,-(SP)          ;SETUP TO CALL TYPE ROUTINE VIA JSR
3248  015406  004767  163502                JSR     PC,$TYPE            ;TYPE ASCII MESSAGE
3249  015412  000000                TYPSBX: OPEN
3250  015414  000763                        BR      TYPSX
3251
3252
3253
3254                                  ;SUBROUTINE FOR OCTAL TO ASCII CONVERSION
3255  015416  013567  000074        OACNV:  MOV     @(5)+,OACNVX        ;GET OCTAL VALUE
3256  015422  012567  000072                MOV     (5)+,OACDST         ;GET DESTINATION ADDRESS
3257  015426  012567  000070                MOV     (5)+,OACNT          ;GET CONVERT COUNT
3258  015432  066767  000064  000060        ADD     OACNT,OACDST        ;DEVELOP ADDRESS TO STORE 1ST CHAR.
3259  015440  016746  000052        OACNVA: MOV     OACNVX,-(SP)
3260  015444  042716  177770                BIC     #177770,@SP         ;ISOLATE LEAST SIGNIFICANT DIGIT
3261  015450  062716  000060                ADD     #60,@SP             ;CONVERT DIGIT TO ASCII
3262  015454  005367  000040                DEC     OACDST
3263  015460  112677  000034                MOVB    (SP)+,@OACDST       ;STORE ASCII CHARACTER
3264  015464  042767  000007  000024        BIC     #7,OACNVX
3265  015472  006067  000020                ROR     OACNVX
3266  015476  006067  000014                ROR     OACNVX
3267  015502  006067  000010                ROR     OACNVX
3268  015506  005367  000010                DEC     OACNT               ;DONE ALL DIGITS?
3269  015512  001352                        BNE     OACNVA              ;BRANCH IF NOT DONE
3270  015514  000205                        RTS     R5                  ;DONE, EXIT
3271  015516  000000                OACNVX: OPEN
3272  015520  000000                OACDST: 0
3273  015522  000000                OACNT:  0
3274
3275
3276
3277                                  ;SUBROUTINE FOR BINARY TO DECIMAL ASCII CONVERSION
3278  015524  104005                BDCNV:  SAV04                       ;SAVE REGS
3279  015526  012700  015702                MOV     #DECVAL,%0          ;SET UP ADDR TO STORE DECIMAL ASCII
```

```
3280  015532  013501                      MOV     @(5)+,R1                ;BINARY VALUE TO R1
3281  015534  012567  000052              MOV     (5)+,BDCNVC             ;DESTINATION ADDR TO BDCNVC
3282  015540  012567  000050              MOV     (5)+,BDCNVD             ;CHARACTER COUNT TO BDCNVD
3283  015544  012702  015670              MOV     #ADTENP,R2              ;ADDR OF TEN POWER STRING
3284  015550  012767  000005  000104      MOV     #5,CNVCTR               ;SET UP FOR 5 POWER CONVERSIONS
3285  015556  012267  000104      BDCNVA: MOV     (2)+,TENPWR             ;MOVE POWER OF TEN VALUE
3286  015562  004767  000034              JSR     PC,SUBTEN               ;PERFORM CONVERSION
3287  015566  005367  000070              DEC     CNVCTR                  ;DONE 5 CONVERSIONS?
3288  015572  001371                      BNE     BDCNVA                  ;BRANCH IF NOT YET 5.
3289  015574  166700  000014              SUB     BDCNVD,%0
3290  015600  010067  000004              MOV     %0,BDCNVB
3291  015604  004567  000100              JSR     R5,BMOVE
3292  015610  000000          BDCNVB:     OPEN
3293  015612  000000          BDCNVC:     OPEN
3294  015614  000000          BDCNVD:     OPEN
3295  015616  104006                      RST04                           ;RESTORE REGS AND EXIT
3296  015620  000205                      RTS     R5
3297  015622  005067  000036      SUBTEN: CLR     DIGIT
3298  015626  166701  000034      SUBTNA: SUB     TENPWR,R1               ;SUBTRACT TEN POWER FROM BINARY VALUE
3299  015632  103403                      BCS     SUBTNB                  ;BRANCH IF UNSUCESSFUL SUBTRACTION
3300  015634  005267  000024              INC     DIGIT
3301  015640  000772                      BR      SUBTNA
3302  015642  066701  000020      SUBTNB: ADD     TENPWR,R1               ;RESTORE SUBTRACTED VALUE.
3303  015646  062767  000060  000010      ADD     #60,DIGIT               ;CONVERT (DIGIT) TO ASCII
3304  015654  116720  000004              MOVB    DIGIT,(0)+              ;MOVE ASCII CHAR TO DECVAL FIELD
3305  015660  000207                      RTS     PC                      ;EXIT
3306  015662  000000          CNVCTR:     OPEN
3307  015664  000000          DIGIT:      OPEN
3308  015666  000000          TENPWR:     OPEN
3309  015670  023420          ADTENP:     10000.
3310  015672  001750                      1000.
3311  015674  000144                      100.
3312  015676  000012                      10.
3313  015700  000001                      1.
3314  015702     040     040     040  DECVAL: .BYTE   040,040,040,040,040,040
3315  015705     040     040     040
3316
3317
3318
3319                                  ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES
3320  015710  104005          BMOVE:  SAV04                           ;SAVE REGS
3321  015712  012501                  MOV     (5)+,R1                 ;GET FROM ADDRESS
3322  015714  012502                  MOV     (5)+,R2                 ;GET TO ADDRESS
3323  015716  012503                  MOV     (5)+,R3                 ;GET COUNT
3324  015720  112122          BMOVA:  MOVB    (1)+,(2)+               ;MOVE BYTE
3325  015722  005303                  DEC     R3                      ;DECREMENT COUNT
3326  015724  001375                  BNE     BMOVA                   ;BRANCH IF NOT DONE
3327  015726  104006                  RST04                           ;RESTORE REGS AND EXIT
3328  015730  000205                  RTS     R5
3329
3330
3331
3332                                  ;UNEXPECTED POWER FAIL SERVICE
3333                                  ;BECAUSE WWP MAY BE SET IN MPR'S AND ALL PROCESSOR REGISTERS
3334                                  ;MAY BE IN USE, CONTINUATION AFTER POWER FAIL IS NOT ATTEMPTED.
3335                                  ;INSTEAD, THE PROGRAM RESTARTS AFTER A POWER FAILURE
```

```
 3336  015732  012737  015776  000024  PWRDN:  MOV     #PWRUP,@#24             ;SET UP FOR POWER UP
 3337  015740  012701  000566          MOV     #MPR0,R1
 3338  015744  032711  000001  1$:     BIT     #1,@R1
 3339  015750  001002                  BNE     .+6
 3340  015752  005071  000000          CLR     @(R1)                   ;CLEAR PARITY REGISTERS IN CASE
 3341  015756  062701  000010          ADD     #10,R1                  ;WWP IS SET
 3342  015762  020127  000766          CMP     R1,#TREG
```

```
3343  015766  103766                        BLO     1$
3344  015770  010667  163264                MOV     SP,SPSAV
3345  015774  000000                        HALT                            ;POWER DOWN HALT
3346  015776  012737  015732  000024  PWRUP: MOV     #PWRDN,@#24             ;SET UP FOR POWER DOWN
3347  016004  016706  163250                MOV     SPSAV,SP
3348  016010  005027  000000                CLR     #0                      ;STALL SO OUTPUT WON'T BE GARBLED
3349  016014  005367  177772                DEC     .-2
3350  016020  001375                        BNE     .-4
3351  016022  104000                        TYPE                            ;TYPE RECOVERY MESSAGE
3352  016024  017344                        MPWRF
3353  016026  000167  163244                JMP     RSTART                  ;RESTART
3354
3355
3356
3357                                  ;EMT HANDLER
3358  016032  011646                EMTINT: MOV     (SP),-(SP)              ;GET SAVED PC
3359  016034  162716  000002                SUB     #2,(SP)                 ;DECREMENT PC BY 2
3360  016040  017616  000000                MOV     @(SP),(SP)              ;GET CALL
3361  016044  121667  000050                CMPB    (SP),EMTLIM             ;CHECK IF CALL WITHIN LIMITS
3362  016050  101402                        BLOS    EMTA
3363  016052  000000                        HALT                            ;CALL IS NOT WITHIN LIMITS
3364  016054  000776                        BR      .-2
3365  016056  006116                EMTA:   ROL     (SP)                    ;EMT ARG X 2
3366  016060  042716  177001                BIC     #177001,(SP)            ;REMOVE 7 MSB
3367  016064  062716  016076                ADD     #EMTTAB,(SP)            ;FORM EMT RTN ADDRESS
3368  016070  017616  000000                MOV     @(SP),(SP)
3369  016074  000136                        JMP     @(SP)+                  ;GO TO EMT RETURN
3370
3371
3372                                  ;EMT DEFINITIONS AND ASSIGNMENTS
3373  016076                        EMTTAB:
3374          104000                        TYPE=EMT+EMTX
3375  016076  001114                        $TYPE
3376          104001                        SCOPE=EMT+EMTX
3377  016100  016356                        SCOPEC
3378          104002                        ERROR=EMT+EMTX
3379  016102  013562                        ERR
3380          104003                        ERRORP=EMT+EMTX
3381  016104  013442                        ERRP
3382          104004                        ERRORS=EMT+EMTX
3383  016106  013364                        ERRST
3384          104005                        SAVO4=EMT+EMTX
3385  016110  016122                        SVO4
3386          104006                        RSTO4=EMT+EMTX
3387  016112  016210                        RSO4
3388          104007                        RSTO5S=EMT+EMTX
3389  016114  016236                        RSO5S
3390          104010                        SAVO5S=EMT+EMTX
3391  016116  016142                        SVO5S
3392  016120  000010                EMTLIM: EMTX-1
3393
3394
3395                                  ;SUBROUTINE TO SAVE REGS 0-4
3396  016122  012666  177764         SVO4:   MOV     (SP)+,-12.(SP)          ;MOVE PC+PS UP STACK
3397  016126  012666  177764                 MOV     (SP)+,-12.(SP)
3398  016132  012767  000002  000040         MOV     #RTI,SVO5C
```

```
                                        BR      SV05B
                                ;SUBROUTINE TO SAVE REGS 0-5 + PLACE EMT PC IN R5
                        SV05S:  MOV     #NOP,SV05C
                                BR      SV05A
                                ;SUBROUTINE TO SAVE REGS 0-5
                        SV05A:  MOV     (SP)+,-14.(SP)
                                MOV     (SP)+,-14.(SP)
                                MOV     R5,-(SP)
                        SV05B:  MOV     R4,-(SP)
                                MOV     R3,-(SP)
                                MOV     R2,-(SP)
                                MOV     R1,-(SP)
                                MOV     %0,-(SP)
                                CMP     -(SP),-(SP)
                        SV05C:  RTI                             ;RTI OR NOP
                                MOV     16.(SP),R5              ;EMT PC TO R5
                                RTI
                                ;SUBROUTINE TO RESTORE REGS 0-4
                        RS04:   CMP     (SP)+,(SP)+
                                MOV     (SP)+,%0
                                MOV     (SP)+,R1
                                MOV     (SP)+,R2
                                MOV     (SP)+,R3
                                MOV     (SP)+,R4
                                MOV     -12.(SP),-(SP)          ;MOVE PC+PS DOWN STACK
                                MOV     -12.(SP),-(SP)
                                RTI
                                ;SUBROUTINE TO RESTORE REGS 0-5
                        RS05S:  MOV     R5,16.(SP)              ;SET EMT PC TO R5
                                CMP     (SP)+,(SP)+
                                MOV     (SP)+,%0
                                MOV     (SP)+,R1
                                MOV     (SP)+,R2
                                MOV     (SP)+,R3
                                MOV     (SP)+,R4
                                MOV     (SP)+,R5
                                MOV     -14.(SP),-(SP)
                                MOV     -14.(SP),-(SP)
                                RTI
                                ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
                                ;LOAD THE STARTING ADDRESS OF THE TEST
                                ;YOU WISH TO RUN (THE ADDRESS OF THE TESTXX
                                ;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
```

```
                                        ;OPTIONS AT THE 2ND HALT.
                                        ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
016272  005037  177776          TESTX:  CLR     @#PS                    ;WAIT FOR STARTING ADDRESS
016276  000000                          HALT
016300  017767  162574  000154        MOV     @SWR,RETURN             ;LOAD STARTING ADDRESS IN RETURN
016306  062767  000002  000146        ADD     #2,RETURN               ;ADD 2 TO POINT TO INSTRUCTION AFTER
016314  000000                        HALT                            ;SET SR OPTIONS
016316  012767  177777  162164        MOV     #-1,TSTX                ;SET FLAG
016324  032777  010000  162546        BIT     #10000,@SWR             ;CHECK SW12
016332  001404                        BEQ     .+12                    ;BRANCH IF NOT SET
016334  042737  000020  177776        BIC     #20,@#PS                ;CLEAR TRACE BIT
016342  000403                        BR      .+10                    ;SKIP NEXT INSTRUCTION
016344  052737  000020  177776        BIS     #20,@#PS                ;SET TRACE BIT
016352  000177  000104                JMP     @RETURN                 ;JUMP TO TEST


                                        ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 64 TIMES
                                        ;A SETUP ROUTINE SHOULD INITIALIZE RETURN AND IMAX
016356  032777  040000  162514  SCOPEC: BIT     #40000,@SWR             ;TEST SR FOR SCOPE
016364  001020                        BNE     SCOPEB                  ;YES, SCOPE
016366  032777  004000  162504        BIT     #4000,@SWR              ;NO-TEST FOR ITERATION
016374  001021                        BNE     SCOPEG                  ;INHIBIT ITERATION
016376  005767  173350                TST     PASCNT                  ;FIRST PASS?
016402  001416                        BEQ     SCOPEG                  ;YES, INHIBIT ITERATIONS
016404  005767  162100                TST     TSTX                    ;USING SINGLE SUBTEST STARTUP?
016410  001006                        BNE     SCOPEB                  ;YES, LOOP
016412  026767  000042  000036        CMP     ICNT,IMAX               ;COMPARE CURRENT COUNT TO MAX NUMBER
016420  100007                        BPL     SCOPEG                  ;EXIT-DONE
016422  005267  000032                INC     ICNT                    ;INCREMENT COUNT
016426  022606                SCOPEB: CMP     (6)+,%6                 ;REPOSITION STACK
016430  012677  161342                MOV     (6)+,@PS                ;RESTORE PREVIOUS PROCESSOR STATUS
016434  000177  000022                JMP     @RETURN                 ;REPEAT TEST
016440  005067  162044        SCOPEG: CLR     TSTX                    ;IF USING TESTX STARTUP, RETURN TO NORMAL FLOW
016444  005067  000010                CLR     ICNT                    ;CLEAR COUNT
016450  011667  000006                MOV     @%6,RETURN              ;SAVE SCOPE RETURN POINTER
016454  000002                        RTI                             ;RETURN INLINE-NEXT TEST
016456  000100                IMAX:   100                             ;ITERATION COUNT
016460  000000                ICNT:   0                               ;COUNT LOCATION FOR ITERATION LOOP
016462  000000                RETURN: 0                               ;ADDRESS OF LAST TEST


                                        ;ASCII MESSAGES
016464                          MEO:
016464  005015  041520  020075  MTNUM:  .ASCII  <15><12>'PC= '
016472  020040  020040  020040  MPC:    .ASCII  '        ICNT= '
016500  020040  041511  052116
016506  020075
016510  020040  020040  020040  MICNT:  .ASCIZ  '      '
016516     000
016517     040  046440  051120  MSTR:   .ASCII  '  MPR= '
016524  020075
016526  020040  020040  020040  MTREG:  .ASCII  '       MPR DATA= '
016534  020040  050115  020122
```

```
3511  016542  040504  040524  020075
3512  016550  020040  020040  020040  MDATA:  .ASCIZ  '        '
3513  016556  020040     000
3514  016561     015  020012  020040  MSTRX:  .ASCII  <15><12>'        TEST LOC= '
3515  016566  020040  020040  052040
3516  016574  051505  020124  047514
3517  016602  036503     040
3518  016605     040  020040  020040  MSTRX1: .ASCII  '      '
3519  016612     040
3520  016613     040  027523  035102          .ASCII  ' S/B: '
3521  016620     040
3522  016621     040  020040  020040  MSTRX3: .ASCII  '      '
3523  016626     040
3524  016627     040  040527  035123          .ASCII  ' WAS: '
3525  016634     040
3526  016635     040  020040  020040  MSTRX5: .ASCIZ  '        '
3527  016642  020040     000
3528  016645     015  051412  052105  MSETSR: .ASCIZ  <15><12>'SET SR OPTIONS'
3529  016652  051440  020122  050117
3530  016660  044524  047117  000123
3531  016666  020054  051120  051505  MCON:   .ASCIZ  ', PRESS CONTNUE'
3532  016674  020123  047503  052116
3533  016702  052516  000105
3534  016706  005015  042523  020124  MMDEV:  .ASCIZ  <15><12>'SET DEVICE ADDRESS IN SR'
3535  016714  042504  044523  042503
3536  016722  040440  042104  042522
3537  016730  051523  044440  020116
3538  016736  051123     000
3539  016741     015  051412  052105  MMADR:  .ASCIZ  <15><12>'SET MEMORY TEST LOC IN SR'
3540  016746  046440  046505  051117
3541  016754  020131  042524  052123
3542  016762  046040  041517  044440
3543  016770  020116  051123     000
3544  016775     015  051412  052105  MMPAT:  .ASCIZ  <15><12>'SET TEST PATTERN IN SR'
3545  017002  052040  051505  020124
3546  017010  040520  052124  051105
3547  017016  020116  047111  051440
3548  017024  000122
3549  017026  005015  046412  046505  MMPRS:  .ASCIZ  <15><12><12>'MEMORY PARITY REGISTERS PRESENT:'<15><12>
3550  017034  051117  020131  040520
3551  017042  044522  054524  051040
3552  017050  043505  051511  042524
3553  017056  051522  050040  042522
3554  017064  042523  052116  006472
3555  017072  000012
3556  017074  005015  040520  044522  MTMAP:  .ASCIZ  <15><12>'PARITY REGISTERS CONTROL MEMORY AS:'<15><12>
3557  017102  054524  051040  043505
3558  017110  051511  042524  051522
3559  017116  041440  047117  051124
3560  017124  046117  046440  046505
3561  017132  051117  020131  051501
3562  017140  006472  000012
3563  017144  020040  020040     000  MTYCOR: .ASCIZ  '  '
3564  017151     055  000040          MDASH:  .ASCIZ  '- '
3565  017154  005015     000          MCR:    .ASCIZ  <15><12>
3566  017157     113     000          MK:     .ASCIZ  'K'
```

```
3567  017161  116     020117  040520  MT:     .ASCIZ  'NO PARITY MEMORY FOUND'<15><12>
3568  017166  044522  054524  046440
3569  017174  046505  051117  020131
3570  017202  047506  047125  006504
3571  017210  000012
3572  017212  047516  050040  051101  MTR:    .ASCIZ  'NO PARITY REGSTER FOUND'<15><12>
3573  017220  052111  020131  042522
3574  017226  051507  042524  020122
3575  017234  047506  047125  006504
3576  017242  000012
3577  017244  020040  020040  020040  MPRAD:  .ASCIZ  '         '<15><12>
3578  017252  020040  005015  000
3579  017257  177     005015  005015  MTIT:   .ASCIZ  <177><15><12><15><12>'CCMFAF0 MS11,MA11-P,MF11-LP PARITY MEMORY TESTS'
3580  017264  041503  043115  043101
3581  017272  020060  051515  030461
3582  017300  046454  030501  026461
3583  017306  026120  043115  030461
3584  017314  046055  020120  040520
3585  017322  044522  054524  046440
3586  017330  046505  051117  020131
3587  017336  042524  052123  000123
3588  017344  005015  047520  042527  MPWRF:  .ASCIZ  <15><12>'POWER FAILED'
3589  017352  020122  040506  046111
3590  017360  042105  000
3591  017363  007             MPGEND: .BYTE   007
3592  017364  005015  047105  020104          .ASCII  <15><12>'END PASS = '
3593  017372  040520  051523  036440
3594  017400  040
3595  017401  040     020040  020040  MPCNT:  .ASCIZ  '      '
3596  017406  000040
3597  017410  006415  041012  042101  MPSER:  .ASCII  <15><15><12>'BAD PAR FOUND IN LOC '
3598  017416  050040  051101  043040
3599  017424  052517  042116  044440
3600  017432  020116  047514  020103
3601  017440  020040  020040  020040  MPSER1: .ASCIZ  '      '
3602  017446  000
3603  017447  015     051012  043505  MX1:    .ASCIZ  <15><12>'REGISTER AT '
3604  017454  051511  042524  020122
3605  017462  052101  000040
3606  017466  005015  047515  044516  NOMON:  .ASCIZ  <15><12>'MONITOR WILL NOT BE RESTORED FOR AN 8K SYSTEM'
3607  017474  047524  020122  044527
3608  017502  046114  047040  052117
3609  017510  041040  020105  042522
3610  017516  052123  051117  042105
3611  017524  043040  051117  040440
3612  017532  020116  045470  051440
3613  017540  051531  042524  000115
3614  017546  047503  052116  047522  MX2:    .ASCIZ  'CONTROLS '
3615  017554  051514  000040
3616  017560  005015  047514  042101  LDRMSG: .ASCIZ  <15><12>'LOADERS RESTORED'
3617  017566  051105  020123  042522
3618  017574  052123  051117  042105
3619  017602  000
3620  017603  015     041012  042101  PSMSG:  .ASCIZ  <15><12>'BAD PARITY SCAN COMPLETE'
3621  017610  050040  051101  052111
3622  017616  020131  041523  047101
```

```
3623  017624  041440  046517  046120
3624  017632  052105  000105
3625  017636  005015  047514  042101  MLDRSV: .ASCIZ  <15><12>'LOADERS SAVED IN BANK 1'
3626  017644  051105  020123  040523
3627  017652  042526  020104  047111
3628  017660  041040  047101  020113
3629  017666  000061
3630  017670  020040  020040  020040  MPRCOR: .ASCIZ  '         - CORE PARITY REGISTER'<15><12>
3631  017676  020040  020055  047503
3632  017704  042522  050040  051101
3633  017712  052111  020131  042522
3634  017720  044507  052123  051105
3635  017726  005015     000
3636  017731     040  020040  020040  MPRMOS: .ASCIZ  '         - MOS PARITY REGISTER'<15><12>
3637  017736  020040  026440  046440
3638  017744  051517  050040  051101
3639  017752  052111  020131  042522
3640  017760  044507  052123  051105
3641  017766  005015     000
3642          017772                  .EVEN
3643          000001                  .END
```

```
ADRPT   000516   425# 1671* 1673* 1680* 1687 1689 1712 1802* 1902* 1904* 1911* 1916 1919
                1932 2763* 2777 2822*
ADRS  = 077400   364#
ADRTYP  001230   660# 2655* 2668 2696 2701* 2702 2711* 2741*
ADTENP  015670  3283 3309#
AE    = 000001   362# 1040 1283 1335 1498 1599 1696 1812 1939 1968 1986 2128 2145
                2163 2315
BDCNV   015524  2782 2797 3278#
BDCNVA  015556  3285# 3288
BDCNVB  015610  3290* 3292#
BDCNVC  015612  3281* 3293#
BDCNVD  015614  3282* 3289 3294#
BEGIN   002106   692  806  815# 2575
BITPT   000520   426# 822* 1668 1670* 1674 1676 1681* 1682 1789* 1790 1793* 1799* 1899
                1901* 1905 1907 1912* 1913 2002 2089* 2098 2101* 2107* 2179 2262* 2597*
                2606* 2608* 2615 2626* 2633* 2634 2638 2649* 2690 2692 2707 2710* 2712
                2728* 2730* 2738 2764* 2777 2808* 2823* 2831*
BIT0  = 000001   346#
BIT1  = 000002   347#
BIT10 = 002000   356#
BIT11 = 004000   357#
BIT12 = 010000   358#
BIT13 = 020000   359# 2901
BIT14 = 040000   360#  402
BIT15 = 100000   361#
BIT2  = 000004   348#
BIT3  = 000010   349#
BIT4  = 000020   350#
BIT5  = 000040   351# 1356 1371
BIT6  = 000100   352# 1441
BIT7  = 000200   353#
BIT8  = 000400   354# 2567
BIT9  = 001000   355#  983
BMOVA   015720  3324# 3326
BMOVE   015710  3291 3320#
CACHE   001224   658# 818* 910*
CACHFL  001222   657# 737* 816 908
CKBKO   010600  2320 2331#
CLRPAR  014012   966  999 1090 1144 1273 1308 1402 1468 1571 1663 1705 1781 1821
                1918 1925 2087 2275 2304 2324 2416 2519 2549 2750 2954# 3106 3147
                3191
CNT16   007314  2052#
CNT17   010206  2157 2239#
CNVCTR  015662  3284* 3287* 3306#
CN16    007334  1980 2063#
CONT    011712  2561 2567#
CONT10  004274  1344 1350 1381#
CONT12  005024  1510 1552#
CONT13  005304  1609 1644#
CONT3   002616   971#  987
CONT4   003074  1052# 1062 1071 1078
COUNT   004336  1337* 1351* 1391#
DDISP = 177570   367#  616
DECVAL  015702  3279 3314#
DIGIT   015664  3297* 3300* 3303* 3304 3307#
DISPLA  001102   616#  732*  832*  884*  963*  997* 1089* 1143* 1272* 1307* 1401* 1467* 1570*
```

```
                    1661*  1777*  1897*  2082*  2083*  2303*  2415*
DISPRE  000174       406#   732
DONE    011624      2266   2270   2273   2540   2547#
DONE10  004322      1323   1387#
DONE11  004542      1419   1455#
DONE12  005040      1485   1557#
DONE13  005320      1584   1649#
DONE14  005616      1685   1707#
DONE15  006262      1796   1823#
DONE16  006650      1915   1925#
DONE17  007576      2104   2116#
DONE20  010564      2324#  2362
DONE4   003140      1013   1079#
DONE5   003322      1103   1131#
DONE6   003662      1156   1262#
DONE7   003774      1282   1288#
DSWR  = 177570       366#   615
EMTA    016056      3362   3365#
EMTINT  016032       397   3358#
EMTLIM  016120      3361   3392#
EMTTAB  016076      3367   3373#
EMTX  = 000011      3373#  3374   3376#  3378#  3380#  3382#  3384#  3386#  3388#  3390#  3392#
ERR     013562      2336   2357   2375   2389   2396   2461   2468   2482   2499   2509   2898#  3379
ERRA    013604      2856   2892   2901#
ERRA1   013632      2284*  2531*  2906#
ERRA2   013634      2285*  2532*  2907#
ERRA3   013644      2286*  2533*  2910#
ERRA4   013646      2287*  2534*  2911#
ERRA5   013656      2288*  2535*  2914#
ERRB    013660      2844*  2866*  2898*  2915#
ERRBX   013662      2845*  2846*  2868*  2869*  2899*  2900*  2916#
ERRC    013666      2902   2918#
ERRD    013710      2903*  2904*  2906   2924#
ERROR = 104002       850    864    922    932   1027   1046   1056   1063   1074   1111   1125   1165   1181
                    1195   1213   1230   1240   1258   1290   1327   1341   1347   1359   1375   1424   1438
                    1445   1491   1503   1514   1522   1537   1545   1591   1603   1613   1622   1638   1718
                    1755   1836   1871   1972   1999   2005   2029   2057   2149   2176   2183   2199   2208
                    2244   2816   3144   3236   3378#
ERRORP= 104003      2020   2046   3380#
ERRORS= 104004      1736   1759   1855   1875   2232   3382#
ERRP    013442      2866#  3381
ERRST   013364      2844#  3383
EX      005460      1677   1679#
FTITLE  000512       423#   700    743    745*
GMPRA   001664       766#   798
GMPRB   002012       764    793#
GMPRC   002020       792    795#
GMPRD   002054       802    805#
HIADR   000530       430#  1687*  1688*  1691   1722   1916*  1917*  1921   2066
HIWORD  000552       440#  2653*  2688   2705   2732*  2757*  2810   2824*
ICNT    016460      2910   3482   3484*  3489*  3493#
IMAX    016456       820*   885*   998*  1662*  3482   3492#
INDC0   000770       556#   763    834    887    907   1007   1316   1409   1475   1991   2168   2445
INDC1   000772       557#
INDC10  001014       566#
INDC11  001016       567#
```

```
INDC12  001020   568#
INDC13  001022   569#
INDC14  001024   570#
INDC15  001026   571#
INDC2   000774   558#
INDC3   000776   559#
INDC4   001000   560#
INDC5   001002   561#
INDC6   001004   562#
INDC7   001006   563#
INDC8   001010   564#
INDC9   001012   565#
INDX17= 010060  2097*  2108*  2181#
INST1   004164  1339#  1364
INST2   004172  1345#  1367
KPAR0   001250   668#  2979#
KPAR1   001252   669#  1785*  1792*  2094*  2100*  2229  2600*  2740*  3015*  3164  3171*  3180*  3188*
                3208
KPAR2   001254   670#
KPAR7   001256   671#  2978#
KPDR0   001240   664#  2975#
KPDR1   001242   665#  2976#
KPDR2   001244   666#
KPDR7   001246   667#  2977#
KTSTAR  001226   659#  2650*  2718  2726*
LDRMSG  017560  3082  3616#
LDRSVD  014416  3052*  3059#  3065
LOCATM  014126  1014  1104  1158  1324  1420  1487  1586  2991#
LOCAT1  014224  3000  3006  3013#
LOGICA  011672   611  2560#
LOOP10  004066  1319#  1331  1386
LOOP11  004416  1411  1415#  1427  1454
LOOP12  004632  1477  1481#  1494  1555
LOOP13  005124  1581#  1595  1647
LOOP14  005436  1672  1674#  1683
LOOP15  006106  1790#  1794  1800
LOOP21  011260  2454#  2477
LOOP4   002762  1008#  1012
LOOP5   003216  1098#  1102
LOOP6   003372  1151#  1155
LOOP7   003734  1279#  1287  1297
LOP15   006116  1792#  1822
LOP4    002770  1010#  1031  1055
LOP5    003224  1100#  1115  1130
LOP6    003400  1153#  1169  1261
LOWFLG  000554   441#  1786*  1795  1797*  2095*  2103  2105*
LSAV    014300  3025*  3026  3029#
LUP10   004060  1317#  1322
LUP11   004402  1410#  1418
LUP12   004616  1476#  1484
LUP13   005116  1578#  1583
LUP14   005446  1676#  1706
LUP16   006526  1900  1904#
LUP7    003726  1277#  1281
MAPMA   011770  2589#
MAPMB   012162  2585  2621#
```

```
MAPMEM  011756   707   967  2585#
MAPRB   012346  2660#  2704  2714  2742
MAPRC   012446  2679#
MAPRD   012546  2682  2700#
MAPRE   012642  2706  2708  2715#
MAPREG  012304   968  2648#
MAP1    014062  1003  1094  1148  1312  1406  1472  1575  1783  2091  2591  2723  2975#  3168
MCON    016666  3531#
MCR     017154  2806  3565#
MDASH   017151  2791  3564#
MDATA   016550  2854  2878  3512#
MEMH    001066   597#  2596*  2609  2625*  2731  3178
MEML    001064   596#  2271  2595*  2598  2624*  2638*  2648  3124  3175
MEMUT   000546   438#  2598*  2609*  2615*  2648*  2712  2731*  2736
MEO     016464  2914  3500#
MICNT   016510  2911  3505#
MK      017157  2804  3566#
MLDRSV  017636   751  3625#
MMADR   016741  3539#
MMDEV   016706  3534#
MMPAT   016775  3544#
MMPRS   017026   760  3549#
MONCNT  011754  2577#
MPC     016472  2907  3502#
MPCNT   017401  2553  3595#
MPGEND  017363  2556  3591#
MPRAD   017244  2770  2775  3577#
MPRCOR  017670   777   780  3630#
MPRMOS  017731   786   789  3636#
MPROK   000542   436#   761*   791*  805
MPRO    000566   478#   762   833  886  906  973  1006  1097  1150  1276  1315  1408  1474
                1577  1693  1726  1747  1808  1845  1863  1936  1948  1964  1983  1990  2125
                2142  2160  2167  2311  2347  2366  2429  2444  2452  2660  2671  2679  2753
                2957  3109  3193  3337
MPR1    000576   482#
MPR10   000706   518#
MPR11   000716   522#
MPR12   000726   526#
MPR13   000736   530#
MPR14   000746   534#
MPR15   000756   538#
MPR2    000606   486#
MPR3    000616   490#
MPR4    000626   494#
MPR5    000636   498#
MPR6    000646   502#
MPR7    000656   506#
MPR8    000666   510#
MPR9    000676   514#
MPSER   017410  3134  3227  3597#
MPSER1  017440  3131  3221  3223  3225*  3601#
MPWRF   017344  3352  3588#
MSETSR  016645  3528#
MSTR    016517  2844  2866  3507#
MSTRX   016561  2868  3514#
MSTRX1  016605  2882  3518#
```

```
MSTRX3  016621  2886  3522#
MSTRX5  016635  2890  3526#
MT      017161  3567#
MTIT    017257   749  3579#
MTMAP   017074  2752  3556#
MTNUM   016464  3501#
MTR     017212   808  3572#
MTREG   016526  2850  2874  3509#
MTYCOR  017144  2784  2789  2799  2802  3563#
MTYFG   000562   445#  684#  722#  964  982#
MX1     017447  2773  3603#
MX2     017546  2787  3614#
NOKT    000550   439#  711  1000  1080  1091  1131  1145  1262  1309  1387  1403  1455  1469
                1558  1572  1650  1778  2084  2589#  2621#  2715  2994  3013  3102
NOMON   017466  2834  3606#
NOP   = 000240   371#  3404
NOREG   002062   807#
NRALL   013712  1002  1093  1147  1311  1405  1471  1574  1782  2090  2590  2722  2929#  3167
OACDST  015520  3256#  3258#  3262#  3263#  3272#
OACNT   015522  3257#  3258  3268#  3273#
OACNV   015416   775  784  2551  2768  2848  2852  2872  2876  2880  2884  2888  2905  2909
                3129  3219  3255#
OACNVA  015440  3259#  3269
OACNVX  015516  3255#  3259  3264#  3265#  3266#  3267#  3271#
ODDFLG  000556   442#  1933#  1945  1955  2063#  2122#  2133  2250#  2426#  2437  2514#
OPEN  = 000000   372#  2915  2916  2924  3249  3271  3292  3293  3294  3306  3307  3308
OUT     010330  1910  2274#
PARPAT  001036   582#  1686  1801  2305
PARVEC= 000114   381#  1039#  1042#  1079#  1274#  1275#  1288#  1336#  1366#  1381#  1495#  1531#  1557#
                1596#  1632#  1649#  1664#  1707#  1788#  1824#  1898#  1926#  2088#  2117#  2310#  2325#
                2547#
PASCNT  011752   685#  723#  2550#  2552  2576#  3478
PDREND  001236   663#  2942
PDRTAB  001232   661#  2936
PERR  = 100000   365#  1284
PMEMH   001072   602#  972#  977#  1798  2106
PMEML   001070   600#  971#  976#  1665  1674  1787  1905  2096  2268
PMEMX   001074   604#  1787#  1790  1798#  2096#  2098  2106#
PS    = 177776   368#  689#  2274#  2301  2335  2356  2374  2388  2395  2413  2460  2467  2481
                2498  2508  2520  3247  3457#  3465#  3467#  3486#
PSADRS  015020  3128#  3130  3157#  3205#  3206#  3218#  3220
PSCAN   014514   708  1740  1764  1879  3090#
PSCANH  015022  3158#  3170#  3173  3183  3185#
PSCANX  014762  3123  3147#  3189
PSCAN1  015024  3103  3164#
PSCX1   015136  3184  3188#
PSLOOP  015060  3172#  3187
PSLUP   015064  3173#  3182
PSMSG   017603   710  3620#
PSNXT   015110  3177  3180#  3204
PSXTST  015144  3176  3179  3190#
PWRDN   015732   686  724  3336#  3346
PWRUP   015776  3336  3346#
RELOC   000564   447#  2651#  2721#  2733#  2734  2740
RESRVD  001030   572#  841#  843#  848  854  860
RESVC   001032   575#  841
```

```
RESVM   001034   576#   843    929
RETURN  016462   815*  2283*  3459*  3460*  3468   3487   3490*  3494#
RSTART  001276   414    683#   3353
RSTLDR  014420   410   2081    3065#
RSTLDX  014510  3066   3076    3083#
RST04 = 104006  3295   3327    3386#
RSTOSS= 104007  3388#
RS04    016210  3387   3424#
RSOSS   016236  3389   3437#
R0SAV   001262   673#
R1SAV   001264   674#
R2SAV   001266   675#
R3SAV   001270   676#
R4SAV   001272   677#
R5SAV   001274   678#
SAVLDR  014302   819   3034#
SAVLDX  014400  3047   3054#
SAV04 = 104005  3278   3320    3384#
SAVOSS= 104010  3390#
SCAN    001356   412    699#    715
SCANA   001400   706#   803
SCANB   001404   703    707#
SCNFLG  001220   655#   704*    706*   800
SCOPE = 104001   831    883     962    996   1088   1142   1271   1306   1400   1466   1569   1660   1776
                1896   2080    2258   3376#
SCOPEB  016426  3475   3481    3485#
SCOPEC  016356  2302   2414    2521   3377   3474#
SCOPEG  016440  3477   3479    3483   3488#
SHDBE   000534   433#  1934*   2123*  2427*  2885
SMLSYS  013344  1667   2831#
SPSAV   001260   672#  3344*   3347
SR0   = 177572   382#   713*   1082*  1133*  1264*  1389*  1457*  1560*  1652*  1823*  2116*  2586   2612*
                2727*  2736*   2980*  3165
START   001436   408    721#    813
START1  001632   705    744     747    759#
STKPT = 000510   373#   683     699    721   2548
SUBTEN  015622  3286   3297#
SUBTNA  015626  3298#  3301
SUBTNB  015642  3299   3302#
SV04    016122  3385   3396#
SV05A   016152  3405   3408#
SV05B   016164  3399   3411#
SV05C   016200  3398*  3404*  3417#
SV05S   016142  3391   3404#
SWR     001100   615#   729    731*   983   2289   2291*  2536   2538*  2567   2901   2920   3135   3228
                3459   3463   3474   3476
SWREG   000176   407#   731
TBANK   000544   437#  2594*  2600   2605*  2610   2652*  2667   2709*  2720*
TEMP    000560   444#
TEMPX   000514   424#   774*   776    783*   785   2767*  2769
TENPWR  015666  3285*  3298   3302   3308#
TESTX   016272  3457#
TEST1   002154   815    831#
TEST10  004016  1289   1306#
TEST11  004340  1390   1400#
TEST12  004554  1466#
```

```
TEST13  005060  1569#
TEST14  005340  1660#  2263
TEST15  006014  1708   1776#
TEST16  006462  1679   1780   1825   1896#
TEST17  007360  1927   2080#
TEST2   002326   883#
TEST20  010452  2283   2293   2301#
TEST21  011054  2326   2413#
TEST3   002564   962#
TEST4   002712   995   996#
TEST5   003160  1080#
TEST6   003334  1142#
TEST7   003674  1271#
TMAP    013020   981   2750#
TMAPA   013034  2754#  2766   2815   2821
TMAPB   013140  2776#  2809   2825
TMAPC   013226  2778   2794#
TMAPD   013272  2780   2793   2795   2808#
TMAPEX  013342  2756   2826#
TNUM  = 000022   824#   832    833#   884    885#   963    964#   997    998#  1089   1090#  1143   1144#
                1272   1273#  1307   1308#  1401   1402#  1467   1468#  1570   1571#  1661   1662#  1777
                1778#  1897   1898#  2083   2084#  2303   2304#  2415   2416#
TPB     001106   618#   647*  2574*
TPCORE  005626  1700   1712#
TPCORX  006276  1817   1829#
TPS     001104   617#   645    903    2572
TRDATA  000540   435#  2847*  2853   2871*  2877
TREG    000766   543#   797    847*   848*   874    876*   898    941    979   1101   1154   1280   1321
                1417   1483   1582   1698   1724*  1732   1735*  1746*  1753   1758*  1815   1843*  1851
                1854*  1862*  1869   1874*  1943   1953   1970   1988   1992*  1997   2011*  2014   2027
                2037   2053*  2055   2131   2147   2165   2169*  2174   2189*  2193   2206   2222   2240*
                2242   2317   2346*  2353   2365*  2372   2435   2450   2453*  2458   2473*  2476   2479
                2489   2504*  2506   2665   2677   2681   2755   2847   2849   2871   2873   2959   3116
                3199   3342
TRFLG   000522   427#  2758*  2779   2781*  2794   2796*
TRP10   004202  1336   1351#
TRP10A  004252  1366   1369#
TRP12   004724  1495   1511#
TRP13   005214  1596   1610#
TRP14   005734  1664   1746#
TRP15   006402  1788   1862#
TRP16   007076  1898   1983#
TRP16A  007102  1984#  1989
TRP17   007762  2088   2160#
TRP17A  007766  2161#  2166
TRP20   010720  2310   2365#
TRP4A   003110  1032   1056#
TRP4B   003116  1039   1063#
TRP4C   003124  1042   1072#
TRP7    004004  1274   1290#
TSTLOC  000532   431#  2870*  2881
TSTX    000510   373    422#   688*   742*  3462*  3480   3488*
TST10   004104  1318   1324#
TST11   004434  1413   1420#
TST12   004650  1479   1487#
TST13   005140  1579   1586#
```

```
TST14   005510          1675    1686#
TST15   006162          1791    1801#
TST4    003002          1009    1014#
TST5    003240          1099    1104#
TST6    003414          1152    1158#
TST7    003750          1278    1283#
TYCOR   000526           429#   2762*   2776*   2783    2798
TYFLG   000524           428#   2760*   2792*   2807*   2812
TYPE  = 104000           709     748     750     759     779     788     807    2555    2751    2772    2774    2786    2788
                        2790    2801    2803    2805    2833    3081    3133    3226    3351    3374#
TYPSAX  015402          3245    3247#
TYPSBX  015412          3243*   3244    3249#
TYPSX   015364          2913    3243#   3250
WAS     000536           434#   1715*   1716    1833*   1834    2017*   2196*   2889
WWP   = 000004           363#   1033    1116    1120    1206    1233    1332    1430    1496    1527    1597    1629    1939
                        1951    1968    1986    2128    2145    2163    2433    2448    2663    2674
WWP16   006666          1923    1932#
WWP16A  006704          1935#   2069
WWP17   007614          2114    2121#
WWP17A  007632          2124#   2257
WWP21   011110          2425#
WWP21A  011124          2428#   2518
XFR1    010252          2086    2118    2258#
XFR2    011520          2524#
$ENDAD  011702           400    2563#
$FILLS  001111           620#
$NULL   001110           619#    639
$TPFLG  001112           621#    624     901     2570
$TYPE   001114           624#   3248    3375
.     = 017772           394#    396#    399#    401#    405#    409#    411#    413#    416#    614#    712     810     817
                         842     849     863     893     904     921     931     984    1026    1073    1081    1110    1124
                        1132    1164    1194    1212    1221    1229    1239    1249    1257    1263    1326    1358    1373
                        1388    1423    1437    1444    1456    1490    1513    1521    1536    1544    1559    1590    1612
                        1621    1637    1651    1666    1678    1695    1717    1728    1749    1779    1811    1835    1847
                        1865    1903    1908    1909    1938    1950    1967    1985    1998    2004    2019    2045    2056
                        2064    2085    2127    2144    2162    2175    2181    2182    2198    2231    2243    2251    2265
                        2269    2272    2314    2334    2341    2349    2368    2432    2447    2459    2466    2507    2515
                        2568    2573    2662    2673    2716    2919    2921    2957    2959    2961    3084    3112    3136
                        3143    3195    3229    3235    3339    3349*   3350    3364    3464    3466    3642#
```

```
EMTDEF     388#   3373   3376   3378   3380   3382   3384   3386   3388   3390
TESTX      390#    832    884    963    997   1089   1143   1272   1307   1401   1467   1570   1661   1777   1897
          2083   2303   2415

. ABS.  017772     000

 ERRORS DETECTED:  0

 CCMFAF,CCMFAF.LST/CRF/SOL=CCMFAF.P11
 RUN-TIME:  4  9  1 SECONDS
 RUN-TIME RATIO: 247/15=16.3
 CORE USED:  9K  (17 PAGES)
```

90N