PDP11/45

COMBINED MS 11, MF11, MA11-P

MD-11-DCMFA-C

EP-DCMFA-C-DL-A

OCT 1976

FICHE 1 OF 1

digital

Made in U.S.A.

```
                                        .REM    *
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DCMFA-C-D
PRODUCT NAME:	COMBINED MS-11 (MOS PARITY) AND MF11-LP,MA11-P (CORE) PARITY MEMORY TESTS (SUPERSEDES DCMS-A-D)
DATE CREATED:	21-DECEMBER-1975
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA
REVISED BY:	PERVEZ ZAKI

```
                                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.

                                4.0     STARTING PROCEDURE

                                4.1     STARTING ADDRESSES

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

                                4.2.1   PROGRAM AND/OR OPERATOR ACTION

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


                                5.0     OPERATING PROCEDURE

                                5.1     OPERATIONAL SWITCH SETTINGS

                                        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 IS 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.
WHEN RUNNING UNDER ACT BANK 0 WILL NOT BE TESTED.

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 ALTERED 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.


                                *
```

H01


```
                                ;MEMORY PARITY TEST
                                ;MAINDEC-11-DCMFA-C
                                ;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                                ;AUTHOR:  JIM KAPADIA
                                        .ABS




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




                                ;SYMBOL DEFINITIONS
000001                          BIT0=1                          ;BIT DEFINITIONS
000002                          BIT1=2
000004                          BIT2=4
000010                          BIT3=10
000020                          BIT4=20
000040                          BIT5=40
000100                          BIT6=100
000200                          BIT7=200
000400                          BIT8=400
001000                          BIT9=1000
002000                          BIT10=2000
004000                          BIT11=4000
010000                          BIT12=10000
020000                          BIT13=20000
040000                          BIT14=40000
100000                          BIT15=100000
000001                          AE=1                            ;ACTION ENABLE
000004                          WWP=4                           ;WRITE WRONG PARITY
077400                          ADRS=77400                      ;ADDRESS OF ERROR
100000                          PERR=100000                     ;PARITY ERROR BIT
177570                          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          015530                          EMTINT
000032          000340                          340
                000046                          .=46
000046          011434                          $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  001226                  JMP     START           ;GO TO START OF PROGRAM
                000210                          .=210
000210          000167  013702                  JMP     RSTLDR          ;GO RESTORE THE LOADERS
                000220                          .=220
000220          000167  001126                  JMP     SCAN            ;SCAN FOR BAD PARITY
                000230                          .=230
000230          000167  001036                  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
```

J01


```
                                                                ;TEST IN SOME SUBTESTS
000534  000000          SHOBE:  0                               ;VALUE EXPECTED
000536  000000          WAS:    0                               ;ACTUAL VALUE FOUND
000540  000000          TROATA: 0
000542  000000          MPROK:  0
000544  000000          PASCNT: 0                               ;PASS COUNT
000546  000000          TBANK:  0
000550  000000          MEMUT:  0
000552  000000          NOKT:   0                               ;SET TO INDICATE NO KT11 PRESENT
000554  000000          HIWORD: 0
000556  000000          LOWFLG: 0
000560  000000          ODDFLG: 0                               ;IF SET INDICATES TESTING HIGH BYTE
                                                                ;OF MEMORY LOCATION
000562  000000          TEMP:   0
000564  000000          MTYFG:  0                               ;SET TO INDICATE MAP OF PARITY MEMORY
                                                                ;ALREADY TYPED
000566  000000          RELOC:  0



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

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

```
000612  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000614  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000616  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000620  172107   MPR3:      172106+1
000622  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000624  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000626  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000630  172111   MPR4:      172110+1
000632  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000634  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000636  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000640  172113   MPR5:      172112+1
000642  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000644  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000646  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000650  172115   MPR6:      172114+1
000652  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000654  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000656  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000660  172117   MPR7:      172116+1
000662  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000664  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000666  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000670  172121   MPR8:      172120+1
000672  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000674  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000676  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000700  172123   MPR9:      172122+1
000702  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000704  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000706  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000710  172125   MPR10:     172124+1
000712  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000714  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000716  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000720  172127   MPR11:     172126+1
000722  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000724  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000726  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000730  172131   MPR12:     172130+1
000732  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000734  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000736  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000740  172133   MPR13:     172132+1
000742  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000744  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000746  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000750  172135   MPR14:     172134+1
000752  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000754  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000756  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
000760  172137   MPR15:     172136+1
000762  000000              0              ;0-64K PARITY MEM UNDER THIS CONTROL
000764  000000              0              ;64-124K PARITY MEM UNDER THIS CONTROL
000766  000000              0              ;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
```

```
551  000770  000000                  TREG:   0                               ;PARITY REGISTER UNDER TEST
552
553
554
555                                  ;INDICATORS FOR CORE OR MOS PARITY REGISTER:
556                                  ;EACH INDICATOR REFERS TO A PARTICULAR PARITY REGISTER.  IF IT IS
557                                  ;A CORE PARITY REGISTER THEN A '1' IS STORED IN THE INDICATOR.
558                                  ;IF IT IS A MOS PARITY REGISTER THEN '-1' GETS STORED.
559                                  ;EX= IF MPR0 (172100) IS FOR CORE AND MPR1 (172102) IS FOR MOS
560                                  ;THEN THE INDICATOR MAP WILL LOOK AS FOLLOWING:
561                                  ;INDC0: 000001
562                                  ;INDC1: 177777
563
564  000772  000000                  INDC0:  0                               ;CORE-MOS PARITY INDICATOR FOR MPR
565  000774  000000                  INDC1:  0                               ;CORE-MOS PARITY IDCICATOR FOR MPR1
566  000776  000000                  INDC2:  0                               ;CORE-MOS PARITY INDICATOR FOR MPR2
567  001000  000000                  INDC3:  0                               ;CORE-MOS PARITY INDICATOR FOR MPR3
568  001002  000000                  INDC4:  0                               ;FOR MPR4
569  001004  000000                  INDC5:  0                               ;FOR MPR5
570  001006  000000                  INDC6:  0                               ;FOR MPR6
571  001010  000000                  INDC7:  0                               ;FOR MPR7
572  001012  000000                  INDC8:  0                               ;FOR MPR8
573  001014  000000                  INDC9:  0                               ;FOR MPR9
574  001016  000000                  INDC10: 0                               ;FOR MPR10
575  001020  000000                  INDC11: 0                               ;FOR MPR11
576  001022  000000                  INDC12: 0                               ;FOR MPR12
577  001024  000000                  INDC13: 0                               ;FOR MPR13
578  001026  000000                  INDC14: 0                               ;FOR MPR14
579  001030  000000                  INDC15: 0                               ;FOR MPR15
580  001032  000000                  RESRVD: 0
581                                                                          ;BIT POSITIONS WHICH ARE RESERVED
582                                                                          ;FOR FUTURE USE IN PARITY REGISTERS
583  001034  070032                  RESVC:  70032                           ;CORE PARITY
584  001036  077772                  RESVM:  77772                           ;MOS PARITY
585
586
587
588
589                                  ;PARITY PATTERNS
590  001040  125325                  PARPAT: 125325                          ;EVEN,ODD BYTES
591  001042  152652                          152652                          ;ODD,EVEN
592  001044  052452                          052452                          ;EVEN,ODD
593  001046  025125                          025125                          ;ODD,EVEN
594  001050  102070                          102070                          ;EVEN,EVEN
595  001052  072527                          072527                          ;ODD,ODD
596  001054  177777                          177777                          ;EVEN,EVEN
597  001056  107030                          107030                          ;ODD,ODD
598  001060  152525                          152525                          ;ODD,EVEN
599  001062  000000                          0                               ;EXTRA PATTERN AREA
600  001064  000000                          0                               ;TERMINATOR, DO NOT USE THIS LOC
601
602
603                                  ;THIS IS A MAP OF THE TOTAL MEMORY PRESENT IN THE SYSTEM.
604  001066  000000                  MEML:   0                               ;0-64K MEM PRESENT IN 4K CONTIGUOUS BLOCKS
605  001070  000000                  MEMH:   0                               ;64-124 MEM PRESENT IN 4K CONTIGOUS BLOCKS
606
```

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



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

        001100                          .=1100
001100  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
```

```
663  001220  000000                  SCNFLG: 0                               ;SCNFLG GETS SET IF USING
664                                                                          ;SCAN ROUTINE (SA=220)
665  001222  000000                  KTSTART:        0
666  001224  000000                  ADRTYP: 0
667  001226  177600                  PDRTAB: 177600
668  001230  172200                          172200
669  001232  172300                  PDREND: 172300
670  001234  172300                  KPDR0:  172300                          ;KERNEL PAGE DESCRIPTOR REGISTER ADDRESSES
671  001236  172302                  KPDR1:  172302
672  001240  172304                  KPDR2:  172304
673  001242  172316                  KPDR7:  172316
674  001244  172340                  KPAR0:  172340                          ;KERNEL PAGE ADDRESS REGISTER ADDRESSES
675  001246  172342                  KPAR1:  172342
676  001250  172344                  KPAR2:  172344
677  001252  172356                  KPAR7:  172356
678  001254  000000                  SPSAV:  0                               ;REGISTER SAVE LOCATIONS
679  001256  000000                  R0SAV:  0
680  001260  000000                  R1SAV:  0
681  001262  000000                  R2SAV:  0
682  001264  000000                  R3SAV:  0
683  001266  000000                  R4SAV:  0
684  001270  000000                  R5SAV:  0
685
686
687
688                                   ;ROUTINE TO RESTART WITHOUT RETYPING MAP AFTER TEST HAS BEEN RUNNING
689  001272  012706  000510          RSTART: MOV     #STKPT,SP               ;SET UP STACK POINTER
690  001276  012767  000001  177260          MOV     #1,MTYFG                ;SET FLAG TO INDICATE MAP HAS BEEN TYPED
691  001304  005067  177234                  CLR     PASCNT                  ;INITIALIZE PASS COUNT
692  001310  012737  015430  000024          MOV     #PWRDN,@#24
693  001316  012737  000340  000026          MOV     #340,@#26
694  001324  005067  177160                  CLR     TSTX
695  001330  005037  177776                  CLR     @#PS                    ;CLEAR PROCESSOR STATUS REGISTER
696  001334  012737  000006  000004          MOV     #6,@#4
697  001342  005037  000006                  CLR     @#6
698  001346  000167  000470                  JMP     BEGIN
699
700
701
702
703                                   ;ROUTINE TO SCAN ALL MEMORY FOR BAD PARITY AND TYPE 18 BIT ADDRESSES OF BAD
704                                   ;LOCATIONS
705  001352  012706  000510          SCAN:   MOV     #STKPT,SP               ;SETUP STACK POINTER
706  001356  005767  177130                  TST     FTITLE                  ;IF TITLE HAS BEEN PRINTED, REGISTERS
707                                                                          ;HAVE ALREADY BEEN LOCATED- GO
708                                                                          ;AND LOCATE IF NOT ALREADY DONE
709  001362  001006                          BNE     SCANB                   ;BRANCH, REGISTERS HAVE ALREADY BEEN LOCATED
710  001364  005267  177630                  INC     SCNFLG                  ;INCREMENT SCNFLG
711  001370  000167  000172                  JMP     START1                  ;GO TO LOCATE THE REGISTERS
712  001374  005067  177620          SCANA:  CLR     SCNFLG                  ;RETURN HERE AFTER LOCATING THE REGISTERS
713  001400  004767  010100          SCANB:  JSR     PC,MAPMEM               ;SETUP MEMORY MAP
714  001404  004767  012602                  JSR     PC,PSCAN                ;SCAN FOR BAD PARITY
715  001410  104000                          TYPE                            ;TYPE MESSAGE "BAD PARITY SCAN COMPLETE"
716  001412  017201                          PSMSG
717  001414  005767  177132                  TST     NOKT
718  001420  001002                          BNE     .+6
```

```
001422  005037  177572          CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
001426  000000                  HALT                    ;END OF PARITY SCAN
001430  000750                  BR      SCAN


                        ;NORMAL STARTUP
001432  012706  000510  START:  MOV     #STKPT,SP       ;SET UP STACK POINTER
001436  005067  177122          CLR     MTYFG           ;CLEAR FLAG WHICH INDICATES MAP TYPED
001442  005067  177076          CLR     PASCNT          ;INITIALIZE PASS COUNT
001446  012737  015430  000024  MOV     #PWRDN,@#24     ;SETUP POWER FAIL RETURN
001454  012737  000340  000026  MOV     #340,@#26
001462  013746  000004          MOV     @#4,-(SP)       ;SAVE THE ERROR VECTOR
001466  013746  000006          MOV     @#6,-(SP)       ;
001472  012737  001506  000004  MOV     #2$,@#4         ;SET UP TIME OUT VECTOR
001500  005777  177374          TST     @SWR            ;TRY TO REFERENCE THE SWITCH REGISTER
001504  000407                  BR      4$              ;BRANCH IF NO TIME OUT
001506  012767  000176  177364  2$:     MOV     #SWREG,SWR      ;POINT TO THE SOFTWARE SWITCH REGISTER
001514  012767  000174  177360          MOV     #DISPREG,DISPLAY ;POINT TO THE SOFTWARE DISPLAY REG.
001522  022626                  CMP     (SP)+,(SP)+     ;RESTORE THE STACK POINTER
001524  012637  000006  4$:     MOV     (SP)+,@#6       ;RESTORE ERROR VECTOR
001530  012637  000004          MOV     (SP)+,@#4       ;
001534  005067  176750          CLR     TSTX
001540  005767  176746  1$:     TST     FTITLE          ;IS TITLE PRINTED YET?
001544  001010                  BNE     START1          ;YES, SKIP OVER
001546  004767  012226          JSR     PC,SAVLDR       ;COPY LOADER TO LOWER 4K
001552  005267  176734          INC     FTITLE          ;SET FLAG
001556  104000                  TYPE                    ;TYPE TITLE "MEMORY PARITY TEST
001560  016746                  MTIT                    ;MAINDEC-11-DCMFA"
001562  104000                  TYPE
001564  017234                  MLDRSV                  ;TYPE "LOADERS SAVED IN BANK 0.
                                                        ;TO RESTORE LOADERS, USE SA 210"


                        ;SEARCH FOR PARITY REGISTERS PRESENT AND TYPE ADDRESSES OF THOSE FOUND
                        ;FAILURE TO LOCATE A REGISTER INDICATES THAT THE ADDRESS TIMED OUT OR THAT
                        ;BITS 5-7 IN THE REGISTER DID NOT SET
001566  104000  START1:  TYPE                    ;TYPE "MEMORY PARITY REGISTERS PRESENT ARE:"
001570  016516          MPRS
001572  005067  176744          CLR     MPROK           ;CLEAR MPR FLAG
001576  012702  000570          MOV     #MPR0,R2        ;SET UP POINTERS
001602  012703  000772          MOV     #INDC0,R3       ;POINTER TO CORE-MOS
001606  012737  001746  000004  MOV     #GMPR8,@#4      ;SET UP TIMEOUT TRAP RETURN
001614  005037  000006          CLR     @#6
001620  042712  000001  GMPRA:  BIC     #1,(2)          ;CLEAR FLAG BIT IN TABLE
001624  005062  000002          CLR     2(R2)           ;INITIALIZE LOCATIONS IN THE TABLE
001630  005062  000004          CLR     4(R2)
001634  005062  000006          CLR     6(R2)
001640  005772  000000          TST     @(2)            ;DOES THIS MPR EXIST? (IF NO, TIMES OUT)
001644  052772  000340  000000  BIS     #340,@(2)       ;YES- IS IT AN MF11-LP OR MA11-P CORE PARITY REG
001652  032772  000340  000000  BIT     #340,@(2)
001660  001414                  BEQ     1$              ;NO, IS IT A MOS-11 PARITY REGISTER?  BRANCH
001662  011267  176626          MOV     (2),TEMPX       ;YES- PRINT REGISTER ADDRESS
```

```
 775  001666  004567  013222                 JSR     R5,OACNV        ;(GET ASCII)
 776  001672  000514                         TEMPX
 777  001674  017326                         MPRCOR
 778  001676  000006                         6
 779  001700  104000                         TYPE                    ;(TYPE ADDRESS)
 780  001702  017326                         MPRCOR
 781  001704  012713  000001                 MOV     #1,(R3)         ;SET INDICATOR FOR CORE PARITY
 782  001710  000413                         BR      2$
 783  001712  011267  176576         1$:     MOV     (2),TEMPX       ;IT IS A MOS REGISTER, PRINT ADDRESS
 784  001716  004567  013172                 JSR     R5,OACNV        ;(GET ASCII)
 785  001722  000514                         TEMPX
 786  001724  017367                         MPRMOS
 787  001726  000006                         6
 788  001730  104000                         TYPE                    ;(TYPE ADDRESS)
 789  001732  017367                         MPRMOS
 790  001734  012713  177777                 MOV     #-1,(R3)        ;SET INDICATOR FOR MOS PARITY
 791  001740  005267  176576         2$:     INC     MPROK           ;SET MPR REGISTER PRESENT FLAG
 792  001744  000403                         BR      GMPRC           ;SKIP NEXT
 793  001746  062626                 GMPR8:  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
 794  001750  052712  000001                 BIS     #1,@R2          ;SET FLAG INDICATING REGISTER NOT PRESENT
 795  001754  062702  000010         GMPRC:  ADD     #10,R2          ;UPDATE POINTER
 796  001760  005723                         TST     (R3)+
 797  001762  022227  000770                 CMP     R2,#TREG        ;DONE YET?
 798  001766  002714                         BLT     GMPRA           ;NO, LOOP
 799  001770  012737  000006  000004         MOV     #6,@#4          ;YES, RESTORE TRAPCATCHER
 800  001776  005767  177216                 TST     SCNFLG          ;ARE YOU IN THE ROUTINE TO SCAN
 801                                                                 ;MEMORY FOR BAD PARITY-(SCAN)
 802  002002  001402                         BEQ     GMPRD           ;NO,BRANCH TO CARRY ON NORMALLY
 803  002004  000167  177364                 JMP     SCANA           ;YES, GO BACK TO THE MEMORY
 804                                                                 ;SCAN ROUTINE
 805  002010  005767  176526         GMPRD:  TST     MPROK           ;ANY PARITY REGISTERS PRESENT?
 806  002014  001012                         BNE     BEGIN           ;YES- GO TEST CONTROLS PRESENT
 807  002016  104000                 nOREG:  TYPE                    ;NO- TYPE "NO PARITY REGISTER FOUND"
 808  002020  016701                         MTR
 809  002022  005737  000042                 TST     @#42            ;LOADED BY MONITOR?
 810  002026  001402                         BEQ     .+6             ;NO, BRANCH
 811  002030  000167  007370                 JMP     LOGICAL         ;YES-  EXIT, NO REGISTERS PRESENT
 812  002034  000000                         HALT                    ;NO REGISTERS TO TEST
 813  002036  000167  177370                 JMP     START           ;IF CONTINUED, TRY AGAIN
 814
 815  002042  012767  002066  014102 BEGIN:  MOV     #TEST1+2,RETURN ;SETUP SCOPE RETURN
 816  002050  012767  000100  014070         MOV     #100,IMAX       ;MAXIMUM ITERATION COUNT
 817  002056  012737  000006  000004         MOV     #6,@#4          ;RESTORE TRAPCATCHER IN TIMEOUT VECTOR
 818
 819
 820
 821                                 ;*****************************************************************
 822                                 ;SHOW THAT BITS 0,2,5-11, AND 15 OF EACH CORE PARITY REGISTER PRESENT
 823                                 ;CAN BE SET AND CLEARED.  BITS 0,2,15 OF EACH MOS PARITY REGISTER
 824                                 ;PRESENT CAN BE SET AND CLEARED
 825                                 ;*****************************************************************
 826  002064  104001                 TEST1:  SCOPE
 827  002066  012777  000001  177006         MOV     #1,@DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY
 828  002074  012700  000570                 MOV     #MPR0,R0        ;LOAD ADDRESS OF TABLE INTO R0
 829  002100  012704  000772                 MOV     #INDC0,R4       ;LOAD ADDRESS OF INDICATOR IN R4
 830  002104  032710  000001         1$:     BIT     #1,@R0          ;IS THIS REGISTER PRESENT?
```

```
831  002110  001042                          BNE     4$                      ;NO- BRANCH TO GET NEXT ADDRESS
832  002112  011001                          MOV     @R0,R1                  ;YES- LOAD R1 WITH ADDRESS OF
833                                                                          ;PARITY REGISTER
834  002114  022714  000001                  CMP     #1,(R4)                 ;IS THIS REGISTER CORE?
835  002120  001004                          BNE     5$              ;NO
836  002122  016767  176706  176702          MOV     RESVC,RESRVD            ;YES, CORE. STORE RESERVED BITS
837  002130  000403                          BR      .+10
838  002132  016767  176700  176672  5$:     MOV     RESVM,RESRVD            ;MOS, STORE RESERVED BITS
839  002140  012702  000001                  MOV     #1,R2                   ;LOAD R2 WITH VALUE OF FIRST BIT
840                                                                          ;TO BE TESTED
841  002144  005011                          CLR     @R1                     ;INITIALIZE PARITY REGISTER
842  002146  011167  176616                  MOV     @R1,TREG                ;READ CONTENTS OF PARITY REGISTER
843  002152  046767  176654  176610          BIC     RESRVD,TREG             ;CLEAR BITS WHICH ARE RESERVED
844  002160  001401                          BEQ     .+4                     ;CHECK OTHER BITS- BRANCH IF OK
845  002162  104002                          ERROR                           ;CLEAR INSTRUCTION DID NOT INITIALIZE
846                                                                          ;ALL USED BITS IN PARITY REGISTER
847                                                                          ;TO ZERO (R1 CONTAINS ADDRESS OF
848                                                                          ;FAILING REGISTER)
849  002164  030267  176642          2$:     BIT     R2,RESRVD               ;IS THIS BIT RESERVED?
850  002170  001010                          BNE     3$                      ;YES- DON'T TEST IT SINCE IT
851                                                                          ;MAY BE ZERO OR ONE
852  002172  010211                          MOV     R2,@R1                  ;NO- SET THIS BIT IN THE PARITY REGISTER
853  002174  011103                          MOV     @R1,R3                  ;READ AND SAVE CONTENTS OF PARITY REGISTER
854  002176  005011                          CLR     @R1                     ;CLEAR PARITY REGISTER
855  002200  046703  176626                  BIC     RESRVD,R3               ;CLEAR BIT LOCATIONS THAT ARE
856                                                                          ;RESERVED
857  002204  020203                          CMP     R2,R3                   ;CHECK REST
858  002206  001401                          BEQ     .+4                     ;BRANCH IF OK
859  002210  104002                          ERROR                           ;PARITY REGISTER WHOSE ADDRESS IS IN R1
860                                                                          ;WAS INCORRECT AFTER THE ... IN R2
861                                                                          ;WAS WRITTEN INTO IT. ACTUAL ...S
862                                                                          ;(WITH UNUSED BITS CLEARED) IS IN R3
863  002212  006302                  3$:     ASL     R2                      ;ROTATE BIT TO BE TESTED
864  002214  103363                          BCC     2$                      ;IF NOT DONE WITH ALL BIT POSITIONS
865                                                                          ;GO TEST THIS ONE
866  002216  062700  000010          4$:     ADD     #10,R0                  ;MOVE R0 TO POINT TO NEXT POSSIBLE ADDRESS
867  002222  005724                          TST     (R4)+
868                                                                          ;OF A PARITY REGISTER
869  002224  020027  000770                  CMP     R0,#TREG                ;AT END OF TABLE?
870  002230  002725                          BLT     1$                      ;NO, BRANCH
871  002232  005067  176532                  CLR     TREG
872
873                                  ;*****************************************************************
874
875                                  ;SHOW THAT RESET CLEARS BITS 0,2, AND 15 OF EACH PARITY REGISTER
876                                  ;PRESENT.
877                                  ;*****************************************************************
878  002236  104001                  TEST2:  SCOPE
879  002240  012777  000002  176634          MOV     #2,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
880  002246  005067  013674                  CLR     IMAX                    ;DON'T ITERATE TEST
881  002252  012700  000570                  MOV     #MPR0,R0                ;LOAD POINTER
882  002256  012703  000772                  MOV     #INDCO,R3               ;POINTER TO INDICATOR
883  002262  032710  000001          1$:     BIT     #1,@R0                  ;IS THIS PARITY REGISTER PRESENT?
884  002266  001012                          BNE     5$                      ;NO BRANCH
885  002270  022713  000001                  CMP     #1,(R3)                 ;IS THIS CORE OR MOS PARITY REGISTER?
886  002274  001404                          BEQ     6$                      ;NO- BRANCH
```

```
887  002276  012770  100015  000000          MOV     #100015,@(R0)           ;MOS-SET ALL DEFINED BITS TO 1
888  002304  000403                          BR      .+10
889  002306  012770  107745  000000  6$:     MOV     #107745,@(R0)           ;CORE- SET ALL DEFINED BITS TO 1
890  002314  062700  000010          5$:     ADD     #10,R0                  ;MOVE POINTER TO POINT TO NEXT APR ADDRESS
891  002320  005723                          TST     (R3)+                   ;INCREMENT POINTER TO INDICATOR
892                                                                          ;OF A PARITY REGISTER
893  002322  020027  000770                  CMP     R0,#TREG                ;AT END OF TABLE?
894
895  002326  002755                          BLT     1$                      ;NO- CONTINUE
896  002330  105767  176556                  TSTB    @TPFLG                  ;YES- TERMINAL AVAILABLE?
897  002334  001003                          BNE     4$                      ;NO- BRANCH
898  002336  105777  176542                  TSTB    @TPS                    ;YES- WAIT FOR TERMINAL TO FINISH
899  002342  100375                          BPL     .-4
900  002344  000005                  4$:     RESET                           ;ISSUE INIT
901  002346  012700  000570                  MOV     #MPR0,R0                ;LOAD ADDRESS OF THE TABLE
902  002352  012703  000772                  MOV     #INDC0,R3               ;POINTER TO INDICATOR
903  002356  032710  000001          2$:     BIT     #1,@R0                  ;IS THIS PARITY REGISTER PRESENT?
904  002362  001030                          BNE     3$                      ;NO- BRANCH
905  002364  022713  000001                  CMP     #1,(R3)                 ;IS THIS A CORE PAR REGISTER?
906  002370  001012                          BNE     7$                      ;NO, BRANCH
907  002372  017002  000000                  MOV     @(R0),R2                ;YES, GET CONTENTS OF REGISTER
908  002376  005070  000000                  CLR     @(R0)                   ;MAKE SURE THAT MMP AND ME ARE CLEAR
909  002402  042702  077772                  BIC     #77772,R2               ;MASK RESERVED BITS FOR CORE PAR
310                                                                          ;-ITY REGISTER. BITS 5-11(ADDRS
911                                                                          ; BITS) ARE ALSO MASKED
912  002406  005702                          TST     R2                      ;CHECK, IF REST WERE CLEARED
913  002410  001401                          BEQ     .+4
914  002412  104002                          ERROR                           ;CORE PARITY REGISTER WHOSE ADDRESS IS
915                                                                          ;POINTED TO BY R0 WAS INCORRECT
916                                                                          ;AFTER A RESET WAS ISSUED- CONTENTS
917                                                                          ;SAVED IN R2 WITH UNUSED BITS MASKED
918  002414  000411                          BR      3$-4
919  002416  017002  000000          7$:     MOV     @(R0),R2                ;MOS, GET CONTENTS OF REGISTER
920  002422  005070  000000                  CLR     @(R0)                   ;MAKE SURE THAT MMP ARE ARE CLEAR
921  002426  046702  176404                  BIC     RESVM,R2                ;MASK RESERVED BITS FOR MOS PAR REG
922  002432  005702                          TST     R2                      ;CHECK REST
923  002434  001401                          BEQ     .+4                     ;RESET DID CLEAR ALL BITS
924  002436  104002                          ERROR                           ;MOS PARITY REGISTER WHOOSE ADDRESS IS
925                                                                          ;POINTED BY R0 WAS INCORRECT.  AFTER
926                                                                          ;ISSUING RESET CONTENTS OF PAR REG
927                                                                          ;WERE AS SHOWN IN R2(UNUSED BITS
928                                                                          ;HAVE BEEN MASKED)
929  002440  005070  000000                  CLR     @(R0)                   ;REINITIALIZE PARITY REGISTER
930  002444  062700  000010          3$:     ADD     #10,R0                  ;MOVE POINTER TO POINT TO ADDRESS
931                                                                          ;OF NEXT REGISTER
932  002450  005723                          TST     (R3)+                   ;INCREMENT POINTER TO INDICATOR
933  002452  020027  000770                  CMP     R0,#TREG                ;DONE?
934  002456  002737                          BLT     2$                      ;NO- LOOP
935
936
937                                  ;*************************************************************************
938                                  ;MAP CORRESPONDENCE BETWEEN PARITY REGISTERS AND MEMORY, AND TYPE RESULTS
939                                  ;NOTE THAT IF PARITY MEMORY IS NOT LOCATED CORRECTLY BY THIS SUBTEST
940                                  ;IT IS DUE TO ONE OF THE FOLLOWING FAILURES:
941                                  ;       -SETTING WRITE WRONG PARITY DID NOT CAUSE BAD PARITY TO BE WRITTEN
942                                  ;       -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 "MMRO" 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.
                                        ;*****************************************************************************
002460  104001                  TEST3:  SCOPE
002462  012777  000003  176412          MOV     #3,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
002470  005767  176070                  TST     MTYFG                   ;IF MAPPING HAS ALREADY BEEN DONE
002474  001044                          BNE     TEST4                   ;SKIP SUBTEST
002476  004767  011006                  JSR     %7,CLRPAR
002502  004767  006776                  JSR     %7,MAPMEM               ;MAP MEMORY
002506  004767  007320                  JSR     %7,MAPREG               ;FIND PARITY MEMORY AND CORRESPONDING
                                                                        ;REGISTERS USING WRITE WRONG PARITY
                                                                        ;WITHOUT ACTION ENABLE SET
002512  005067  176354          CONT3:  CLR     PMEML                   ;INITIALIZE LOCATIONS INDICATING
002516  005067  176352                  CLR     PMEMH                   ;TOTAL PARITY MEMORY PRESENT
002522  012701  000570                  MOV     #MPRO,R1
002526  032711  000001          1$:     BIT     #1,@R1
002532  001006                          BNE     2$
002534  056167  000002  176330          BIS     2(R1),PMEML             ;FLAG EXISTING PARITY MEMORY (LOW 64K)
002542  056167  000004  176324          BIS     4(R1),PMEMH             ;FLAG EXISTING PARITY MEMORY (HIGH 64K)
002550  062701  000010          2$:     ADD     #10,R1
002554  020127  000770                  CMP     R1,#TREG
002560  103762                          BLO     1$
002562  004767  007760                  JSR     %7,TMAP                 ;TYPE MAP
002566  005267  175772                  INC     MTYFG                   ;INDICATE MAPPING DONE
002572  032777  001000  176300          BIT     #BIT9,@SWR              ;SWITCH 9 SET?
002600  001402                          BEQ     .+6                     ;NO- BRANCH
002602  000000                          HALT                            ;YES- SWITCH 9 SET INDICATING HALT
                                                                        ;AFTER TYPING PARITY MEMORY MAP
002604  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
                                        ;*****************************************************************************
002606  104001                  TEST4:  SCOPE
002610  012777  000004  176264          MOV     #4,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
002616  012767  000100  013322          MOV     #100,IMAX
002624  004767  010660                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
002630  005767  175716                  TST     NOKT                    ;KT11 PRESENT?
002634  001004                          BNE     1$                      ;NO-BRANCH
002636  004767  010546                  JSR     %7,NRALL                ;YES-MAP ALL PAGES NON RESIDENT
002642  004767  010712                  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
002646  012700  000570          1$:     MOV     #MPRO,R0                ;SETUP TO FIND REGISTERS PRESENT
```

```
  999  002652  012702  000772                    MOV     #INDCO,R2
 1000  002656  032710  000001          LOOP4:    BIT     #1,@R0          ;IS THIS REGISTER PRESENT?
 1001  002662  001405                             BEQ     TST4            ;YES-BRANCH TO TEST IT
 1002  002664  062700  000010          LOP4:     ADD     #10,R0          ;NO-CHECK FOR ANOTHER ONE
 1003  002670  005722                             TST     (R2)+           ;INCREMENT PTR%LT
 1004  002672  103771                             BLO     LOOP4
 1005  002674  000457                             BR      DONE4           ;BRANCH WHEN ALL REGISTERS HAVE BEEN TESTED
 1006  002676  004767  010722          TST4:     JSR     %7,LOCATM       ;LOCATE MEMORY CORRESPONDING TO
 1007                                                                    ;THIS REGISTER- IF NO KT11, R1 SHOULD
 1008                                                                    ;BE RETURNED CONTAINING THE ADDRESS
 1009                                                                    ;OF THE 1ST LOCATION CONTROLLED BY THIS
 1010                                                                    ;REGISTER (INCLUDING EXTERNAL INTERLEAVE
 1011                                                                    ;OFFSET IF NEEDED)
 1012                                                                    ;IF KT11 IS PRESENT, R1 SHOULD BE RETURNED
 1013                                                                    ;POINTING TO THE 1ST LOCATION CONTROLLED
 1014                                                                    ;BY THIS REGISTER, MAPPED THRU KERNEL
 1015                                                                    ;PAGE 1. KERNEL PAGE 1 SHOULD BE
 1016                                                                    ;MAPPED TO THE CORRECT BANK
 1017  002702  032701  000001                    BIT     #1,R1           ;IS ERROR RETURN INDICATED?
 1018  002706  001403                             BEQ     .+10            ;NO- BRANCH
 1019  002710  104002                             ERROR                   ;MAP INDICATES NO PARITY MEMORY
 1020                                                                    ;IS CONTROLLED BY THIS REGISTER
 1021                                                                    ;R0 POINTS TO THE ADDRESS OF THE
 1022                                                                    ;PARITY REGISTER
 1023  002712  000167  177746                    JMP     LOP4
 1024  002716  012737  003004  000114          MOV     #TRP4A,@#114    ;SETUP PARITY TRAP RETURN
 1025  002724  012770  000004  000000          MOV     #4#P,2(R0)      ;SET WRITE WRONG PARITY
 1026  002732  011111                           MOV     @R1,@R1         ;WRITE CONTENTS OF LOCATION WITH
 1027                                                                    ;WRONG PARITY
 1028  002734  005070  000000                    CLR     @(R0)           ;CLEAR PARITY REGISTER
 1029  002740  005711                           TST     @R1             ;READ BAD PARITY WITH ACTION ENABLE
 1030                                                                    ;CLEARED- NO TRAP EXPECTED
 1031  002742  012737  003012  000114          MOV     #TRP4B,@#PARVEC ;CHANGE PARITY TRAP RETURN
 1032  002750  052770  000001  000000          BIS     #AE,@(R0)       ;SET ACTION ENABLE WITH PARITY ERROR
 1033                                                                    ;ALREADY SET- SHOULDN'T TRAP YET
 1034  002756  012737  003020  000114          MOV     #TRP4C,@#PARVEC ;CHANGE PARITY TRAP RETURN
 1035  002764  005711                           TST     @R1             ;READ LOCATION AGAIN- SHOULD GET
 1036                                                                    ;A PARITY TRAP DUE TO READING BAD
 1037                                                                    ;PARITY WITH ACTION ENABLE SET
 1038  002766  104002                             ERROR                   ;NO PARITY TRAP AFTER READING LOCATION
 1039                                                                    ;WHICH SHOULD CONTAIN BAD PARITY-
 1040                                                                    ;R1 CONTAINS ADDRESS OF MEMORY LOCATION
 1041                                                                    ;(VIRTUAL, THRU KERNEL PAGE 1, IF KT11
 1042                                                                    ;PRESENT). R0 POINTS TO THE ADDRESS OF
 1043                                                                    ;THE PARITY REGISTER IN WHICH AE WAS SET
 1044  002770  005070  000000          CONT4:    CLR     @(R0)           ;CLEAR PARITY REGISTER
 1045  002774  005511                           ADC     @R1             ;CLEAR BAD PARITY
 1046  002776  005070  000000                    CLR     @(R0)           ;CLEAR PARITY ERROR BIT
 1047  003002  000730                             BR      LOP4            ;GO TO TEST NEXT REGISTER
 1048  003004  104002                  TRP4A:    ERROR                   ;PARITY TRAP OCCURRED WITH ACTION
 1049                                                                    ;ENABLE CLEAR- R0 POINTS TO THE ADDRESS
 1050                                                                    ;OF THE PARITY REGISTER UNDER TEST
 1051                                                                    ;R1 CONTAINS THE ADDRESS OF THE MEMORY
 1052                                                                    ;UNDER TEST (VIRTUAL IF KT11 IS PRESENT)
 1053  003006  022626                             CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
 1054  003010  000767                             BR      CONT4
```

```
1055  003012  104002                  TRP4B:  ERROR                           ;PARITY TRAP OCCURRED WHEN ACTION
1056                                                                          ;ENABLE WAS SET WITH PARITY ERROR
1057                                                                          ;ALREADY SET
1058                                                                          ;R0 POINTS TO THE ADDRESS OF THE
1059                                                                          ;PARITY REGISTER UNDER TEST
1060                                                                          ;R1 CONTAINS THE MEMORY ADDRESS UNDER
1061                                                                          ;TEST (VIRTUAL IF KT11 IS PRESENT)
1062  003014  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1063  003016  000764                          BR      CONT4
1064  003020  005770  000000          TRP4C:  TST     @(R0)                   ;ERROR BIT SET AFTER PARITY TRAP?
1065  003024  100401                          BMI     .+4                     ;YES- BRANCH
1066  003026  104002                          ERROR                           ;ERROR BIT NOT SET AFTER PARITY
1067                                                                          ;TRAP- R0 POINTS TO THE ADDRESS
1068                                                                          ;OF THE PARITY REGISTER UNDER TEST
1069  003030  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1070  003032  000756                          BR      CONT4
1071  003034  012737  000116  000114  DONE4:  MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
1072  003042  005767  175504                  TST     NOKT
1073  003046  001002                          BNE     .+6
1074  003050  005037  177572                  CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1075
1076
1077                                          ;*************************************************************************
1078                                          ;SHOW THAT READING GOOD PARITY AFTER BAD PARITY DOESN'T CLEAR PARITY ERROR BIT
1079                                          ;*************************************************************************
1080  003054  104001                  TEST5:  SCOPE
1081  003056  012777  000005  176016          MOV     #5,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
1082  003064  004767  010420                  JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1083  003070  005767  175456                  TST     NOKT                    ;KT11 PRESENT?
1084  003074  001004                          BNE     1$                      ;NO, BRANCH
1085  003076  004767  010306                  JSR     PC,MRALL                ;YES, MAP ALL PAGES NON-RESIDENT
1086  003102  004767  010452                  JSR     PC,MAP1                 ;THEN MAP KERNEL 0 TO BANK 0. MAP KERNEL
1087                                                                          ;7 TO THE EXTERNAL BANK. SET KERNEL
1088                                                                          ;0,1, AND 7 RW, AND TURN ON KT11
1089  003106  012700  000570          1$:     MOV     #MPR0,R0                ;SETUP TO FIND REGISTERS PRESENT
1090  003112  032710  000001          LOOP5:  BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
1091  003116  001406                          BEQ     TST5                    ;YES- BRANCH TO TEST IT
1092  003120  062700  000010          LOP5:   ADD     #10,R0                  ;NO- CHECK FOR ANOTHER ONE
1093  003124  020027  000770                  CMP     R0,#TREG
1094  003130  103770                          BLO     LOOP5
1095  003132  000431                          BR      DONE5
1096  003134  004767  010464          TST5:   JSR     PC,LOCATM               ;EXIT WHEN ALL REGISTERS HAVE BEEN TESTED
1097                                                                          ;LOCATE MEMORY CORRESPONDING TO  THIS
1098                                                                          ;REGISTER- R1 WILL BE RETURNED CONTAINING
1099                                                                          ;THE ADDRESS OF THE FIRST LOCATION
1100                                                                          ;CONTROLLED BY THIS REGISTER (MAPPED
1101  003140  032701  000001                  BIT     #1,R1                   ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1102  003144  001403                          BEQ     .+10                    ;IS ERROR RETURN INDICATED?
1103  003146  104002                          ERROR                           ;NO- BRANCH
1104                                                                          ;MAP INDICATES NO PARITY MEMORY IS
1105                                                                          ;CONTROLLED BY THIS REGISTER. R0
1106                                                                          ;POINTS TO THE ADDRESS OF THE
1107  003150  000167  177744                  JMP     LOP5                    ;PARITY REGISTER
1108  003154  012770  000004  000000          MOV     #WWP,@(R0)              ;SET WWP IN THIS REGISTER
1109  003162  011111                          MOV     @R1,@R1                 ;WRITE CONTENTS OF LOCATION WITH
1110                                                                          ;WRONG PARITY
```

```
1111  003164  005711                          TST     @R1                     ;DETECT WRONG PARITY
1112  003166  042770  000004  000000          BIC     #WWP,@(R0)              ;CLEAR WWP IN PARITY REGISTER
1113  003174  011111                          MOV     @R1,@R1                 ;RESTORE GOOD PARITY
1114  003176  005711                          TST     @R1                     ;REREAD LOCATION
1115  003200  005770  000000                  TST     @(R0)                   ;READ CONTENTS OF PARITY REGISTER
1116  003204  100401                          BMI     .+4                     ;BRANCH IF PARITY ERROR IS STILL SET
1117  003206  104002                          ERROR                           ;PARITY ERROR BIT CLEARED BY READING
1118                                                                          ;GOOD PARITY (OR POSSIBLY WHILE DOING
1119                                                                          ;THE BIC TO CLEAR WWP). R0 POINTS TO
1120                                                                          ;THE PARITY REGISTER ADDRESS.
1121  003210  005070  000000                  CLR     @(R0)                   ;CLEAR THE PARITY REGISTER
1122  003214  000741                          BR      LOP5
1123  003216  005767  175330          DONE5:  TST     NOKT
1124  003222  001002                          BNE     .+6
1125  003224  005037  177572                  CLR     @#SR0                   ;TURN OFF KT11 IF PRESENT
1126
1127
1128                                  ;*****************************************************************
1129                                  ;SHOW THAT PARITY GENERATE AND DETECT LOGIC WORKS CORRECTLY FOR EACH BYTE
1130                                  ;SHOW THAT WRITE WRONG PARITY WORKS FOR HIGH AND LOW BYTES
1131                                  ;SHOW THAT WRITING INTO LOCATION WHEN WRITE WRONG PARITY IS NOT SET
1132                                  ;RESTORES GOOD PARITY
1133                                  ;*****************************************************************
1134  003230  104001                  TEST6:  SCOPE
1135  003232  012777  000006  175642          MOV     #6,@DISPLAY             ;LOAD THE TEST NUMBER INTO THE DISPLAY
1136  003240  004767  010244                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1137  003244  005767  175302                  TST     NOKT                    ;KT11 PRESENT?
1138  003250  001004                          BNE     1$                      ;NO, BRANCH
1139  003252  004767  010132                  JSR     %7,MRALL                ;YES- MAP IT (KERNEL 0 TO BANK 0, RW;
1140  003256  004767  010276                  JSR     %7,MAP1                 ;KERNEL 7 TO EXTERNAL BANK, RW; KERNEL 1 RW)
1141                                                                          ;AND TURN IT ON
1142  003262  012700  000570          1$:     MOV     #MPR0,R0                ;SETUP TO FIND REGISTERS PRESENT
1143  003266  032710  000001          LOOP6:  BIT     #1,@R0                  ;IS THIS REGISTER PRESENT?
1144  003272  001406                          BEQ     TST6                    ;YES- BRANCH TO TEST IT
1145  003274  062700  000010          LOP6:   ADD     #10,R0                  ;NO- CHECK FOR ANOTHER ONE
1146  003300  020027  000770                  CMP     R0,#TREG
1147  003304  103770                          BLO     LOOP6
1148  003306  000523                          BR      DONE6                   ;BRANCH TO DONE IF ALL REGISTERS
1149                                                                          ;HAVE BEEN TESTED
1150  003310  004767  010310          TST6:   JSR     %7,LOCATM               ;LOCATE MEMORY CORRESPONDING TO
1151                                                                          ;THIS REGISTER- R1 SHOULD BE RETURNED
1152                                                                          ;CONTAINING THE ADDRESS OF THE FIRST
1153                                                                          ;LOCATION CONTROLLED BY THIS REGISTER
1154                                                                          ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 PRESENT)
1155  003314  032701  000001                  BIT     #1,R1                   ;IF NO MEMORY WAS FOUND TO CORRESPOND
1156  003320  001403                          BEQ     .+10                    ;ODD ADDRESS IS RETURNED-BRANCH IF OK
1157  003322  104002                          ERROR                           ;MAP INDICATES NO PARITY MEMORY IS
1158                                                                          ;CONTROLLED BY THIS REGISTER
1159                                                                          ;R0 POINTS TO THE ADDRESS OF THE
1160                                                                          ;PARITY REGISTER
1161  003324  000167  177744                  JMP     LOP6                    ;AFTER ERROR, CHECK FOR NEXT REGISTER
1162
1163                                  ;FIRST SHOW THAT IF THE PARITY REGISTER IS CLEARED INITIALLY,
1164                                  ;PARITY ERROR DOESN'T SET
1165  003330  005011                          CLR     @R1                     ;INITIALIZE LOCATION UNDER TEST
1166  003332  005070  000000                  CLR     @(R0)                   ;INITIALLY CLEAR PARITY REGISTER
```

```
1167  003336  005002                          CLR     R2              ;R2 CONTAINS VALUE TO BE LOADED
1168                                                                  ;INTO MEMORY
1169  003340  110211                    1$:   MOVB    R2,@R1          ;WRITE VALUE INTO LOW BYTE
1170  003342  005711                          TST     @R1             ;READ WORD TO CHECK PARITY
1171  003344  005770  000000                  TST     @(R0)           ;CHECK PARITY REGISTER
1172  003350  100006                          BPL     5$              ;BRANCH IF ERROR NOT SET
1173  003352  104002                          ERROR                   ;PARITY ERROR SET WHEN VALUE IN R2 WAS
1174                                                                  ;WRITTEN AND READ BACK FROM LOW BYTE OF
1175                                                                  ;LOCATION WHOSE ADDRESS IS CONTAINED IN
1176                                                                  ;R1 (WMP WAS NOT SET)
1177  003354  005070  000000                  CLR     @(R0)           ;CLEAR ERROR BIT
1178  003360  005011                          CLR     @R1             ;REINITIALIZE TEST LOCATION
1179  003362  005002                          CLR     R2              ;REINITIALIZE VALUE TO BE USED
1180  003364  000402                          BR      2$
1181  003366  105202                    5$:   INCB    R2              ;INCREMENT VALUE TO BE LOADED
1182  003370  001363                          BNE     1$              ;LOOP UNTIL ALL VALUES HAVE BEEN USED
1183  003372  110261  000001            2$:   MOVB    R2,1(R1)        ;WRITE ALL VALUES INTO HIGH BYTE
1184  003376  005711                          TST     @R1             ;READ WORD TO CHECK PARITY
1185  003400  005770  000000                  TST     @(R0)           ;CHECK PARITY REGISTER
1186  003404  100002                          BPL     .+6             ;BRANCH IF ERROR NOT SET
1187  003406  104002                          ERROR                   ;PARITY ERROR SET WHEN VALUE IN R2
1188                                                                  ;WAS WRITTEN AND READ BACK FROM HIGH BYTE
1189                                                                  ;OF LOCATION WHOSE ADDRESS IS CONTAINED
1190  003410  000402                          BR      6$              ;IN R1 (WMP WAS NOT SET)
1191  003412  105202                          INCB    R2              ;INCREMENT VALUE TO BE LOADED
1192  003414  001366                          BNE     2$              ;LOOP UNTIL ALL VALUES HAVE BEEN USED
1193
1194                                    ;TEST PARITY GENERATE AND DETECT LOGIC BY SETTING WRITE WRONG PARITY AND
1195                                    ;WRITING EACH POSSIBLE VALUE TO THE LOW BYTE, THEN TO THE HIGH BYTE
1196  003416  005011                    6$:   CLR     @R1             ;INITIALIZE LOCATION UNDER TEST
1197  003420  005002                          CLR     R2              ;INITIALIZE VALUE TO BE WRITTEN
1198  003422  012770  000004  000000    3$:   MOV     #WMP,@(R0)      ;SET WRITE WRONG PARITY
1199  003430  110211                          MOVB    R2,@R1          ;WRITE WRONG PARITY IN LOW BYTE
1200  003432  005070  000000                  CLR     @(R0)           ;CLEAR WRITE WRONG PARITY, AND CLEAR
1201                                                                  ;PARITY ERROR IF SET
1202  003436  005711                          TST     @R1             ;READ BACK WRONG PARITY
1203  003440  005770  000000                  TST     @(R0)           ;PARITY ERROR SET?
1204  003444  100402                          BMI     .+6             ;YES-BRANCH
1205  003446  104002                          ERROR                   ;PARITY ERROR DID NOT SET WHEN THE
1206                                                                  ;LOCATION UNDER TEST WAS WRITTEN
1207                                                                  ;AND READ BACK WITH WRITE WRONG PARITY
1208                                                                  ;SET. R0 POINTS TO ADDRESS OF PARITY
1209                                                                  ;REGISTER. R1 CONTAINS ADDRESS OF LOCATION
1210                                                                  ;BEING TESTED (VIRTUAL, THRU KERNEL
1211                                                                  ;PAGE 1 IF KT11 IS PRESENT). R2
1212                                                                  ;CONTAINS THE VALUE WRITTEN
1213  003450  000402                          BR      .+6             ;EXIT LOOP AFTER ERROR
1214  003452  105202                          INCB    R2              ;INCREMENT DATA
1215  003454  001362                          BNE     3$              ;LOOP TILL DONE WITH ALL VALUES
1216  003456  005011                          CLR     @R1             ;REINITIALIZE LOCATION TO CLEAR BAD PARITY
1217  003460  005070  000000                  CLR     @(R0)           ;CLEAR ERROR IF SET
1218  003464  005711                          TST     @R1             ;READ LOCATION, WHICH SHOULD NOW HAVE
1219                                                                  ;GOOD PARITY
1220  003466  005770  000000                  TST     @(R0)           ;PARITY ERROR SET?
1221  003472  100001                          BPL     .+4             ;NO, BRANCH
1222  003474  104002                          ERROR                   ;GOOD PARITY WAS NOT RESTORED BY
```

```
1223
1224
1225  003476  012770  000004  000000  4$:     MOV     #WWP,@(R0)      ;WRITING INTO THE LOCATION WITH
1226  003504  110261  000001                  MOVB    R2,1(R1)        ;WRITE WRONG PARITY CLEARED
1227  003510  005070  000000                  CLR     @(R0)           ;SET WRITE WRONG PARITY
1228                                                                  ;WRITE WRONG PARITY IN HIGH BYTE
1229  003514  005711                          TST     @R1             ;CLEAR WRITE WRONG PARITY AND PARITY
1230  003516  005770  000000                  TST     @(R0)           ;ERROR IF SET
1231  003522  100402                          BMI     .+6             ;READ BACK WRONG PARITY
1232  003524  104002                          ERROR                   ;PARITY ERROR SET?
1233                                                                  ;YES-BRANCH
1234                                                                  ;PARITY ERROR DID NOT SET WHEN THE LOCATION
1235                                                                  ;UNDER TEST WAS WRITTEN AND READ BACK WITH
1236                                                                  ;WRITE WRONG PARITY SET, R0 POINTS
1237                                                                  ;TO THE ADDRESS OF THE PARITY REGISTER.
1238                                                                  ;THE VALUE IN R2 WAS WRITTEN INTO THE HIGH
1239                                                                  ;BYTE OF THE LOCATION WHOSE ADDRESS IS IN R1
1240                                                                  ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 PRESENT)
1241  003526  000402                          BR      .+6             ;THEN THE PARITY REGISTER WAS
1242  003530  105202                          INCB    R2              ;CLEARED AND THE WORD WAS READ BACK
1243  003532  001361                          BNE     4$              ;EXIT LOOP AFTER ERROR
1244  003534  005011                          CLR     @R1             ;INCREMENT DATA
1245  003536  005070  000000                  CLR     @(R0)           ;LOOP TILL DONE WITH ALL VALUES
1246  003542  005711                          TST     @R1             ;CLEAR BAD PARITY IN TEST LOCATION
1247                                                                  ;CLEAR PARITY ERROR BIT IF SET
1248  003544  005770  000000                  TST     @(R0)           ;READ LOCATION, WHICH SHOULD NOW
1249  003550  100001                          BPL     .+4             ;HAVE GOOD PARITY
1250  003552  104002                          ERROR                   ;CHECK PARITY ERROR BIT
1251                                                                  ;BRANCH IF CLEAR
1252                                                                  ;WRITING INTO LOCATION WHEN WRITE
1253  003554  000647                          BR      LOP6            ;WRONG PARITY WAS NOT SET DID NOT
1254  003556  005767  174770          DONE6:  TST     NOKT            ;WRITE GOOD PARITY
1255  003562  001002                          BNE     .+6             ;GO CHECK FOR ANOTHER PARITY REGISTER
1256  003564  005037  177572                  CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1257
1258
1259
1260                                          ;*************************************************************************
1261                                          ;SHOW THAT SETTING PARITY ERROR AFTER SETTING ACTION ENABLE WON'T CAUSE A TRAP
1262                                          ;*************************************************************************
1263  003570  104001          TEST7:  SCOPE
1264  003572  012777  000007  175302          MOV     #7,@DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY
1265  003600  004767  007704                  JSR     %7,CLRPAR       ;INITIALLY CLEAR ALL PARITY REGISTERS
1266  003604  012737  003700  000114          MOV     #TRP7,@#PARVEC  ;SETUP PARITY TRAP RETURN
1267  003612  005037  000116                  CLR     @#PARVEC+2
1268  003616  012700  000570                  MOV     #MPR0,R0        ;SETUP TO GET ADDRESS OF REGISTER PRESENT
1269  003622  032710  000001          LUP7:   BIT     #1,@R0
1270  003626  001406                          BEQ     TST7            ;BRANCH TO TEST REGISTER
1271  003630  062700  000010          LOOP7:  ADD     #10,R0
1272  003634  020027  000770                  CMP     R0,#TREG
1273  003640  103770                          BLO     LUP7
1274  003642  000412                          BR      DONE7           ;BRANCH IF DONE TESTING ALL REGISTERS
1275  003644  012770  000001  000000  TST7:   MOV     #AE,@(R0)       ;SET ACTION ENABLE
1276  003652  052770  100000  000000          BIS     #PERR,@(R0)     ;SET PARITY ERROR
1277  003660  000240                          NOP                     ;SHOULD NOT TRAP
1278  003662  005070  000000                  CLR     @(R0)           ;CLEAR PARITY REGISTER
```

```
1279  003666  000760                          BR      LOOP7                   ;GO CHECK NEXT REGISTER
1280  003670  012737  000116  000114  DONE7:  MOV     #PARVEC+2,@#PARVEC
1281  003676  000405                          BR      TEST10
1282  003700  104002                  TRP7:   ERROR                           ;TRAP OCCURRED WHEN PARITY ERROR BIT
1283                                                                          ;WAS SET VIA A BIS INSTRUCTION
1284                                                                          ;WITH ACTION ENABLE ALREADY SET.
1285                                                                          ;R0 POINTS TO THE ADDRESS OF THE
1286                                                                          ;PARITY REGISTER
1287  003702  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
1288  003704  005070  000000                  CLR     @(R0)                   ;CLEAR PARITY REGISTER
1289  003710  000747                          BR      LOOP7
1290
1291
1292
1293                                  ;*************************************************************************
1294                                  ;SHOW THAT REPEATED PARITY ERRORS WILL CAUSE REPEATED TRAPS IF ACTION
1295                                  ;ENABLE IS SET AND PARITY ERROR IS LEFT SET.
1296                                  ;SHOW THAT THE ERROR ADDRESS BITS (11-5) TRACK (ONLY FOR CORE PARITY REGISTERS)
1297                                  ;*************************************************************************
```

```
1298  003712  104001                    TEST13: SCOPE
1299  003714  012777  000010  175160            MOV     #10,@DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY
1300  003722  004767  007562                    JSR     %7,CLRPAR           ;INITIALLY CLEAR ALL PARITY REGISTERS
1301  003726  005767  174620                    TST     NOKT                ;KT11 PRESENT?
1302  003732  001004                            BNE     1$                  ;NO- BRANCH
1303  003734  004767  007450                    JSR     %7,NRALL            ;YES- INITIALLY MAP ALL PAGES NR
1304  003740  004767  007614                    JSR     %7,MAP1             ;MAP KERNEL 0 TO BANK 0,RW
1305                                                                        ;KERNEL 7 TO THE EXTERNAL BANK, RW
1306                                                                        ;MAKE KERNEL PAGE 1 RW AND TURN ON THE KT11
1307  003744  012700  000570          1$:       MOV     #MPR0,R0
1308  003750  012702  000772                    MOV     #INDC0,R2
1309  003754  032710  000001          LUP10:    BIT     #1,@R0
1310  003760  001407                            BEQ     TST10               ;BRANCH TO TEST REGISTER IF PRESENT
1311  003762  062700  000010          LOOP10:   ADD     #10,R0
1312  003766  005722                            TST     (R2)+
1313  003770  020027  000770                    CMP     R0,#TREG
1314  003774  103767                            BLO     LUP10
1315  003776  000507                            BR      DONE10              ;BRANCH IF ALL REGISTERS HAVE BEEN TESTED
1316  004000  004767  007620          TST10:    JSR     %7,LOCATM
1317  004004  032701  000001                    BIT     #1,R1               ;ERROR RETURN INDICATED?
1318  004010  001403                            BEQ     .+10                ;BRANCH IF NO
1319  004012  104002                            ERROR                       ;MAP INDICATES THERE IS NO MEMORY
1320                                                                        ;CORRESPONDING TO THIS REGISTER
1321                                                                        ;R0 POINTS TO THE ADDRESS OF
1322                                                                        ;THE PARITY REGISTER
1323  004014  000167  177742                    JMP     LOOP10
1324  004020  012770  000004  000000            MOV     #WWP,@(R0)          ;SET WRITE WRONG PARITY
1325  004026  011111                            MOV     @R1,@R1             ;WRITE WRONG PARITY IN FIRST LOCATION
1326  004030  016161  004000  004000            MOV     4000(R1),4000(R1)   ;WRITE WRONG PARITY IN SECOND LOCATION
1327  004036  012770  000001  000000            MOV     #AE,@(R0)           ;SET ACTION ENABLE AND CLEAR REST
1328  004044  012737  004076  000114            MOV     #TRP10,@#PARVEC     ;SETUP PARITY TRAP RETURN
1329  004052  012767  000010  000152            MOV     #10,COUNT           ;SETUP COUNTER TO EXECUTE INSTRUCTION 1
1330                                                                        ;(INST1) TEN TIMES
1331  004060  005711                  INST1:    TST     @R1                 ;READ WRONG PARITY WITH AE SET- SHOULD
1332                                                                        ;TRAP TO TRP10
1333  004062  104002                            ERROR                       ;NO PARITY TRAP OCCURRED. R0 POINTS TO
1334                                                                        ;ADDRESS OF THE PARITY REGISTER BEING
1335                                                                        ;TESTED.
1336  004064  000441                            BR      CONT10
1337  004066  005761  004000          INST2:    TST     4000(R1)            ;READ WRONG PARITY FROM SECOND ADDRESS
1338                                                                        ;WITH AE SET- SHOULD TRAP TO TRP10A
1339  004072  104002                            ERROR                       ;NO PARITY TRAP OCCURRED. R0 POINTS TO
1340                                                                        ;THE ADDRESS OF THE PARITY REGISTER
1341                                                                        ;BEING TESTED
1342  004074  000435                            BR      CONT10
1343  004076  005367  000130          TRP10:    DEC     COUNT               ;HAS PARITY TRAP OCCURRED TEN TIMES?
1344  004102  001413                            BEQ     1$                  ;YES- BRANCH
1345  004104  022712  000001                    CMP     #1,(R2)             ;IS THIS A CORE PAR REG?
1346  004110  001005                            BNE     2$                  ;NO, BRANCH (NO ERROR
1347                                                                        ;ADDRESS BITS FOR MOS PAR
1348  004112  032770  000040  000000            BIT     #BIT5,@(R0)         ;IF ERROR ADDRESS BITS ARE TRACKING,
1349                                                                        ;BIT 5 SHOULD BE CLEAR (ONLY FOR CORE PARITY)
1350  004120  001401                            BEQ     .+4
1351  004122  104002                            ERROR                       ;PARITY ERROR ADDRESS BITS INCORRECT
1352                                                                        ;R0 POINTS TO THE ADDRESS OF THE PARITY
1353                                                                        ;REGISTER. R1 CONTAINS THE ADDRESS
```

```
1354                                                    ;REFERENCED TO CAUSE A PARITY TRAP
1355                                                    ;(VIRTUAL IF KT11 IS PRESENT)
1356  004124  012716  004060          2$:     MOV     #INST1,@SP      ;GO EXECUTE INSTRUCTION 1 AGAIN
1357  004130  000002                  RTI
1358  004132  012737  004146  000114  1$:     MOV     #TRP10A,@#PARVEC        ;CHANGE PARITY TRAP RETURN
1359  004140  012716  004066          MOV     #INST2,@SP      ;GO EXECUTE INSTRUCTION 2
1360  004144  000002                  RTI
1361  004146  022712  000001          TRP10A: CMP     #1,(R2)         ;IS THIS A CORE REG?
1362  004152  001005                  BNE     1$              ;NO, BRANCH
1363  004154  032770  000040  000000  BIT     #BIT5,@(R0)     ;PARITY TRAP OCCURRED- CHECK PARITY
1364                                                    ;ERROR ADDRESS BITS
1365  004162  001001                  BNE     .+4             ;BRANCH IF OK (IF THE PARITY ERROR
1366                                                    ;ADDRESS BITS TRACKED, BIT 5 WILL BE SET)
1367  004164  104002                  ERROR                   ;PARITY ERROR ADDRESS BITS INCORRECT
1368                                                    ;R0 POINTS TO THE ADDRESS OF THE
1369                                                    ;PARITY REGISTER. THE ADDRESS REFERENCED
1370                                                    ;TO CAUSE THE ERROR WAS THAT IN
1371                                                    ;R1 PLUS 4000 (OCTAL).
1372  004166  022626          1$:     CMP     (SP)+,(SP)+
1373  004170  012737  000116  000114  CONT10: MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAP CATCHER
1374  004176  005070  000000          CLR     @(R0)           ;CLEAR PARITY REGISTER
1375  004202  005511                  ADC     @R1             ;CLEAR BAD PARITY
1376  004204  005561  004000          ADC     4000(R1)
1377  004210  005070  000000          CLR     @(R0)           ;CLEAR PARITY ERROR BIT IF SET
1378  004214  000662                  BR      LOOP10
1379  004216  005767  174330          DONE10: TST     NOKT
1380  004222  001002                  BNE     .+6
1381  004224  005037  177572          CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1382  004230  000401                  BR      TEST11
1383  004232  000000          COUNT:  0
1384
1385
1386
1387                                                    ;*****************************************************************************
1388                                                    ;IF MULTIPLE PARITY ERRORS OCCUR DURING ONE INSTRUCTION (WITH ACTION ENABLE
1389                                                    ;NOT SET) THE ERROR ADDRESS BITS WILL RECORD THE LAST ERROR (ONLY FOR CORE PARITY
1390                                                    ;REGISTERS)
1391                                                    ;*****************************************************************************
1392  004234  104001                  TEST11: SCOPE
1393  004236  012777  000011  174636          MOV     #11,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
1394  004244  004767  007240          JSR     %7,CLRPAR       ;INITIALLY CLEAR ALL PARITY REGISTERS
1395  004250  005767  174276          TST     NOKT            ;KT11 PRESENT?
1396  004254  001004                  BNE     1$              ;NO- BRANCH
1397  004256  004767  007126          JSR     %7,NRALL        ;YES, MAP KERNEL PAGE 0 TO BANK 0,RW
1398  004262  004767  007272          JSR     %7,MAP1         ;MAP KERNEL PAGE 7 TO THE EXTERNAL BANK, RW
1399                                                    ;SET KERNEL PAGE 1 RW AND TURN ON KT11
1400  004266  012700  000570          1$:     MOV     #MPR0,R0        ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1401  004272  012703  000772          MOV     #INDC0,R3
1402  004276  032710  000001          LUP11:  BIT     #1,@R0
1403  004302  001003                  BNE     LOOP11          ;IF THIS REG NOT PRESENT, SKIP
1404  004304  022713  000001          CMP     #1,(R3)         ;IS THIS A CORE PAR REG?
1405  004310  001407                  BEQ     TST11           ;YES, THEN TEST IT
1406                                                    ;IF NOT CORE, SKIP THIS REGISTER
1407  004312  062700  000010          LOOP11: ADD     #10,R0
1408  004316  005723                  TST     (R3)+
1409  004320  020027  000770          CMP     R0,#TREG
```

```
1410  004324  103764                          BLO     LUP11
1411  004326  000443                          BR      DONE11          ;BRANCH OUT IF ALL REGISTERS HAVE BEEN TESTED
1412  004330  004767  007270          TST11:  JSR     %7,LOCATM       ;GET THE ADDRESS OF A MEMORY LOCATION
1413                                                                  ;CORRESPONDING TO THIS PARITY REGISTER
1414  004334  032701  000001                  BIT     #1,R1           ;ERROR RETURN INDICATED?
1415  004340  001403                          BEQ     .+10            ;BRANCH IF NOT
1416  004342  104002                          ERROR                   ;NO MEMORY IN MAP CORRESPONDING TO
1417                                                                  ;THIS PARITY REGISTER. R0 POINTS
1418                                                                  ;TO THE ADDRESS OF THE PARITY REGISTER
1419  004344  000167  177742                  JMP     LOOP11
1420  004350  010102                          MOV     R1,R2           ;SETUP SECOND TEST ADDRESS LOCATION
1421  004352  062702  010000                  ADD     #10000,R2
1422  004356  012770  000004  000000          MOV     #WWP,@(R0)      ;SET WRITE WRONG PARITY
1423  004364  011111                          MOV     @R1,@R1         ;WRITE WRONG PARITY IN FIRST TEST LOCATION
1424  004366  011212                          MOV     @R2,@R2         ;WRITE WRONG PARITY IN SECOND TEST LOCATION
1425  004370  005070  000000                  CLR     @(R0)           ;CLEAR PARITY REGISTER
1426  004374  021112                          CMP     @R1,@R2         ;READ FIRST TEST LOCATION, AND
1427                                                                  ;THEN READ SECOND TEST LOCATION
1428  004376  005770  000000                  TST     @(R0)           ;MAKE SURE PARITY ERROR SET
1429  004402  100401                          BMI     .+4
1430  004404  104002                          ERROR                   ;PARITY ERROR NOT SET AFTER
1431                                                                  ;READING TWO LOCATIONS WHICH
1432                                                                  ;SHOULD HAVE BAD PARITY
1433  004406  032770  000100  000000          BIT     #BIT6,@(R0)     ;CHECK ERROR ADDRESS- IF THE LAST
1434                                                                  ;ADDRESS WAS RECORDED, BIT 6 WILL
1435                                                                  ;BE SET
1436  004414  001001                          BNE     .+4
1437  004416  104002                          ERROR                   ;PARITY ERROR ADDRESS BITS INCORRECT
1438                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER
1439                                                                  ;R2 CONTAINS ADDRESS OF LAST BAD PARITY
1440                                                                  ;LOCATION REFERENCED (IF KT11 PRESENT,
1441                                                                  ;ADDRESS IS VIRTUAL THRU KERNEL PAGE 1)
1442  004420  005070  000000                  CLR     @(R0)           ;CLEAR PARITY REGISTER
1443  004424  005511                          ADC     @R1             ;CLEAR BAD PARITY
1444  004426  005512                          ADC     @R2
1445  004430  005070  000000                  CLR     @(R0)           ;CLEAR PARITY ERROR BIT
1446  004434  000726                          BR      LOOP11
1447  004436  005767  174110          DONE11: TST     NOKT
1448  004442  001002                          BNE     .+6
1449  004444  005037  177572                  CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1450
1451
1452
1453                                          ;*************************************************************************
1454                                          ;SHOW THAT IF AN INSTRUCTION DOING A DATIP GETS A PARITY ERROR,
1455                                          ;THE ORIGINAL DATA IS REWRITTEN IF ACTION ENABLE IS SET, AND IS
1456                                          ;ALTERRED IF ACTION ENABLE IS CLEAR
1457                                          ;*************************************************************************
1458  004450  104001          TEST12: SCOPE
1459  004452  012777  000012  174422          MOV     #12,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
1460  004460  004767  007024                  JSR     %7,CLRPAR
1461  004464  005767  174062                  TST     NOKT            ;KT11 PRESENT?
1462  004470  001004                          BNE     1$              ;NO- BRANCH
1463  004472  004767  006712                  JSR     %7,MRALL        ;YES, MAP KERNEL 0 TO BANK 0, RW
1464  004476  004767  007056                  JSR     %7,MAP1         ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
1465                                                                  ;SET KERNEL 1 RW AND TURN ON KT11
```

```
1466  004502  012700  000570          1$:     MOV     #MPRO,R0        ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1467  004506  012702  000772                  MOV     #INDCO,R2
1468  004512  032710  000001          LUP12:  BIT     #1,@R0
1469  004516  001003                          BNE     LOOP12          ;SKIP, IF THIS REG NOT PRESENT
1470  004520  022712  000001                  CMP     #1,(R2)         ;REG PRESENT, IS IT CORE?
1471  004524  001407                          BEQ     TST12           ;YES, DO# THIS TEST
1472                                                                  ;SKIP, IF THIS REG IS NOT CORE
1473  004526  062700  000010          LOOP12: ADD     #10,R0
1474  004532  005722                          TST     (R2)+
1475  004534  020027  000770                  CMP     R0,#TREG
1476  004540  103764                          BLO     LUP12
1477  004542  000474                          BR      DONE12          ;BRANCH TO DONE IF ALL REGISTERS
1478                                                                  ;HAVE BEEN TESTED
1479  004544  004767  007054          TST12:  JSR     %7,LOCATM       ;LOCATE MEMORY CORRESPONDING TO THIS
1480                                                                  ;REGISTER
1481  004550  032701  000001                  BIT     #1,R1           ;ERROR RETURN INDICATED?
1482  004554  001403                          BEQ     .+10            ;NO- BRANCH
1483  004556  104002                          ERROR                   ;NO MEMORY IN MAP CORRESPONDING TO
1484                                                                  ;THIS REGISTER. R0 POINTS TO
1485                                                                  ;THE ADDRESS OF THE PARITY REGISTER
1486  004560  000167  177742                  JMP     LOOP12
1487  004564  012737  004620  000114          MOV     #TRP12,@#PARVEC ;SET UP PARITY TRAP RETURN
1488  004572  012770  000004  000000          MOV     #WWP,2(R0)      ;SET WRITE WRONG PARITY
1489  004600  012711  125252                  MOV     #125252,@R1     ;WRITE WRONG PARITY IN TEST LOCATION
1490  004604  012770  000001  000000          MOV     #AE,2(R0)       ;SET ACTION ENABLE AND CLEAR
1491                                                                  ;WRITE WRONG PARITY
1492  004612  005211                          INC     @R1             ;DO DATIP,DATO WITH ACTION ENABLE
1493                                                                  ;SET- SHOULD ABORT ON DATIP AND
1494                                                                  ;RESTORE ORIGINAL DATA
1495  004614  104002                          ERROR                   ;NO ABORT OCCURRED ON READING LOCATION
1496                                                                  ;WHICH SHOULD CONTAIN BAD PARITY
1497                                                                  ;(WITH AE SET).
1498                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER.
1499                                                                  ;R1 CONTAINS ADDRESS OF TEST LOCATION
1500                                                                  ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11 IS
1501                                                                  ;PRESENT)
1502  004616  000440                          BR      CONT12
1503  004620  005070  000000          TRP12:  CLR     2(R0)           ;PARITY TRAP OCCURRED- CLEAR PARITY REGISTER
1504  004624  021127  125252                  CMP     @R1,#125252     ;ORIGINAL DATA RESTORED?
1505  004630  001401                          BEQ     .+4             ;YES, BRANCH
1506  004632  104002                          ERROR                   ;NO- DATIP WHICH GOT A PARITY ERROR
1507                                                                  ;TRAP ALTERED CONTENTS OF LOCATION
1508                                                                  ;READ. ADDRESS OF TEST LOCATION IS IN R1
1509                                                                  ;(IF KT11 IS PRESENT, ADDRESS IN R1
1510                                                                  ;IS VIRTUAL THRU KERNEL PAGE 1)
1511                                                                  ;R0 POINTS TO ADDRESS OF PARITY REGISTER
1512  004634  005770  000000                  TST     2(R0)           ;MAKE SURE PARITY ERROR SET WHEN
1513  004640  100401                          BMI     .+4             ;DATA WAS REREAD IN THE ABOVE CMP
1514  004642  104002                          ERROR                   ;DATIP WHICH GOT A PARITY ERROR TRAP
1515                                                                  ;ALTERED THE PARITY OF THE LOCATION READ
1516                                                                  ;R1 CONTAINS ADDRESS OF TEST LOCATION
1517                                                                  ;(VIRTUAL THRU KERNEL 1 IF KT11 PRESENT)
1518  004644  022626                  CONT12: CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1519  004646  012770  000004  000000          MOV     #WWP,2(R0)      ;SET WRITE WRONG PARITY AND CLEAR
1520                                                                  ;PARITY ERROR
1521  004654  012711  125252                  MOV     #125252,@R1     ;REWRITE DATA WITH WRONG PARITY
```

```
1522  004660  005070  000000                    CLR     @(R0)           ;CLEAR PARITY REGISTER
1523  004664  012737  000116  000114            MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAPCATCHER
1524  004672  005211                            INC     @R1             ;SINCE AE IS CLEAR, INSTRUCTION SHOULD
1525                                                                    ;COMPLETE AND SHOULD CLEAR BAD PARITY
1526  004674  005070  000000                    CLR     @(R0)           ;CLEAR PARITY ERROR BIT
1527  004700  022711  125253                    CMP     #125253,@R1     ;CHECK DATA
1528  004704  001401                            BEQ     .+4
1529  004706  104002                            ERROR

1530                                                                    ;DATIP, DATO TO A LOCATION CONTAINING BAD
1531                                                                    ;PARITY WITHOUT AE SET LEFT INCORRECT
1532                                                                    ;DATA. R0 POINTS TO THE ADDRESS OF
1533                                                                    ;THE PARITY REGISTER. R1 CONTAINS THE
1534                                                                    ;ADDRESS OF THE TEST LOCATION (VIRTUAL
1535  004710  005770  000000                    TST     @(R0)           ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1536  004714  100001                            BPL     .+4             ;CHECK PARITY ERROR BIT
1537  004716  104002                            ERROR

1538                                                                    ;DATIP, DATO WITH AE CLEAR DID
1539                                                                    ;NOT CLEAR BAD PARITY IN LOCATION
1540                                                                    ;ADDRESSED. R0 POINTS TO THE ADDRESS
1541                                                                    ;OF THE PARITY REGISTER. R1 CONTAINS
1542                                                                    ;THE ADDRESS OF THE TEST LOCATION
1543                                                                    ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11
1544  004720  005070  000000            CONT12: CLR     @(R0)           ;IS PRESENT)
1545  004724  005011                            CLR     @R1             ;CLEAR PARITY REGISTER
1546  004726  005070  000000                    CLR     @(R0)           ;CLEAR LOCATION TO RESTORE GOOD PARITY
1547  004732  000675                            BR      LOOP12          ;CLEAR PARITY ERROR IF SET
1548                                                                    ;GO CHECK FOR ANOTHER PARITY
1549  004734  012737  000116  000114  DONE12:   MOV     #PARVEC+2,@#PARVEC      ;REGISTER
1550  004742  005767  173604                    TST     NOKT                    ;RESTORE TRAPCATCHER
1551  004746  001002                            BNE     .+6
1552  004750  005037  177572                    CLR     @#SR0           ;TURN OFF KT11 IF PRESENT
1553
1554
1555
1556                                            ;*************************************************************************
1557                                            ;SHOW THAT IF AN INSTRUCTION DOING A DATI (BUT NO DATO TO THE SAME LOCATION)
1558                                            ;GETS A PARITY ERROR, THE ORIGINAL DATA IS UNALTERRED, WHETHER OR NOT ACTION
1559                                            ;ENABLE IS SET
1560                                            ;*************************************************************************
1561  004754  104001                    TEST13: SCOPE
1562  004756  012777  000013  174116            MOV     #13,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
1563  004764  004767  006520                    JSR     %7,CLRPAR
1564  004770  005767  173556                    TST     NOKT            ;KT11 PRESENT?
1565  004774  001004                            BNE     1$              ;NO- BRANCH
1566  004776  004767  006406                    JSR     %7,MRALL        ;YES, MAP KERNEL 0 TO BANK 0,KERNEL
1567  005002  004767  006552                    JSR     %7,MAP1         ;7 TO THE EXTERNAL BANK, AND KERNEL
1568                                                                    ;0,1,AND 7 RW
1569  005006  012700  000570            1$:     MOV     #MPR0,R0        ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1570  005012  032710  000001            LUP13:  BIT     #1,@R0
1571  005016  001406                            BEQ     TST13           ;IF THIS REGISTER IS PRESENT, GO
1572                                                                    ;TEST IT
1573  005020  062700  000010            LOOP13: ADD     #10,R0
1574  005024  020027  000770                    CMP     R0,#TREG
1575  005030  103770                            BLO     LUP13
1576  005032  000470                            BR      DONE13          ;BRANCH IF ALL REGISTERS HAVE BEEN
1577                                                                    ;TESTED
```

E03


```
1578  005034  004767  006564          TST13:  JSR     %7,LOCATM       ;LOCATE MEMORY CORRESPONDING TO THIS
1579                                                                  ;REGISTER, AND IF KT11 IS PRESENT
1580                                                                  ;MAP KE  L 1 TO THAT MEMORY
1581  005040  032701  000001                  BIT     #1,R1           ;ERROR  URN INDICATED?
1582  005044  001403                          BEQ     .+10            ;NO- E  CH
1583  005046  104002                          ERROR                   ;MAP  ICATES NO MEMORY WAS FOUND
1584                                                                  ;CORR  ONDI G TO THIS REGISTER
1585                                                                  ;R0 POINTS TO THE ADDRESS OF THE
1586                                                                  ;PARITY REGISTER
1587  005050  000167  177744                  JMP     LOOP13
1588  005054  012737  005110  000114          MOV     #TRP13,@#PARVEC ;SETUP PARITY TRAP RETURN
1589  005062  012770  000004  000000          MOV     #WRP,@(R0)      ;SET WRITE  RONG PARITY
1590  005070  012711  125252                  MOV     #125252,@R1     ;WRITE  RONG  ITY
1591  005074  012770  000001  000000          MOV     #AE,@(R0)       ;SET ACTI  E  E AND CLEAR
1592                                                                  ;WRITE  RONG P, ITY
1593  005102  005711                          TST     @R1             ;DATI WITH ACTI  ENABLE SET SHOULD
1594                                                                  ;ABORT LEAVING  TA U  HANGED
1595  005104  104002                          ERROR                   ;NO ABORT ON R   ING BAD P RITY
1596                                                                  ;WITH ACTION E  LE SET. R0 POINTS
1597                                                                  ;TO THE AC  ESS OF THE PARITY REGISTER.
1598                                                                  ;R1 CONTAINS THE ADD  SS OF THE TEST
1599                                                                  ;LOCATION (VIR  AL  RU KERNEL PAGE
1600                                                                  ;1 IF KT11 IS PR SENT).
1601  005106  000434                          BR      CONT13
1602  005110  005070  000000          TRP13:  CLR     @(R0)           ;ABORT OCCUR ED AS EXPECTED- CLEAR REGISTER
1603  005114  021127  125252                  CMP     @R1,#125252     ;ORIGINAL DATA RESTOR D?
1604  005120  001401                          BEQ     .+4             ;YES, B NCH
1605  005122  104002                          ERROR                   ;DATI WHICH GOT A PARITY ERROR ALTERRED
1606                                                                  ;THE CONTENTS OF THE LOCATION ADDRESSED.
1607                                                                  ;(R1 CONTAINS THE ADDRESS OF
1608                                                                  ;MEMORY BEI  TESTED- IF KT11 IS
1609                                                                  ;PRESENT, ADDRESS IN R1 IS VIRTUAL)
1610                                                                  ;R0 POINTS TO THE ADDRESS OF THE
1611                                                                  ;PARITY R GISTER
1612  005124  005770  000000                  TST     @(R0)           ;CHECK PARITY REGISTER
1613  005130  100401                          BMI     .+4             ;BR NCH IF PARITY E  OR SET
1614  005132  104002                          ERROR                   ;PARITY ERROR NOT SET AFTER READING
1615                                                                  ;DATA WITH BAD PARITY
1616                                                                  ;R0 POINTS TO THE ADDRESS OF THE PARITY
1617                                                                  ;REGISTER. R1 CONTAINS THE ADDRESS
1618                                                                  ;OF THE TEST LOCATION (VIRTUAL THRU
1619                                                                  ;KERNEL PAGE 1 IF KT11 PRESENT)
1620  005134  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1621  005136  012770  000004  000000          MOV     #WRP,@(R0)      ;SET WRITE WRONG PARITY, CLEAR PARITY ERROR
1622  005144  012711  125252                  MOV     #125252,@R1     ;REWRITE DATA WITH WRONG PARITY
1623  005150  005070  000000                  CLR     @(R0)           ;CLEAR PARITY REGISTER
1624  005154  012737  000116  000114          MOV     @#PARVEC+2,@#PARVEC
                                                                      ;RESTORE TRAPCATCHER
1625  005162  005711                          TST     @R1             ;DATI TO LOCATION WITH BAD PARITY
1626                                                                  ;AE NOT SET-INSTRUCTION SHOULD COMPLETE
1627  005164  005070  000000                  CLR     @(R0)           ;CLEAR PARITY ERROR BIT
1628  005170  022711  125252                  CMP     #125252,@R1     ;CHECK DATA
1629  005174  001401                          BEQ     .+4
1630  005176  104002                          ERROR
1631                                                                  ;DATI TO LOCATION WITH BAD PARITY
1632                                                                  ;WITHOUT ACTION ENABLE SET LEFT INCORRECT DATA
1633                                                                  ;R1 CONTAINS THE ADDRESS OF THE TEST
                                                                      ;LOCATION (VIRTUAL THRU KERNEL PAGE 1
```

```
                                                                ;IF KT11 IS PRESENT). R0 POINTS TO THE
                                                                ;ADDRESS OF THE PARITY REGISTER.
005200  005070  000000          CONT13: CLR     @(R0)           ;CLEAR PARITY REGISTER
005204  005011                          CLR     @R1             ;CLEAR LOCATION
005206  005070  000000                  CLR     @(R0)           ;CLEAR PARITY ERROR IF SET
005212  000702                          BR      LOOP13          ;GO CHECK FOR ANOTHER PARITY
                                                                ;REGISTER
005214  012737  000116  000114  DONE13: MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAPCATCHER
005222  005767  173324                  TST     NOKT
005226  001002                          BNE     .+6
005230  005037  177572                  CLR     @#SR0           ;TURN OFF KT11 IF PRESENT


                                        ;*****************************************************************
                                        ;CHECK PARITY MEMORY WITH SERIES OF PATTERNS FROM 4K TO 28K
                                        ;ENABLE PARITY TRAP
                                        ;*****************************************************************
005234  104001                  TEST14: SCOPE
005236  012777  000014  173636          MOV     #14,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
005244  005067  010676                  CLR     IMAX            ;DON'T ITERATE THE REST OF THE SUBTESTS
005250  004767  006234                  JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
005254  012737  005556  000114          MOV     #TRP14,@#PARVEC ;SETUP PARITY TRAP RETURN
005262  012767  000002  173230          MOV     #2,BITPT        ;INITIALIZE BANK INDICATOR TO BANK 1
005270  012767  020000  173220          MOV     #20000,ADRPT    ;INITIALIZE MEMORY STARTING ADDRESS
005276  036767  173216  173366  LOOP14: BIT     BITPT,PHEML     ;DOES THIS 4K HAVE PARITY?
005304  001012                          BNE     TST14           ;YES, TEST IT
005306  062767  020000  173202  LUP14:  ADD     #20000,ADRPT    ;NO- UPDATE MEMORY ADDRESS
005314  006367  173200                  ASL     BITPT           ;UPDATE BIT POINTER
005320  022767  000200  173172          CMP     #200,BITPT      ;THIS 2 K DONE?
005326  003363                          BGT     LOOP14          ;NO  BRANCH TO SEE IF NEXT 4K
                                                                ;SHOULD BE TESTED
005330  000443                          BR      DONE14          ;YES, EXIT
005332  012704  001040          TST14:  MOV     #PARPAT,R4      ;INITIALIZE PATTERN POINTER
005336  016767  173154  173164          MOV     ADRPT,HIADR     ;SET UPPER LIMIT FOR THIS 4K
005344  062767  020000  173156          ADD     #20000,HIADR
005352  016705  173140                  MOV     ADRPT,R5
005356  005025                  2$:     CLR     (5)+            ;INITIALLY CLEAR CORE BLOCK UNDER TEST
005360  020527  173144                  CMP     R5,HIADR
005364  103774                          BLO     2$
005366  012701  000570                  MOV     #MPRO,R1        ;INITIALIZE TO SET AE IN ALL REGISTERS
005372  032711  000001          3$:     BIT     #1,@R1
005376  001003                          BNE     .+10
005400  012771  000001  000000          MOV     #AE,@(R1)       ;SET ACTION ENABLE IF REGISTER IS PRESENT
005406  062701  000010                  ADD     #10,R1
005412  020127  000770                  CMP     R1,#TREG
005416  103765                          BLO     3$
005420  004767  000024          4$:     JSR     %7,TPCORE       ;GO TO ROUTINE TO EXERCISE THIS 4K
                                                                ;WITH THE CURRENT PATTERN
005424  005724                          TST     (4)+            ;UPDATE PATTERN
005426  005714                          TST     (4)             ;LAST PATTERN?
005430  001373                          BNE     4$              ;NO, LOOP
005432  004767  006022                  JSR     PC,CLRPAR       ;YES, CLEAR ALL PARITY REGISTERS
005436  000723                          BR      LUP14           ;UPDATE AND CHECK NEXT 4K
005440  012737  000116  000114  DONE14: MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAP CATCHER
005446  000473                          BR      TEST15          ;GO TO NEXT TEST
```

```
1690
1691                                        ;ROUTINE TO WRITE AND CHECK EACH LOCATION IN 4K (STARTING AT ADDRESS
1692                                        ;IN AORPT) WITH VALUE POINTED TO BY R4
1693  005450  016705  173042        TPCORE: MOV     AORPT,R5            ;SETUP R5 TO ADDRESS MEMORY
1694                                                                    ;LOCATION BEING CHECKED
1695  005454  011415                1$:     MOV     (4),(5)             ;WRITE PATTERN INTO MEMORY
1696  005456  011567  173054              MOV     (5),WAS             ;READ TEST LOCATION
1697  005462  021467  173050              CMP     (4),WAS             ;DATA OK?
1698  005466  001401                      BEQ     .+4                 ;YES- BRANCH
1699  005470  104002                      ERROR                       ;DATA INCORRECT IN LOCATION WHOSE
1700                                                                    ;ADDRESS IS IN R5. R4 POINTS TO THE
1701                                                                    ;DATA WRITTEN.
1702  005472  005725                      TST     (5)+                ;UPDATE ADDRESS POINTER
1703  005474  020567  173030              CMP     R5,HIADR            ;THIS 4K DONE?
1704  005500  103765                      BLO     1$                  ;NO, BRANCH TO TEST NEXT LOCATION
1705  005502  005067  173262              CLR     TREG                ;YES, DID ANY PARITY ERRORS OCCUR
1706                                                                    ;WITHOUT TRAPPING?
1707  005506  012701  000570              MOV     #MPRO,R1
1708  005512  032711  000001        2$:     BIT     #1,(1)
1709  005516  001003                      BNE     .+10
1710  005520  005771  000000              TST     @(R1)
1711  005524  100406                      BMI     3$                  ;YES- BRANCH
1712  005526  062701  000010              ADD     #10,R1
1713  005532  020127  000770              CMP     R1,#TREG
1714  005536  103765                      BLO     2$
1715  005540  000207                      RTS     %7                  ;NO, RETURN
1716  005542  011167  173222        3$:     MOV     @R1,TREG            ;STORE ADDRESS OF REGISTER GETTING ERROR
1717  005546  104004                      ERRORS                      ;PARITY ERROR SET (WITH AE SET) AND
1718                                                                    ;NO TRAP OCCURRED. TREG CONTAINS
1719                                                                    ;ADDRESS OF PARITY REGISTER WHICH
1720                                                                    ;HAS ERROR BIT SET.
1721  005550  004767  006436              JSR     %7,PSCAN            ;SCAN FOR PARITY ERRORS AND PRINT
1722                                                                    ;18 BIT ADDRESSES OF THOSE FOUND.
1723                                                                    ;AFTER REPORTING EACH ERROR CLEAR IT
1724  005554  000207                      RTS     %7
1725
1726                                        ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
1727  005556  005067  173206        TRP14:  CLR     TREG
1728  005562  012701  000570              MOV     #MPRO,R1            ;FIND PARITY REGISTER INDICATING PARITY ERROR
1729  005566  032711  000001        1$:     BIT     #1,(R1)
1730  005572  001003                      BNE     .+10
1731  005574  005771  000000              TST     @(R1)
1732  005600  100407                      BMI     2$                  ;BRANCH IF PARITY ERROR SET
1733  005602  062701  000010              ADD     #10,R1
1734  005606  020127  000770              CMP     R1,#TREG
1735  005612  103765                      BLO     1$
1736  005614  104002                      ERROR                       ;PARITY TRAP TO 114 OCCURRED DURING
1737  005616  000405                      BR      3$                  ;TEST 14 BUT NO REGISTERS HAVE
1738                                                                    ;PARITY ERROR SET
1739  005620  011167  173144        2$:     MOV     @R1,TREG            ;STORE ADDRESS OF REGISTER GETTING ERROR
1740  005624  104004                      ERRORS                      ;PARITY TRAP TO 114 OCCURRED DUE TO
1741                                                                    ;PARITY ERROR WHILE EXERCISING MEMORY
1742                                                                    ;R1 POINTS TO THE ADDRESS OF THE
1743                                                                    ;PARITY REGISTER HAVING PARITY ERROR
1744                                                                    ;BIT SET
1745  005626  004767  006360              JSR     %7,PSCAN            ;SCAN FOR BAD PARITY AND TYPE 18 BIT ADDRESSES
```

```
1746
1747  005632  022626                    3$:     CMP     (SP)+,(SP)+             ;OF LOCATIONS FOUND BAD
1748  005634  000207                            RTS     %7                      ;RESTORE STACK POINTER
1749                                                                            ;RETURN (FROM JSR TO TPCORE) TO
1750                                                                            ;CHECK NEXT PATTERN
1751
1752
1753                                  ;*************************************************************************
1754                                  ;CHECK PARITY MEMORY WITH SERIES OF PATTERNS ABOVE 28K
1755                                  ;ENABLE PARITY ERROR TRAPPING
1756                                  ;*************************************************************************
1757  005636  104001                  TEST15: SCOPE
1758  005640  012777  000015  173234          MOV     #15,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
1759  005646  005767  172700                  TST     NOKT                    ;KT11 PRESENT?
1760  005652  001402                          BEQ     .+6                     ;YES- BRANCH
1761  005654  000167  000424                  JMP     TEST16                  ;NO, SKIP TEST
1762  005660  004767  000424                  JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
1763  005664  004767  000420                  JSR     %7,MPALL                ;MAP KERNEL 0 TO BANK 0,RW
1764  005670  004767  005664                  JSR     %7,MAP1                 ;MAP KERNEL 7 TO THE EXTERNAL BANK
1765                                                                          ;SET KERNEL 1 RW AND TURN ON KT11
1766  005674  012777  001600  173344          MOV     #1600,@KPAR1            ;MAP KERNEL PAGE 1 TO BEGINNING OF 28-32K
1767  005702  005067  172650                  CLR     LOWFLG                  ;CLEAR FLAG TO INDICATE CHECKING LOWER 64K
1768  005706  016767  173160  173162          MOV     PMEML,PMEMX
1769  005714  012737  006224  000114          MOV     #TRAP1S,@#PARVEC        ;SETUP PARITY TRAP RETURN
1770  005722  012767  000200  172570          MOV     #200,BITPT              ;INITIALIZE BIT POINTER
1771  005730  036767  172564  173140  LOOP15: BIT     BITPT,PMEMX             ;DOES THIS 4K HAVE PARITY?
1772  005736  001022                          BNE     TST15                   ;YES, BRANCH TO TEST IT
1773  005740  062777  000200  173300  LOP15:  ADD     #200,@KPAR1             ;NO- MAP TO NEXT 4K
1774  005746  006367  172546                  ASL     BITPT                   ;UPDATE BIT POINTER
1775  005752  103366                          BCC     LOOP15                  ;BRANCH IF NOT DONE WITH 64K
1776  005754  005767  172576                  TST     LOWFLG                  ;DONE WITH 128K?
1777  005760  001051                          BNE     DONE15                  ;YES, BRANCH
1778  005762  005267  172570                  INC     LOWFLG                  ;NO, SET FLAG INDICATING UPPER 64K
1779  005766  016767  173102  173102          MOV     PMEMH,PMEMX             ;SETUP PARITY MAP WORD
1780  005774  012767  000001  172516          MOV     #1,BITPT                ;SETUP BIT POINTER FOR UPPER 64K
1781  006002  000752                          BR      LOOP15                  ;CONTINUE
1782  006004  012704  001040          TST15:  MOV     #PARPAT,R4              ;INITIALIZE PATTERN POINTER
1783  006010  012767  020000  172500          MOV     #20000,ADRPT            ;INITIALIZE VIRTUAL ADDRESS OF MEMORY
1784                                                                          ;BEING TESTED
1785  006016  012705  020000                  MOV     #20000,R5
1786  006022  005025                  2$:     CLR     (5)+                    ;INITIALLY CLEAR CORE BLOCK UNDER TEST
1787  006024  020527  040000                  CMP     R5,#40000
1788  006030  103774                          BLO     2$
1789  006032  012701  000570                  MOV     #MPR0,R1                ;INITIALIZE TO SET ACTION ENABLE IN ALL
1790                                                                          ;PARITY REGISTERS
1791  006036  032711  000001          3$:     BIT     #1,@R1
1792  006042  001003                          BNE     .+10
1793  006044  012771  000001  000000          MOV     #AE,@(R1)               ;SET ACTION ENABLE IF THIS REGISTER
1794                                                                          ;IS PRESENT
1795  006052  062701  000010                  ADD     #10,R1
1796  006056  020127  000770                  CMP     R1,#TREG
1797  006062  103765                          BLO     3$
1798  006064  004767  000030          4$:     JSR     %7,TPCORX               ;EXERCISE THIS 4K
1799  006070  005724                          TST     (4)+                    ;UPDATE PATTERN
1800  006072  005714                          TST     (4)                     ;LAST PATTERN?
1801  006074  001373                          BNE     4$                      ;NO, LOOP
```

```
1802  006076  004767  005406                  JSR     PC,CLRPAR       ;YES, CLEAR ALL PARITY REGISTERS
1803  006102  000716                          BR      LOP15           ;UPDATE AND CHECK NEXT 4K
1804  006104  005037  177572          DONE15: CLR     @#SRO           ;TURN OFF KT11 WHEN DONE
1805  006110  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC      ;RESTORE TRAPCATCHER
1806  006116  000472                          BR      TEST16          ;GO TO NEXT TEST
1807
1808                                  ;PARITY MEMORY TEST ROUTINE USING KT11 AND TESTING MEMORY ABOVE 28K
1809                                  ;WRITES AND CHECKS EACH LOCATION IN 4K USING KERNEL PAGE 1 MAPPED TO CURRENT BANK
1810  006120  000240                  TPCORX: NOP
1811  006122  012705  020000                  MOV     #20000,R5       ;SETUP R5 TO POINT TO THE LOCATION
1812                                                                  ;UNDER TEST (VIRTUAL ADDRESS)
1813  006126  011415                  1$:     MOV     (4),(5)         ;WRITE PATTERN
1814  006130  011567  172402                  MOV     (5),WAS         ;READ TEST LOCATION
1815  006134  021467  172376                  CMP     (4),WAS         ;DATA OK?
1816  006140  001401                          BEQ     .+4             ;YES- BRANCH
1817  006142  104002                          ERROR                   ;NO- DATA INCORRECT IN LOCATION WHOSE
1818                                                                  ;VIRTUAL ADDRESS IS IN R1 (GOES THRU
1819                                                                  ;KERNEL PAGE 1). R4 POINTS TO
1820                                                                  ;THE VALUE WRITTEN.
1821  006144  005725                          TST     (5)+            ;UPDATE ADDRESS POINTER
1822  006146  020527  040000                  CMP     R5,#40000       ;THIS 4K DONE?
1823  006152  103765                          BLO     1$              ;NO, BRANCH TO TEST NEXT LOCATION
1824  006154  005067  172610                  CLR     TREG            ;YES, CHECK TO SEE IF ANY PARITY
1825                                                                  ;ERRORS OCCURRED WITHOUT TRAPPING
1826  006160  012701  000570                  MOV     #MPR0,R1
1827  006164  032711  000001          2$:     BIT     #1,(1)          ;IS THIS PARITY REGISTER PRESENT?
1828  006170  001003                          BNE     .+10            ;NO, GET NEXT ONE
1829  006172  005771  000000                  TST     @(R1)           ;YES- DID ERROR SET?
1830  006176  100406                          BMI     3$              ;YES- BRANCH
1831  006200  062701  000010                  ADD     #10,R1          ;NO- GET NEXT REGISTER
1832  006204  020127  000770                  CMP     R1,#TREG
1833  006210  103765                          BLO     2$
1834  006212  000207                          RTS     %7              ;NO ERRORS- EXIT
1835  006214  011167  172550          3$:     MOV     @R1,TREG        ;STORE ADDRESS OF REGISTER GETTING ERROR
1836  006220  104004                          ERRORS                  ;PARITY ERROR SET (AE ALREADY SET)
1837                                                                  ;AND NO TRAP OR TIMEOUT OCCURRED
1838                                                                  ;R1 POINTS TO THE ADDRESS OF THE
1839                                                                  ;PARITY REGISTER
1840  006222  000207                          RTS     %7
1841
1842                                  ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
1843  006224  005067  172540          TRP15:  CLR     TREG
1844  006230  012701  000570                  MOV     #MPR0,R1        ;LOCATE PARITY REGISTER INDICATING ERROR
1845  006234  032711  000001          1$:     BIT     #1,(R1)
1846  006240  001003                          BNE     .+10
1847  006242  005771  000000                  TST     @(R1)
1848  006246  100407                          BMI     2$              ;BRANCH IF PARITY ERROR IS SET
1849  006250  062701  000010                  ADD     #10,R1
1850  006254  020127  000770                  CMP     R1,#TREG
1851  006260  103765                          BLO     1$
1852  006262  104002                          ERROR                   ;TRAP TO 114 OCCURRED DURING TEST 15 BUT
1853                                                                  ;NO PARITY REGISTERS HAVE PARITY ERROR SET
1854  006264  000405                          BR      3$
1855  006266  011167  172476          2$:     MOV     @R1,TREG        ;STORE ADDRESS OF REGISTER GETTING ERROR
1856  006272  104004                          ERRORS                  ;PARITY TRAP TO 114 OCCURRED DUE TO
1857                                                                  ;PARITY ERROR WHILE EXERCISING MEMORY
```

```
 1858                                                                    ;"TREG" CONTAINS ADDRESS OF PARITY REGISTER
 1859                                                                    ;HAVING PARITY ERROR BIT SET
 1860  006274  004767  005712                JSR     %7,PSCAN          ;SCAN MEMORY FOR BAD PARITY AND PRINT 18
 1861                                                                    ;BIT ADDRESSES OF LOCATIONS FOUND
 1862                                                                    ;CLEAR BAD PARITY IN EACH AFTER
 1863                                                                    ;REPORTING IT
 1864  006300  022626                3$:     CMP     (SP)+,(SP)+       ;RESTORE STACK POINTER
 1865  006302  000207                        RTS     %7                ;RETURN (FROM JSR TO TPCORX) TO
 1866                                                                    ;TEST NEXT PATTERN
 1867
 1868
 1869
 1870                                ;*************************************************************************
 1871                                ;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY FROM 4K TO 28K
 1872                                ;WRITE WRONG PARITY AND READ IT BACK WITH ACTION ENABLE SET, MAKING
 1873                                ;SURE THAT A TRAP OCCURS. THEN WRITE GOOD PARITY AND MAKE SURE THAT
 1874                                ;NO TRAP OCCURS WHEN IT IS READ. MAKE SURE THAT THE ERROR ADDRESS BITS
 1875                                ;(PARITY REGISTER BITS 5-11) ARE CORRECT.
 1876                                ;*************************************************************************
 1877  006304  104001                TEST16: SCOPE
 1878  006306  012777  000016  172566        MOV     #16,@#DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY
 1879  006314  012737  006662  000114        MOV     #TRP16,@#PARVEC   ;SET UP TRAP RETURN
 1880  006322  012767  000002  172170        MOV     #2,BITPT          ;INIT 4K BIT POINTER TO BANK 1
 1881  006330  012767  020000  172160        MOV     #20000,ADRPT      ;INIT MEMORY STARTING ADDRESS
 1882  006336  036767  172156  172526  1$:   BIT     BITPT,PMEML       ;DOES THIS 4K HAVE PARITY?
 1883  006344  001012                        BNE     3$                ;YES, BRANCH TO TEST IT
 1884  006346  062767  020000  172142  2$:   ADD     #20000,ADRPT      ;NO, UPDATE MEMORY ADDRESS BY 4K
 1885  006354  006367  172140                ASL     BITPT             ;UPDATE BIT POINTER
 1886  006360  022767  000200  172132        CMP     #200,BITPT        ;THIS 28K DONE?
 1887  006366  003363                        BGT     1$                ;NO, CHECK NEXT 4K
 1888  006370  000421                        BR      DONE16            ;YES, EXIT
 1889  006372  016767  172120  172130  3$:   MOV     ADRPT,HIADR       ;SET UPPER LIMIT THIS 4K
 1890  006400  062767  020000  172122        ADD     #20000,HIADR
 1891  006406  004767  005076                JSR     %7,CLRPAR         ;CLEAR ALL PARITY REGISTERS
 1892  006412  016705  172100                MOV     ADRPT,R5
 1893  006416  005025                5$:     CLR     (5)+              ;CLEAR BANK UNDER TEST
 1894  006420  020567  172104                CMP     R5,HIADR
 1895  006424  103774                        BLO     5$
 1896  006426  004767  000020        6$:     JSR     %7,WWP16          ;GO WRITE WRONG PARITY IN EACH BYTE
 1897  006432  000745                        BR      2$                ;UPDATE AND CHECK NEXT 4K
 1898  006434  004767  005050        DONE16: JSR     %7,CLRPAR         ;CLEAR ALL PARITY REGISTERS IF DONE
 1899  006440  012737  000116  000114        MOV     #PARVEC+2,@#PARVEC        ;RESTORE TRAP CATCHER
 1900  006446  000167  000472                JMP     TEST17            ;GO TO NEXT TEST
 1901
 1902                                ;WRITE WRONG PARITY TEST ROUTINE - TESTS EACH BYTE IN 4K
 1903                                ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
 1904                                ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
 1905  006452  016705  172040        WWP16:  MOV     ADRPT,R5          ;SET TEST ADDRESS POINTER
 1906  006456  005067  172076                CLR     ODDFLG            ;INDICATE TESTING LOW BYTE
 1907  006462  012767  125253  172044        MOV     #125253,SHDBE     ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
 1908  006470  012715  125253        WWP16A: MOV     #125253,@R5       ;INITIALIZE TEST LOCATION
 1909  006474  012701  000570                MOV     #MPRO,R1          ;SETUP TO LOAD PARITY REGISTERS
 1910  006500  032711  000001        1$:     BIT     #1,(1)
 1911  006504  001003                        BNE     .+10
 1912  006506  012771  000005  000000        MOV     #WWP+AE,@(R1)     ;SET WRITE WRONG PARITY AND ACTION
 1913                                                                    ;ENABLE IF THIS PARITY REGISTER
```

```
1914
1915  006514  062701  000010                  ADD     #10,R1                  ;IS PRESENT
1916  006520  020127  000770                  CMP     R1,#TREG
1917  006524  103765                          BLO     1$
1918  006526  005767  172026                  TST     ODDFLG                  ;WRITING HIGH BYTE?
1919  006532  100425                          BMI     2$                      ;YES, BRANCH
1920  006534  112715  000253                  MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
1921  006540  012701  000570          5$:     MOV     #MPR0,R1                ;THIS CODE CLEARS WWP BIT IN ALL
1922  006544  032711  000001                  BIT     #1,(1)                  ;PARITY REGISTERS
1923  006550  001003                          BNE     .+10                    ;
1924  006552  042771  000004  000000          BIC     #WWP,@(R1)
1925  006560  062701  000010                  ADD     #10,R1
1926  006564  020127  000770                  CMP     R1,#TREG
1927  006570  103765                          BLO     5$+4
1928  006572  005767  171762                  TST     ODDFLG                  ;TESTING HIGH OR LOW BYTE?
1929  006576  100407                          BMI     6$                      ;BRANCH,IF HIGH BYTE
1930  006600  142715  000377                  BICB    #377,@R5                ;DETECT WRONG PARITY WITH DATIP-
1931                                                                          ;SHOULD TRAP TO TRP16 BEFORE DOING THE DATOB
1932  006604  000407                          BR      3$
1933  006606  112765  000252  000001  2$:     MOVB    #252,1(R5)              ;WRITE WRONG PARITY IN HIGH BYTE
1934  006614  000751                          BR      5$
1935  006616  142765  000377  000001  6$:     BICB    #377,1(R5)              ;DETECT WRONG PARITY WITH DATIP-
1936                                                                          ;SHOULD TRAP TO TRP16 BEFORE DOING THE DATOB
1937  006624  012701  000570          3$:     MOV     #MPR0,R1                ;NO TRAP OCCURRED- SETUP TO CLEAR
1938                                                                          ;AE AND WWP
1939  006630  032711  000001          4$:     BIT     #1,@R1
1940  006634  001003                          BNE     .+10
1941  006636  042771  000005  000000          BIC     #WWP+AE,@(R1)           ;CLEAR AE AND WWP IN ALL PARITY REGISTERS
1942  006644  062701  000010                  ADD     #10,R1
1943  006650  020127  000770                  CMP     R1,#TREG
1944  006654  103765                          BLO     4$
1945  006656  104002                          ERROR                           ;ERROR, NO TRAP AFTER WRITING AND
1946                                                                          ;READING WRONG PARITY IN LOCATION
1947                                                                          ;WHOSE ADDRESS IS IN R5 (AE AND WWP
1948                                                                          ;WERE SET IN ALL PARITY REGISTERS)
1949                                                                          ;TESTING LOW BYTE IF ODDFLG IS POSITIVE
1950                                                                          ;TESTING HIGH BYTE IF ODDFLG IS NEGATIVE
1951                                                                          ;NOTE THAT AE AND WWP WERE CLEARED
1952                                                                          ;BEFORE TYPING THE ERROR PRINTOUT
1953  006660  000517                          BR      CN16
1954
1955                                  ;WHEN WRONG PARITY DATA IS READ BACK SHOULD ENTER HERE VIA TRAP TO 114
1956  006662  012701  000570          TRP16:  MOV     #MPR0,R1                ;PARITY TRAP OCCURRED- FIRST CLEAR
1957  006666  032711  000001          TRP16A: BIT     #1,@R1                  ;WWP AND AE IN ALL REGISTERS
1958  006672  001003                          BNE     .+10
1959  006674  042771  000005  000000          BIC     #AE+WWP,@(R1)
1960  006702  062701  000010                  ADD     #10,R1
1961  006706  020127  000770                  CMP     R1,#TREG
1962  006712  103765                          BLO     TRP16A
1963  006714  012701  000570                  MOV     #MPR0,R1                ;FIND THE REGISTER THAT SENSED THE ERROR
1964  006720  012703  000772                  MOV     #INDCO,R3
1965  006724  005067  172040                  CLR     TREG
1966  006730  032711  000001          1$:     BIT     #1,@R1                  ;DOES THIS CONTROL EXIST?
1967  006734  001017                          BNE     2$                      ;NO, BRANCH
1968  006736  005771  000000                  TST     @(R1)                   ;YES- IS ERROR SET?
1969  006742  100014                          BPL     2$                      ;NO, BRANCH
```

```
1970  006744  005767  172020              TST     TREG            ;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
1971  006750  001401                      BEQ     .+4             ;NO- BRANCH
1972  006752  104002                      ERROR                   ;ERROR SET IN MORE THAN ONE PARITY REGISTER
1973                                                              ;AFTER WRITING WRONG PARITY IN LOCATION
1974                                                              ;WHOSE ADDRESS IS IN R5
1975  006754  036761  171540  000002      BIT     BITPT,2(R1)     ;DOES MAP INDICATE THIS PARITY REGISTER
1976                                                              ;CONTROLS THIS MEMORY?
1977  006762  001001                      BNE     .+4             ;YES, BRANCH
1978  006764  104002                      ERROR                   ;PARITY REGISTER RESPONDED TO MEMORY
1979                                                              ;NOT INCLUDED IN ITS MAP
1980                                                              ;PARITY REGISTER'S ADDRESS IS POINTED
1981                                                              ;TO BY R1. ADDRESS OF LOCATION CAUSING
1982                                                              ;PARITY ERROR IS IN R5
1983  006766  010304                      MOV     R3,R4
1984  006770  011167  171774              MOV     @R1,TREG        ;STORE REGISTER ADDRESS
1985  006774  062701  000010      2$:     ADD     #10,R1
1986  007000  005723                      TST     (R3)+
1987  007002  020127  000770              CMP     R1,#TREG
1988  007006  103750                      BLO     1$              ;BRANCH UNTIL ALL THE PARITY
1989                                                              ;REGISTERS HAVE BEEN CHECKED
1990  007010  011567  171522              MOV     (5),WAS         ;SAVE DATA FROM LOCATION UNDER TEST
1991  007014  022715  125253              CMP     #125253,@R5     ;DID BICB CHANGE DATA?
1992  007020  001401                      BEQ     .+4             ;NO, CONTINUE
1993  007022  104003                      ERRORP                  ;DATA WAS MODIFIED BY THE BICB WHICH
1994                                                              ;GOT A PARITY ERROR TRAP- SINCE PARITY ERROR
1995                                                              ;TRAP OCCURRED, CONTENTS SHOULD NOT HAVE
1996                                                              ;BEEN MODIFIED, R5 CONTAINS ADDRESS
1997                                                              ;OF TEST LOCATION. "TREG" CONTAINS
1998                                                              ;ADDRESS OF PARITY REGISTER SENSING
1999                                                              ;ERROR
2000  007024  005767  171740              TST     TREG            ;WAS PARITY ERROR SET IN ANY REGISTERS?
2001  007030  001002                      BNE     3$              ;YES- BRANCH
2002  007032  104002                      ERROR                   ;PARITY TRAP OCCURRED ON READING
2003                                                              ;WRONG PARITY (WITH AE SET) BUT NO
2004                                                              ;REGISTERS HAD PARITY ERROR BIT SET.
2005                                                              ;R5 CONTAINS THE ADDRESS OF THE
2006                                                              ;TEST LOCATION.
2007  007034  000420                      BR      4$
2008  007036  022714  000001      3$:     CMP     #1,(R4)
2009  007042  001015                      BNE     4$
2010  007044  017701  171720              MOV     @TREG,R1        ;GET PARITY REGISTER CONTENTS
2011  007050  042701  170037              BIC     #170037,R1      ;MASK OFF ALL BUT ERROR ADDRESS BITS
2012  007054  010502                      MOV     R5,R2           ;GET ADDRESS OF LOCATION UNDER TEST
2013  007056  042702  003777              BIC     #3777,R2        ;POSITION BITS IN R2
2014  007062  000302                      SWAB    R2
2015  007064  006302                      ASL     R2
2016  007066  006302                      ASL     R2
2017  007070  020102                      CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
2018  007072  001401                      BEQ     .+4
2019  007074  104003                      ERRORP                  ;ERROR ADDRESS BITS (PARITY REGISTER
2020                                                              ;BITS 11-5) ARE INCORRECT. R5 CONTAINS
2021                                                              ;THE ADDRESS OF THE TEST LOCATION.
2022                                                              ;"TREG" CONTAINS THE ADDRESS OF THE
2023                                                              ;PARITY REGISTER DETECTING THE ERROR
2024  007076  022626              4$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
2025  007100  011515              CNT16:  MOV     (5),(5)         ;RESTORE TEST LOCATION TO FIX BAD PARITY
```

```
2026  007102  005077  171662          CLR     @TREG           ;CLEAR ERROR BIT IN PARITY REGISTER
2027  007106  005715                  TST     @R5             ;READ LOCATION TO SEE IF PARITY IS GOOD
2028  007110  005777  171654          TST     @TREG           ;IS PARITY ERROR SET?
2029  007114  100001                  BPL     .+4             ;NO- BRANCH
2030  007116  104002                  ERROR                   ;WRITING LOCATION WITH WRITE WRONG PARITY
2031                                                          ;CLEAR DIDN'T CLEAR BAD PARITY
2032                                                          ;R5 CONTAINS ADDRESS OF THE TEST
2033                                                          ;LOCATION. "TREG" CONTAINS THE ADDRESS
2034                                                          ;OF THE PARITY REGISTER DETECTING
2035                                                          ;THE ERROR
2036  007120  005167  171434  CN16:    COM     ODDFLG          ;TOGGLE BYTE INDICATOR
2037  007124  100401                  BMI     .+4             ;BRANCH IF READY TO TEST HIGH BYTE
2038  007126  005725                  TST     (5)+            ;UPDATE ADDRESS POINTER
2039  007130  020567  171374          CMP     R5,HIADR        ;THIS 4K DONE?
2040  007134  103401                  BLO     1$              ;NO, TEST NEXT LOCATION
2041  007136  000207                  RTS     %7              ;RETURN TO TEST NEXT BANK
2042  007140  000167  177324  1$:     JMP     WWP16A
2043
2044
2045
2046                          ;*****************************************************************
2047                          ;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY ABOVE 28K
2048                          ;WRITE WRONG PARITY AND READ IT WITH ACTION ENABLE SET, MAKING SURE
2049                          ;THAT A TRAP OCCURS. THEN WRITE AND READ THE SAME LOCATION WITH GOOD PARITY
2050                          ;(USING SAME DATA) TO SHOW THAT THE PARITY BIT TOGGLES. MAKE SURE THAT
2051                          ;THE ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11) ARE CORRECT.
2052                          ;*****************************************************************
2053  007144  104001          TEST17: SCOPE
2054  007146  012777  000017  171726          MOV     #17,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY
2055  007154  005767  171372          TST     NOKT            ;KT11 PRESENT?
2056  007160  001402                  BEQ     .+6             ;YES, BRANCH
2057  007162  000167  000636          JMP     XFR1            ;NO, SKIP TO NEXT TEST
2058  007166  004767  004316          JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2059  007172  012737  007534  000114  MOV     #TRP17,@#PARVEC ;SETUP FOR PARITY TRAP
2060  007200  012767  000200  171312  MOV     #200,BITPT      ;INITIALIZE 4K BIT POINTER
2061  007206  004767  004176          JSR     %7,MRALL        ;INITIALIZE ALL PAGES TO NON-RESIDENT
2062  007212  004767  004342          JSR     %7,MAP1         ;MAP KERNEL 0 TO BANK 0,RW; KERNEL 7
2063                                                          ;TO THE EXTERNAL BANK, RW. TURN ON
2064                                                          ;THE KT11 AND MAKE KERNEL 1 RW
2065  007216  012777  001600  172022          MOV     #1600,@KPAR1    ;INITIALIZE KERNEL PAGE 1 TO 28K
2066  007224  005067  171326          CLR     LOWFLG          ;CLEAR FLAG TO INDICATE TESTING LOWER 64K
2067  007230  016767  171636  171640  MOV     PMEML,PMEMX
2068  007236  012767  000002  000366  MOV     #2,INDX17       ;SETUP OFFSET TO CHECK LOWER 64K OF MAPS
2069  007244  036767  171250  171624  1$:     BIT     BITPT,PMEMX     ;DOES THIS 4K HAVE PARITY?
2070  007252  001025                  BNE     3$              ;YES, BRANCH TO TEST IT
2071  007254  062777  000200  171764  2$:     ADD     #200,@KPAR1     ;NO, MAP TO NEXT 4K
2072  007262  006367  171232          ASL     BITPT           ;UPDATE BIT POINTER
2073  007266  103366                  BCC     1$              ;GO CHECK TO SEE IF THIS 4K HAS PARITY
2074  007270  005767  171262          TST     LOWFLG          ;END OF 128K?
2075  007274  001025                  BNE     DONE17          ;YES, EXIT
2076  007276  005267  171254          INC     LOWFLG          ;NO, SET FLAG TO INDICATE SHIFT TO
2077  007302  016767  171566  171566  MOV     PMEMH,PMEMX     ;HIGH 64K AND CHANGE MAPS
2078  007310  012767  000001  171202  MOV     #1,BITPT
2079  007316  012767  000004  000306  MOV     #4,INDX17       ;SETUP OFFSET TO CHECK UPPER 64K OF MAPS
2080  007324  000747                  BR      1$
2081  007326  012705  020000          3$:     MOV     #20000,R5
```

```
2082  007332  005025                  5$:     CLR     (5)+                    ;CLEAR BANK UNDER TEST
2083  007334  020527  040000                  CMP     R5,#40000
2084  007340  103774                  6$:     BLO     5$
2085  007342  004767  000020                  JSR     %7,WWP17                ;GO WRITE WRONG PARITY AND CHECK IT
2086  007346  000742                          BR      2$                      ;UPDATE AND CHECK NEXT 4K
2087  007350  005037  177572          DONE17: CLR     @#SR0                   ;TURN OFF KT11 WHEN DONE
2088  007354  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
2089  007362  000167  000436                  JMP     XFR1                    ;GO TO SETUP FOR NEXT TEST
2090
2091                                  ;WRITE WRONG PARITY TEST ROUTINE TO TEST MEMORY ABOVE 28K
2092  007366  012705  020000          WWP17:  MOV     #20000,R5               ;SET TEST ADDRESS POINTER
2093  007372  005067  171162                  CLR     ODDFLG                  ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
2094  007376  012767  125253  171130          MOV     #125253,SHOBE           ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
2095  007404  012715  125253          WWP17A: MOV     #125253,@R5             ;INITIALIZE LOCATION
2096  007410  012701  000570                  MOV     #MPR0,R1                ;INITIALIZE REGISTER ADDRESS POINTER
2097  007414  032711  000001          1$:     BIT     #1,(1)                  ;DOES THIS CONTROL EXIST?
2098  007420  001003                          BNE     .+10                    ;NO, GET NEXT
2099  007422  012771  000005  000000          MOV     #WWP+AE,@(R1)           ;YES- SET WRITE WRONG PARITY
2100                                                                          ;AND ACTION ENABLE
2101  007430  062701  000010                  ADD     #10,R1
2102  007434  020127  000770                  CMP     R1,#TREG                ;ALL REGISTERS SETUP?
2103  007440  103765                          BLO     1$                      ;NO- LOOP
2104  007442  005767  171112                  TST     ODDFLG                  ;YES- TESTING HIGH BYTE"
2105  007446  100405                          BMI     2$                      ;YES, BRANCH
2106  007450  112715  000253                  MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
2107  007454  142715  000377                  BICB    #377,@R5                ;DETECT WRONG PARITY WITH DATIP-
2108                                                                          ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
2109  007460  000406                          BR      3$
2110  007462  112765  000252  000001  2$:     MOVB    #252,1(R5)              ;WRITE WRONG PARITY IN HIGH BYTE
2111  007470  142765  000377  000001          BICB    #377,1(R5)              ;DETECT WRONG PARITY WITH DATIP
2112                                                                          ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
2113  007476  012701  000570          3$:     MOV     #MPR0,R1                ;IF NO TRAP, CLEAR AE AND WWP IN ALL
2114  007502  032711  000001          4$:     BIT     #1,@R1                  ;PARITY REGISTERS
2115  007506  001003                          BNE     .+10
2116  007510  042771  000005  000000          BIC     #AE+WWP,@(R1)
2117  007516  062701  000010                  ADD     #10,R1
2118  007522  020127  000770                  CMP     R1,#TREG
2119  007526  103765                          BLO     4$
2120  007530  104002                          ERROR                           ;NO TRAP AFTER WRITING AND READING
2121                                                                          ;WRONG PARITY WITH AE SET- VIRTUAL
2122                                                                          ;ADDRESS OF LOCATION IS IN R5
2123                                                                          ;(MAPPED THRU KERNEL PAGE 1)
2124                                                                          ;WROTE LOW BYTE IF ODDFLG IS POSITIVE
2125                                                                          ;WROTE HIGH BYTE IF IT IS NEGATIVE
2126                                                                          ;NOTE THAT WWP AND AE WERE CLEARED
2127                                                                          ;BEFORE ERROR PRINTOUT
2128  007532  000512                          BR      CNT17
2129
2130                                  ;WHEN WRONG PARITY DATA IS READ BACK, SHOULD ENTER HERE VIA .RAP TO 114
2131  007534  012701  000570          TRP17:  MOV     #MPR0,R1                ;PARITY TRAP OCCURRED- BEFORE CHECKING
2132  007540  032711  000001          TRP17A: BIT     #1,@R1                  ;IT, CLEAR WWP AND AE IN ALL REGISTERS
2133  007544  001003                          BNE     .+10
2134  007546  042771  000005  000000          BIC     #AE+WWP,@(R1)
2135  007554  062701  000010                  ADD     #10,R1
2136  007560  020127  000770                  CMP     R1,#TREG
2137  007564  103765                          BLO     TRP17A
```

```
2138  007566  012701  000570              MOV     #MPRO,R1        ;FIND REGISTER THAT SENSED THE EPROR
2139  007572  012703  000772              MOV     #INDCO,R3               ;SETUP PTR TO INDICATOR
2140  007576  005067  171166              CLR     TREG
2141  007602  032711  000001      1$:     BIT     #1,@R1          ;DOE` THIS CONTROL EXIST?
2142  007606  001017                      BNE     2$              ;NO, BRANCH
2143  007610  005771  000000              TST     @(R1)           ;YES- IS ERROR SET?
2144  007614  100014                      BPL     2$              ;NO, BRANCH
2145  007616  005767  171146              TST     TREG            ;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
2146  007622  001401                      BEQ     .+4             ;NO- BRANCH
2147  007624  104002                      ERROR                   ;ERROR SET IN MORE THAN ONE PARITY REGISTER
2148                                                              ;AFTER READING WRONG PARITY IN LOCATION
2149                                                              ;WHOSE VIRTUAL ADDRESS IS IN R5
2150  007626  036761  170666  000002      BIT     BITPT,2(R1)     ;DOES MAP INDICATE THAT THIS REGISTER
2151                                                              ;CONTROLS THIS MEMORY?
2152          007632                      INDX17=.-2
2153  007634  001001                      BNE     .+4             ;YES- BRANCH
2154  007636  104002                      ERROR                   ;PARITY REGISTER RESPONDED TO MEMORY
2155                                                              ;NOT INCLUDED IN ITS MAP. R1 POINTS TO
2156                                                              ;THE PARITY REGISTER'S ADDRESS. R5
2157                                                              ;CONTAINS VIRTUAL ADDRESS OF THE LOCATION
2158                                                              ;BEING TESTED (MAPPED THRU KERNEL
2159                                                              ;PAGE 1)
2160  007640  011167  171124              MOV     @R1,TREG        ;STORE REGISTER ADDRESS
2161  007644  010304                      MOV     R3,R4           ;STORE PTR TO INDICATOR
2162  007646  062701  000010      2$:     ADD     #10,R1
2163  007652  005723                      TST     (R3)+           ;INCREMENT PTR TO INDICATOR
2164  007654  020177  000770              CMP     R1,#TREG
2165  007660  103750                      BLO     1$              ;LOOP UNTIL ALL THE PARITY REGISTERS
2166                                                              ;HAVE BEEN CHECKED
2167  007662  011567  170650              MOV     (5),WAS         ;SAVE CONTENTS OF LOCATION UNDER TEST
2168  007666  022715  125253              CMP     #125253,@R5     ;DID BICB CHANGE DATA?
2169  007672  001401                      BEQ     .+4             ;NO, CONTINUE
2170  007674  104002                      ERROR                   ;DATA WAS MODIFIED BY THE BICB WHICH
2171                                                              ;GOT A PARITY ERROR TRAP- SINCE PARITY
2172                                                              ;TRAP OCCURRED, CONTENTS SHOULD NOT
2173                                                              ;HAVE BEEN MODIFIED. R5 CONTAINS TEST
2174                                                              ;LOCATION ADDRESS (VIRTUAL, MAPPED THRU
2175                                                              ;KERNEL PAGE 1). "TREG" CONTAINS ADDRESS OF
2176                                                              ;PARITY REGISTER DETECTING PARITY ERROR
2177  007676  005767  171066              TST     TREG            ;WAS PARITY ERROR SET IN ANY REGISTER?
2178  007702  001003                      BNE     3$              ;YES- BRANCH
2179  007704  104002                      ERROR                   ;PARITY TRAP OCCURRED ON READING
2180                                                              ;WRONG PARITY WITH AE SET BUT NO
2181                                                              ;REGISTER HAS THE PARITY ERROR BIT
2182                                                              ;SET. R5 CONTAINS VIRTUAL ADDRESS
2183                                                              ;OF THE TEST LOCATION (MAPPED THRU
2184                                                              ;KERNEL PAGE 1)
2185  007706  000423                      BR      4$
2186  007710  001022                      BNE     4$              ;NO, BRANCH. DO NOT CHECK THE
2187  007712  022714  000001      3$:     CMP     #1,(R4)         ;IS THIS A CORE PARITY REGISTER?
2188  007716  001017                      BNE     4$              ;NO, BRANCH. DO NOT CHECK THE
2189                                                              ;ADDRESS BITS OF THE LAST PARITY
2190                                                              ;ERROR
2191                                                              ;IF CORE REG, CHECK IF ADDRESS
2192                                                              ;BITS OF LAST PAR ERR WERE STORED
2193  007720  017701  171044              MOV     @TREG,R1        ;GET CONTENTS OF REGISTER DETECTING PARITY ERROR
```

```
2194  007724  042701  170037          BIC     #170037,R1      ;MASK OFF ALL BUT ADDRESS BITS
2195  007730  010502                  MOV     R5,R2           ;CALCULATE TOP 7 BITS OF ERROR ADDRESS
2196  007732  042702  163777          BIC     #163777,R2
2197  007736  000302                  SWAB    R2
2198  007740  006302                  ASL     R2
2199  007742  006302                  ASL     R2
2200  007744  062702  171276          ADD     @KPAR1,R2
2201  007750  020102                  CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
2202  007752  001401                  BEQ     .+4
2203  007754  104004                  ERRORS                  ;PARITY ERROR ADDRESS BITS (PARITY
2204                                                          ;REGISTER BITS 11-5) INCORRECT
2205                                                          ;"TREG" CONTAINS ADDRESS OF PARITY
2206                                                          ;REGISTER, R5 CONTAINS THE VIRTUAL
2207                                                          ;ADDRESS OF THE TEST LOCATION
2208                                                          ;(MAPPED THRU KERNEL PAGE 1)
2209  007756  022626          4$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
2210  007760  011515          CNT17:  MOV     (5),(5)         ;RESTORE TEST LOCATION TO FIX BAD PARITY
2211  007762  005077  171002          CLR     @TREG           ;CLEAR ERROR BIT IN THE PARITY REGISTER
2212  007766  005715                  TST     @R5             ;READ LOCATION TO SEE IF PARITY IS GOOD
2213  007770  005777  170774          TST     @TREG           ;IS PARITY ERROR SET?
2214  007774  100001                  BPL     .+4             ;NO, BRANCH
2215  007776  104002                  ERROR                   ;WRITING LOCATION WITH WRITE WRONG
2216                                                          ;PARITY CLEAR DIDN'T CLEAR BAD PARITY
2217                                                          ;"TREG" CONTAINS THE ADDRESS OF THE
2218                                                          ;PARITY REGISTER. R5 CONTAINS THE
2219                                                          ;VIRTUAL ADDRESS OF THE TEST LOCATION
2220                                                          ;(MAPPED THRU KERNEL PAGE 1)
2221  010000  005167  170554          COM     ODDFLG          ;TOGGLE BYTE INDICATOR
2222  010004  100401                  BMI     .+4             ;BRANCH IF HIGH BYTE NOT YET TESTED
2223  010006  005725                  TST     (5)+            ;UPDATE ADDRESS POINTER
2224  010010  020527  040000          CMP     R5,#40000       ;THIS 4K DONE?
2225  010014  103401                  BLO     1$              ;NO, TEST NEXT LOCATION
2226  010016  000207                  RTS     %7              ;YES, RETURN TO CHECK FOR NEXT
2227                                                          ;BANK TO BE TESTED
2228  010020  000167  177360  1$:     JMP     @MP17A          ;GO AND TEST NEXT LOCATION
2229  010024  104001          XFR1:   SCOPE
2230                                  ;IF THE FIRST BANK (BANK 0) IS PARITY MEMORY, SETUP TO TEST IT
2231                                  ;COPY THE FIRST 4K TO THE SECOND 4K, MOVE THE STACK POINTER TO BANK 1,
2232                                  ;AND THEN JUMP TO THE COPY OF TEST20 IN BANK 1
2233  010026  005737  000042          TST     @#42
2234  010032  001402                  BEQ     .+6
2235  010034  000167  001324          JMP     DONE            ;IF THE PROGRAM IS RUNNING UNDER ACT THEN
2236                                                          ;GO TO DONE
2237  010040  032767  000001  171024  BIT     #1,PMEML
2238  010046  001002                  BNE     .+6             ;BRANCH IF YES
2239  010050  000167  001310          JMP     DONE            ;NO- DONE WITH TEST
2240  010054  032767  000002  171004  BIT     #2,MEML         ;IS THERE A SECOND 4K(BANK 1)?
2241  010062  001002                  BNE     .+6             ;YES, BRANCH
2242  010064  000167  001274          JMP     DONE            ;NO, EXIT
2243  010070  005037  177776          CLR     @#PS            ;CLEAR STATUS REGISTER
2244  010074  004767  003410          JSR     %7,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2245  010100  012700  010000          MOV     #10000,R0       ;R0 IS COUNTER TO MOVE 4K
2246  010104  005001                  CLR     R1              ;R1 POINTS TO LOCATION IN BANK 0
2247  010106  011161  020000  1$:     MOV     @R1,20000(R1)   ;COPY FROM BANK 0 TO BANK 1
2248  010112  005721                  TST     (R1)+           ;MOVE POINTER
2249  010114  005300                  DEC     R0              ;DONE WITH 4K?
```

```
2250  010116  001373                        BNE     1$                      ;NO- BR YCH
2251  010120  062706  020000                ADD     #20000,SP               ;YES, MOVE STACK POINTER TO POINT TO BANK 1
2252  010124  012767  032222  026020        MOV     #TEST20+20010,RETURN+20000      ;UPDATE SCOPE RETURN IN BANK 1
2253  010132  062767  020000  023200        ADD     #20000,ERRA1+20000      ;UPDATE ADDRESSES USED IN ERROR TYPEOUT
2254  010140  062767  020000  023174        ADD     #20000,ERRA2+20000
2255  010146  062767  020000  023176        ADD     #20000,ERRA3+20000
2256  010154  062767  020000  023172        ADD     #20000,ERRAH+20000
2257  010162  062767  020000  023174        ADD     #20000,ERRAS+20000
2258  010170  022767  177570  170702        CMP     #177570,SWR             ;DOES THE CONSOLE HAVE A SWITCH REG. ?
2259  010176  001403                        BEQ     2$                      ;IF SO THEN GO TO 2$
2260  010200  062767  020000  010672        ADD     #20000,SWR+20000        ;OTHERWISE THERE IS A SOFTWARE SWITCH LOCATION
2261                                                                        ;AND ITS ADDRESS SHOULD BE UPDATED
2262  010206  000167  020010          2$:   JMP     TEST20+20010            ;GO TO TEST 20 IN BANK 1
2263
2264
2265
2266                                  ;*****************************************************************
2267                                  ;IF FIRST 4K IS PARITY MEMORY, CHECK IT WITH A SERIES OF PATTERNS
2268                                  ;THIS SUBTEST IS RUN IN BANK 1 (20000 ABOVE THE ADDRESSES IN THE LISTING)
2269                                  ;*****************************************************************
2270  010212  013746  177776          TEST20: MOV     @#PS,-(SP)              ;THESE 2 LINES DO THE SAME AS A SCOPE WITHOUT
2271  010216  004767  005632                  JSR     PC,SCOPEC               ;USING AN EMT
2272  010222  012777  000020  170652          MOV     #20,@#DISPLAY           ;LOAD THE TEST NUMBER INTO THE DISPLAY
2273  010230  004767  003254                  JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2274  010234  012704  021040                  MOV     #PARPAT+20000,R4        ;INITIALIZE PATTERN POINTER
2275  010240  005005                          CLR     R5
2276  010242  005025                  1$:     CLR     (5)+                    ;INITIALLY CLEAR BANK 0
2277  010244  020527  020000                  CMP     R5,#20000
2278  010250  103774                          BLO     1$
2279  010252  012737  030460  000114          MOV     #TRP20+20000,@#PARVEC   ;SETUP TRAP RETURN
2280  010260  012701  020570                  MOV     #MPR0+20000,R1          ;SETUP TO SET ACTION ENABLE IN ALL
2281                                                                        ;PARITY REGISTERS PRESENT
2282  010264  032711  000001          2$:     BIT     #1,@R1
2283  010270  001003                          BNE     .+10
2284  010272  012771  000001  000000          MOV     #AE,@(R1)               ;SET ACTION ENABLE IF REGISTER IS PRESENT
2285  010300  062701  000010                  ADD     #10,R1
2286  010304  020127  020770                  CMP     R1,#TREG+20000
2287  010310  103765                          BLO     2$                      ;BRANCH UNTIL ALL REGISTER ADDRESSES
2288                                                                        ;HAVE BEEN CHECKED
2289  010312  004767  000022          3$:     JSR     %7,CKBK0                ;EXERCISE THIS 4K
2290  010316  005724                          TST     (4)+                    ;UPDATE PATTERN
2291  010320  005714                          TST     (4)                     ;LAST PATTERN?
2292  010322  001373                          BNE     3$                      ;NO, LOOP
2293  010324  004767  003160          DONE20: JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2294  010330  012737  000116  000114          MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAP CATCHER
2295  010336  000526                          BR      TEST21                  ;GO TO NEXT TEST
2296
```

```
2297                                    ;PARITY MEMORY TEST ROUTINE
2298                                    ;WRITES AND CHECKS EACH LOCATION IN BANK 0 (EXCEPT 114 AND 116)
2299                                    ;WITH VALUE POINTED TO BY R4
2300  010340  005005                    CKBK0:  CLR     R5                      ;SET ADDRESS POINTER
2301  010342  011415                    1$:     MOV     (4),(5)                 ;WRITE PATTERN
2302  010344  021415                            CMP     (4),(5)                 ;DATA OK?
2303  010346  001404                            BEQ     .+12                    ;YES- BRANCH
2304  010350  013746  177776                    MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
2305  010354  004767  002710                    JSR     PC,ERR                  ;ERROR- DATA INCORRECT IN LOCATION
2306                                                                            ;WHOSE ADDRESS IS IN R5. R4 POINTS
2307                                                                            ;TO THE VALUE WRITTEN.
2308  010360  005725                            TST     (5)+                    ;UPDATE ADDRESS POINTER
2309  010362  020527  000114                    CMP     R5,#114                 ;DON'T CHANGE CONTENTS OF 114 AND 116
2310  010366  001002                            BNE     .+6
2311  010370  062705  000004                    ADD     #4,R5
2312  010374  020527  020000                    CMP     R5,#20000               ;HAS THE WHOLE BANK BEEN TESTED WITH
2313                                                                            ;THIS PATTERN?
2314  010400  103760                            BLO     1$                      ;NO, BRANCH TO TEST NEXT LOCATION
2315  010402  005067  170362                    CLR     TREG                    ;YES, DID ANY PARITY ERROR BITS SET?
2316  010406  012701  020570                    MOV     #MPR0+20000,R1
2317  010412  032711  000001            2$:     BIT     #1,(1)
2318  010416  001003                            BNE     .+10
2319  010420  005771  000000                    TST     @(R1)
2320  010424  100406                            BMI     3$                      ;YES- BRANCH
2321  010426  062701  000010                    ADD     #10,R1
2322  010432  020127  020770                    CMP     R1,#TREG+20000
2323  010436  103765                            BLO     2$
2324  010440  000207                            RTS     %7                      ;NO- RETURN
2325  010442  013746  177776            3$:     MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
2326  010446  004767  002616                    JSR     PC,ERR                  ;ERROR- PARITY ERROR BIT SET AND NO
2327                                                                            ;PARITY TRAP OCCURRED
2328                                                                            ;(AE WAS SET) - R1 POINTS TO ADDRESS
2329                                                                            ;OF PARITY REGISTER
2330  010452  022626                            CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
2331  010454  000167  177644                    JMP     DONE20
2332
2333                                    ;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
2334  010460  005067  170304            TRP20:  CLR     TREG
2335  010464  012701  020570                    MOV     #MPR0+20000,R1          ;FIND THE REGISTER RECORDING A PARITY ERROR
2336  010470  032711  000001            1$:     BIT     #1,(R1)
2337  010474  001003                            BNE     .+10
2338  010476  005771  000000                    TST     @(R1)
2339  010502  100412                            BMI     2$                      ;BRANCH IF ERROR IS SET
2340  010504  062701  000010                    ADD     #10,R1
2341  010510  020127  020770                    CMP     R1,#TREG+20000
2342  010514  103765                            BLO     1$
2343  010516  013746  177776                    MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA A JSR
2344  010522  004767  002542                    JSR     PC,ERR                  ;ERROR- PARITY TRAP TO 114 OCCURRED DURING
2345                                                                            ;TEST 20 BUT NO REGISTERS HAVE PARITY
2346  010526  000430                            BR      5$                      ;ERROR SET
2347  010530  017102  000000            2$:     MOV     @(R1),R2                ;SAVE CONTENTS OF PARITY REGISTER
2348  010534  005003                            CLR     R3                      ;INITIALIZE ADDRESS POINTER
2349  010536  005071  000000            3$:     CLR     @(R1)                   ;CLEAR PARITY REGISTER
2350  010542  005713                            TST     @R3                     ;READ LOCATION
2351  010544  005771  000000                    TST     @(R1)                   ;PARITY ERROR SET?
2352  010550  100413                            BMI     4$                      ;YES- BRANCH
```

```
2353  010552  005723                  TST     (R3)+                   ;MOVE POINTER
2354  010554  020327  020000          CMP     R3,#20000
2355  010560  103766                  BLO     3$
2356  010562  010271  000000          MOV     R2,@(R1)                ;RESTORE PARITY REGISTER CONTENTS
2357  010566  013746  177776          MOV     @#PS,-(SP)              ;SETUP TO CALL ERR VIA JSR
2358  010572  004767  002472          JSR     PC,ERR                  ;PARITY ERROR OCCURRED WHILE TESTING
2359                                                                  ;BANK 0 BUT NO BAD PARITY WAS FOUND
2360                                                                  ;DURING SCAN OF BANK 0. R1 POINTS
2361                                                                  ;TO THE ADDRESS OF THE PARITY REGISTER
2362                                                                  ;DETECTING THE ERROR
2363  010576  000404                  BR      5$
2364  010600  013746  177776  4$:     MOV     @#PS,-(SP)              ;SETUP TO DO ERROR CALL VIA JSR
2365  010604  004767  002460          JSR     PC,ERR                  ;PARITY ERROR WHILE EXERCISING MEMORY
2366                                                                  ;BANK 0- R1 POINTS TO ADDRESS OF PARITY
2367                                                                  ;REGISTER DETECTING THE ERROR. R3 CONTAINS
2368                                                                  ;THE ADDRESS OF THE LOCATION HAVING
2369                                                                  ;BAD PARITY
2370  010610  022626          5$:     CMP     (SP)+,(SP)+
2371  010612  000207                  RTS     %7                      ;RETURN TO CHECK USING THE NEXT PATTERN
2372
2373
2374
2375                                  ;*************************************************************
2376                                  ;FORCE WRONG PARITY IN EACH LOCATION IN BANK 0
2377                                  ;NOTE THAT THIS SUBTEST IS EXECUTED IN BANK 1 (20000 ABOVE ADDRESSES
2378                                  ;IN THE LISTING). MAKE SURE THAT WRONG PARITY IN EACH BYTE CAN BE DETECTED,
2379                                  ;AND THAT WHEN GOOD PARITY IS WRITTEN AND READ NO PARITY ERROR IS DETECTED.
2380                                  ;CHECK ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11)
2381                                  ;*************************************************************
2382  010614  013746  177776  TEST21: MOV     @#PS,-(SP)              ;SAME AS SCOPE WITHOUT DOING EMT
2383  010620  004767  005230          JSR     PC,SCOPEC
2384  010624  012777  000021  170250  MOV     #21,@DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY
2385  010632  004767  002652          JSR     PC,CLRPAR               ;CLEAR ALL PARITY REGISTERS
2386  010636  005005                  CLR     R5
2387  010640  005025          1$:     CLR     (R5)+                   ;CLEAR PARITY ERRORS IN BANK 0
2388  010642  020527  020000          CMP     R5,#20000
2389  010646  103774                  BLO     1$
2390
2391                                  ;WRITE WRONG PARITY TEST ROUTINE
2392                                  ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
2393                                  ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
2394  010650  005005          WWP21:  CLR     R5                      ;SET TEST ADDRESS POINTER
2395  010652  005067  167702          CLR     @DDFLG                  ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
2396  010656  012767  125253  167650  MOV     #125253,SHDBE           ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
2397  010664  012715  125253  WWP21A: MOV     #125253,@R5             ;INITIALIZE LOCATION
2398  010670  012701  020570          MOV     #WWPR0+20000,R1         ;SETUP TO SET WRITE WRONG PARITY
2399                                                                  ;IN ALL REGISTERS
2400  010674  032711  000001  1$:     BIT     #1,(1)
2401  010700  001003                  BNE     .+10
2402  010702  012771  000004  000000  MOV     #WWP,@(R1)              ;SET WRITE WRONG PARITY
2403  010710  062701  000010          ADD     #10,R1
2404  010714  020127  020770          CMP     R1,#TREG+20000
2405  010720  103765                  BLO     1$
2406  010722  005767  167632          TST     @DDFLG                  ;TESTING HIGH BYTE?
2407  010726  100404                  BMI     2$                      ;YES, BRANCH
2408  010730  112715  000253          MOVB    #253,@R5                ;NO, WRITE WRONG PARITY IN LOW BYTE
```

```
2409  010734  005715                          TST     (R5)            ;DETECT WRONG PARITY WITH DATI
2410  010736  000405                          BR      3$
2411  010740  112765  000252  000001  2$:     MOVB    #252,1(R5)      ;WRITE WRONG PARITY IN HIGH BYTE
2412  010746  105765  000001                  TSTB    1(R5)           ;DETECT WRONG PARITY WITH DATI
2413  010752  012701  020370          3$:     MOV     #MPR0+20000,R1  ;SETUP TO CLEAR MAP IN ALL PARITY REGISTERS
2414  010756  012703  000772                  MOV     #INDCO,R3       ;POINTER TO INDICATOR
2415  010762  032711  000001          4$:     BIT     #1,@R1
2416  010766  001003                          BNE     .+10
2417  010770  042771  000004  000000          BIC     #4,@(R1)        ;CLEAR MAP IN ALL PARITY REGISTERS
2418  010776  062701  000010                  ADD     #10,R1
2419  011002  020127  020770                  CMP     R1,#TREG+20000
2420  011006  103765                          BLO     4$
2421  011010  012701  020570                  MOV     #MPR0+20000,R1  ;CHECK TO SEE IF ANY REGISTER DETECTED
2422  011014  005067  167750                  CLR     TREG            ;THE PARITY ERROR
2423  011020  032711  000001          LOOP21: BIT     #1,@R1
2424  011024  001024                          BNE     5$
2425  011026  005771  000000                  TST     @(R1)           ;CHECK PARITY REGISTER
2426  011032  100021                          BPL     5$              ;BRANCH IF PARITY ERROR IS NOT SET
2427  011034  005767  167730                  TST     TREG            ;WAS PARITY ERROR SET IN ANY OTHER REGISTER?
2428  011040  001404                          BEQ     .+12            ;NO- BRANCH
2429  011042  013746  177776                  MOV     @#PS,-(SP)      ;SETUP TO DO ERROR CALL VIA A JSR
2430  011046  004767  002216                  JSR     PC,ERR          ;ERROR- PARITY ERROR SET IN MORE THAN
2431                                                                  ;ONE REGISTER AFTER WRITING WRONG PARITY
2432                                                                  ;IN LOCATION WHOSE ADDRESS IS IN R5
2433  011052  032761  000001  000002          BIT     #1,2(R1)        ;DOES MAP INDICATE THAT THIS REGISTER
2434                                                                  ;CONTROLS THIS MEMORY?
2435  011060  001004                          BNE     .+12            ;YES, BRANCH
2436  011062  013746  177776                  MOV     @#PS,-(SP)      ;NO- SETUP TO DO ERROR CALL VIA A JSR
2437  011066  004767  002176                  JSR     PC,ERR          ;ERROR- PARITY REGISTER RESPONDED
2438                                                                  ;TO MEMORY WHICH IS NOT IN ITS MAP.
2439                                                                  ;R1 POINTS TO PARITY REGISTER'S ADDRESS.
2440                                                                  ;R5 CONTAINS ADDRESS OF LOCATION BEING
2441                                                                  ;TESTED
2442  011072  011167  167672                  MOV     @R1,TREG        ;STORE REGISTER ADDRESS
2443  011076  062701  000010          5$:     ADD     #10,R1
2444  011102  005723                          TST     (R3)+
2445  011104  020127  020770                  CMP     R1,#TREG+20000
2446  011110  103743                          BLO     LOOP21          ;BRANCH UNTIL ALL REGISTERS HAVE BEEN
2447                                                                  ;CHECKED
2448  011112  005767  167652                  TST     TREG            ;WAS PARITY ERROR SET IN ANY REGISTER?
2449  011116  001005                          BNE     6$              ;YES- BRANCH
2450  011120  013746  177776                  MOV     @#PS,-(SP)      ;NO- SETUP TO DO ERROR CALL VIA A JSR
2451  011124  004767  002140                  JSR     PC,ERR          ;ERROR- NO REGISTER HAS PARITY ERROR
2452                                                                  ;SET AFTER READING WRONG PARITY IN LOCATION
2453                                                                  ;WHOSE ADDRESS IS IN R5
2454  011130  000423                          BR      7$
2455  011132  032713  000001          6$:     CMP     #1,(R3)         ;IS THIS A CORE PAR REG?
2456  011136  001020                          BNE     7$              ;NO, BRANCH(MOS PAR REG DOES NOT
2457                                                                  ;STORE ADDR BITS OF PAR ERROR)
2458  011140  017701  167624                  MOV     @TREG,R1        ;GET PARITY REGISTER CONTENTS
2459  011144  042701  170037                  BIC     #170037,R1      ;MASK ALL BUT ERROR ADDRESS BITS
2460  011150  010502                          MOV     R5,R2           ;GET ADDRESS OF LOCATION UNDER TEST
2461  011152  042702  003777                  BIC     #3777,R2        ;POSITION BITS IN R2
2462  011156  000302                          SWAB    R2
2463  011160  006302                          ASL     R2
2464  011162  006302                          ASL     R2
```

```
2465  011164  020102                    CMP     R1,R2           ;PARITY ERROR ADDRESS BITS CORRECT?
2466  011166  001424                    BEQ     7$              ;BRANCH IF YES
2467  011170  013746  177776            MOV     @#PS,-(SP)      ;NO- SETUP TO DO ERROR CALL VIA A JSR
2468  011174  004767  002070            JSR     PC,ERR          ;ERROR- ADDRESS BITS (PARITY REGISTER
2469                                                            ;BITS 5-11) INCORRECT- ADDRESS OF PARITY
2470                                                            ;REGISTER IS CONTAINED IN LOCATION "TREG"
2471                                                            ;ADDRESS OF TEST LOCATION IS IN R5
2472  011200  011515            7$:     MOV     @R5,@R5         ;RESTORE TEST LOCATION TO FIX BAD PARITY
2473  011202  005077  167562            CLR     @TREG           ;CLEAR ERROR BIT IN PARITY REGISTER
2474  011206  005715                    TST     @R5             ;RE-D LOCATION TO SET IF PARITY IS GOOD
2475  011210  005777  167554            TST     @TREG           ;CHECK PARITY ERROR BIT
2476  011214  100004                    BPL     .+12            ;BRANCH IF NO
2477  011216  013746  177776            MOV     @#PS,-(SP)      ;SETUP TO DO       CALL VIA A JSR
2478  011222  004767  002042            JSR     PC,ERR          ;ERROR- WRITE       CATION WITH WRITE
2479                                                            ;WRONG PARITY BUT DIDN'T CLEAR BAD
2480                                                            ;PARITY (PC       F LOCATION IS IN R5)
2481                                                            ;"TREG" CONTAINS THE ADDRESS
2482                                                            ;OF THE PARITY REGISTER
2483  011226  005167  167326            COM     ODDFLG          ;TOGGLE BYTE INDICATOR
2484  011232  100401                    BMI     .+4             ;BRANCH IF HIGH BYTE NOT YET TESTED
2485  011234  005725                    TST     (R5)+           ;UPDATE ADDRESS POINTER
2486  011236  020527  020000            CMP     R5,#20000       ;THIS 4K DONE?
2487  011242  103610                    BLO     @#P21A          ;LOOP TILL ALL 4K HAS BEEN TESTED
2488  011244  004767  002240            JSR     PC,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2489  011250  013746  177776            MOV     @#PS,-(SP)      ;SETUP TO CALL SCOPE VIA JSR
2490  011254  004767  004574            JSR     PC,SCOPEC       ;SCOPE
2491
2492                                    ;COPY SECOND 4K BANK BACK TO FIRST 4K AND RETURN TO FIRST 4K BANK
2493  011260  012700  010000    XFR2:   MOV     #10000,R0       ;R0 IS USED AS A COUNTER
2494  011264  005001                    CLR     R1              ;R1 POINTS TO THE CURRENT LOCATION
2495  011266  016111  020000    1$:     MOV     20000(R1),@R1   ;COPY BANK 1 TO BANK 0
2496  011272  005721                    TST     (R1)+
2497  011274  005300                    DEC     R0
2498  011276  001373                    BNE     1$
2499  011300  162706  020000            SUB     #20000,SP       ;RESTORE STACK POINTER
2500  011304  162737  020000  013340    SUB     #20000,@#ERRA1
2501  011312  162737  020000  013342    SUB     #20000,@#ERRA2
2502  011320  162737  020000  013352    SUB     #20000,@#ERRA3
2503  011326  162737  020000  013354    SUB     #20000,@#ERRA4
2504  011334  162737  020000  013364    SUB     #20000,@#ERRA5
2505  011342  022737  177570  001100    CMP     #177570,@#SWR   ;DOES THE CONSOLE HAVE A SWITCH REG. ?
2506  011350  001403                    BEQ     2$              ;IF SO THEN GO TO 2$
2507  011352  162737  020000  001100    SUB     #20000,@#SWR    ;OTHERWISE RESTORE THE ADDRESS OF SOFTWARE
2508                                                            ;SWITCH REGISTER
2509  011360  000137  011364    2$:     JMP     @#DONE          ;RETURN TO BANK 0
2510
2511
2512
2513                                    ;*************************************************************************
2514                                    ;AT THIS POINT EXECUTION RETURNS TO BANK 0
2515                                    ;*************************************************************************
2516  011364  012737  000116  000114 DONE:   MOV     #PARVEC+2,@#PARVEC              ;RESTORE TRAPCATCHER
2517  011372  012706  000510            MOV     #STKPT,SP       ;REINITIALIZE STACK POINTER
2518  011376  004767  002106            JSR     %7,CLRPAR       ;CLEAR ALL PARITY REGISTERS
2519  011402  005267  167136            INC     PASCNT          ;KEEP TRACK OF PASSES COMPLETED
2520  011406  004567  003502            JSR     R5,DACNV
```

```
2521  011412  000544                          PASCNT
2522  011414  017057                          MPCNT
2523  011416  000006                          6
2524  011420  104000                          TYPE                            ;TYPE BELL, "END PASS=" AND PASS COUNT
2525  011422  017041                          MPGEND
2526  011424  013705  000042          LOGICAL:        MOV     @#42,R5         ;LOADED BY MONITOR?
2527  011430  001405                          BEQ     CONT                    ;BRANCH IF NO
2528  011432  000005                          RESET                           ;SETUP FOR MONITOR EXIT
2529  011434  004715                  SENDAD: JSR     7,(5)
2530  011436  000240                          NOP
2531  011440  000240                          NOP
2532  011442  000240                          NOP
2533  011444  032777  000400  167426  CONT:   BIT     #BITB,@SWR              ;SWITCH B SET?
2534  011452  001401                          BEQ     .+4
2535  011454  000000                          HALT                            ;HALT AT END OF PASS SET
2536  011456  105767  167430                  TSTB    $TPFLG
2537  011462  001006                          BNE     1$                      ;IF NO TERMINAL, SKIP
2538  011464  105777  167414                  TSTB    @TPS                    ;WAIT FOR TTY TO FINISH SO THAT RESET
2539  011470  100375                          BPL     .-4                     ;WON'T CLOBBER THE BELL
2540  011472  112777  000000  167406          MOVB    #0,@TPB                 ;OUTPUT A NULL
2541  011500  000167  170336          1$:     JMP     BEGIN
2542
2543
2544
2545
2546                                  ;*************************************************************************
2547                                  ;CREATE MAP INDICATING WHERE 4K BLOCKS OF MEMORY ARE PRESENT
2548                                  ;*************************************************************************
2549  011504  012737  011710  000004  MAPMEM: MOV     #MAPMB,@#4              ;SET NO MEM MANAGEMENT TRAP
2550  011512  005737  177572                  TST     @#SR0                   ;IS KT PRESENT? (TIMEOUT IF NO)
2551
2552                                  ;MAP MEMORY USING KT11 - MAX OF 124K POSSIBLE
2553  011516  005067  167030          MAPMA:  CLR     NOKT                    ;INDICATE KT11 PRESENT
2554  011522  004767  001662                  JSR     %7,NRALL                ;INITIALLY SET ALL PAGES NONRESIDENT, BANK 0
2555  011526  004767  002026                  JSR     %7,MAP1                 ;MAP KERNEL 0 TO BANK 0, RW
2556                                                                          ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
2557                                                                          ;MAP KERNEL 1 RW, AND TURN ON KT11
2558  011532  005067  167010                  CLR     TBANK
2559  011536  012767  177777  167322          MOV     #177777,MEML            ;SET UP CORE MAPS
2560  011544  012767  077777  167316          MOV     #77777,MEMH
2561  011552  012767  000001  166740          MOV     #1,BITPT                ;SET UP 4K POINTER
2562  011560  012767  001066  166762          MOV     #MEML,MEMUT
2563  011566  012737  011676  000004          MOV     #5$,@#4                 ;SET UP FOR TIME OUTS
2564  011574  016777  166746  167444  2$:     MOV     TBANK,@KPAR1            ;MAP KERNEL PAGE 1 TO BANK BEING TESTED
2565  011602  005737  021000                  TST     @#21000                 ;1ST K PRESENT?
2566  011606  005737  025000                  TST     @#25000                 ;2ND K PRESENT?
2567  011612  005737  031000                  TST     @#31000                 ;3RD K PRESENT
2568  011616  005737  035000                  TST     @#35000                 ;4TH K PRESENT?
2569  011622  062767  000200  166716  3$:     ADD     #200,TBANK              ;UPDATE TEST ADDRESS
2570  011630  006367  166664                  ASL     BITPT                   ;UPDATE BANK POINTER
2571  011634  103006                          BCC     4$                      ;BRANCH IF NOT DONE WITH 64K SECTION
2572  011636  012767  000001  166654          MOV     #1,BITPT                ;YES, DO MEMH(64-124K)
2573  011644  012767  001070  166676          MOV     #MEMH,MEMUT
2574  011652  022767  007600  166666  4$:     CMP     #7600,TBANK             ;EXTERNAL BANK YET?
2575  011660  003345                          BGT     2$                      ;NO, NOT YET
2576  011662  005037  177572                  CLR     @#SR0                   ;YES- DISABLE KT11
```

```
2577  011666  012737  000006  000004          MOV     #6,@#4                  ;RESTORE TRAPCATCHER
2578  011674  000207                          RTS     %7                      ;RETURN
2579  011676  046777  166616  166644  5$:     BIC     BITPT,@MEMUT            ;TIMEOUT OCCURRED-CLEAR BIT TO INDICATE
2580                                                                          ;4K BLOCK NOT PRESENT
2581  011704  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK
2582  011706  000745                          BR      3$                      ;CHECK NEXT BLOCK
2583
2584                                  ;NO KT PRESENT - MAP MAX OF 28K IN 4K CONTIGUOUS BLOCKS
2585  011710  012767  000001  166634  MAPMB:  MOV     #1,NOKT                 ;SET FLAG TO INDICATE KT11 NOT PRESENT
2586  011716  022626                          CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
2587  011720  012737  012020  000004          MOV     #3$,@#4                 ;SET UP TIMEOUT RETURN
2588  011726  012767  000177  167132          MOV     #177,MEML               ;INITIALIZE MAP
2589  011734  005067  167130                  CLR     MEMH
2590  011740  012767  000001  166552          MOV     #1,BITPT                ;SETUP 4K POINTER
2591  011746  005001                          CLR     R1                      ;INITIALIZE 4K ADDRESS
2592  011750  005761  001000          1$:     TST     1000(1)                 ;1ST K PRESENT
2593  011754  005761  005000                  TST     5000(1)                 ;2ND K PRESENT
2594  011760  005761  011000                  TST     11000(1)                ;3RD K PRESENT
2595  011764  005761  015000                  TST     15000(1)                ;4TH K PRESENT
2596  011770  062701  020000          2$:     ADD     #20000,R1               ;UPDATE TEST ADDRESS
2597  011774  006367  166520                  ASL     BITPT                   ;UPDATE POINTER TO NEXT 4K
2598  012000  022767  000200  166512          CMP     #200,BITPT              ;28K CHECKED YET?
2599  012006  003360                          BGT     1$                      ;NO, CHECK NEXT 4K BLOCK
2600  012010  012737  000006  000004          MOV     #6,@#4
2601  012016  000207                          RTS     %7
2602  012020  046767  166474  167040  3$:     BIC     BITPT,MEML              ;TIMEOUT OCCURRED- CLEAR BIT TO
2603                                                                          ;INDICATE 4K BLOCK NOT PRESENT
2604  012026  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK POINTER
2605  012030  000757                          BR      2$
2606
2607
2608
2609                                  ;*************************************************************************
2610                                  ;MAP PARITY CORE AND CORRESPONDENCE TO ASSOCIATED REGISTERS
2611                                  ;*************************************************************************
2612  012032  013767  001066  166510  MAPREG: MOV     @#MEML,MEMUT            ;LOAD MAP OF MEMORY PRESENT IN LOWER 64K
2613  012040  012767  000001  166452          MOV     #1,BITPT                ;INITIALIZE 4K POINTER
2614  012046  012767  000001  167146          MOV     #1,KTSTART              ;INDICATE KT11 NOT IN USE
2615  012054  005067  166506                  CLR     RELOC
2616  012060  005067  166462                  CLR     TBANK                   ;INITIALIZE ADDRESS OF TEST BANK
2617  012064  005067  166464                  CLR     HIWORD                  ;CLEAR FLAG TO INDICATE FIRST 64K
2618                                                                          ;BEING CHECKED
2619  012070  005067  167130                  CLR     ADRTYP
2620
2621                                  ;SET WRITE WRONG PARITY IN ALL REGISTERS PRESENT
2622                                  ;THEN WRITE TEST LOCATION VIA DATO AND READ TEST LOCATION VIA DATI
2623                                  ;THEN CLEAR WRITE WRONG PARITY IN ALL REGISTERS
2624  012074  012702  000570          MAPRB:  MOV     #MPR0,R2                ;LOAD ADDRESS OF TABLE
2625  012100  032712  000001          1$:     BIT     #1,(2)                  ;IS THIS REGISTER PRESENT?
2626  012104  001003                          BNE     .+10                    ;NO, GET NEXT ONE
2627  012106  012772  000004  000000          MOV     #WWP,@(2)               ;YES, SET WRITE WRONG PARITY AND CLEAR REST
2628  012114  062702  000010                  ADD     #10,R2
2629  012120  020227  000770                  CMP     R2,#TREG                ;DONE WITH TABLE?
2630  012124  103765                          BLO     1$                      ;BRANCH IF NOT
2631  012126  016703  166414                  MOV     TBANK,R3                ;LOAD ADDRESS OF 4K BANK UNDER TEST
2632  012132  066703  167066                  ADD     ADRTYP,R3               ;ADD ADDRESS OFFSET (EITHER 0,2,4,OR 6)
```

```
2633  012136  011313                          MOV     (3),(3)          ;WRITE WRONG PARITY
2634  012140  005713                          TST     (3)              ;READ WRONG PARITY
2635  012142  012702  000570                  MOV     #MPRO,R2
2636  012146  032712  000001          2$:     BIT     #1,(2)
2637  012152  001003                          BNE     .+10
2638  012154  042772  000004  000000          BIC     #WWP,@(2)        ;CLEAR WRITE WRONG PARITY IN ALL
2639                                                                   ;PARITY REGISTERS
2640  012162  062702  000010                  ADD     #10,R2
2641  012166  020227  000770                  CMP     R2,#TREG
2642  012172  103765                          BLO     2$
2643  012174  012702  000750          MAPRC:  MOV     #MPRO-10,R2      ;INIT FOR ERROR CHECKS
2644  012200  062702  000010          1$:     ADD     #10,R2
2645  012204  020227  000770                  CMP     R2,#TREG
2646  012210  002031                          BGE     MAPRD            ;BRANCH IF DONE WITH TABLE
2647  012212  032712  000001                  BIT     #1,(2)           ;IS THIS REGISTER PRESENT?
2648  012216  001370                          BNE     1$               ;NO, GET NEXT ADDRESS
2649  012220  005772  000000                  TST     @(R2)            ;YES, DID THIS CONTROLLER GET A
2650                                                                   ;PARITY ERROR?
2651  012224  100365                          BPL     1$               ;NO, CHECK NEXT
2652  012226  005767  166322                  TST     HIWORD           ;YES, WHICH 64K IS UNDER TEST?
2653  012232  001004                          BNE     2$               ;BRANCH IF UPPER 64K
2654  012234  056762  166260  000002          BIS     BITPT,2(2)       ;SET BIT IN MAP FOR THIS PARITY REGISTER
2655  012242  000403                          BR      3$
2656  012244  056762  166250  000004  2$:     BIS     BITPT,4(2)       ;SET BIT IN MAP FOR THIS PARITY REGISTER
2657  012252  105762  000007          3$:     TSTB    7(2)             ;IS THIS THE FIRST ADDRESS FOUND FOR
2658                                                                   ;THIS PARITY REGISTER?
2659  012256  001005                          BNE     4$               ;NO, BRANCH
2660  012260  116762  166740  000006          MOVB    ADRTYP,6(2)      ;YES, RECORD LOW BYTE OF ADDRESS (0,2,4,OR 6)
2661  012266  105262  000007                  INCB    7(2)             ;INDICATE AN ADDRESS HAS BEEN FOUND FOR
2662                                                                   ;THIS REGISTER
2663  012272  000742                  4$:     BR      1$
2664  012274  011313          MAPRD:  MOV     @R3,@R3                  ;CLEAR BAD PARITY
2665  012276  062767  000002  166720          ADD     #2,ADRTYP        ;CHECK FIRST 4 ADDRESSES IN EACH 4K
2666  012304  026727  166714  000010          CMP     ADRTYP,#10
2667  012312  001402                          BEQ     1$               ;BRANCH IF FIRST 4 ADDRESSES TESTED
2668  012314  000167  177554                  JMP     MAPRB            ;IF NOT, GO TEST NEXT ONE
2669  012320  005767  166230          1$:     TST     HIWORD           ;IS LOWER MEMORY DONE?
2670  012324  001021                          BNE     MAPRE            ;YES, BRANCH
2671  012326  026727  166166  000100          CMP     BITPT,#100       ;DONE WITH 1ST 28K?
2672  012334  103015                          BHIS    MAPRE            ;YES, BRANCH
2673  012336  062767  020000  166202          ADD     #20000,TBANK     ;NO- ADD 4K TO ADDRESS
2674  012344  006367  166150                  ASL     BITPT            ;SHIFT BIT POINTER
2675  012350  005067  166650                  CLR     ADRTYP           ;START WITH FIRST ADDRESS IN BANK
2676  012354  036767  166140  166166          BIT     BITPT,MEMUT      ;DOES THIS 4K BLOCK EXIST?
2677  012362  001756                          BEQ     1$               ;NO- BRANCH
2678  012364  000167  177504                  JMP     MAPRB            ;YES, TEST IT
2679  012370  005767  166156          MAPRE:  TST     NOKT             ;KT11 PRESENT?
2680  012374  001401                          BEQ     .+4              ;YES, BRANCH
2681  012376  000207                          RTS     %7               ;NO, DONE
2682  012400  005767  166616                  TST     KTSTART          ;KT11 ALREADY ON?
2683  012404  001417                          BEQ     1$               ;YES, BRANCH
2684  012406  012767  020000  166132          MOV     #20000,TBANK     ;NO, INIT TBANK TO SELECT KERNEL PAGE 1
2685  012414  012767  001400  166144          MOV     #1400,RELOC      ;SET UP FOR ACCESS TO 28K BANK
2686  012422  004767  000762                  JSR     %7,NRALL         ;INITIALLY MAP ALL PAGES NR, BANK 0
2687  012426  004767  001126                  JSR     %7,MAP1          ;MAP KERNEL 0 TO BANK 0, RW
2688                                                                   ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
```

```
                                                                ;SET KERNEL 1 RW AND TURN ON KT11
012432  005067  166564          CLR     KTSTART               ;INDICATE KT11 NOW IN USE
012436  012737  000001  177572  MOV     #1,@#SR0                      ;TURN ON KT11
012444  006367  166050    1$:   ASL     BITPT                 ;SHIFT BANK INDICATOR
012450  103011                  BCC     2$                    ;BRANCH IF FIRST 64K NOT DONE
012452  012767  000001  166040  MOV     #1,BITPT              ;IF FIRST 64K DONE, SETUP FOR
012460  013767  001070  166062  MOV     @#MEMH, MEMUT         ;SECOND 64K
012466  012767  000001  166060  MOV     #1,HIWORD             ;INDICATE NOW TESTING HIGH 64K
012474  062767  000200  166064  2$:    ADD     #200,RELOC
012502  022767  007600  166056  CMP     #7600,RELOC           ;UP TO EXTERNAL BANK YET?
012510  003003                  BGT     3$                    ;NO, CONTINUE
012512  005037  177572          CLR     @#SR0                 ;YES, TURN OFF KT11 AND EXIT
012516  000207                  RTS     %7
012520  036767  165774  166022  3$:    BIT     BITPT,MEMUT   ;IS THIS 4K PRESENT?
012526  001746                  BEQ     1$                    ;NO- BRANCH
012530  016777  166032  166510  MOV     RELOC,@KPAR1          ;YES, MAP PAGE 1 TO THIS BANK
012536  005067  166462          CLR     ADRTYP
012542  000167  177326          JMP     MAPR8                 ;GO TEST FOR PARITY MEMORY


                        ;*************************************************************************
                        ;ROUTINE TO TYPE MAP OF WHERE PARITY MEMORY IS PRESENT
                        ;AND WHICH CONTROL REGISTERS CONTROL WHICH MEMORY
                        ;*************************************************************************
012546  004767  000736  TMAP:   JSR     %7,CLRPAR             ;CLEAR ALL PARITY REGISTERS PRESENT
012552  104000                  TYPE                          ;TYPE "THE PARITY REGISTERS CONTROL MEMORY
012554  016564                  MTMAP                         ;AS FOLLOWS:"
012556  012701  000560          MOV     #MPR0-10,R1           ;SET UP POINTER
012562  062701  000010  TMAPA:  ADD     #10,R1
012566  020127  000770          CMP     R1,#TREG              ;DONE WITH MAP?
012572  002136                  BGE     TMAPEX                ;YES, BRANCH
012574  005067  165754          CLR     HIWORD
012600  005067  165716          CLR     TRFLG                 ;INITIALIZE TRANSITION FLAG (USED TO
                                                              ;FIGURE MEMORY LIMITS)
012604  005067  165714          CLR     TYFLG                 ;INITIALIZE TO INDICATE NOTHING TYPED FOR
                                                              ;THIS REGISTER YET
012610  012767  177774  165710  MOV     #-4,TYCOR
012616  016167  000002  165672  MOV     2(1),ADRPT            ;GET LOW 64K MEMORY MAP WORD
012624  012767  000001  165666  MOV     #1,BITPT              ;INITIALIZE 4K POINTER
012632  032711  000001          BIT     #1,(1)                ;DOES THIS CONTROL EXIST?
012636  001351                  BNE     TMAPA                 ;NO, GET ADDRESS OF NEXT ONE
012640  011167  165650          MOV     (1),TEMPX             ;YES, PRINT ITS ADDRESS
012644  004567  002244          JSR     R5,@ACNV
012650  000514                  TEMPX
012652  016733                  MPRAD
000006                          6
104000                          TYPE
017125                          MX1
104000                          TYPE
016733                          MPRAD
012666  062767  000004  165632  TMAPB:  ADD     #4,TYCOR              ;KEEP TRACK OF # OF K OF CORE
012674  036767  165620  165614          BIT     BITPT,ADRPT           ;DOES THIS PARITY REGISTER CONTROL THIS 4K?
012702  001424                  BEQ     TMAPC                 ;NO- BRANCH
012704  005767  165612          TST     TRFLG                 ;YES, DOES IT CONTROL PREVIOUS 4K?
012710  001043                  BNE     TMAPD                 ;YES- DON'T TYPE IT
```

```
2745  012712  012767  000001  165602          MOV     #1,TRFLG                ;NO- SET FLAG INDICATING TRANSITION
2746  012720  004567  002276                  JSR     R5,BDCNV                ;CONVERT K CORE TO ASCII
2747  012724  000526                          TYCOR
2748  012726  016633                          MTYCOR
2749  012730  000003                          3
2750  012732  104000                          TYPE                            ;TYPE "CONTROLS", AND ADDRESS OF CORE
2751  012734  017144                          MX2
2752  012736  104000                          TYPE
2753  012740  016633                          MTYCOR
2754  012742  104000                          TYPE
2755  012744  016640                          MDASH
2756  012746  005267  165552                  INC     TYFLG                   ;INDICATE TYPED
2757  012752  000422                          BR      TMAPD
2758  012754  005767  165542          TMAPC:  TST     TRFLG                   ;DID THIS PARITY REGISTER CONTROL PREVIOUS 4K?
2759  012760  001417                          BEQ     TMAPD                   ;NO, SKIP PRINTING
2760  012762  005067  165534                  CLR     TRFLG                   ;YES, TRANSITION OCCURRED- CLEAR FLAG
2761  012766  004567  002230                  JSR     R5,BDCNV                ;CONVERT K CORE TO ASCII
2762  012772  000526                          TYCOR
2763  012774  016633                          MTYCOR
2764  012776  000003                          3
2765  013000  104000                          TYPE
2766  013002  016633                          MTYCOR                          ;TYPE RIGHT AND RETURN
2767  013004  104000                          TYPE
2768  013006  016646                          MK
2769  013010  104000                          TYPE
2770  013012  016643                          MCR
2771  013014  005267  165504                  INC     TYFLG                   ;INDICATE TYPED
2772  013020  006367  165474          TMAPD:  ASL     BITPT                   ;UPDATE BIT POINTER TO NEXT 4K
2773  013024  103320                          BCC     TMAPB                   ;TEST NEXT 4K IF NOT DONE WITH 1ST 64K
2774  013026  005767  165522                  TST     HIWORD                  ;64-124K DONE?
2775  013032  001405                          BEQ     1$                      ;NO, BRANCH
2776  013034  005767  165464                  TST     TYFLG                   ;YES, WAS ANY PARITY MEMORY
2777                                                                          ;FOUND FOR THIS REGISTER?
2778
2779  013040  001250                          BNE     TMAPA
2780  013042  104002                          ERROR                           ;NO PARITY MEMORY WAS FOUND FOR THIS
2781                                                                          ;REGISTER- EITHER WRITE WRONG PARITY
2782                                                                          ;FAILED, PARITY ERROR GENERATE OR
2783                                                                          ;DETECT FAILED, OR THE PARITY ERROR
2784                                                                          ;BIT FAILED TO SET
2785  013044  000646                          BR      TMAPA
2786  013046  016167  000004  165442  1$:     MOV     4(1),ADRPT              ;UPDATE TO MAP WORD FOR THE UPPER 64K
2787  013054  012767  000001  165436          MOV     #1,BITPT                ;RESET BIT POINTER
2788  013062  005267  165466                  INC     HIWORD                  ;INDICATE LOW 64K DONE
2789  013066  000677                          BR      TMAPB                   ;LOOP
2790  013070  000207                  TMAPEX: RTS     %7                      ;RETURN WHEN DONE
2791
2792
2793
2794                                          ;*****************************************************************************
2795                                          ;ERROR HANDLER
2796                                          ;*****************************************************************************
2797                                          ;ERRORS CALL ENTERS HERE
2798                                          ;TYPES PC, ICNT, MPR ADDRESS, AND MPR CONTENTS
2799                                          ;TREG SHOULD CONTAIN ADDRESS OF PARITY REGISTER
2800  013072  012767  016207  000266  ERRST:  MOV     #MSTR,ERRB              ;SETUP TO TYPE MPR ADDRESS AND CONTENTS
```

```
2801  013100  012767  177777  000262          MOV     #-1,ERRBX
2802  013106  012767  000240  000256          MOV     #240,ERRBX+2            ;NOP LOCATION AFTER MESSAGE
2803  013114  017767  165650  165416          MOV     @TREG,TRDATA            ;SETUP DATA
2804  013122  004567  001766                  JSR     R5,OACNV                ;CONVERT TO ASCII
2805  013126  000770                          TREG
2806  013130  016216                          MTREG
2807  013132  000006                          6
2808  013134  004567  001754                  JSR     R5,OACNV                ;CONVERT TO ASCII
2809  013140  000540                          TRDATA
2810  013142  016240                          MDATA
2811  013144  000006                          6
2812  013146  000461                          BR      ERRA
2813
2814
2815                                  ;ERRORP CALL ENTERS HERE
2816                                  ;TYPES PC, ICNT, MPR ADDRESS, MPR CONTENTS
2817                                  ;TEST LOCATION ADDRESS, VALUE EXPECTED, VALUE FOUND
2818                                  ;R5 MUST CONTAIN ADDRESS OF TEST LOCATION
2819                                  ;SHOBE MUST CONTAIN EXPECTED VALUE
2820                                  ;WAS MUST CONTAIN ACTUAL DATA
2821                                  ;TREG MUST CONTAIN ADDRESS OF PARITY REGISTER
2822  013150  012767  016207  000210  ERRP:   MOV     #MSTR,ERRB              ;IN ADDITION TO BASIC PRINTOUT, TYPE
2823                                                                          ;MPR ADDRESS AND CONTENTS
2824  013156  012767  016251  000204          MOV     #MSTRX,ERRBX            ;ALSO OUTPUT DATA EXPECTED AND ACTUAL
2825  013164  012767  177777  000200          MOV     #-1,ERRBX+2             ;NOP LOCATION AFTER MESSAGE
2826  013172  010567  165334                  MOV     R5,TSTLOC               ;STORE ADDRESS BEING TESTED
2827  013176  017767  165566  165334          MOV     @TREG,TRDATA
2828  013204  004567  001704                  JSR     R5,OACNV
2829  013210  000770                          TREG
2830  013212  016216                          MTREG
2831  013214  000006                          6
2832  013216  004567  001672                  JSR     R5,OACNV
2833  013222  000540                          TRDATA
2834  013224  016240                          MDATA
2835  013226  000006                          6
2836  013230  004567  001660                  JSR     R5,OACNV
2837  013234  000532                          TSTLOC
2838  013236  016275                          MSTRX1
2839  013240  000006                          6
2840  013242  004567  001646                  JSR     R5,OACNV
2841  013246  000534                          SHOBE
2842  013250  016311                          MSTRX3
2843  013252  000006                          6
2844  013254  004567  001634                  JSR     R5,OACNV
2845  013260  000536                          WAS
2846  013262  016325                          MSTRX5
2847  013264  000006                          6
2848  013266  000411                          BR      ERRA
2849
2850
2851
2852                                  ;ERROR CALL ENTERS HERE
2853                                  ;TYPE PC AND ICNT ONLY
2854  013270  012767  177777  000070  ERR:    MOV     #-1,ERRB                ;SET UP ONE MESSAGE CALL
2855  013276  012767  000240  000064          MOV     #240,ERRBX
2856  013304  012767  000240  000060          MOV     #240,ERRBX+2
```

```
2857  013312  032777  020000  165560  ERRA:   BIT     @#BIT13,@SWR            ;INHIBIT ERROR PRINT?
2858  013320  001025                          BNE     ERRC                    ;YES- BRANCH
2859  013322  011667  000060                  MOV     (SP),ERRD               ;NO- DEVELOP CALLING ADDRESS
2860  013326  162767  000002  000052          SUB     #2,ERRD
2861  013334  004567  001554                  JSR     R5,OACNV                ;GO TO OCTAL TO ASCII CONVERT
2862  013340  013406                  ERRA1:  ERRD                            ;SOURCE ADDRESS
2863  013342  016162                  ERRA2:  MPC                             ;DESTINATION ADDRESS
2864  013344  000006                          6                               ;#OF DIGITS TO CONVERT
2865  013346  004567  001542                  JSR     R5,OACNV                ;CONVERT ICNT TO ASCII
2866  013352  016150                  ERRA3:  ICNT
2867  013354  016200                  ERRA4:  MICNT
2868  013356  000006                          6
2869  013360  004567  001476                  JSR     R5,TYPSX                ;TYPE MESSAGE
2870  013364  016154                  ERRA5:  MEO                             ;ERROR HEADER
2871  013366  000000                  ERRB:   OPEN                            ;ADDITIONAL ERROR MESSAGES IF ANY
2872  013370  000000                  ERRBX:  OPEN
2873  013372  177777                          -1
2874  013374  005777  165500          ERRC:   TST     @SWR                    ;HALT ON ERROR SET?
2875  013400  100001                          BPL     .+4                     ;NO- BRANCH
2876  013402  000000                          HALT                            ;YES- ERROR OCCURRED SO HALT
2877  013404  000002                          RTI
2878  013406  000000                  ERRD:   OPEN
2879
2880
2891
2892                                  ;MAP ALL PAGES NON-RESIDENT, BANK 0
2883  013410  013746  000004          MRALL:  MOV     @#4,-(SP)
2884  013414  013746  000006                  MOV     @#6,-(SP)
2885  013420  012737  000006  000004          MOV     #6,@#4
2886  013426  012737  000002  000006          MOV     #RTI,@#6
2887  013434  010146                          MOV     R1,-(SP)
2888  013436  010246                          MOV     R2,-(SP)
2889  013440  010346                          MOV     R3,-(SP)
2890  013442  012701  001226                  MOV     #PORTAB,R1
2891  013446  012703  000040          1$:     MOV     #32.,R3
2892  013452  012102                          MOV     (R1)+,R2
2893  013454  005022                  2$:     CLR     (R2)+
2894  013456  005303                          DEC     R3
2895  013460  001375                          BNE     2$
2896  013462  020127  001232                  CMP     R1,#POREND
2897  013466  003767                          BLE     1$
2898  013470  012603                          MOV     (SP)+,R3
2899  013472  012602                          MOV     (SP)+,R2
2900  013474  012601                          MOV     (SP)+,R1
2901  013476  012637  000006                  MOV     (SP)+,@#6
2902  013502  012637  000004                  MOV     (SP)+,@#4
2903  013506  000207                          RTS     %7
2904
2905
2906
2907                                  ;ROUTINE TO CLEAR ALL PARITY REGISTERS PRESENT
2908  013510  010146                  CLRPAR: MOV     R1,-(SP)
2909  013512  010246                          MOV     R2,-(SP)
2910  013514  010701                          MOV     PC,R1
2911  013516  062701  165052                  ADD     #MPR0-.,R1
2912  013522  010702                          MOV     PC,R2
```

```
2913  013524  062702  165244                  ADD     #TREG-.,R2
2914  013530  032711  000001          1$:     BIT     #1,@R1                  ;IS THIS REGISTER PRESENT?
2915  013534  001002                          BNE     .+6
2916  013536  005071  000000                  CLR     @(R1)                   ;CLEAR ALL PARITY REGISTERS
2917  013542  062701  000010                  ADD     #10,R1
2918  013546  020102                          CMP     R1,R2
2919  013550  103767                          BLO     1$
2920  013552  012602                          MOV     (SP)+,R2
2921  013554  012601                          MOV     (SP)+,R1
2922  013556  000207                          RTS     %7
2923
2924
2925
2926                                  ;ROUTINE TO MAP KERNEL 0 TO BANK 0 READ/WRITE.
2927                                  ;KERNEL 1 READ/WRITE BUT BANK MAPPED BY CALLING ROUTINE,
2928                                  ;AND KERNEL 7 TO EXTERNAL BANK, READ/WRITE
2929  013560  012777  077406  165446  MAP1:   MOV     #77406,@KPDR0
2930  013566  012777  077406  165442          MOV     #77406,@KPDR1
2931  013574  012777  077406  165440          MOV     #77406,@KPDR7
2932  013602  012777  007600  165442          MOV     #7600,@KPAR7
2933  013610  005077  165430                  CLR     @KPAR0
2934  013614  012737  000001  177572          MOV     #1,@#SR0
2935  013622  000207                          RTS     %7
2936
2937
2938
2939                                  ;ROUTINE TO LOCATE THE FIRST PARITY MEMORY ADDRESS (ABOVE BANK 0)
2940                                  ;CORRESPONDING TO A GIVEN PARITY REGISTER-REQUIRES THAT THE ROUTINES
2941                                  ;MAPMEM AND MAPREG HAVE ALREADY BEEN RUN
2942                                  ;TO USE, PUT THE ADDRESS OF THE ADDRESS OF THE REGISTER IN R0 (I.E. POINT TO
2943                                  ;TO MAP TABLE)-THE DESIRED ADDRESS IS RETURNED IN R1, USING KERNEL PAGE 1 IF
2944                                  ;KT11 IS PRESENT
2945  013624  010246                  LOCATM: MOV     R2,-(SP)
2946  013626  012702  000200                  MOV     #200,R2
2947  013632  012701  000002                  MOV     #2,R1                   ;SKIP USE OF BANK 0
2948  013636  005767  164710          1$:     TST     NOKT                    ;KT11 PRESENT?
2949  013642  001403                          BEQ     3$                      ;YES, BRANCH
2950  013644  022701  000200                  CMP     #200,R1                 ;NO, CHECK ONLY FOR MEMORY IN FIRST 28K
2951  013650  101420                          BLOS    4$                      ;IF NO MEMORY FOUND IN 1ST 28K, GIVE
2952                                                                          ;ERROR RETURN
2953  013652  030160  000002          3$:     BIT     R1,2(R0)                ;DOES THIS 4K CORRESPOND TO THIS REGISTER?
2954  013656  001021                          BNE     LOCAT1                  ;YES, BRANCH
2955  013660  062702  000200                  ADD     #200,R2                 ;NO, CHECK TO SEE IF NEXT 4K CORRESPONDS
2956  013664  006301                          ASL     R1
2957  013666  103363                          BCC     1$
2958  013670  012701  000001                  MOV     #1,R1
2959  013674  030160  000004          2$:     BIT     R1,4(R0)                ;CHECK HIGH 64K
2960  013700  001010                          BNE     LOCAT1
2961  013702  062702  000200                  ADD     #200,R2
2962  013706  006301                          ASL     R1
2963  013710  103371                          BCC     2$
2964  013712  012701  000001          4$:     MOV     #1,R1                   ;NO PARITY MEMORY CORRESPONDS TO
2965  013716  012602                          MOV     (SP)+,R2                ;THIS REGISTER- RETURN WITH ERROR
2966  013720  000207                          RTS     %7                      ;INDICATION
2967  013722  005767  164624          LOCAT1: TST     NOKT                    ;KT11 PRESENT?
2968  013726  001005                          BNE     1$                      ;NO- BRANCH OVER
```

```
2969  013730  010277  165312                  MOV     R2,@#PAR1               ;YES- SETUP R1 TO REFERENCE 4K
2970  013734  012701  020000                  MOV     #20000,R1               ;BANK USING KERNEL PAGE 1
2971  013740  000407                          BR      2$
2972  013742  010201                  1$:     MOV     R2,R1                   ;SETUP R1 TO REFERENCE 4K BANK
2973  013744  006301                          ASL     R1                      ;WITHOUT KT11
2974  013746  006301                          ASL     R1
2975  013750  006301                          ASL     R1
2976  013752  006301                          ASL     R1
2977  013754  006301                          ASL     R1
2978  013756  006301                          ASL     R1
2979  013760  116067  000006  000010  2$:     MOVB    6(R0),LSAV
2980  013766  066701  000004                  ADD     LSAV,R1
2981  013772  012602                          MOV     (SP)+,R2                ;RESTORE R2
2982  013774  000207                          RTS     %7                      ;AND RETURN
2983  013776  000000                  LSAV:   0
2984
2985
2986
2987                                          ;SAVE LOADER IN TOP OF FIRST 4K
2988  014000  013746  000004          SAVLDR: MOV     @#4,-(SP)               ;SAVE CONTENTS OF TIMEOUT VECTOR
2989  014004  013746  000006                  MOV     @#6,-(SP)
2990  014010  012737  000006  000004          MOV     #6,@#4
2991  014016  012737  000002  000006          MOV     #RTI,@#6                ;SETUP TO RTI ON TIMEOUT
2992  014024  010146                          MOV     R1,-(SP)                ;SAVE REGISTERS
2993  014026  010246                          MOV     R2,-(SP)
2994  014030  012701  157500                  MOV     #157500,R1
2995  014034  000261                  1$:     SEC
2996  014036  005711                          TST     @R1                     ;IF TIMEOUT, C BIT WILL STILL BE SET
2997  014040  103006                          BCC     2$                      ;IF NO TIMEOUT, C BIT WILL BE CLEAR
2998  014042  162701  020000                  SUB     #20000,R1               ;TIMEOUT OCCURRED, CHECK FOR NEXT LOWER BANK
2999  014046  020127  020000                  CMP     R1,#20000
3000  014052  101370                          BHI     1$
3001  014054  000410                          BR      SAVLDX                  ;ONLY 4K OF 1ST 28K PRESENT- EXIT
3002  014056  012702  017500          2$:     MOV     #17500,R2               ;THIS BANK IS HIGHEST OF 1ST 28K- COPY
3003  014062  012122                  3$:     MOV     (R1)+,(R2)+             ;LOADER AREA TO BANK 0
3004  014064  020127  020000                  CMP     R2,#20000
3005  014070  103774                          BLO     3$
3006  014072  005267  000016                  INC     LDRSVD                  ;INDICATE LOADER HAS BEEN MOVED TO
3007                                                                          ;THE TOP OF THE FIRST 4K
3008  014076  012602                  SAVLDX: MOV     (SP)+,R2
3009  014100  012601                          MOV     (SP)+,R1
3010  014102  012637  000006                  MOV     (SP)+,@#6
3011  014106  012637  000004                  MOV     (SP)+,@#4
3012  014112  000207                          RTS     %7
3013  014114  000000                  LDRSVD: 0
3014
3015
3016
3017                                          ;ROUTINE TO RESTORE THE LOADER FROM BANK 0 (WHERE IT WAS SAVED) TO THE
3018                                          ;HIGHEST BANK IN THE FIRST 28K OF MEMORY
3019  014116  005767  177772          RSTLDR: TST     LDRSVD
3020  014122  001431                          BEQ     RSTLDX
3021  014124  012737  000006  000004          MOV     #6,@#4
3022  014132  012737  000002  000006          MOV     #RTI,@#6
3023  014140  012701  157500                  MOV     #157500,R1
3024  014144  000261                  1$:     SEC
```

```
3025  014146  005711                          TST     @R1                     ;IF TIMEOUT, C BIT WILL BE SET
3026  014150  103006                          BCC     2$                      ;IF NO TIMEOUT, C BIT WILL BE CLEAR
3027  014152  162701  020000                  SUB     #20000,R1               ;TIMEOUT OCCURRED, CHECK FOR NEXT
3028  014156  020127  020000                  CMP     R1,#20000               ;LOWER BANK
3029  014162  101370                          BHI     1$
3030  014164  000410                          BR      RSTLDX
3031  014166  012702  017500          2$:     MOV     #17500,R2
3032  014172  012221                  3$:     MOV     (R2)+,(R1)+
3033  014174  020227  020000                  CMP     R2,#20000
3034  014200  103774                          BLO     3$
3035  014202  104000                          TYPE                            ;TYPE MESSAGE "LOADER RESTORED"
3036  014204  017156                          LDRMSG
3037  014206  000000                  RSTLDX: HALT                            ;LOADER HAS BEEN RESTORED
3038  014210  000776                          BR      .-2                     ;TO HIGHEST BANK IN FIRST 28K
3039
3040
3041
3042                                          ;SCAN ALL MEMORY FOR BAD PARITY, TYPE 18 BIT ADDRESSES OF
3043                                          ;LOCATIONS FOUND TO BE BAD, AND WRITE INTO LOCATIONS WITH GOOD PARITY
3044  014212  010146                  PSCAN:  MOV     R1,-(SP)                ;STORE REGISTERS AND LOCATIONS TO BE
3045  014214  010246                          MOV     R2,-(SP)                ;ALTERRED
3046  014216  010346                          MOV     R3,-(SP)
3047  014220  010446                          MOV     R4,-(SP)
3048  014222  013746  000004                  MOV     @#4,-(SP)
3049  014226  013746  000006                  MOV     @#6,-(SP)
3050  014232  013746  000114                  MOV     @#114,-(SP)
3051  014236  013746  000116                  MOV     @#116,-(SP)
3052  014242  012737  000006  000004          MOV     #6,@#4                  ;SETUP TIMEOUT TRAPCATCHER
3053  014250  005037  000006                  CLR     @#6
3054  014254  012737  000116  000114          MOV     #116,@#114              ;SETUP PARITY TRAP TRAPCATCHER
3055  014262  005037  000116                  CLR     @#116
3056  014266  005767  164260                  TST     NOKT                    ;KT11 PRESENT?
3057  014272  001513                          BEQ     PSCAN1                  ;YES, BRANCH
3058  014274  005002                          CLR     R2                      ;R2 CONTAINS TEST ADDRESS
3059  014276  012703  000001                  MOV     #1,R3                   ;R3 USED AS A BIT POINTER
3060  014302  004767  177202          1$:     JSR     %7,CLRPAR               ;CLEAR ALL PARITY REGISTERS
3061  014306  005712                          TST     @R2                     ;READ LOCATION TO CHECK FOR BAD PARITY
3062  014310  000240                          NOP
3063  014312  012701  000570                  MOV     #MPRO,R1                ;SETUP TO SCAN REGISTERS FOR PARITY
3064                                                                          ;ERROR SET
3065  014316  032711  000001          3$:     BIT     #1,@R1
3066  014322  001003                          BNE     .+10
3067  014324  005771  000000                  TST     @(R1)                   ;PARITY ERROR SET?
3068  014330  100424                          BMI     6$                      ;YES- BRANCH
3069  014332  062701  000010                  ADD     #10,R1                  ;NO- CHECK NEXT REGISTER
3070  014336  020127  000770                  CMP     R1,#TREG
3071  014342  103765                          BLO     3$                      ;LOOP UNTIL ALL REGISTERS HAVE BEEN CHECKED
3072  014344  062702  000002          4$:     ADD     #2,R2                   ;MOVE ADDRESS POINTER
3073  014350  032702  017777                  BIT     #17777,R2               ;DONE WITH 4K?
3074  014354  001362                          BNE     1$                      ;NO, CONTINUE
3075  014356  005303                  5$:     DEC     R3                      ;YES, CHECK FOR TESTING NEXT 4K
3076  014360  020327  000200                  CMP     R3,#200
3077  014364  103035                          BHIS    PSCANX                  ;EXIT IF DONE WITH 28K
3078  014366  030367  164474                  BIT     R3,MEML                 ;IS THIS MEMORY PRESENT?
3079  014372  001343                          BNE     1$                      ;YES, GO TEST IT
3080  014374  062702  020000                  ADD     #20000,R2               ;NO, UPDATE ADDRESS
```

```
3091  014400  000766                        BR      5$                      ;LOOP
3082  014402  010267  000110          6$:   MOV     R2,PSADRS               ;PARITY ERROR OCCURRED
3083  014406  004567  000502                JSR     R5,OACNV                ;GET ASCII OF ADDRESS CONTAINING BAD PARITY
3084  014412  014516                        PSADRS
3085  014414  017116                        MPSERI
3086  014416  000006                        6
3087  014420  104000                        TYPE                            ;TYPE MESSAGE "BAD PARITY FOUND IN LOCATION"
      014422  017066                        MPSER                           ;AND ADDRESS OF FAILING LOCATION
      014424  032777  002000  164446        BIT     #2000,@SWR              ;SWITCH 10 SET?
      014432  001401                        BEQ     .+4                     ;NO- CONTINUE
      014434  000000                        HALT                            ;HALT ON BAD PARITY SET
      014436  011212                        MOV     @R2,@R2                 ;WRITE INTO LOCATION- SHOULD
                                                                            ;CLEAR BAD PARITY
      014440  005071  000000                CLR     @(R1)                   ;CLEAR CORRESPONDING PARITY REGISTER
      014444  005712                        TST     @R2                     ;READ LOCATION TO SEE IF BAD PARITY WAS
      014446  005771  000000                TST     @(R1)                   ;CLEARED
      014452  100001                        BPL     .+4                     ;OK- BRANCH
      014454  104002                        ERROR                           ;BAD PARITY DIDN'T CLEAR WHEN LOCATION
                                                                            ;WAS REWRITTEN
3100  014456  000732                        BR      4$                      ;GO CHECK NEXT LOCATION
3101  014460  004767  177024          PSCANX: JSR     PC,CLRPAR               ;DONE- CLEAR ALL PARITY REGISTERS
3102  014464  012637  000116                MOV     (SP)+,@#116             ;RESTORE LOCATIONS ALTERRED
3103  014470  012637  000114                MOV     (SP)+,@#114
3104  014474  012637  000006                MOV     (SP)+,@#6
3105  014500  012637  000004                MOV     (SP)+,@#4
3106  014504  012604                        MOV     (SP)+,R4
3107  014506  012603                        MOV     (SP)+,R3
3108  014510  012602                        MOV     (SP)+,R2
3109  014512  012601                        MOV     (SP)+,R1
3110  014514  000207                        RTS     PC                      ;RETURN
3111  014516  000000            PSADRS: 0
3112  014520  000000            PSCANH: 0
3113
3114
3115
3116                                        ;SCAN ALL MEMORY FOR BAD PARITY USING KT11
3117                                        ;TYPE 18 BIT ADDRESSES OF LOCATIONS FOUND BAD, AND WRITE GOOD PARITY BACK IN
3118  014522  017746  164520          PSCAN1: MOV     @KPAR1,-(SP)            ;SAVE CONTENTS OF KERNEL PAR1
3119  014526  032737  000001  177572        BIT     #1,@#SR0                        ;SKIP IF KT11 IS ALREADY ON
3120  014534  001004                        BNE     1$
3121  014536  004767  176646                JSR     PC,MRALL                ;MAP KERNEL 0 TO BANK 0,RW
3122  014542  004767  177012                JSR     PC,MAP1                 ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
3123                                                                        ;MAP KERNEL 1 RW, AND TURN ON KT11
3124  014546  005067  177746          1$:   CLR     PSCANH                  ;CLEAR FLAG TO INDICATE CHECKING FIRST 64K
3125  014552  005077  164470                CLR     @KPAR1                  ;INITIALIZE TO BANK 0
3126  014556  012703  000001          PSLOOP: MOV     #1,R3                   ;R3 IS USED AS A BIT POINTER
3127  014562  005767  177732          PSLUP:  TST     PSCANH                  ;TESTING TOP 64K?
3128  014566  001004                        BNE     2$                      ;YES, BRANCH
3129  014570  030367  164272                BIT     R3,MEML                 ;NO, IS PARITY MEMORY PRESENT IN THIS 4K?
3130  014574  001022                        BNE     PSXTST                  ;YES- GO TEST IT
3131  014576  000403                        BR      PSNXT                   ;NO- CHECK FOR NEXT 4K
3132  014600  030367  164264          2$:   BIT     R3,MEMH                 ;IS PARITY MEMORY PRESENT IN THIS 4K?
3133  014604  001016                        BNE     PSXTST                  ;YES- GO TEST IT
3134  014606  062777  000200  164432  PSNXT:  ADD     #200,@KPAR1             ;NO- MAP TO NEXT 4K
3135  014614  006303                        ASL     R3
3136  014616  103361                        BCC     PSLUP                   ;BRANCH IF NOT END OF 64K
```

```
3137  014620  005767  177674            TST     PSCANH          ;END OF TOP 64K?
3138  014624  001003                    BNE     PSCX1           ;YES, GET READY TO EXIT
3139  014626  005267  177666            INC     PSCANH          ;NO, SET FLAG INDICATING DONE WITH
3140                                                            ;LOWER 64K
3141  014632  000751                    BR      PSLOOP
3142  014634  012677  164406  PSCX1:    MOV     (SP)+,@KPAR1
3143  014640  000707                    BR      PSCANX
3144  014642  012702  020000  PSXTST:   MOV     #20000,R2       ;R2 USED AS ADDRESS POINTER
3145  014646  004767  176636  1$:       JSR     X7,CLRPAR       ;CLEAR ALL PARITY REGISTERS
3146  014652  005712                    TST     @R2             ;READ LOCATION
3147  014654  012701  000570            MOV     #MPRO,R1        ;SETUP TO SCAN REGISTERS FOR PARITY ERROR SET
3148  014660  032711  000001  2$:       BIT     #1,@R1
3149  014664  001003                    BNE     .+10
3150  014666  005771  000000            TST     @(R1)           ;PARITY ERROR SET?
3151  014672  100413                    BMI     4$              ;YES, BRANCH
3152  014674  062701  000010            ADD     #10,R1          ;NO, CHECK NEXT
3153  014700  020127  000770            CMP     R1,#TREG
3154  014704  10    5                   BLO     2$              ;LOOP UNTIL ALL REGISTERS HAVE BEEN CHECKED
3155  014706  0     2  000002  3$:       ADD     #2,R2           ;UPDATE TEST ADDRESS POINTER
3156  014712  0     7  040000            CMP     R2,#40000       ;DONE WITH BANK?
3157  014716  10    53                  BLO     1$              ;NO- LOOP
3158  014720  0    732                  BR      PSNXT           ;YES- GO CHECK FOR ANOTHER BANK
3159  014723  0     7  177570  4$:       MOV     R2,PSADRS       ;PARITY ERROR OCCURRED- GET 18 BIT
3160  014726  0     67 160000  177562    BIC     #160000,PSADRS  ;OCTAL ADDRESS OF BAD LOCATION
3161  014734  0L    46                  CLR     -(SP)
3162  014736  017746  164304            MOV     @KPAR1,-(SP)
3163  014742  0     5316                ASL     @SP
3164  014744  0     5316                ASL     @SP
3165  014746  006316                    ASL     @SP
3166  014750  006316                    ASL     @SP
3167  014752  006316                    ASL     @SP
3168  014754  006166  000002            ROL     2(SP)
3169  014760  006316                    ASL     @SP
3170  014762  006166  000002            ROL     2(SP)
3171  014766  006366  000002            ASL     2(SP)
3172  014772  062667  177520            ADD     (SP)+,PSADRS
3173  014776  004567  000112            JSR     R5,OACNV        ;CONVERT LOW 16 OCTAL BITS TO ASCII
3174  015002  014516                    PSADRS
3175  015004  017116                    MPSER1
3176  015006  000006                    6
3177  015010  116704  002102            MOVB    MPSER1,R4
3178  015014  062604                    ADD     (SP)+,R4        ;CHANGE TO ASCII FOR 18 BITS
3179  015016  110467  002074            MOVB    R4,MPSER1
3180  015022  104000                    TYPE                    ;TYPE ADDRESS OF LOCATION WITH BAD PARITY
3181  015024  017066                    MPSER
3182  015026  032777  002000  164044    BIT     #2000,@SWR      ;SWITCH 10 SET?
3183  015034  001401                    BEQ     .+4             ;NO- BRANCH
3184  015036  000000                    HALT                    ;HALT ON BAD PARITY SET
3185  015040  011212                    MOV     @R2,@R2         ;REWRITE LOCATION CONTAINING BAD PARITY
3186  015042  005071  000000            CLR     @(R1)           ;CLEAR PARITY ERROR BIT
3187  015046  005712                    TST     @R2             ;READ LOCATION TO SEE IF PARITY IS NOW GOOD
3188  015050  005771  000000            TST     @(R1)           ;CHECK PARITY ERROR BIT
3189  015054  100001                    BPL     .+4
3190  015056  104002                    ERROR                   ;REWRITING LOCATION DID NOT CLEAR BAD PARITY
3191  015060  000712                    BR      3$              ;GO TEST NEXT LOCATION
3192
```

```
3193
3194
3195
3196                                    ;PIC ROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES (CALLED VIA JSR R5)
3197  015062  012567  000022          TYPSX:  MOV     (R5)+,TYPSBX            ;GET ADDRESS OF MESSAGE
3198  015066  022767  177777  000014          CMP     #-1,TYPSBX              ;TERMINATOR?
3199  015074  001001                          BNE     TYPSAX                  ;NO, BRANCH
3200  015076  000205                          RTS     R5                      ;YES, RETURN
3201  015100  013746  177776          TYPSAX: MOV     @#PS,-(SP)              ;SETUP TO CALL TYPE ROUTINE VIA JSR
3202  015104  004767  164004                  JSR     PC,$TYPE                ;TYPE ASCII MESSAGE
3203  015110  000000                  TYPSBX: .OPEN
3204  015112  000763                          BR      TYPSX
3205
3206
3207
3208                                    ;SUBROUTINE FOR OCTAL TO ASCII CONVERSION
3209  015114  013567  000074          OACNV:  MOV     @(5)+,OACNVX            ;GET OCTAL VALUE
3210  015120  012567  000072                  MOV     (5)+,OACDST             ;GET DESTINATION ADDRESS
3211  015124  012567  000070                  MOV     (5)+,OACNT              ;GET CONVERT COUNT
3212  015130  066767  000064  000060          ADD     OACNT,OACDST            ;DEVELOP ADDRESS TO STORE 1ST CHAR.
3213  015136  016746  000052          OACNVA: MOV     OACNVX,-(SP)
3214  015142  042716  177770                  BIC     #177770,@SP             ;ISOLATE LEAST SIGNIFICANT DIGIT
3215  015146  062716  000060                  ADD     #60,@SP                 ;CONVERT DIGIT TO ASCII
3216  015152  005367  000040                  DEC     OACDST
3217  015156  116377  000034                  MOVB    (SP)+,@OACDST           ;STORE ASCII CHARACTER
3218  015162  042767  000007  000024          BIC     #7,OACNVX
3219  015170  006067  000020                  ROR     OACNVX
3220  015174  006067  000014                  ROR     OACNVX
3221  015200  006067  000010                  ROR     OACNVX
3222  015204  005367  000010                  DEC     OACNT                   ;DONE ALL DIGITS?
3223  015210  001352                          BNE     OACNVA                  ;BRANCH IF NOT DONE
3224  015212  000205                          RTS     R5                      ;DONE, EXIT
3225  015214  000000                  OACNVX: .OPEN
3226  015216  000000                  OACDST: 0
3227  015220  000000                  OACNT:  0
3228
3229
3230
3231                                    ;SUBROUTINE FOR BINARY TO DECIMAL ASCII CONVERSION
3232  015222  104005                  BDCNV:  SAV04                           ;SAVE REGS
3233  015224  012700  015400                  MOV     #DECVAL,R0              ;SET UP ADDR TO STORE DECIMAL ASCII
3234  015230  013501                          MOV     @(5)+,R1                ;BINARY VALUE TO R1
3235  015232  012567  000052                  MOV     (5)+,BDCNVC             ;DESTINATION ADDR TO BDCNVC
3236  015236  012567  000050                  MOV     (5)+,BDCNVD             ;CHARACTER COUNT TO BDCNVD
3237  015242  012702  015366                  MOV     #BOTEMP,R2              ;ADDR OF TEN POWER STRING
3238  015246  012767  000005  000104          MOV     #5,CNVCTR               ;SET UP FOR 5 POWER CONVERSIONS
3239  015254  012267  000104  BDCNVA: MOV     (2)+,TENPWR             ;MOVE POWER OF TEN VALUE
3240  015260  004767  000034                  JSR     PC,SUBTEN               ;PERFORM CONVERSION
3241  015264  005367  000070                  DEC     CNVCTR                  ;DONE 5 CONVERSIONS?
3242  015270  001371                          BNE     BDCNVA                  ;BRANCH IF NOT YET 5.
3243  015272  166700  000014                  SUB     BDCNVD,R0
3244  015276  010067  000004                  MOV     R0,BDCNVB
3245  015302  004567  000100                  JSR     R5,BMOVE
3246  015306  000000                  BDCNVB: .OPEN
3247  015310  000000                  BDCNVC: .OPEN
3248  015312  000000                  BDCNVD: .OPEN
```

```
3249  015314  104006                          RSTO4                           ;RESTORE REGS AND EXIT
3250  015316  000205                          RTS     R5
3251  015320  005067  000036          SUBTEN: CLR     DIGIT
3252  015324  166701  000034          SUBTNA: SUB     TENPWR,R1               ;SUBTRACT TEN POWER FROM BINARY VALUE
3253  015330  103403                          BCS     SUBTNB                  ;BRANCH IF UNSUCESSFUL SUBTRACTION
3254  015332  005267  000024                  INC     DIGIT
3255  015336  000772                          BR      SUBTNA
3256  015340  066701  000020          SUBTNB: ADD     TENPWR,R1               ;RESTORE SUBTRACTED VALUE.
3257  015344  062767  000060  000010          ADD     #60,DIGIT               ;CONVERT (DIGIT) TO ASCII
3258  015352  116720  000004                  MOVB    DIGIT,(0)+              ;MOVE ASCII CHAR TO DECVAL FIELD
3259  015356  000207                          RTS     PC                      ;EXIT
3260  015360  000000          CNVCTR: OPEN
3261  015362  000000          DIGIT:  OPEN
3262  015364  000000          TENPWR: OPEN
3263  015366  023420          ADTENP: 10000.
3264  015370  001750                  1000.
3265  015372  000144                  100.
3266  015374  000012                  10.
3267  015376  000001                  1.
3268  015400     040     040     040  DECVAL: .BYTE   040,040,040,040,040,040
3269  015403     040     040     040
3270
3271
3272
3273                                  ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES
3274  015406  104005                  BMOVE:  SAVO4                           ;SAVE REGS
3275  015410  012501                          MOV     (5)+,R1                 ;GET FROM ADDRESS
3276  015412  012502                          MOV     (5)+,R2                 ;GET TO ADDRESS
3277  015414  012503                          MOV     (5)+,R3                 ;GET COUNT
3278  015416  112122                  BMOVA:  MOVB    (1)+,(2)+               ;MOVE BYTE
3279  015420  005303                          DEC     R3                      ;DECREMENT COUNT
3280  015422  001375                          BNE     BMOVA                   ;BRANCH IF NOT DONE
3281  015424  104006                          RSTO4                           ;RESTORE REGS AND EXIT
3282  015426  000205                          RTS     R5
3283
3284
3285
3286                                  ;UNEXPECTED POWER FAIL SERVICE
3287                                  ;BECAUSE WWP MAY BE SET IN MPR'S AND ALL PROCESSOR REGISTERS
3288                                  ;MAY BE IN USE, CONTINUATION AFTER POWER FAIL IS NOT ATTEMPTED.
3289                                  ;INSTEAD, THE PROGRAM RESTARTS AFTER A POWER FAILURE
3290  015430  012737  015474  000024  PWRDN:  MOV     #PWRUP,@#24             ;SET UP FOR POWER UP
3291  015436  012701  000570                  MOV     #MPR0,R1
3292  015442  032711  000001          1$:     BIT     #1,@R1
3293  015446  001002                          BNE     .+6
3294  015450  005071  000000                  CLR     @(R1)                   ;CLEAR PARITY REGISTERS IN CASE
3295  015454  062701  000010                  ADD     #10,R1                  ;WWP IS SET
3296  015460  020127  000770                  CMP     R1,#TREG
```

```
3297  015464  103766                          BLO     1$
3298  015466  010667  163562                  MOV     SP,SPSAV
3299  015472  000000                          HALT                            ;POWER DOWN HALT
3300  015474  012737  015430  000024  PWRUP:  MOV     #PWRDN,@#24             ;SET UP FOR POWER DOWN
3301  015502  016706  163546                  MOV     SPSAV,SP
3302  015506  005027  000000                  CLR     #0                      ;STALL SO OUTPUT WON'T BE GARBLED
3303  015512  005367  177772                  DEC     .-2
3304  015516  001375                          BNE     .-4
3305  015520  104000                          TYPE                            ;TYPE RECOVERY MESSAGE
3306  015522  017022                          MPWRF
3307  015524  000167  163542                  JMP     RSTART                  ;RESTART
3308
3309
3310
3311                                  ;EMT HANDLER
3312  015530  011646                  EMTINT: MOV     (SP),-(SP)              ;GET SAVED PC
3313  015532  162716  000002                  SUB     #2,(SP)                 ;DECREMENT PC BY 2
3314  015536  017616  000000                  MOV     @(SP),(SP)              ;GET CALL
3315  015542  121667  000050                  CMPB    (SP),EMTLIM             ;CHECK IF CALL WITHIN LIMITS
3316  015546  101402                          BLOS    EMTA
3317  015550  000000                          HALT                            ;CALL IS NOT WITHIN LIMITS
3318  015552  000776                          BR      .-2
3319  015554  006116                  EMTA:   ROL     (SP)                    ;EMT ARG X 2
3320  015556  042716  177001                  BIC     #177001,(SP)            ;REMOVE 7 MSB
3321  015562  062716  015574                  ADD     #EMTTAB,(SP)            ;FORM EMT RTN ADDRESS
3322  015566  017616  000000                  MOV     @(SP),(SP)
3323  015572  000136                          JMP     @(SP)+                  ;GO TO EMT RETURN
3324
3325
3326                                  ;EMT DEFINITIONS AND ASSIGNMENTS
3327  015574                          EMTTAB:
3328          104000                          TYPE=EMT+EMTX
3329  015574  001114                          $TYPE
3330          104001                          SCOPE=EMT+EMTX
3331  015576  016054                          SCOPEC
3332          104002                          ERROR=EMT+EMTX
3333  015600  013270                          ERR
3334          104003                          ERRORP=EMT+EMTX
3335  015602  013150                          ERRP
3336          104004                          ERRORS=EMT+EMTX
3337  015604  013072                          ERRST
3338          104005                          SAVO4=EMT+EMTX
3339  015606  015620                          SVO4
3340          104006                          RSTO4=EMT+EMTX
3341  015610  015706                          RSO4
3342          104007                          RSTOSS=EMT+EMTX
3343  015612  015734                          RSOSS
3344          104010                          SAVOSS=EMT+EMTX
3345  015614  015640                          SVOSS
3346  015616  000010                  EMTLIM: EMTX-1
3347
3348
3349                                  ;SUBROUTINE TO SAVE REGS 0-4
3350  015620  012666  177764          SVO4:   MOV     (SP)+,-12.(SP)          ;MOVE PC+PS UP STACK
3351  015624  012666  177764                  MOV     (SP)+,-12.(SP)
3352  015630  012767  000002  000040          MOV     #RTI,SVOSC
```

```
 3353  015636  000411                          BR      SVOSB
 3354
 3355
 3356
 3357                                  ;SUBROUTINE TO SAVE REGS 0-5 + PLACE EMT PC IN R5
 3358  015640  012767  000240  000030  SVOSS:  MOV     #NOP,SVOSC
 3359  015646  000400                          BR      SVOSA
 3360
 3361                                  ;SUBROUTINE TO SAVE REGS 0-5
 3362  015650  012666  177762          SVOSA:  MOV     (SP)+,-14.(SP)
 3363  015654  012666  177762                  MOV     (SP)+,-14.(SP)
 3364  015660  010546                          MOV     R5,-(SP)
 3365  015662  010446                  SVOSB:  MOV     R4,-(SP)
 3366  015664  010346                          MOV     R3,-(SP)
 3367  015666  010246                          MOV     R2,-(SP)
 3368  015670  010146                          MOV     R1,-(SP)
 3369  015672  010046                          MOV     %0,-(SP)
 3370  015674  024646                          CMP     -(SP),-(SP)
 3371  015676  000002                  SVOSC:  RTI                             ;RTI OR NOP
 3372  015700  016505  000020                  MOV     16.(SP),R5              ;EMT PC TO R5
 3373  015704  000002                          RTI
 3374
 3375
 3376
 3377                                  ;SUBROUTINE TO RESTORE REGS 0-4
 3378  015706  022626                  RSO4:   CMP     (SP)+,(SP)+
 3379  015710  012600                          MOV     (SP)+,%0
 3380  015712  012601                          MOV     (SP)+,R1
 3381  015714  012602                          MOV     (SP)+,R2
 3382  015716  012603                          MOV     (SP)+,R3
 3383  015720  012604                          MOV     (SP)+,R4
 3384  015722  016646  177764                  MOV     -12.(SP),-(SP)          ;MOVE PC+PS DOWN STACK
 3385  015726  016646  177764                  MOV     -12.(SP),-(SP)
 3386  015732  000002                          RTI
 3387
 3388
 3389
 3390                                  ;SUBROUTINE TO RESTORE REGS 0-5
 3391  015734  010566  000020          RSOSS:  MOV     R5,16.(SP)              ;SET EMT PC TO R5
 3392  015740  022626                          CMP     (SP)+,(SP)+
 3393  015742  012600                          MOV     (SP)+,%0
 3394  015744  012601                          MOV     (SP)+,R1
 3395  015746  012602                          MOV     (SP)+,R2
 3396  015750  012603                          MOV     (SP)+,R3
 3397  015752  012604                          MOV     (SP)+,R4
 3398  015754  012605                          MOV     (SP)+,R5
 3399  015756  016646  177762                  MOV     -14.(SP),-(SP)
 3400  015762  016646  177762                  MOV     -14.(SP),-(SP)
 3401  015766  000002                          RTI
 3402
 3403
 3404
 3405                                  ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
 3406                                  ;LOAD THE STARTING ADDRESS OF THE TEST
 3407                                  ;YOU WISH TO RUN (THE ADDRESS OF THE TESTXX
 3408                                  ;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
```

```
3409                                        ;OPTIONS AT THE 2ND HALT.
3410                                        ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
3411  015770  005037  177776          TESTX:  CLR     @#PS
3412  015774  000000                          HALT                            ;WAIT FOR STARTING ADDRESS
3413  015776  017767  163076  000146        MOV     @SWR,RETURN             ;LOAD STARTING ADDRESS IN RETURN
3414  016004  062767  000002  000140        ADD     #2,RETURN               ;ADD 2 TO POINT TO INSTRUCTION AFTER
3415  016012  000000                          HALT                            ;SET SR OPTIONS
3416  016014  012767  177777  162466        MOV     #-1,TSTX                ;SET FLAG
3417  016022  032777  010000  163050        BIT     #10000,@SWR             ;CHECK SW12
3418  016030  001404                          BEQ     .+12                    ;BRANCH IF NOT SET
3419  016032  042737  000020  177776        BIC     #20,@#PS                ;CLEAR TRACE BIT
3420  016040  000403                          BR      .+10                    ;SKIP NEXT INSTRUCTION
3421  016042  052737  000020  177776        BIS     #20,@#PS                ;SET TRACE BIT
3422  016050  000177  000076                JMP     @RETURN                 ;JUMP TO TEST
3423
3424
3425
3426                                        ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 64 TIMES
3427                                        ;A SETUP ROUTINE SHOULD INITIALIZE RETURN AND IMAX
3428  016054  032777  040000  163016  SCOPEC: BIT     #40000,@SWR             ;TEST SR FOR SCOPE
3429  016062  001015                          BNE     SCOPEB                  ;YES, SCOPE
3430  016064  032777  004000  163006        BIT     #4000,@SWR              ;NO-TEST FOR ITERATION
3431  016072  001016                          BNE     SCOPEG                  ;INHIBIT ITERATION
3432  016074  005767  162410                TST     TSTX                    ;USING SINGLE SUBTEST STARTUP?
3433  016100  001006                          BNE     SCOPEB                  ;YES, LOOP
3434  016102  026767  000042  000036        CMP     ICNT,IMAX               ;COMPARE CURRENT COUNT TO MAX NUMBER
3435  016110  100007                          BPL     SCOPEG                  ;EXIT-DONE
3436  016112  005267  000032                INC     ICNT                    ;INCREMENT COUNT
3437  016116  022606                SCOPEB: CMP     (6)+,%6                 ;REPOSITION STACK
3438  016120  012677  161652                MOV     (6)+,@PS                ;RESTORE PREVIOUS PROCESSOR STATUS
3439  016124  000177  000022                JMP     @RETURN                 ;REPEAT TEST
3440  016130  005067  162354          SCOPEG: CLR     TSTX                    ;IF USING TESTX STARTUP, RETURN TO NORMAL FLOW
3441  016134  005067  000010                CLR     ICNT                    ;CLEAR COUNT
3442  016140  011667  000006                MOV     @%6,RETURN              ;SAVE SCOPE RETURN POINTER
3443  016144  000002                          RTI                             ;RETURN INLINE-NEXT TEST
3444  016146  000100                IMAX:   100                             ;ITERATION COUNT
3445  016150  000000                ICNT:   0                               ;COUNT LOCATION FOR ITERATION LOOP
3446  016152  000000                RETURN: 0                               ;ADDRESS OF LAST TEST
3447
3448
3449
3450
3451                                        ;ASCII MESSAGES
3452  016154                                MEO:
3453  016154  005015  041520  020075  MTNUM:  .ASCII  <15><12>'PC= '
3454  016162  020040  020040  020040  MPC:    .ASCII  '         ICNT= '
3455  016170  020040  041511  052116
3456  016176  020075
3457  016200  020040  020040  020040  MICNT:  .ASCIZ  '      '
3458  016206     000
3459  016207     040  046440  051120  MSTR:   .ASCII  '  MPR= '
3460  016214  020075
3461  016216  020040  020040  020040  MTREG.  .ASCII  '        MPR DATA= '
3462  016224  020040  050115  020122
3463  016232  040504  040524  020075
3464  016240  020040  020040  020040  MDATA:  .ASCIZ  '        '
```

```
3465  016246  020040     000
3466  016251     015  020012  020040  MSTRX:  .ASCII  <15><12>'       TEST LOC= '
3467  016256  020040  020040  052040
3468  016264  051505  020124  047514
3469  016272  036503     040
3470  016275     040  020040  020040  MSTRX1: .ASCII  '      '
3471  016302     040
3472  016303     040  027523  035102          .ASCII  ' S/B: '
3473  016310     040
3474  016311     040  020040  020040  MSTRX3: .ASCII  '      '
3475  016316     040
3476  016317     040  040527  035123          .ASCII  ' WAS: '
3477  016324     040
3478  016325     040  020040  020040  MSTRX5: .ASCIZ  '      '
3479  016332  020040     000
3480  016335     015  051412  052105  MSETSR: .ASCIZ  <15><12>'SET SR OPTIONS'
3481  016342  051440  020122  050117
3482  016350  044524  047117  000123
3483  016356  020054  051120  051505  MCON:   .ASCIZ  ', PRESS CONTNUE'
3484  016364  020123  047503  052116
3485  016372  052516  000105
3486  016376  005015  042523  020124  MMDEV:  .ASCIZ  <15><12>'SET DEVICE ADDRESS IN SR'
3487  016404  042504  044526  042503
3488  016412  040440  042104  042522
3489  016420  051523  044440  020116
3490  016426  051123     000
3491  016431     015  051412  052105  MMADR:  .ASCIZ  <15><12>'SET MEMORY TEST LOC IN SR'
3492  016436  046440  046505  051117
3493  016444  020131  042524  052123
3494  016452  046040  041517  044440
3495  016460  020116  051123     000
3496  016465     015  051412  052105  MMPAT:  .ASCIZ  <15><12>'SET TEST PATTERN IN SR'
3497  016472  052040  051505  020124
3498  016500  040520  052124  051105
3499  016506  020116  047111  051440
3500  016514  000122
3501  016516  005015  046412  046505  MMPRS:  .ASCIZ  <15><12><12>'MEMORY PARITY REGISTERS PRESENT:'<15><12>
3502  016524  051117  020131  040520
3503  016532  044522  054524  051040
3504  016540  043505  051511  042524
3505  016546  051522  050040  042522
3506  016554  042523  052116  006472
3507  016562  000012
3508  016564  005015  040520  052122  MTMAP:  .ASCIZ  <15><12>'PARTY REGISTERS CONTROL MEMORY AS:'<15><12>
3509  016572  020131  042522  044507
3510  016600  052123  051105  020123
3511  016606  047503  052116  047522
3512  016614  020114  042515  047515
3513  016622  054522  040440  035123
3514  016630  005015     000
3515  016633     040  020040  000040  MTYCOR: .ASCIZ  '    '
3516  016640  020055     000          MDASH:  .ASCIZ  '- '
3517  016643     015  000012          MCR:    .ASCIZ  <15><12>
3518  016646  000113                  MK:     .ASCIZ  'K'
3519  016650  047516  050040  051101  MT:     .ASCIZ  'NO PARITY MEMORY FOUND'<15><12>
3520  016656  052111  020131  042515
```

```
3521  016664  047515  054522  043040
3522  016672  052517  042116  005015
3523  016700     000
3524  016701     116  020117  040520  MTR:    .ASCIZ  'NO PARITY REGSTER FOUND'<15><12>
3525  016706  044522  054524  051040
3526  016714  043505  052123  051105
3527  016722  043040  052517  042116
3528  016730  005015     000
3529  016733     040  020040  020040  MPRAD:  .ASCIZ  '       '<15><12>
3530  016740  020040  006440  000012
3531  016746  005015  005015  042515  MTIT:   .ASCIZ  <15><12><15><12>'MEMORY PARITY TEST - MAINDEC-11-DCMFA-C'
3532  016754  047515  054522  050040
3533  016762  051101  052111  020131
3534  016770  042524  052123  026440
3535  016776  046440  044501  042116
3536  017004  041505  030455  026461
3537  017012  041504  043115  026501
3538  017020  000103
3539  017022  005015  047520  042527  MPWRF:  .ASCIZ  <15><12>'POWER FAILED'
3540  017030  020122  040506  046111
3541  017036  042105     000
3542  017041     007                  MPGEND: .BYTE   007
3543  017042  005015  047105  020104          .ASCII  <15><12>'END PASS = '
3544  017050  040520  051523  036440
3545  017056     040
3546  017057     040  020040  020040  MPCNT:  .ASCIZ  '     '
3547  017064  000040
3548  017066  006415  041012  042101  MPSER:  .ASCII  <15><15><12>'BAD PAR FOUND IN LOC '
3549  017074  050040  051101  043040
3550  017102  052517  042116  044440
3551  017110  020116  047514  020103
3552  017116  020040  020040  020040  MPSER1: .ASCIZ  '     '
3553  017124     000
3554  017125     015  051012  043505  MX1:    .ASCIZ  <15><12>'REGISTER AT '
3555  017132  051511  042524  020122
3556  017140  052101  000040
3557  017144  047503  052116  047522  MX2:    .ASCIZ  'CONTROLS '
3558  017152  051514  000040
3559  017156  005015  047514  042101  LDRMSG: .ASCIZ  <15><12>'LOADERS RESTORED'
3560  017164  051105  020123  042522
3561  017172  052123  051117  042105
3562  017200     000
3563  017201     015  041012  042101  PSMSG:  .ASCIZ  <15><12>'BAD PARITY SCAN COMPLETE'
3564  017206  050040  051101  052111
3565  017214  020131  041523  047101
3566  017222  041440  046517  046120
3567  017230  052105  000105
3568  017234  005015  047514  042101  MLDRSV: .ASCII  <15><12>'LOADERS SAVED IN BANK 0'
3569  017242  051105  020123  040523
3570  017250  042526  020104  047111
3571  017256  041040  047101  020113
3572  017264     060
3573  017265     015  052012  020117          .ASCIZ  <15><12>'TO RESTORE LOADERS I'SE SA 210.'
3574  017272  042522  052123  051117
3575  017300  020105  047514  042101
3576  017306  051105  020123  051525
```

```
 3577  017314  020105  040523  031040
 3578  017322  030061  000056
 3579  017326  020040  020040  020040  MPRCOR: .ASCIZ  '        - CORE PARITY REGISTER'<15><12>
 3580  017334  020040  020055  047503
 3581  017342  042522  050040  051101
 3582  017350  052111  020131  042522
 3583  017356  044507  052123  051105
 3584  017364  005015     000
 3585  017367     040  020040  020040  MPRMOS: .ASCIZ  '        - MOS PARITY REGISTER'<15><12>
 3586  017374  020040  026440  046440
 3587  017402  051517  050040  051101
 3588  017410  052111  020131  042522
 3589  017416  044507  052123  051105
 3590  017424  005015     000
 3591          017430                          .EVEN
 3592          000001                          .END
```

```
ADRPT   000516          432#  1658*  1661*  1668   1670   1693   1783*  1881*  1884*  1889   1892   1905   2727*
                       2741   2786*
ADRS  = 077400           371#
ADRTYP  001224           666#  2619*  2632   2660   2665*  2666   2675*  2705*
ADTEMP  015366          3237   3263#
AE    = 000001           369#  1032   1275   1327   1490   1591   1677   1793   1912   1941   1959   2099   2116
                       2134   2284
BDCNV   015222          2746   2761   3232#
BDCNVA  015254          3239#  3242
BDCNVB  015306          3244*  3246#
BDCNVC  015310          3235*  3247#
BDCNVD  015312          3236*  3243   3248#
BEGIN   002042           698    806    815#  2541
BITPT   000520           433#  1657*  1659   1662*  1663   1770*  1771   1774*  1780*  1890*  1892   1895*  1896
                       1975   2060*  2069   2072*  2078*  2150   2561*  2570*  2572*  2579   2590*  2597*  2598
                       2602   2613*  2654   2656   2671   2674*  2676   2692*  2694*  2702   2728*  2741   2772*
                       2787*
BIT0  = 000001           353#
BIT1  = 000002           354#
BIT10 = 002000           363#
BIT11 = 004000           364#
BIT12 = 010000           365#
BIT13 = 020000           366#  2857
BIT14 = 040000           367#   409
BIT15 = 100000           368#
BIT2  = 000004           355#
BIT3  = 000010           356#
BIT4  = 000020           357#
BIT5  = 000040           358#  1348   1363
BIT6  = 000100           359#  1433
BIT7  = 000200           360#
BIT8  = 000400           361#  2533
BIT9  = 001000           362#   975
P"OVA   015416          3278#  3280
B OVE   015426          3245   3274#
CKBKO   010340          2289   2300#
CLRPAR  013510           958    991   1082   1136   1265   1300   1394   1460   1563   1655   1686   1762   1802
                       1891   1898   2058   2244   2273   2293   2385   2488   2518   2714   2908#  3060   3101
                       3145
CNT16   007100          2025#
CNT17   007760          2128   2210#
CNVCTR  015360          3238*  3241*  3260#
CN16    007120          1953   2036#
CONT    011444          2527   2533#
CONT10  004170          1336   1342   1373#
CONT12  004720          1502   1544#
CONT13  005200          1601   1636#
CONT3   002512           963#   979
CONT4   002770          1044#  1054   1063   1070
COUNT   004232          1329*  1343*  1383#
DDISP = 177570           374#   624
DECVAL  015400          3233   3268#
DIGIT   015362          3251*  3254*  3257*  3258   3261#
DISPLA  001102           624#   738*   827*   879*   955*   989*  1081*  1135*  1264*  1299*  1393*  1459*  1562*
                       1653*  1758*  1878*  2054*  2272*  2384*
DISPRE  000174           413#   738
```

```
DONE    011364          2235    2239    2242    2509    2516#
DONE10  004216          1315    1379#
DONE11  004436          1411    1447#
DONE12  004734          1477    1549#
DONE13  005214          1576    1641#
DONE14  005440          1666    1693#
DONE15  006104          1777    1804#
DONE16  006434          1888    1898#
DONE17  007350          2075    2087#
DONE20  010324          2293#   2331
DONE4   003034          1005    1071#
DONE5   003216          1095    1123#
DONE6   003456          1148    1254#
DONE7   003570          1274    1280#
DSPAR = 177570           373#    623
EMTA    015554          3316    3319#
EMTINT  015530           404    3312#
EMTLIM  015616          3315    3346#
EMTTAB  015574          3321    3327#
EMTX  = 000011          3327#   3328    3330#   3332#   3334#   3336#   3338#   3340#   3342#   3344#   3346#
ERR     013270          2305    2326    2344    2358    2365    2430    2437    2451    2468    2478    2954#   3333
ERRA    013312          2812    2849    2857#
ERRA1   013340          2823#   2870#   2872#
ERRA2   013342          2824#   2871#   2873#
ERRA3   013352          2255#   2502#   2886#
ERRA4   013354          2256#   2503#   2887#
ERRA5   013364          2257#   2504#   2870#
ERRA6   013366          2800#   2852#   2854#   2871#
ERRADX  013370          2801#   2852#   2824#   2825#   2855#   2856#   2872#
ERRC    013374          2858    2874#
ERRD    013406          2863#   2870#   2862    2878#
ERROR = 104002           845     859     914     924    1019    1038    1048    1055    1066    1103    1117    1157    1173
                        1187    1205    1222    1232    1250    1282    1319    1333    1339    1351    1367    1416    1430
                        1437    1483    1495    1506    1514    1529    1537    1583    1595    1605    1614    1630    1699
                        1736    1817    1852    1945    1972    1978    2002    2030    2120    2147    2154    2170    2179
                        2215    2780    3098    3190    3332#
ERRORP= 104003          1993    2019    3334#
ERRORS= 104004          1717    1740    1836    1856    2203    3336#
ERRP    013150          2822#   3335
ERRST   013072          2800#   3337
FTITLE  000512           430#    706     743     746#
GMPRA   001620           766#    798
GMPRB   001746           764     793#
GMPRC   001754           792     795#
GMPRD   002010           802     805#
HIADR   000530           437#   1668#   1669#   1672    1703    1889#   1890#   1894    2039
HIWORD  000554           448#   2617#   2652    2669    2696#   2721#   2774    2788#
ICNT    016150          2866    3434    3436#   3441#   3445#
IMAX    016146           816#    880#    990#   1654#   3434    3444#
INDCO   000772           564#    763     829     882     902     999    1308    1401    1467    1964    2139    2414
INDC1   000774           565#
INDC10  001016           574#
INDC11  001020           575#
INDC12  001022           576#
INDC13  001024           577#
INDC14  001026           578#
```

```
INDC15  001030            579#
INDC2   000776            566#
INDC3   001000            567#
INDC4   001002            568#
INDC5   001004            569#
INDC6   001006            570#
INDC7   001010            571#
INDC8   001012            572#
INDC9   001014            573#
INDX17= 007632           2068#   2079#   2152#
INST1   004060           1331#   1356
INST2   004066           1337#   1359
KPAR0   001244            674#   2933#
KPAR1   001246            675#   1766#   1773#   2065#   2071#   2200    2564#   2704#   2969#   3118    3125#   3134#   3142#
                         3162
KPAR2   001250            676#
KPAR7   001252            677#   2932#
KPDR0   001234            670#   2929#
KPDR1   001236            671#   2930#
KPDR2   001240            672#
KPDR7   001242            673#   2931#
KTSTAR  001222            665#   2614#   2682    2690#
LDRMSG  017156           3036    3559#
LDRSVD  014114           3006#   3013#   3019
LOCATM  013624           1006    1096    1150    1316    1412    1479    1578    2945#
LOCAT1  013722           2954    2960    2967#
LOGICA  011424            811    2526#
LOOP10  003762           1311#   1323    1378
LOOP11  004312           1403    1407#   1419    1446
LOOP12  004526           1469    1473#   1486    1547
LOOP13  005020           1573#   1587    1639
LOOP14  005276           1659#   1664
LOOP15  005730           1771#   1775    1781
LO  1   011020           2423#   2446
LOOP4   002656           1000#   1004
LOOP5   003112           1090#   1094
LOOP6   003366           1143#   1147
LOOP7   003630           1271#   1279    1289
LOP15   00 740           1773#   1803
LOP4    002664           1002#   1023    1047
LOP5    003120           1092#   1107    1122
LOP6    003274           1145#   1161    1253
LOWFLG  000056            449#   1767#   1776    1778#   2066#   2074    2076#
LSAV    013776           2979#   2980    2983#
LUP10   003754           1309#   1314
LUP11   004276           1402#   1410
LUP12   004512           1468#   1476
LUP13   005012           1570#   1575
LUP14   005306           1661#   1687
LUP7    003622           1269#   1273
MAPMA   011516           2553#
MAPMB   011710           2549    2585#
MAPMEM  011504            713     959    2549#
MAPRB   012074           2624#   2668    2678    2706
MAPRC   012174           2643#
MAPRD   012274           2646    2664#
```

```
MAPRE   012370          2670    2672    2679#
MAPREG  012032           960    2612#
MAPI    013560           995    1086    1140    1304    1398    1464    1567    1764    2062    2555    2687    2929#   3122
MCON    016356          3483#
MCR     016643          2770    3517#
MDASH   016640          2755    3516#
MDATA   016240          2810    2834    3464#
MEMH    001070           605#   2[illegible]0#  2573    2589#   2695    3132
MEML    001066           604#   2[illegible]0   2559#   2562    2588#   2602#   2612    3078    3129
MEMUT   000550           446#   2[illegible]3#  2573#   2579#   2612#   2676    2695#   2702
MEO     016154          2870    3[illegible]3#
MICNT   016200          2[illegible]57  3457#
MK      016646          2768    3518#
MLDRSV  017234           750    3[illegible]3#
MMAOR   016431          3491#
MMDEV   016376          3486#
MMPAT   016465          3496#
MMPRS   016516           760    3501#
MPC     016162          2[illegible]53  3454#
MPCNT   017057          2522    3546#
MPGEND  017041          2525    3542#
MPRAD   016733          2734    2739    3529#
MPRCOR  017[illegible]5  777     780    3579#
MPRMOS  017367           786     789    3585#
MPROK   000542           443#    761#    791#    805
MPRO    000570           486#    762     828     881     901     965     998    1089    1142    1268    1307    1400    1466
                        1569    1674    1707    1728    1789    1826    1844    1909    1921    1937    1956    1963    2096
                        2113    2131    2138    2280    2316    2335    2398    2413    2421    2624    2635    2643    2717
                        2911    3063    3147    3291
MPR1    000600           490#
MPR10   000710           526#
MPR11   000720           530#
MPR12   000730           534#
MPR13   000740           538#
MPR14   000750           542#
MPR15   000760           546#
MPR2    000610           494#
MPR3    000620           498#
MPR4    000630           502#
MPR5    000640           506#
MPR6    000650           510#
MPR7    000660           514#
MPR8    000670           518#
MPR9    000700           522#
MPSER   017066          3088    3181    3548#
MPSER1  017116          3085    3175    3177    3179#   3552#
MPSF    017022          3306    3539#
MSETSR  016335          3480#
MSTR    016207          2800    2822    3459#
MSTRX   016251          2824    3466#
MSTRX1  016275          2838    3470#
MSTRX3  016311          2842    3474#
MSTRX5  016325          2846    3478#
MT      016650          3519#
MTIT    016746           748    3531#
MTMAP   016564          2716    3508#
```

```
MTNUM   016154          3453#
MTR     016701           809    3524#
MTREG   016216          2836    2930    3461#
MTYCOR  016633          2748    2753    2763    2766    3515#
MTYFG   000564           453#    690*    728*     956     974*
MX1     017125          2737    3534#
MX2     017144          2751    3557#
NOKT    000552           447#     717     992    1072    1083    1123    1137    1254    1301    1379    1395    1447    1461
                        1550    1564    1642    1759    2055    2553*   2585*   2679    2948    2967    3056
NOP   = 000240           378#    3358
NOREG   002016           807#
NRALL   013410           994    1095    1139    1303    1397    1463    1566    1763    2061    2554    2686    2883#   3121
OACOST  015216          3210*   3212*   3216*   3217*   3226#
OACNT   015220          3211*   3212    3232*   3227#
OACNV   015114           775     784    2520    2732    2804    2808    2828    2832    2836    2840    2844    2061    2865
                        3083    3173    3209#
OACNVA  015136          3213#   3223
OACNVX  015214          3209*   3213    3218*   3219*   3220*   3221*   3225#
OODFLG  000560           450#   1906*   1918    1928    2036*   2093*   2104    2221*   2395*   2406    2483*
OPEN  = 000000           379#   2871    2872    2878    3203    3225    3246    3247    3248    3260    3261    3262
PARPAT  001040           590#   1667    1782    2274
PARVEC= 000114           358#   1031*   1034*   1071*   1266*   1267*   1280*   1328*   1358*   1373*   1487*   1523*   1549*
                        1568*   1624*   1641*   1656*   1688*   1769*   1805*   1879*   1899*   2059*   2088*   2279*   2294*
                        2516*
PASCNT  000544           444#    691*    729*   2519*   2521
PC    =X000007           376#    643*    651*    657*    713*    714*    745*   1082*   1085*   1086*   1096*   1655*   1686*
                        1762*   1802*   2058*   2271*   2293*   2305*   2326*   2344*   2358*   2365*   2383*   2385*   2430*
                        2437*   2451*   2468*   2478*   2488*   2490*   2910    2912    3101*   3121*   3122*   3202*   3240*
                        3259*
PDREND  001232           669#   2896
PDRTAB  001226           667#   2890
PERR  = 100000           372#   1276
PMEMH   001074           610#    524*    969*   1779    2077
PMEML   001072           608#    963*    968*   1659    1768    1882    2067    2237
PMEMX   001076           612#   1768*   1771    1779*   2067*   2069    2077*
PS    = 177776           375#    695*   2243*   2270    2304    2325    2343    2357    2364    2382    2429    2436    2450
                        2467    2477    2489    3201    3411*   3419*   3421*   3438*
PSADRS  014516          3082*   3084    3111#   3159*   3160*   3172*   3174
PSCAN   014212           714    1721    1745    1860    3044#
PSCANH  014520          3112#   3124*   3127    3137    3139*
PSCANX  014460          3077    3101#   3143
PSCAN1  014522          3057    3118#
PSCX1   014634          3138    3142#
PSLOOP  014656          3126#   3141
PSLUP   014652          3127#   3136
PSMSG   017201           716    3563#
PSNXT   014606          3131    3134#   3158
PSXTST  014642          3130    3133    3144#
PWRON   015430           692     730    3290#   3300
PWRUP   015474          3290    3300#
RELOC   000566           455#   2615*   2685*   2697*   2698    2704
RESRVD  001032           580#    836*    838*    843     849     855
RESVC   001034           583#    836
RESVM   001036           584#    838     921
RETURN  016152           815*   2252*   3413*   3414*   3422    3439    3442*   3446#
RSTART  001272           421     689#   3307
```

```
                    3177*  3178*  3179   3365   3383*  3397*
R4SAV   001266       683*
R5    =x000005       386*   775*   784*  1670*  1672   1693*  1703   1785*  1787   1811*  1922   1892*  1894
                    1905*  1908*  1920*  1930*  1933*  1935*  1991   2012   2027   2039   2051*  2073   2092*
                    2095*  2106*  2107*  2110*  2111*  2168   2195   2212   2254   2275*  2277   2300*  2309
                    2311*  2312   2386*  2387*  2388   2394*  2397*  2408*  2503   2411*  2412   2460   2472*
                    2474   2485   2486   2520*  2526*  2732*  2746*  2761*  2804*  2808*  2826   2828*  2832*
                    2836*  2840*  2844*  2861*  2865*  2869*  3083*  3173*  3197   3224*  3245*  3250*  3282*
                    3364   3372*  3391   3398*
R5SAV   001270       684*
R6    =x000006       387*
SAVLDR  014000       745    2988*
SAVLDX  014076      3001    3008*
SAVO4 = 104005      3232    3274   3338*
SAVOSS= 104010      3344*
SCAN    001352       419    705*   721
SCANA   001374       712*   803
SCANB   001400       709    713*
SCNFLG  001220       663*   710*   712*   800
SCOPE = 104001       826    878    954    988   1080   1134   1263   1298   1392   1458   1561   1652   1757
                    1877   2053   2229   3330*
SCOPEB  016116      3429   3433   3437*
SCOPEC  016054      2271   2383   2490   3331   3428*
SCOPEG  016130      3431   3435   3440*
SHOBE   000534       440*  1907*  2094*  2396*  2841
SP    =x000006       377*   635*   636    637*   639    640    641*   644    647*   649*   655    689*   705*
                     727*   732*   733*   739    740    741    793   1053   1062   1069   1287   1356*  1359*
                    1372   1518   1620   1747   1864   2024   2209   2251*  2270*  2304*  2325*  2330   2343*
                    2357*  2364*  2370   2382*  2429*  2436*  2450*  2467*  2477*  2489*  2499*  2517*  2581
                    2596   2604   2659   2633*  2884*  2887*  2888*  2889*  2898   2899   2900   2901   2902
                    2908*  2909*  2920   2921   2945*  2965   2981   2988*  2989*  2992*  2993*  3008   3009
                    3010   3011   3044*  3045*  3046*  3047*  3048*  3049*  3050*  3051*  3102   3103   3104
                    3105   3106   3107   3108   3109   3118*  3142   3161*  3162*  3163*  3164*  3165*  3166*
                    3167*  3168*  3169*  3170*  3171*  3172   3178   3201*  3213*  3214*  3215*  3217   3298
                    3301*  3312*  3313*  3314*  3315   3319*  3320*  3321*  3322*  3323   3350*  3351*  3362*
                    3363*  3364*  3365*  3366*  3367*  3368*  3369*  3370   3372   3378   3379   3380   3381
                    3382   3383   3384*  3385*  3391*  3392   3393   3394   3395   3396   3397   3398   3399*
                    3400*
SPSAV   001254       678*   3298*  3301
SR0   = 177572       389*   719*  1074*  1125*  1256*  1381*  1449*  1552*  1644*  1804*  2087*  2550   2576*
                    2691*  2700*  2934*  3119
START   001432       415    727*   813
START1  001566       711    744    759*
STKPT = 000510       380*   689    705    727   2517
SUBTEN  015320      3240   3251*
SUBTNA  015324      3252*  3255
SUBTNB  015340      3253   3256*
SVO4    015620      3339   3350*
SVOSA   015650      3359   3362*
SVOSB   015662      3353   3365*
SVOSC   015676      3352*  3358*  3371*
SVOSS   015640      3345   3358*
SWR     001100       623*   735    737*   975   2258   2260*  2505   2507*  2533   2857   2874   3089   3182
                    3413   3417   3428   3430
SWREG   000176       414*   737
TBWNK   000546       445*  2558*  2564   2569*  2574   2616*  2631   2673*  2684*
```

```
TEMP    000562            452#
TEMPX   000514            431#    774*    776     783*    785    2731*   2733
TEMPWR  015364           3239*   3252    3256    3262#
TESTX   015770           3411#
TEST1   002064            815     826#
TEST10  003712           1281    1298#
TEST11  004234           1382    1392#
TEST12  004450           1458#
TEST13  004754           1561#
TEST14  005234           1652#
TEST15  005636           1689    1757#
TEST16  006304           1761    1806    1877#
TEST17  007144           1900    2053#
TEST2   002236            878#
TEST20  010212           2252    2262    2270#
TEST21  010614           2295    2382#
TEST3   002460            954#
TEST4   002606            957     988#
TEST5   003054           1080#
TEST6   003230           1134#
TEST7   003570           1263#
TMAP    012546            973    2714#
TMAPA   012562           2718#   2730    2779    2785
TMAPB   012566           2740#   2773    2789
TMAPC   012754           2742    2758#
TMAPD   013020           2744    2757    2759    2772#
TMAPEX  013070           2720    2790#
TNUM  = 000022            819#    827     828#    879     880#    955     956#    989     990#   1081    1082#   1135    1136#
                         1264    1265#   1299    1300#   1393    1394#   1459    1460#   1562    1563#   1653    1654#   1758
                         1759#   1878    1879#   2054    2055#   2272    2273#   2384    2385#
TPB     001106            626#    655*   2540*
TPCORE  005450           1681    1693#
TPCORX  006120           1798    1810#
TPS     001104            625#    653     898    2538
TRDATA  000540            442#   2803*   2809    2827*   2833
TREG    000770            551#    797     842*    843*    869     871*    893     933     971    1093    1146    1272    1313
                         1409    1475    1574    1679    1705*   1713    1716*   1727*   1734    1739*   1796    1824*   1832
                         1835*   1843*   1850    1855*   1916    1926    1943    1961    1965*   1970    1984*   1987    2000
                         2010    2026*   2028    2102    2118    2136    2140*   2145    2160*   2164    2177    2193    2211*
                         2213    22 6    2315*   2322    2334*   2341    2404    2419    2422*   2427    2442*   2445    2448
                         2458    2473*   2475    2629    2641    2645    2719    2803    2805    2827    2829    2913    3070
                         3153    3296
TRFLG   000522            434#   2722*   2743    2745*   2758    2760*
TRP10   004076           1328    1343#
TRP10A  004146           1358    1361#
TRP12   004620           1487    1503#
TRP13   005110           1588    1602#
TRP14   005556           1656    1727#
TRP15   006224           1769    1843#
TRP16   006662           1879    1956#
TRP16A  006666           1957#   1962
TRP17   007534           2059    2131#
TRP17A  007540           2132#   2137
TRP20   010460           2279    2334#
TRP4A   003004           1024    1048#
TRP4B   003012           1031    1055#
```

```
TRP4C   003020          1034    1064#
TRP7    003700          1266    1282#
TSTLOC  000532           438#   2826*  2837
TSTX    000510           380     429#   694*   742*  3416*  3432  3440*
TST10   004000          1310    1316#
TST11   004330          1405    1412#
TST12   004544          1471    1479#
TST13   005034          1571    1578#
TST14   005332          1660    1667#
TST15   006004          1772    1782#
TST4    002676          1001    1006#
TST5    003134          1091    1096#
TST6    003310          1144    1150#
TST7    003644          1270    1275#
TYCOR   000526           436#   2726*  2740*  2747   2762
TYFLG   000524           435#   2724*  2756*  2771*  2776
TYPE  = 104000           715     747    749    759    779   788   807  2524  2715  2736  2738  2750  2752
                        2754    2765   2767   2769   3035  3087  3180  3305  3328#
TYPSAX  015100          3199    3201#
TYPSBX  015110          3197*   3198   3203#
TYPSX   015062          2869    3197#  3204
WAS     000536           441#   1696*  1697   1814*  1815  1990*  2167*  2845
WAP   = 000004           370#   1025   1108   1112   1198  1225   1324   1422  1488  1519  1589  1621  1912
                        1924    1941   1959   2099   2116  2134   2402   2417  2627  2638
WAP16   006452          1896    1905#
WAP16A  006470          1908#   2042
WAP17   007366          2085    2092#
WAP17A  007404          2095#   2228
WAP21   010650          2394#
WAP21A  010664          2397#   2487
XFR1    010024          2057    2089   2229#
XFR2    011260          2493#
$ENDAD  011434           407    2529#
$FILLS  001111           628#
$NULL   001110           627#    647
$TPFLG  001112           629#    632    896    2536
$TYPE   001114           632#   3202   3329
.     = 017430           401#    403#   406#   408#   412#  416#  418#  420#  423#  622#  718   810   837
                         844     858    888    899    913   923   976  1018  1065  1073  1102  1116  1124
                        1156    1186   1204   1213   1221  1231  1241  1249  1255  1318  1350  1365  1380
                        1415    1429   1436   1448   1482  1505  1513  1528  1536  1551  1582  1604  1613
                        1629    1643   1676   1698   1709  1730  1760  1792  1816  1828  1846  1911  1923
                        1940    1958   1971   1977   1992  2018  2029  2037  2056  2098  2115  2133  2146
                        2152    2153   2169   2202   2214  2222  2234  2238  2241  2283  2303  2310  2318
                        2337    2401   2416   2428   2435  2476  2484  2534  2539  2626  2637  2680  2875
                        2911    2913   2915   3038   3066  3090  3097  3149  3183  3189  3293  3303*  3304
                        3318    3418   3420   3591#
```

```
EMTDEF     395#   3327   3330   3332   3334   3336   3338   3340   3342   3344
TESTX      397#    827    879    955    989   1081   1135   1264   1299   1393   1459   1562   1653   1758   1878
          2054   2272   2384
```

```
                2643  2664  2684  2685  2691  2694  2695  2696  2704  2717  2726  2727  2728  2731  2745
                2786  2787  2800  2801  2802  2803  2823  2824  2825  2826  2827  2828  2829  2830  2859
                2883  2894  2885  2886  2887  2888  2889  2890  2891  2892  2898  2899  2900  2901  2902
                2908  2909  2910  2912  2920  2921  2929  2930  2931  2932  2933  2936  2937  2947  2958
                2964  2965  2969  2970  2972  2981  2988  2989  2990  2991  2992  2993  2994  3002  3003
                3008  3009  3010  3011  3021  3022  3023  3031  3032  3044  3045  3046  3047  3048  3049
                3050  3051  3052  3054  3059  3063  3082  3092  3102  3103  3104  3105  3106  3107  3108
                3109  3118  3126  3142  3144  3147  3154  3162  3196  3197  3201  3209  3210  3211  3213
                3233  3234  3235  3236  3237  3238  3239  3244  3275  3276  3277  3290  3291  3298  3300
                3301  3312  3314  3322  3350  3351  3352  3358  3362  3363  3364  3365  3366  3367  3368
                3369  3372  3379  3380  3381  3382  3383  3384  3385  3391  3393  3394  3395  3396  3397
                3398  3399  3400  3413  3416  3438  3442
MOVB             637   655  1169  1183  1199  1226  1920  1933  2106  2110  2408  2411  2540  2660  2979
                3177  3179  3217  3258  3278
NOP             1277  1810  2530  2531  2532  3062
RESET            900  2528
ROL             3168  3170  3319
ROR             3219  3220  3221
RTI              642  1357  1360  2877  2896  2991  3022  3352  3371  3373  3386  3401  3443
RTS              657  1715  1724  1748  1834  1840  1865  2041  2226  2324  2371  2578  2601  2681  2701
                2790  2903  2922  2935  2966  2982  3012  3110  3200  3224  3250  3259  3282
SEC             2995  3024
SUB             2499  2500  2501  2502  2503  2504  2507  2860  2998  3027  3243  3252  3313
SWAB            2014  2197  2462
TST              639   706   717   735   743   770   796   800   805   809   867   891   912   922   932
                 956   992  1003  1029  1035  1064  1072  1083  1111  1114  1115  1123  1137  1170  1171
                1184  1185  1202  1203  1218  1220  1229  1230  1246  1248  1254  1301  1312  1331  1337
                1379  1395  1408  1428  1447  1461  1474  1512  1535  1550  1564  1593  1612  1625  1642
                1683  1684  1702  1710  1731  1759  1776  1799  1800  1821  1829  1847  1918  1928  1968
                1970  1986  2000  2027  2028  2038  2055  2074  2104  2143  2145  2163  2177  2212  2213
                2223  2233  2248  2290  2291  2308  2319  2338  2350  2351  2353  2406  2409  2425  2427
                2444  2448  2474  2475  2485  2496  2550  2565  2566  2567  2568  2592  2593  2594  2595
                2634  2649  2652  2669  2679  2682  2743  2758  2774  2776  2874  2948  2967  2996  3019
                3025  3056  3061  3067  3095  3096  3127  3137  3146  3150  3187  3188  3432
TSTB             632   653   896   898  2412  2536  2538  2657
.ABS             331
.ASCII          3453  3454  3459  3461  3466  3470  3472  3474  3476  3543  3548  3568
.ASCIZ          3457  3464  3478  3480  3483  3486  3491  3496  3501  3508  3515  3516  3517  3518  3519
                3524  3529  3531  3539  3546  3552  3554  3557  3559  3563  3573  3579  3585
.BYTE            627   628   629   630  3268  3542
.END            3592
.EVEN           3591
.LIST            332   401   819   828   880   956   990  1082  1136  1265  1300  1394  1460  1563  1654
                1759  1879  2055  2273  2385  3327  3330  3332  3334  3336  3338  3340  3342  3344  3346
.MACR            395
.MACRO           397
.NLIST           332   401   819   828   880   956   990  1082  1136  1265  1300  1394  1460  1563  1654
                1759  1879  2055  2273  2385  3327  3330  3332  3334  3336  3338  3340  3342  3344  3346
.REM               1
.REPT            401
.TITLE           332
.WORD            413   414   623   624
```

```
DCMFA.C MACY11 27(732)  10-SEP-76  09:52  PAGE 83
DCMFAC.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *DCMFAC,DCMFAC.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DCMFAC.P11
 RUN-TIME: 11 25 4 SECONDS
 RUN-TIME RATIO: 63/43=1.4
 CORE USED:  11K  (21 PAGES)
```