

PDP11/45

COMBINED MS 11, MF11, MA11 P
MD-11-DCMFA-C

EP-DCMFA C-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

This microfiche card contains a grid of frames, each containing technical data. The data is organized into columns and rows, with some frames containing diagrams or tables. The text is small and difficult to read, but appears to be technical specifications or test results for the PDP11/45 system. The frames are arranged in a regular grid pattern, with some frames containing more detailed information than others.

.REM *

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

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

COPYRIGHT (C) 1973, 1975
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORTATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

101
102
103
104
105
106
107

1.0 ABSTRACT

THIS PROGRAM LOCATES THE PARITY MEMORY REGISTERS FOR BOTH THE CORE AND MOS PARITY MEMORIES AND PERFORMS 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 BK 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)

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163

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 ENTS 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 STYPE (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.

164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219

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=ZZZZZ PC OF FAILING ERROR CALL. REFER TO THIS ADDRESS IN THE LISTING FOR AN EXPLANATION OF THE ERROR.

ICNT=YYYYY CURRENT ITERATION COUNT OF FAILING TEST.
MPR=XXXXXX ADDRESS OF PARITY REGISTER UNDER TEST.
MPR DATA=VVVVV 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

F01

220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275

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 ZEROS (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 ZEROS TO THE RIGHT OF THE NUMBER IN KPAR1. THIS THE VALUE 2400 INDICATES THAT THE BANK STARTS AT 240000. SECOND, CALCULATE THE OFFSET WITHIN THE BANK. THE VIRTUAL ADDRESS 32000 BREAKS DOWN INTO 1 (TOP 3 BITS) WHICH REFERENCES KPAR1, AND 12000 (LOWER 13 BITS) WHICH IS THE OFFSET. ADD THE OFFSET (12000) TO THE BANK ADDRESS (240000) TO GET THE ACTUAL PHYSICAL ADDRESS BEING TESTED (252000).

6.3 ERROR RECOVERY

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

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

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

7.0 RESTRICTIONS

THE PROGRAM REQUIRES A MINIMUM OF 8K MEMORY TO RUN. 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

GO1

276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326

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 M11 & M11 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 (INDCO-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.

*

327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382

```
: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).
```

000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000
000001
000004
077400
100000
177570
177570
177776
000007
000006
000240
000000
000510
000000
000001

```
:SYMBOL DEFINITIONS
BIT0=1
BIT1=2
BIT2=4
BIT3=10
BIT4=20
BIT5=40
BIT6=100
BIT7=200
BIT8=400
BIT9=1000
BIT10=2000
BIT11=4000
BIT12=10000
BIT13=20000
BIT14=40000
BIT15=100000
AE=1
WMP=4
ADRS=77400
PERR=100000
DSMR=177570
DDISP=177570
PS=177776
PC=%7
SP=%6
NOP=240
OPEN=0
STKPT=TSTX
RO=%0
RI=%1
```

:BIT DEFINITIONS

```
:ACTION ENABLE
:WRITE WRONG PARITY
:ADDRESS OF ERROR
:PARITY ERROR BIT
:HARDWARE SWITCH REGISTER
:HARDWARE DISPLAY REGISTER
```



```

383      000002
384      000003
385      000004
386      000005
387      000006
388      000114
389      177572
390
391
392
393
394
395
396
397
398
399
400
401
402
403      000030
404      000030 015530
405      000032 000340
406      000046
407      000046 011434
408      000052
409      000052 040000
410
411
412      000174 000174
413      000174 000000
414      000176 000000
415      000200 000167 001226
416      000210 000210
417      000210 000167 013702
418      000220 000220
419      000220 000167 001126
420      000230 000230
421      000230 000167 001036
422
423      000510
424
425
426
427
428
429      000510 000000
430      000512 000000
431      000514 000000
432      000516 000000
433      000520 000000
434      000522 000000
435      000524 000000
436      000526 000000
437      000530 000000
438      000532 000000
    
```

```

R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
    
```

```

PARVEC=114
SRO=177572
    
```

```

;PARITY ERROR TRAP VECTOR
;ADDRESS OF MEM MGMT REGISTER SRO
    
```

```

;MACRO DEFINITIONS
    
```

```

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

```

;LOAD ENT VECTOR
    
```

```

.=30
ENTINT
340
.=46
SENDAD
.=52
BIT14
    
```

```

;LOAD STARTING ADDRESS AREA
    
```

```

.=174
DISPREG: .WORD 0
SWREG:   .WORD 0
JMP     START
.=210
JMP     RSTLDR
.=220
JMP     SCAN
.=230
JMP     RSTART
.=510
    
```

```

;THIS IS THE SOFTWARE DISPLAY REGISTER
;THIS IS THE SOFTWARE SWITCH REGISTER
;GO TO START OF PROGRAM
;GO RESTORE THE LOADERS
;SCAN FOR BAD PARITY
;RESTART WITHOUT RETYPING MAP INFORMATION
    
```

```

;GENERAL DATA AREA
    
```

```

TSTX: 0
FTITLE: 0
TEMPX: 0
ADRPT: 0
BITPT: 0
TRFLG: 0
TYFLG: 0
TYCOR: 0
HIADR: 0
TSTLOC: 0
    
```

```

;TITLE PRINTED = 1
    
```

```

;MAPPING- ADDRESS POINTER
;MAPPING- BIT POINTER INDICATING BANK
;MAPPING- TRANSITION FLAG
;MAPPING- TYPED FLAG
;MAPPING- K CORE ACCUMULATOR
;USED TO CHECK WHEN DONE TESTING A BANK
;LOADED WITH ADDRESS OF LOCATION UNDER
    
```

```

439
440 000534 000000
441 000536 000000
442 000540 000000
443 000542 000000
444 000544 000000
445 000546 000000
446 000550 000000
447 000552 000000
448 000554 000000
449 000556 000000
450 000560 000000
451
452 000562 000000
453 000564 000000
454
455 000566 000000
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486 000570 172101
487 000572 000000
488 000574 000000
489 000576 000000
490 000600 172103
491 000602 000000
492 000604 000000
493 000606 000000
494 000610 172105

```

```

SHOBE: 0
WAS: 0
TRDATA: 0
MPROK: 0
PASCNT: 0
TBANK: 0
MEMUT: 0
NOKT: 0
HIWORD: 0
LOWFLG: 0
ODDFLG: 0
TEMP: 0
MTYFG: 0
RELOC: 0

```

```

;TEST IN SOME SUBTESTS
;VALUE EXPECTED
;ACTUAL VALUE FOUND

;PASS COUNT

;SET TO INDICATE NO KT11 PRESENT

;IF SET INDICATES TESTING HIGH BYTE
;OF MEMORY LOCATION

;SET TO INDICATE MAP OF PARITY MEMORY
;ALREADY TYPED

```

```

;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 MPRO AND MPR1 EXIST, CONTROLLING INTERLEAVED MEMORY
;FROM 0 TO 16K, AND THAT MPRO CONTROLS THE ADDRESSES ENDING IN 0 AND 4.
;THE MAP WOULD THEN LOOK AS FOLLOWS:

```

```

MPRO: 172100
      17
      0
      400

MPR1: 172102
      17
      0
      402

```

```

;BIT 0 IS CLEAR SINCE REGISTER IS PRESENT
;REGISTER CONTROLS 1ST 16K (=4 BANKS)

;LOW BYTE SHOWS THAT FIRST ADDRESS
;ENDS IN 0 (OCTAL)
;HIGH BYTE CONTAINS A 1 TO INDICATE
;THAT AN ADDRESS WAS FOUND
;BIT 0 IS CLEAR SINCE REGISTER IS PRESENT
;REGISTER CONTROLS 1ST 16K

;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

```

```

MPRO: 172100+1
      0
      0
      0
MPR1: 172102+1
      0
      0
      0
MPR2: 172104+1

```

```

;PARITY STATUS REGISTERS
;0-64K PARITY MEM UNDER THIS CONTROL
;64-124K PARITY MEM UNDER THIS CONTROL
;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)

;0-64K PARITY MEM UNDER THIS CONTROL
;64-124K PARITY MEM UNDER THIS CONTROL
;ADDRESS RESPONSE THIS CONTROL (0,2,4,6)

```

K01

495	000612	000000	0		
496	000614	000000	0		
497	000616	000000	0		
498	000620	172107	MPR3:	172106+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
499	000622	000000	0		
500	000624	000000	0		
501	000626	000000	0		
502	000630	172111	MPR4:	172110+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
503	000632	000000	0		
504	000634	000000	0		
505	000636	000000	0		
506	000640	172113	MPR5:	172112+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
507	000642	000000	0		
508	000644	000000	0		
509	000646	000000	0		
510	000650	172115	MPR6:	172114+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
511	000652	000000	0		
512	000654	000000	0		
513	000656	000000	0		
514	000660	172117	MPR7:	172116+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
515	000662	000000	0		
516	000664	000000	0		
517	000666	000000	0		
518	000670	172121	MPR8:	172120+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
519	000672	000000	0		
520	000674	000000	0		
521	000676	000000	0		
522	000700	172123	MPR9:	172122+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
523	000702	000000	0		
524	000704	000000	0		
525	000706	000000	0		
526	000710	172125	MPR10:	172124+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
527	000712	000000	0		
528	000714	000000	0		
529	000716	000000	0		
530	000720	172127	MPR11:	172126+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
531	000722	000000	0		
532	000724	000000	0		
533	000726	000000	0		
534	000730	172131	MPR12:	172130+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
535	000732	000000	0		
536	000734	000000	0		
537	000736	000000	0		
538	000740	172133	MPR13:	172132+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
539	000742	000000	0		
540	000744	000000	0		
541	000746	000000	0		
542	000750	172135	MPR14:	172134+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
543	000752	000000	0		
544	000754	000000	0		
545	000756	000000	0		
546	000760	172137	MPR15:	172136+1	:0-64K PARITY MEM UNDER THIS CONTROL :64-124K PARITY MEM UNDER THIS CONTROL :ADDRESS RESPONSE THIS CONTROL (0,2,4,6)
547	000762	000000	0		
548	000764	000000	0		
549	000766	000000	0		
550					

L01

551 000770 000000

TREG: 0

;PARITY REGISTER UNDER TEST

552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606

000772 000000
000774 000000
000776 000000
001000 000000
001002 000000
001004 000000
001006 000000
001010 000000
001012 000000
001014 000000
001016 000000
001020 000000
001022 000000
001024 000000
001026 000000
001030 000000
001032 000000

001034 070032
001036 077772

001040 125325
001042 152652
001044 052452
001046 025125
001050 102070
001052 072527
001054 177777
001056 107030
001060 152525
001062 000000
001064 000000

001066 000000
001070 000000

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

INDC0: 0
INDC1: 0
INDC2: 0
INDC3: 0
INDC4: 0
INDC5: 0
INDC6: 0
INDC7: 0
INDC8: 0
INDC9: 0
INDC10: 0
INDC11: 0
INDC12: 0
INDC13: 0
INDC14: 0
INDC15: 0
RESRVD: 0

;CORE-MOS PARITY INDICATOR FOR MPR
;CORE-MOS PARITY INDICATOR FOR MPR1
;CORE-MOS PARITY INDICATOR FOR MPR2
;CORE-MOS PARITY INDICATOR FOR MPR3
;FOR MPR4
;FOR MPR5
;FOR MPR6
;FOR MPR7
;FOR MPR8
;FOR MPR9
;FOR MPR10
;FOR MPR11
;FOR MPR12
;FOR MPR13
;FOR MPR14
;FOR MPR15

RESVC: 70032
RESVM: 77772

;BIT POSITIONS WHICH ARE RESERVED
;FOR FUTURE USE IN PARITY REGISTERS
;CORE PARITY
;MOS PARITY

;PARITY PATTERNS
PARPAT: 125325
152652
052452
025125
102070
072527
177777
107030
152525
0
0

;EVEN, ODD BYTES
;ODD, EVEN
;EVEN, ODD
;ODD, EVEN
;EVEN, EVEN
;ODD, ODD
;EVEN, EVEN
;ODD, ODD
;ODD, EVEN
;EXTRA PATTERN AREA
;TERMINATOR, DO NOT USE THIS LOC

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

MO1

DCNFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 12
DCNFAC.P11

```

607          ;THIS IS A MAP OF THE TOTAL PARITY MEMORY PRESENT IN THE SYSTEM.
608 001072 000000 PMEML: 0          ;0-64K PARITY MEMORY PRESENT
609          ;(IN 4K CONTIGUOUS BLOCKS)
610 001074 000000 PMEMH: 0        ;64-124K PARITY MEMORY PRESENT
611          ;(IN 4K CONTIGUOUS BLOCKS)
612 001076 000000 PMEMX: 0        ;TEMP TO HOLD CONTENTS OF EITHER
613          ;LOW OR HIGH MAP
614
615
616
617          ;ROUTINE TO TYPE ASCII MESSAGES, MESSAGE MUST TERMINATE WITH A 0 BYTE.
618          ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
619          ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
620          ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
621
622          .=1100
623 001100 177570 SWR: .WORD DSWR          ;ADDRESS OF THE SWITCH REGISTER
624 001102 177570 DISPLAY: .WORD DDISP    ;ADDRESS OF THE DISPLAY REGISTER
625 001104 177564 TPS: 177564            ;PRINTER STATUS REGISTER ADDRESS
626 001106 177566 TPB: 177566            ;PRINTER BUFFER REGISTER ADDRESS
627 001110 000 SNUL: .BYTE 0             ;CONTAINS NULL CHARACTER FOR FILLS
628 001111 002 SFILLS: .BYTE 2          ;CONTAINS # OF FILLER CHARACTERS REQUIRED
629 001112 000 STPFLG: .BYTE 0          ;"TERMINAL AVAILABLE" FLAG (0=YES)
630 001113 000          .BYTE 0          ;RESERVED
631
632 001114 105767 177772 STYPE: TSTB STPFLG          ;IS THERE A TERMINAL?
633 001120 001401 BEQ 6$                ;BR IF YES
634 001122 000000 HALT 6$                ;HALT HERE IF NO TERMINAL
635 001124 010046 6$: MOV RO, -(SP)       ;SAVE RO
636 001126 017600 000002 MOV @2(SP), RO  ;GET ADDRESS OF ASCIZ STRING
637 001132 112046 1$: MOVB (RO)+, -(SP)  ;PUSH CHARACTER TO BE TYPED ONTO STACK
638 001134 001005 BNE 2$                ;BR IF IT ISN'T THE TERMINATOR
639 001136 005726 TST (SP)+            ;IF TERMINATOR POP IT OFF THE STACK
640 001140 012600 MOV (SP)+, RO        ;RESTORE RO
641 001142 062716 000002 7$: ADD #2, (SP) ;ADJUST RETURN PC
642 001146 000002 RTI                    ;RETURN
643 001150 004767 000026 2$: JSR PC, 5$  ;GO TYPE THIS CHARACTER
644 001154 122726 000012 3$: CMPB #12, (SP)+ ;CHECK IF THE CHARACTER TYPED
645          ;WAS A LINE FEED
646 001160 001364 BNE 1$                ;GO GET NEXT CHARACTER IF NOT LINE FEED
647 001162 016746 177722 MOV $NULL, -(SP) ;GET # OF FILLER CHARACTERS NEEDED
648          ;AND THE NULL CHARACTER
649 001166 105366 000001 4$: DECB 1(SP)  ;DOES A NULL NEED TO BE TYPED?
650 001172 002770 BLT 3$                ;BR IF NO--GO POP THE NULL OF THE STACK
651 001174 004767 000002 JSR PC, 5$      ;GO TYPE A NULL
652 001200 000772 BR 4$                ;LOOP
653 001202 105777 177676 5$: TSTB @TPS ;WAIT UNTIL PRINTER IS READY
654 001206 100375 BPL 5$
655 001210 116677 000002 177670 MOVB 2(SP), @TPB ;LOAD CHARACTER TO BE
656          ;TYPED INTO DATA REGISTER
657 001216 000207 RTS PC
658
659
660
661          ;GENERAL DATA AREA
662

```

NO1

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 13
DCMFAC.P11

663	001220	000000		SCNFLG: 0	
664					; SCNFLG GETS SET IF USING
665	001222	000000		KTSTART: 0	; SCAN ROUTINE (SA=220)
666	001224	000000		ADRTYP: 0	
667	001226	177600		PDRTAB: 177600	
668	001230	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		ROSAV: 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					
689	001272	012706	000510		; ROUTINE TO RESTART WITHOUT RETYPING MAP AFTER TEST HAS BEEN RUNNING
690	001276	012767	000001	177260	RSTART: MOV #STKPT, SP ; SET UP STACK POINTER
691	001304	005067	177234		MOV #1, MYFG ; SET FLAG TO INDICATE MAP HAS BEEN TYPED
692	001310	012737	015430	000024	CLR PASCNT ; INITIALIZE PASS COUNT
693	001316	012737	000340	000026	MOV #PWRDN, a#24
694	001324	005067	177160		MOV #340, a#26
695	001330	005037	177776		CLR TSTX
696	001334	012737	000006	000004	CLR a#PS ; CLEAR PROCESSOR STATUS REGISTER
697	001342	005037	000006		MOV #6, a#4
698	001346	000167	000470		CLR a#6
699					JMP BEGIN
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			; BRANCH, REGISTERS HAVE ALREADY BEEN LOCATED
710	001364	005267	177630		BNE SCANB ; INCREMENT SCNFLG
711	001370	000167	000172		INC SCNFLG
712	001374	005067	177620		JMP START1 ; GO TO LOCATE THE REGISTERS
713	001400	004767	010100	SCANA:	CLR SCNFLG ; RETURN HERE AFTER LOCATING THE REGISTERS
714	001404	004767	012602	SCANB:	JSR PC, MAPMEM ; SETUP MEMORY MAP
715	001410	104000			JSR PC, PSCAN ; SCAN FOR BAD PARITY
716	001412	017201			TYPE PMSG ; TYPE MESSAGE "BAD PARITY SCAN COMPLETE"
717	001414	005767	177132		TST NOKT
718	001420	001002			BNE .+6

719	001422	005037	177572		CLR	28SR0		:TURN OFF KT11 IF PRESENT
720	001426	000000			HALT			:END OF PARITY SCAN
721	001430	000750			BR	SCAN		
722								
723								
724								
725								
726								
727	001432	012706	000510		:NORMAL STARTUP			
728	001436	005067	177122		START: MOV	8STKPT, SP		:SET UP STACK POINTER
729	001442	005067	177076		CLR	MTYFG		:CLEAR FLAG WHICH INDICATES MAP TYPED
730	001446	012737	015430	000024	CLR	PASCNT		:INITIALIZE PASS COUNT
731	001454	012737	000340	000026	MOV	8PHRON, 2824		:SETUP POWER FAIL RETURN
732	001462	013746	000004		MOV	8340, 2826		
733	001466	013746	000006		MOV	284, -(SP)		:SAVE THE ERROR VECTOR
734	001472	012737	001506	000004	MOV	286, -(SP)		
735	001500	005777	177374		MOV	825, 284		:SET UP TIME OUT VECTOR
736	001504	000407			TST	25WR		:TRY TO REFERENCE THE SWITCH REGISTER
737	001506	012767	000176	177364	BR	45		:BRANCH IF NO TIME OUT
738	001514	012767	000174	177360	25: MOV	8SWREG, SWR		:POINT TO THE SOFTWARE SWITCH REGISTER
739	001522	022626			MOV	8DISPREG, DISPLAY		:POINT TO THE SOFTWARE DISPLAY REG.
740	001524	012637	000006		45: CMP	(SP)+, (SP)+		:RESTORE THE STACK POINTER
741	001530	012637	000004		MOV	(SP)+, 286		:RESTORE ERROR VECTOR
742	001534	005067	176750		MOV	(SP)+, 284		
743	001540	005767	176746		CLR	TSTX		
744	001544	001010			15: TST	FTITLE		:IS TITLE PRINTED YET?
745	001546	004767	012226		BNE	START1		:YES, SKIP OVER
746	001552	005267	176734		JSR	PC, SAVLDR		:COPY LOADER TO LOWER 4K
747	001556	104000			INC	FTITLE		:SET FLAG
748	001560	016746			TYPE			:TYPE TITLE "MEMORY PARITY TEST
749	001562	104000			MTIT			:MAINDEC-11-DCMFA"
750	001564	017234			TYPE			
751					MIDRSV			:TYPE "LOADERS SAVED IN BANK D.
752								:TO RESTORE LOADERS, USE SA 210"
753								
754								
755								
756								
757								
758								
759	001566	104000			:SEARCH FOR PARITY REGISTERS PRESENT AND TYPE ADDRESSES OF THOSE FOUND			
760	001570	016516			:FAILURE TO LOCATE A REGISTER INDICATES THAT THE ADDRESS TIMED OUT OR THAT			
761	001572	005067	176744		:BITS 5-7 IN THE REGISTER DID NOT SET			
762	001576	012702	000570		START1: TYPE			:TYPE "MEMORY PARITY REGISTERS PRESENT ARE:"
763	001602	012703	000772		MIPRS			
764	001606	012737	001746	000004	CLR	MPROK		:CLEAR MPR FLAG
765	001614	005037	000006		MOV	8MPRO, R2		:SET UP POINTERS
766	001620	042712	000001		MOV	8INDCO, R3		:POINTER TO CORE-MOS
767	001624	005062	000002		MOV	8GMPRB, 284		:SET UP TIMEOUT TRAP RETURN
768	001630	005062	000004		CLR	286		
769	001634	005062	000006		GMPRA: BIC	81, (2)		:CLEAR FLAG BIT IN TABLE
770	001640	005772	000000		CLR	2(R2)		:INITIALIZE LOCATIONS IN THE TABLE
771	001644	052772	000340	000000	CLR	4(R2)		
772	001652	032772	000340	000000	CLR	6(R2)		
773	001660	001414			TST	2(2)		:DOES THIS MPR EXIST? (IF NO, TIMES OUT)
774	001662	011267	176626		771: BIS	8340, 2(2)		:YES- IS IT AN MF11-LP OR MA11-P CORE PARITY REG
					BIT	8340, 2(2)		
					BEQ	15		:NO, IS IT A MOS-11 PARITY REGISTER? BRANCH
					MOV	(2), TEMPX		:YES- PRINT REGISTER ADDRESS

```

775 001666 004567 013222 JSR RS,0ACNV ;(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$: MOV (2),TEMPX ;IT IS A MOS REGISTER, PRINT ADDRESS
784 001716 004567 013172 JSR RS,0ACNV ;(GET ASCII)
785 001722 000514 TEMPX
786 001724 017367 MPRMOS
787 001726 000006 6
788 001730 104000 TYPE ;(TYPE ADDRESS)
789 001732 017367 MPRMOS
790 001734 012713 177777 MOV #-1,(R3) ;SET INDICATOR FOR MOS PARITY
791 001740 005267 176576 2$: INC MPROK ;SET MPR REGISTER PRESENT FLAG
792 001744 000403 BR GMPRC ;SKIP NEXT
793 001746 022626 GMPRB: 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 020227 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 002002 001402 BEQ GMPRO ;MEMORY FOR BAD PARITY-(SCAN)
802 002004 000167 177364 JMP SCANA ;NO BRANCH TO CARRY ON NORMALLY
803 002010 005767 176526 GMPRO: TST MPROK ;YES, GO BACK TO THE MEMORY
804 002014 001012 BNE BEGIN ;SCAN ROUTINE
805 002016 104000 NOREG: TYPE ;ANY PARITY REGISTERS PRESENT?
806 002020 016701 MTR ;YES- GO TEST CONTROLS PRESENT
807 002022 005737 000042 TST #42 ;NO- TYPE "NO PARITY REGISTER FOUND"
808 002026 001402 BEQ +6 ;LOADED BY MONITOR?
809 002030 000167 007370 JMP LOGICAL ;NO BRANCH
810 002034 000000 HALT ;YES- EXIT, NO REGISTERS PRESENT
811 002036 000167 177370 JMP START ;NO REGISTERS TO TEST
812 002042 012767 002066 014102 BEGIN: MOV #TEST1+2,RETURN ;IF CONTINUED, TRY AGAIN
813 002050 012767 000100 014070 MOV #100,IMAX ;SETUP SCOPE RETURN
814 002056 012737 000006 000004 MOV #6,#4 ;MAXIMUM ITERATION COUNT
815 002064 104001 TEST1: SCOPE ;RESTORE TRAPCATCHER IN TIMEOUT VECTOR
816 002066 012777 000001 177006 MOV #1,DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY
817 002074 012700 000570 MOV #MPRO,R0 ;LOAD ADDRESS OF TABLE INTO R0
818 002100 012704 000772 MOV #INDCO,R4 ;LOAD ADDRESS OF INDICATOR IN R4
819 002104 032710 000001 1$: BIT #1,R0 ;IS THIS REGISTER PRESENT?

```


831	002110	001042			BNE	45		:NO- BRANCH TO GET NEXT ADDRESS
832	002112	011001			MOV	3R0,R1		:YES- LOAD R1 WITH ADDRESS OF
833								:PARITY REGISTER
834	002114	022714	000001		CMP	81,(R4)		:IS THIS REGISTER CORE?
835	002120	001004			BNE	55	;NO	
836	002122	016767	176706	176702	MOV	RESVC,RESVD		:YES, CORE. STORE RESERVED BITS
837	002130	000403			BR	.+10		
838	002132	016767	176700	176672	MOV	RESVM,RESVD	55:	:MOS, STORE RESERVED BITS
839	002140	012702	000001		MOV	81,R2		:LOAD R2 WITH VALUE OF FIRST BIT
840								:TO BE TESTED
841	002144	005011			CLR	3R1		:INITIALIZE PARITY REGISTER
842	002146	011167	176616		MOV	3R1,TREG		:READ CONTENTS OF PARITY REGISTER
843	002152	046767	176654	176610	BIC	RESVD,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		BIT	R2,RESVD	25:	:IS THIS BIT RESERVED?
850	002170	001010			BNE	35		:YES- DON'T TEST IT SINCE IT
851								:MAY BE ZERO OR ONE
852	002172	010211			MOV	R2,3R1		:NO- SET THIS BIT IN THE PARITY REGISTER
853	002174	011103			MOV	3R1,R3		:READ AND SAVE CONTENTS OF PARITY REGISTER
854	002176	005011			CLR	3R1		:CLEAR PARITY REGISTER
855	002200	046703	176626		BIC	RESVD,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 VALUE IN R2
861								:WAS WRITTEN INTO IT. ACTUAL CONTENTS
862								: (WITH UNUSED BITS CLEARED) IS IN R3
863	002212	006302			ASL	R2	35:	:ROTATE BIT TO BE TESTED
864	002214	103363			BCC	25		:IF NOT DONE WITH ALL BIT POSITIONS
865								:GO TEST THIS ONE
866	002216	062700	000010		ADD	#10,R0	45:	: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	15		:NO, BRANCH
871	002232	005067	176532		CLR	TREG		
872								
873								:*****
874								:SHOW THAT RESET CLEARS BITS 0,2, AND 15 OF EACH PARITY REGISTER
875								:PRESENT.
876								:*****
877								TEST2: SCOPE
878	002236	104001						
879	002240	012777	000002	176634	MOV	#2,3DISPLAY		:LOAD THE TEST NUMBER INTO THE DISPLAY
880	002246	005067	013674		CLR	IMAX		:DON'T ITERATE TEST
881	002252	012700	000570		MOV	#MPRO,R0		:LOAD POINTER
882	002256	012703	000772		MOV	#INDCO,R3		:POINTER TO INDICATOR
883	002262	032710	000001		BIT	81,3R0	15:	:IS THIS PARITY REGISTER PRESENT?
884	002266	001012			BNE	55		:NO BRANCH
885	002270	022713	000001		CMP	81,(R3)		:IS THIS CORE OR MOS PARITY REGISTER?
886	002274	001404			BEQ	65		:NO- BRANCH

887	002276	012770	100015	000000		MOV	#100015,2(R0)	:MOS-SET ALL DEFINED BITS TO 1
888	002304	000403				BR	.+10	
889	002306	012770	107745	000000	6S:	MOV	#107745,2(R0)	:CORE- SET ALL DEFINED BITS TO 1
890	002314	062700	000010		5S:	ADD	#10,R0	:MOVE POINTER TO POINT TO NEXT AFR 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			4S:	RESET		:ISSUE INIT
901	002346	012700	000570			MOV	#MPRO,R0	:LOAD ADDRESS OF THE TABLE
902	002352	012703	000772			MOV	#INDCO,R3	:POINTER TO INDICATOR
903	002356	032710	000001		2S:	BIT	#1,2R0	: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	2(R0),R2	:YES, GET CONTENTS OF REGISTER
908	002376	005070	000000			CLR	2(R0)	:MAKE SURE THAT MIP AND AE ARE CLEAR
909	002402	042702	077772			BIC	#77772,R2	:MASK RESERVED BITS FOR CORE PAR
910								:ITY REGISTER. BITS 5-11(ADORS
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		7S:	MOV	2(R0),R2	:MOS, GET CONTENTS OF REGISTER
920	002422	005070	000000			CLR	2(R0)	:MAKE SURE THAT MIP BAE 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 WHOSE 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	2(R0)	:REINITIALIZE PARITY REGISTER
930	002444	062700	000010		3S:	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								
939								
940								
941								
942								

:MAP CORRESPONDENCE BETWEEN PARITY REGISTERS AND MEMORY, AND TYPE RESULTS
:NOTE THAT IF PARITY MEMORY IS NOT LOCATED CORRECTLY BY THIS SUBTEST
:IT IS DUE TO ONE OF THE FOLLOWING FAILURES:
:-SETTING WRITE WRONG PARITY DID NOT CAUSE BAD PARITY TO BE WRITTEN
:-PARITY GENERATE OR DETECT LOGIC FAILED

943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998

```
002460 104001
002462 012777 000003 176412
002470 005767 176070
002474 001044
002476 004767 011006
002502 004767 006776
002506 004767 007320

002512 005067 176354
002516 005067 176352
002522 012701 000570
002526 032711 000001
002530 001006
002534 056167 000002 176330
002542 056167 000004 176324
002550 062701 000010
002554 020127 000770
002560 103762
002562 004767 007760
002566 005267 175772
002572 032777 001000 176300
002600 001402
002602 000000

002604 000742
```

```
:-PARITY ERROR BIT FAILED TO SET
:-PARITY BITS IN MEMORY LOCATION FAILED (I.E. BIT STUCK AT GOOD PARITY VALUE)
NOTE THAT SETTING SWITCH REGISTER SWITCH 9 WILL CAUSE A HALT AFTER THE MAP
IS TYPED. IF YOU WISH TO CHANGE THE MAP TO ISOLATE THE CAUSE OF A MAPPING
FAILURE, YOU CAN DO THIS ONCE THE PROCESSOR IS HALTED. SEE THE DESCRIPTION
IN THE LISTING (PRECEDING THE MAP TAG "MPRO" AT LOCATION 600) FOR THE MEANING
OF THE MAP CONTENTS. AFTER MAKING THE DESIRED CHANGES, PRESS CONTINUE. THE NEW
MAP WILL BE TYPED AND IF SWITCH 9 IS LEFT SET THE PROCESS WILL BE REPEATED.
IF SWITCH 9 IS NOT LEFT SET THE PROGRAM WILL PROCEED TO TEST THE PARITY MEMORY
AND REGISTERS AS RECORDED IN THE NEW MAP.
```

```
*****
TEST3: SCOPE
MOV #3,DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY
TST MTYFG ;IF MAPPING HAS ALREADY BEEN DONE
BNE TEST4 ;SKIP SUBTEST
JSR X7,CLPAR
JSR X7,MAPMEM ;MAP MEMORY
JSR X7,MAPREG ;FIND PARITY MEMORY AND CORRESPONDING
;REGISTERS USING WRITE WRONG PARITY
;WITHOUT ACTION ENABLE SET
;INITIALIZE LOCATIONS INDICATING
;TOTAL PARITY MEMORY PRESENT

CONT3: CLR PHEML
CLR PHEMH
MOV #MPRO,R1
1S: BIT #1,R1
BNE 2S
BIS 2(R1),PHEML ;FLAG EXISTING PARITY MEMORY (LOW 64K)
BIS 4(R1),PHEMH ;FLAG EXISTING PARITY MEMORY (HIGH 64K)
2S: ADD #10,R1
CMP R1,#TREG
BLO 1S
JSR X7,THAP ;TYPE MAP
INC MTYFG ;INDICATE MAPPING DONE
BIT #BIT9,JSR ;SWITCH 9 SET?
BEQ .+6 ;NO- BRANCH
HALT ;YES- SWITCH 9 SET INDICATING HALT
;AFTER TYPING PARITY MEMORY MAP
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
*****
```

```
TEST4: SCOPE
MOV #4,DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY
MOV #100,IMAX
JSR X7,CLPAR ;CLEAR ALL PARITY REGISTERS
TST NOKT ;KT11 PRESENT?
BNE 1S ;NO-BRANCH
JSR X7,NRALL ;YES-MAP ALL PAGES NON RESIDENT
JSR X7,MAP1 ;THEN MAP KERNEL 0 TO BANK 0, KERNEL
;7 TO EXTERNAL BANK, SET KERNEL
;0,1, AND 7 RW, AND TURN ON KT11
1S: 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 PTR9LT
1004	002672	103771			BLO	LOOP4			
1005	002674	000457			BR	DONE4			: BRANCH WHEN ALL REGISTERS HAVE BEEN TESTED
1006	002676	004767	010722	TST4:	JSR	X7,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			: SETUP PARITY TRAP RETURN
1024	002716	012737	003004	000114	MOV	#TRP4A,#114			: SET WRITE WRONG PARITY
1025	002724	012770	000004	000000	MOV	#MWP,#(R0)			: WRITE CONTENTS OF LOCATION WITH
1026	002732	011111			MOV	R1,R1			: WRONG PARITY
1027									: CLEAR PARITY REGISTER
1028	002734	005070	000000		CLR	(R0)			: READ BAD PARITY WITH ACTION ENABLE
1029	002740	005711			TST	R1			: CLEARED- NO TRAP EXPECTED
1030									: CHANGE PARITY TRAP RETURN
1031	002742	012737	003012	000114	MOV	#TRP4B,#PARVEC			: SET ACTION ENABLE WITH PARITY ERROR
1032	002750	052770	000001	000000	BIS	#AE,(R0)			: ALREADY SET- SHOULDN'T TRAP YET
1033									: CHANGE PARITY TRAP RETURN
1034	002756	012737	003020	000114	MOV	#TRP4C,#PARVEC			: READ LOCATION AGAIN- SHOULD GET
1035	002764	005711			TST	R1			: A PARITY TRAP DUE TO READING BAD
1036									: PARITY WITH ACTION ENABLE SET
1037									: NO PARITY TRAP AFTER READING LOCATION
1038	002766	104002			ERROR				: WHICH SHOULD CONTAIN BAD PARITY-
1039									: R1 CONTAINS ADDRESS OF MEMORY LOCATION
1040									: (VIRTUAL, THRU KERNEL PAGE 1, IF KT11
1041									: PRESENT). R0 POINTS TO THE ADDRESS OF
1042									: THE PARITY REGISTER IN WHICH AE WAS SET
1043									: CLEAR PARITY REGISTER
1044	002770	005070	000000	CONT4:	CLR	(R0)			: CLEAR BAD PARITY
1045	002774	005511			ADC	R1			: CLEAR PARITY ERROR BIT
1046	002776	005070	000000		CLR	(R0)			: GO TO TEST NEXT REGISTER
1047	003002	000730			BR	LOP4			: PARITY TRAP OCCURRED WITH ACTION
1048	003004	104002		TRP4A:	ERROR				: ENABLE CLEAR- R0 POINTS TO THE ADDRESS
1049									: OF THE PARITY REGISTER UNDER TEST
1050									: R1 CONTAINS THE ADDRESS OF THE MEMORY
1051									: UNDER TEST (VIRTUAL IF KT11 IS PRESENT)
1052									: RESTORE STACK POINTER
1053	003006	022626			CMP	(SP)+,(SP)+			
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)+
1063 003016 000764          BR      CONT4
1064 003020 005770 000000    TRP4C: TST      @ (R0)
1065 003024 100401          BMI     .+4
1066 003026 104002          ERROR
1067                                     ;ERROR BIT SET AFTER PARITY TRAP?
1068                                     ;YES- BRANCH
1069                                     ;ERROR BIT NOT SET AFTER PARITY
1070                                     ;TRAP- R0 POINTS TO THE ADDRESS
1071                                     ;OF THE PARITY REGISTER UNDER TEST
1072                                     ;RESTORE STACK POINTER
1073 003030 022626          CMP      (SP)+,(SP)+
1074 003032 000756          BR      CONT4
1075 003034 012737 000116 000114  DONE4: MOV     @PARVEC+2,@PARVEC          ;RESTORE TRAP CATCHER
1076 003042 005767          TST     NOKT
1077 003046 001002          BNE     .+6
1078 003050 005037 177572    CLR     @SR0          ;TURN OFF KT11 IF PRESENT
1079                                     ;*****
1080                                     ;SHOW THAT READING GOOD PARITY AFTER BAD PARITY DOESN'T CLEAR PARITY ERROR BIT
1081                                     ;*****
1082 003054 104001          TEST5: SCOPE
1083 003056 012777 000005 176016  MOV     @5,@DISPLAY
1084 003064 004767 010420    JSR     PC,CLAPAR
1085 003070 005767 175456    TST     NOKT
1086 003074 001004          BNE     IS
1087 003076 004767 010306    JSR     PC,NRALL
1088 003102 004767 010452    JSR     PC,MAP1
1089                                     ;LOAD THE TEST NUMBER INTO THE DISPLAY
1090                                     ;CLEAR ALL PARITY REGISTERS
1091                                     ;KT11 PRESENT?
1092                                     ;NO- BRANCH
1093                                     ;YES, MAP ALL PAGES NON-RESIDENT
1094                                     ;THEN MAP KERNEL 0 TO BANK 0. MAP KERNEL
1095                                     ;7 TO THE EXTERNAL BANK. SET KERNEL
1096                                     ;0,1, AND 7 RW, AND TURN ON KT11
1097                                     ;SETUP TO FIND REGISTERS PRESENT
1098 003106 012700 000570    IS:    MOV     @MPRO,R0
1099 003112 032710 000001    LOPS:  BIT     @1,@R0
1100 003116 001406          BEQ     TST5
1101 003120 062700 000010    LOPS:  ADD     @10,R0
1102 003124 020027 000770    CMP     R0,@TREG
1103 003130 103770          BLO     LOPS
1104 003132 000431          BR      DONES
1105 003134 004767 010464    TST5:  JSR     PC,LOCATM
1106                                     ;EXIT WHEN ALL REGISTERS HAVE BEEN TESTED
1107                                     ;LOCATE MEMORY CORRESPONDING TO THIS
1108                                     ;REGISTER- R1 WILL BE RETURNED CONTAINING
1109                                     ;THE ADDRESS OF THE FIRST LOCATION
1110                                     ;CONTROLLED BY THIS REGISTER (MAPPED
1111                                     ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1112 003140 032701 000001          BIT     @1,R1
1113 003144 001403          BEQ     .+10
1114 003146 104002          ERROR
1115                                     ;IS ERROR RETURN INDICATED?
1116                                     ;NO- BRANCH
1117                                     ;MAP INDICATES NO PARITY MEMORY IS
1118                                     ;CONTROLLED BY THIS REGISTER. R0
1119                                     ;POINTS TO THE ADDRESS OF THE
1120                                     ;PARITY REGISTER
1121 003150 000167 177744          JMP     LOPS
1122 003154 012770 000004 000000  MOV     @MWP,@(R0)
1123 003162 011111          MOV     @R1,@R1
1124                                     ;SET MWP IN THIS REGISTER
1125                                     ;WRITE CONTENTS OF LOCATION WITH
1126                                     ;WRONG PARITY

```


1167	003336	005002			CLR	R2			:R2 CONTAINS VALUE TO BE LOADED
1168									:INTO MEMORY
1169	003340	110211		1S:	MOVB	R2,R1			:WRITE VALUE INTO LOW BYTE
1170	003342	005711			TST	R1			:READ WORD TO CHECK PARITY
1171	003344	005770	000000		TST	R(R)			:CHECK PARITY REGISTER
1172	003350	100006			BPL	5S			: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 (HWP WAS NOT SET)
1177	003354	005070	000000		CLR	R(R)			: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	2S			
1181	003366	105202		5S:	INCB	R2			:INCREMENT VALUE TO BE LOADED
1182	003370	001363			BNE	1S			:LOOP UNTIL ALL VALUES HAVE BEEN USED
1183	003372	110261	000001	2S:	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	R(R)			: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	6S			:IN R1 (HWP WAS NOT SET)
1191	003412	105202			INCB	R2			:INCREMENT VALUE TO BE LOADED
1192	003414	001366			BNE	2S			: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		6S:	CLR	R1			:INITIALIZE LOCATION UNDER TEST
1197	003420	005002			CLR	R2			:INITIALIZE VALUE TO BE WRITTEN
1198	003422	012770	000004	3S:	MOV	HWP,R(R)			:SET WRITE WRONG PARITY
1199	003430	110211			MOVB	R2,R1			:WRITE WRONG PARITY IN LOW BYTE
1200	003432	005070	000000		CLR	R(R)			: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	R(R)			: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	3S			:LOOP TILL DONE WITH ALL VALUES
1216	003456	005011			CLR	R1			:REINITIALIZE LOCATION TO CLEAR BAD PARITY
1217	003460	005070	000000		CLR	R(R)			:CLEAR ERROR IF SET
1218	003464	005711			TST	R1			:READ LOCATION, WHICH SHOULD NOW HAVE
1219									:GOOD PARITY
1220	003466	005770	000000		TST	R(R)			:PARITY ERROR SET?
1221	003472	100001			BPL	+4			:NO, BRANCH
1222	003474	104002			ERROR				:GOOD PARITY WAS NOT RESTORED BY

L02

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 24
DCMFAC.P11

```
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                                     ;RD POINTS TO THE ADDRESS OF THE
1286                                     ;PARITY REGISTER
1287 003702 022626          CMP    (SP)+,(SP)+    ;RESTORE STACK POINTER
1288 003704 005070 000000          CLR    @ (RD)        ;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			TEST10:	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, MAPI			: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	#MPRO, R0			
1308	003750	012702	000772			MOV	#INDCO, 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	#MWP, @ (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	#BITS, @ (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      2S:  MOV      #INST1,SP      ;GO EXECUTE INSTRUCTION 1 AGAIN
1357 004130 000002                                     ;
1358 004132 012737 004146 000114 1S:  MOV      #TRP10A,2#PARVEC ;CHANGE PARITY TRAP RETURN
1359 004140 012716 004066                                     ;GO EXECUTE INSTRUCTION 2
1360 004144 000002                                     ;
1361 004146 022712 000001      TRP10A: CMP      #1,(R2)      ;IS THIS A CORE REG?
1362 004152 001005                                     ;NO BRANCH
1363 004154 032770 000040 000000      BNE      1S          ;PARITY TRAP OCCURRED- CHECK PARITY
1364                                     ;ERROR ADDRESS BITS
1365                                     ;BRANCH IF OK (IF THE PARITY ERROR
1366                                     ;ADDRESS BITS TRACKED, BIT 5 WILL BE SET)
1367 004162 001001      BNE      .+4          ;PARITY ERROR ADDRESS BITS INCORRECT
1368                                     ;RO 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                                     ;
1373 004170 012737 000116 000114 1S:  CONT10: CMP      (SP)+,(SP)+ ;RESTORE TRAPCATCHER
1374 004176 005070 000000      MOV      #PARVEC+2,2#PARVEC ;CLEAR PARITY REGISTER
1375 004202 005511      CLR      2(R0)          ;CLEAR BAD PARITY
1376 004204 005561 004000      ADC      2R1          ;
1377 004210 005070 000000      ADC      4000(R1)     ;
1378 004214 000662      CLR      2(R0)          ;CLEAR PARITY ERROR BIT IF SET
1379 004216 005767 174330      BR       LOOP10      ;
1380 004222 001002      DONE10: TST      NOKT ;
1381 004224 005037 177572      BNE      .+6          ;TURN OFF KT11 IF PRESENT
1382 004230 000401      CLR      2#SRO      ;
1383 004232 000000      BR       TEST11     ;
1384      COUNT: 0
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      #11,2DISPLAY ;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      1S         ;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      1S:  MOV      #MPRO,RO ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1401 004272 012703 000772      MOV      #INDCO,R3
1402 004276 032710 000001      LUP11: BIT      #1,2RO ;
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      TEST11     ;YES, THEN TEST IT
1406                                     ;IF NOT CORE, SKIP THIS REGISTER
1407 004312 062700 000010      LOOP11: ADD     #10,RO ;
1408 004316 005723      TST      (R3)+
1409 004320 020027 000770      CMP      RO,#TREG

```

1410	004324	103764			BLO	LUP11			
1411	004326	000443			BR	DONE11			
1412	004330	004767	007270		TST11: JSR	%7,LOCATM			: BRANCH OUT IF ALL REGISTERS HAVE BEEN TESTED
1413									: GET THE ADDRESS OF A MEMORY LOCATION
1414	004334	032701	000001		BIT	#1,R1			: CORRESPONDING TO THIS PARITY REGISTER
1415	004340	001403			BEQ	+.10			: ERROR RETURN INDICATED?
1416	004342	104002			ERROR				: BRANCH IF NOT
1417									: NO MEMORY IN MAP CORRESPONDING TO
1418									: THIS PARITY REGISTER. R0 POINTS
1419	004344	000167	177742		JMP	LOOP11			: TO THE ADDRESS OF THE PARITY REGISTER
1420	004350	010102			MOV	R1,R2			: ; SETUP SECOND TEST ADDRESS LOCATION
1421	004352	062702	010000		ADD	#10000,R2			
1422	004356	012770	000004	000000	MOV	#MAP,2(R0)			: ; SET WRITE WRONG PARITY
1423	004364	011111			MOV	2R1,2R1			: ; WRITE WRONG PARITY IN FIRST TEST LOCATION
1424	004366	011212			MOV	2R2,2R2			: ; WRITE WRONG PARITY IN SECOND TEST LOCATION
1425	004370	005070	000000		CLR	2(R0)			: ; CLEAR PARITY REGISTER
1426	004374	021112			CMP	2R1,2R2			: ; READ FIRST TEST LOCATION, AND
1427									: ; THEN READ SECOND TEST LOCATION
1428	004376	005770	000000		TST	2(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	#BIT6,2(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	2(R0)			: ; CLEAR PARITY REGISTER
1443	004424	005511			ADC	2R1			: ; CLEAR BAD PARITY
1444	004426	005512			ADC	2R2			
1445	004430	005070	000000		CLR	2(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	2#SR0			: ; TURN OFF KT11 IF PRESENT
1450									
1451									
1452									
1453									
1454									
1455									
1456									
1457									
1458	004450	104001			TEST12: SCOPE				
1459	004452	012777	000012	174422	MOV	#12,2DISPLAY			: ; LOAD THE TEST NUMBER INTO THE DISPLAY
1460	004460	004767	007024		JSR	%7,CLPAR			
1461	004464	005767	174062		TST	NOKT			: ; KT11 PRESENT?
1462	004470	001004			BNE	1\$: ; NO- BRANCH
1463	004472	004767	006712		JSR	%7,NRALL			: ; 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

: SHOW THAT IF AN INSTRUCTION DOING A DATIP GETS A PARITY ERROR,
: THE ORIGINAL DATA IS REWRITTEN IF ACTION ENABLE IS SET, AND IS
: ALTERED IF ACTION ENABLE IS CLEAR
: *****

1466	004502	012700	000570	IS:	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	#MWP, #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, #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	#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	#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			CMP	(SP)+, (SP)+	:RESTORE STACK POINTER
1519	004646	012770	000004	000000	MOV	#MWP, #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      2(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      2(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          ;THRU KERNEL PAGE 1 IF KT11 IS PRESENT)
1536 004710 005770 000000          TST     2(R0)          ;CHECK PARITY ERROR BIT
1537 004714 100001          BPL     .+4
1538 004716 104002          ERROR
1539          ;DATIP, DATO WITH AE CLEAR DID
1540          ;NOT CLEAR BAD PARITY IN LOCATION
1541          ;ADDRESSED. R0 POINTS TO THE ADDRESS
1542          ;OF THE PARITY REGISTER. R1 CONTAINS
1543          ;THE ADDRESS OF THE TEST LOCATION
1544          ;(VIRTUAL THRU KERNEL PAGE 1 IF KT11
1545          ;IS PRESENT)
1546 004720 005070 000000          CONT12: CLR     2(R0)   ;CLEAR PARITY REGISTER
1547 004724 005011          CLR     R1           ;CLEAR LOCATION TO RESTORE GOOD PARITY
1548 004726 005070 000000          CLR     2(R0)        ;CLEAR PARITY ERROR IF SET
1549 004732 000675          BR      LOOP12       ;GO CHECK FOR ANOTHER PARITY
1550          ;REGISTER
1551 004734 012737 000116 000114    DONE12: MOV     @PARVEC+2,@PARVEC ;RESTORE TRAPCATCHER
1552 004742 005767 173604          TST     NOKT
1553 004746 001002          BNE     .+6
1554 004750 005037 177572          CLR     @SRO         ;TURN OFF KT11 IF PRESENT
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 UNALTERED, 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
1565 004774 001004          BNE    1$
1566 004776 004767 006406          JSR    %7,NRALL
1567 005002 004767 006552          JSR    %7,MAP1
1568          ;KT11 PRESENT?
1569          ;NO- BRANCH
1570          ;YES, MAP KERNEL 0 TO BANK 0, KERNEL
1571          ;7 TO THE EXTERNAL BANK, AND KERNEL
1572          ;0,1, AND 7 RM
1573          ;SETUP TO GET ADDRESSES OF REGISTERS PRESENT
1574 005006 012700 000570          1$: MOV     @MPRO,R0
1575 005012 032710 000001    LUP13: BIT     @1,@R0
1576 005016 001406          BEQ    TST13
1577          ;IF THIS REGISTER IS PRESENT, GO
1578          ;TEST IT
1579
1580          ;*****
1581          ;*****
1582          ;*****
1583          ;*****
1584          ;*****
1585          ;*****
1586          ;*****
1587          ;*****
1588          ;*****
1589          ;*****
1590          ;*****
1591          ;*****
1592          ;*****
1593          ;*****
1594          ;*****
1595          ;*****
1596          ;*****
1597          ;*****
1598          ;*****
1599          ;*****
1600          ;*****
1601          ;*****
1602          ;*****
1603          ;*****
1604          ;*****
1605          ;*****
1606          ;*****
1607          ;*****
1608          ;*****
1609          ;*****
1610          ;*****
1611          ;*****
1612          ;*****
1613          ;*****
1614          ;*****
1615          ;*****
1616          ;*****
1617          ;*****
1618          ;*****
1619          ;*****
1620          ;*****
1621          ;*****
1622          ;*****
1623          ;*****
1624          ;*****
1625          ;*****
1626          ;*****
1627          ;*****
1628          ;*****
1629          ;*****
1630          ;*****
1631          ;*****
1632          ;*****
1633          ;*****
1634          ;*****
1635          ;*****
1636          ;*****
1637          ;*****
1638          ;*****
1639          ;*****
1640          ;*****
1641          ;*****
1642          ;*****
1643          ;*****
1644          ;*****
1645          ;*****
1646          ;*****
1647          ;*****
1648          ;*****
1649          ;*****
1650          ;*****
1651          ;*****
1652          ;*****
1653          ;*****
1654          ;*****
1655          ;*****
1656          ;*****
1657          ;*****
1658          ;*****
1659          ;*****
1660          ;*****
1661          ;*****
1662          ;*****
1663          ;*****
1664          ;*****
1665          ;*****
1666          ;*****
1667          ;*****
1668          ;*****
1669          ;*****
1670          ;*****
1671          ;*****
1672          ;*****
1673          ;*****
1674          ;*****
1675          ;*****
1676          ;*****
1677          ;*****
1678          ;*****
1679          ;*****
1680          ;*****
1681          ;*****
1682          ;*****
1683          ;*****
1684          ;*****
1685          ;*****
1686          ;*****
1687          ;*****
1688          ;*****
1689          ;*****
1690          ;*****
1691          ;*****
1692          ;*****
1693          ;*****
1694          ;*****
1695          ;*****
1696          ;*****
1697          ;*****
1698          ;*****
1699          ;*****
1700          ;*****
1701          ;*****
1702          ;*****
1703          ;*****
1704          ;*****
1705          ;*****
1706          ;*****
1707          ;*****
1708          ;*****
1709          ;*****
1710          ;*****
1711          ;*****
1712          ;*****
1713          ;*****
1714          ;*****
1715          ;*****
1716          ;*****
1717          ;*****
1718          ;*****
1719          ;*****
1720          ;*****
1721          ;*****
1722          ;*****
1723          ;*****
1724          ;*****
1725          ;*****
1726          ;*****
1727          ;*****
1728          ;*****
1729          ;*****
1730          ;*****
1731          ;*****
1732          ;*****
1733          ;*****
1734          ;*****
1735          ;*****
1736          ;*****
1737          ;*****
1738          ;*****
1739          ;*****
1740          ;*****
1741          ;*****
1742          ;*****
1743          ;*****
1744          ;*****
1745          ;*****
1746          ;*****
1747          ;*****
1748          ;*****
1749          ;*****
1750          ;*****
1751          ;*****
1752          ;*****
1753          ;*****
1754          ;*****
1755          ;*****
1756          ;*****
1757          ;*****
1758          ;*****
1759          ;*****
1760          ;*****
1761          ;*****
1762          ;*****
1763          ;*****
1764          ;*****
1765          ;*****
1766          ;*****
1767          ;*****
1768          ;*****
1769          ;*****
1770          ;*****
1771          ;*****
1772          ;*****
1773          ;*****
1774          ;*****
1775          ;*****
1776          ;*****
1777          ;*****
1778          ;*****
1779          ;*****
1780          ;*****
1781          ;*****
1782          ;*****
1783          ;*****
1784          ;*****
1785          ;*****
1786          ;*****
1787          ;*****
1788          ;*****
1789          ;*****
1790          ;*****
1791          ;*****
1792          ;*****
1793          ;*****
1794          ;*****
1795          ;*****
1796          ;*****
1797          ;*****
1798          ;*****
1799          ;*****
1800          ;*****
1801          ;*****
1802          ;*****
1803          ;*****
1804          ;*****
1805          ;*****
1806          ;*****
1807          ;*****
1808          ;*****
1809          ;*****
1810          ;*****
1811          ;*****
1812          ;*****
1813          ;*****
1814          ;*****
1815          ;*****
1816          ;*****
1817          ;*****
1818          ;*****
1819          ;*****
1820          ;*****
1821          ;*****
1822          ;*****
1823          ;*****
1824          ;*****
1825          ;*****
1826          ;*****
1827          ;*****
1828          ;*****
1829          ;*****
1830          ;*****
1831          ;*****
1832          ;*****
1833          ;*****
1834          ;*****
1835          ;*****
1836          ;*****
1837          ;*****
1838          ;*****
1839          ;*****
1840          ;*****
1841          ;*****
1842          ;*****
1843          ;*****
1844          ;*****
1845          ;*****
1846          ;*****
1847          ;*****
1848          ;*****
1849          ;*****
1850          ;*****
1851          ;*****
1852          ;*****
1853          ;*****
1854          ;*****
1855          ;*****
1856          ;*****
1857          ;*****
1858          ;*****
1859          ;*****
1860          ;*****
1861          ;*****
1862          ;*****
1863          ;*****
1864          ;*****
1865          ;*****
1866          ;*****
1867          ;*****
1868          ;*****
1869          ;*****
1870          ;*****
1871          ;*****
1872          ;*****
1873          ;*****
1874          ;*****
1875          ;*****
1876          ;*****
1877          ;*****
1878          ;*****
1879          ;*****
1880          ;*****
1881          ;*****
1882          ;*****
1883          ;*****
1884          ;*****
1885          ;*****
1886          ;*****
1887          ;*****
1888          ;*****
1889          ;*****
1890          ;*****
1891          ;*****
1892          ;*****
1893          ;*****
1894          ;*****
1895          ;*****
1896          ;*****
1897          ;*****
1898          ;*****
1899          ;*****
1900          ;*****
1901          ;*****
1902          ;*****
1903          ;*****
1904          ;*****
1905          ;*****
1906          ;*****
1907          ;*****
1908          ;*****
1909          ;*****
1910          ;*****
1911          ;*****
1912          ;*****
1913          ;*****
1914          ;*****
1915          ;*****
1916          ;*****
1917          ;*****
1918          ;*****
1919          ;*****
1920          ;*****
1921          ;*****
1922          ;*****
1923          ;*****
1924          ;*****
1925          ;*****
1926          ;*****
1927          ;*****
1928          ;*****
1929          ;*****
1930          ;*****
1931          ;*****
1932          ;*****
1933          ;*****
1934          ;*****
1935          ;*****
1936          ;*****
1937          ;*****
1938          ;*****
1939          ;*****
1940          ;*****
1941          ;*****
1942          ;*****
1943          ;*****
1944          ;*****
1945          ;*****
1946          ;*****
1947          ;*****
1948          ;*****
1949          ;*****
1950          ;*****
1951          ;*****
1952          ;*****
1953          ;*****
1954          ;*****
1955          ;*****
1956          ;*****
1957          ;*****
1958          ;*****
1959          ;*****
1960          ;*****
1961          ;*****
1962          ;*****
1963          ;*****
1964          ;*****
1965          ;*****
1966          ;*****
1967          ;*****
1968          ;*****
1969          ;*****
1970          ;*****
1971          ;*****
1972          ;*****
1973          ;*****
1974          ;*****
1975          ;*****
1976          ;*****
1977          ;*****
1978          ;*****
1979          ;*****
1980          ;*****
1981          ;*****
1982          ;*****
1983          ;*****
1984          ;*****
1985          ;*****
1986          ;*****
1987          ;*****
1988          ;*****
1989          ;*****
1990          ;*****
1991          ;*****
1992          ;*****
1993          ;*****
1994          ;*****
1995          ;*****
1996          ;*****
1997          ;*****
1998          ;*****
1999          ;*****
2000          ;*****

```

E03

Address	OpCode	Op1	Op2	Op3	Op4	Label	Instruction	Comments
1578	005034	004767	006564			TST13:	JSR X7,LOCATM	:LOCATE MEMORY CORRESPONDING TO THIS REGISTER, AND IF KT11 IS PRESENT
1579								:MAP KERNEL 1 TO THAT MEMORY
1580								:ERROR RETURN INDICATED?
1581	005040	032701	000001				BIT #1,R1	:NO BRANCH
1582	005044	001403					BEQ .+10	:MAP INDICATES NO MEMORY WAS FOUND
1583	005046	104002					ERROR	:CORRESPONDING TO THIS REGISTER
1584								:RD POINTS TO THE ADDRESS OF THE
1585								:PARITY REGISTER
1586								
1587	005050	000167	177744				JMP LOOP13	:SETUP PARITY TRAP RETURN
1588	005054	012737	005110	000114			MOV #TRP13,2#PARVEC	:SET WRITE WRONG PARITY
1589	005062	012770	000004	000000			MOV #MWP,2(R0)	:WRITE WRONG PARITY
1590	005070	012711	125252				MOV #125252,2R1	:SET ACTION ENABLE AND CLEAR
1591	005074	012770	000001	000000			MOV #AE,2(R0)	:WRITE WRONG PARITY
1592								:DATA WITH ACTION ENABLE SET SHOULD
1593	005102	005711					TST 2R1	:ABORT LEAVING DATA UNCHANGED
1594								:NO ABORT ON READING BAD PARITY
1595	005104	104002					ERROR	:WITH ACTION ENABLE SET, RD POINTS
1596								:TO THE ADDRESS OF THE PARITY REGISTER.
1597								:R1 CONTAINS THE ADDRESS OF THE TEST
1598								:LOCATION (VIRTUAL THRU KERNEL PAGE
1599								:1 IF KT11 IS PRESENT).
1600								
1601	005106	000434					BR CONT13	:ABORT OCCURRED AS EXPECTED- CLEAR REGISTER
1602	005110	005070	000000			TRP13:	CLR 2(R0)	:ORIGINAL DATA RESTORED?
1603	005114	021127	125252				CMP 2R1,#125252	:YES BRANCH
1604	005120	001401					BEQ .+4	:DATA WHICH GOT A PARITY ERROR ALTERED
1605	005122	104002					ERROR	:THE CONTENTS OF THE LOCATION ADDRESSED.
1606								: (R1 CONTAINS THE ADDRESS OF
1607								:MEMORY BEING TESTED- IF KT11 IS
1608								:PRESENT, ADDRESS IN R1 IS VIRTUAL)
1609								:RD POINTS TO THE ADDRESS OF THE
1610								:PARITY REGISTER
1611								:CHECK PARITY REGISTER
1612	005124	005770	000000				TST 2(R0)	:BRANCH IF PARITY ERROR SET
1613	005130	100401					BMI .+4	:PARITY ERROR NOT SET AFTER READING
1614	005132	104002					ERROR	:DATA WITH BAD PARITY
1615								:RD POINTS TO THE ADDRESS OF THE PARITY
1616								:REGISTER, R1 CONTAINS THE ADDRESS
1617								:OF THE TEST LOCATION (VIRTUAL THRU
1618								:KERNEL PAGE 1 IF KT11 PRESENT)
1619								:RESTORE STACK POINTER
1620	005134	022626					CMP (SP)+,(SP)+	:SET WRITE WRONG PARITY, CLEAR PARITY ERROR
1621	005136	012770	000004	000000			MOV #MWP,2(R0)	:REWRITE DATA WITH WRONG PARITY
1622	005144	012711	125252				MOV #125252,2R1	:CLEAR PARITY REGISTER
1623	005150	005070	000000				CLR 2(R0)	:RESTORE TRAPCATCHER
1624	005154	012737	000116	000114			MOV #PARVEC+2,2#PARVEC	:DATA TO LOCATION WITH BAD PARITY
1625	005162	005711					TST 2R1	:AE NOT SET-INSTRUCTION SHOULD COMPLETE
1626								:CLEAR PARITY ERROR BIT
1627	005164	005070	000000				CLR 2(R0)	:CHECK DATA
1628	005170	022711	125252				CMP #125252,2R1	
1629	005174	001401					BEQ .+4	
1630	005176	104002					ERROR	
1631								:DATA 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

F03

```

1634
1635
1636 005200 005070 000000          CONT13: CLR      2(R0)
1637 005204 005011                    CLR      2R1
1638 005206 005070 000000          CLR      2(R0)
1639 005212 000702                    BR       LOOP13
1640
1641 005214 012737 000116 000114  DONE13: MOV      @PARVEC+2,@PARVEC
1642 005222 005767 173324                    TST     NOKT
1643 005226 001002                    BNE     +6
1644 005230 005037 177572                    CLR     2@SRO
1645
1646
1647
1648
1649
1650
1651
1652 005234 104001                    TEST14: SCOPE
1653 005236 012777 000014 173636          MOV     814,@DISPLAY
1654 005244 005067 010676                    CLR     IMAX
1655 005250 004767 006234                    JSR     PC,CLAPAR
1656 005254 012737 005556 000114          MOV     8TRAP14,@PARVEC
1657 005262 012767 000002 173230          MOV     82,BITPT
1658 005270 012767 020000 173220          MOV     82000,ADRPT
1659 005276 036767 173216 173566  LOOP14: BIT     BITPT,PHEML
1660 005304 001012                    BNE     TST14
1661 005306 062767 020000 173202  LUP14: ADD     82000,ADRPT
1662 005314 006367 173200                    ASL     BITPT
1663 005320 022767 000200 173172          CMP     8200,BITPT
1664 005326 003363                    BGT     LOOP14
1665
1666 005330 000443                    BR      DONE14
1667 005332 012704 001040          TST14: MOV     @PARPAT,R4
1668 005336 016767 173154 173164          MOV     ADRPT,HIADR
1669 005344 062767 020000 173156          ADD     82000,HIADR
1670 005352 016705 173140          MOV     ADRPT,R5
1671 005356 005025          2S:   CLR     (5)+
1672 005360 020567 173144          CMP     R5,HIADR
1673 005364 103774                    BLO     2S
1674 005366 012701 000570          MOV     @MPRO,R1
1675 005372 032711 000001          3S:   BIT     81,2R1
1676 005376 001003                    BNE     .+10
1677 005400 012771 000001 000000          MOV     @AE,2(R1)
1678 005406 062701 000010          ADD     810,R1
1679 005412 020127 000770          CMP     R1,@TREG
1680 005416 103765                    BLO     3S
1681 005420 004767 000024          4S:   JSR     %7,TPCORE
1682
1683 005424 005724                    TST     (4)+
1684 005426 005714                    TST     (4)
1685 005430 001373                    BNE     4S
1686 005432 004767 006052          JSR     PC,CLAPAR
1687 005436 000723                    BR      LUP14
1688 005440 012737 000116 000114  DONE14: MOV     @PARVEC+2,@PARVEC
1689 005446 000473                    BR      TEST15

```

```

;IF KT11 IS PRESENT), RD POINTS TO THE
;ADDRESS OF THE PARTIY REGISTER.
;CLEAR PARITY REGISTER
;CLEAR LOCATION
;CLEAR PARITY ERROR IF SET
;GO CHECK FOR ANOTHER PARITY
;REGISTER
;RESTORE TRAPCATCHER
;TURN OFF KT11 IF PRESENT

```

```

*****
;CHECK PARITY MEMORY WITH SERIES OF PATTERNS FROM 4K TO 28K
;ENABLE PARITY TRAP
*****

```

```

;LOAD THE TEST NUMBER INTO THE DISPLAY
;DON'T ITERATE THE REST OF THE SUBTESTS
;CLEAR ALL PARITY REGISTERS
;SETUP PARITY TRAP RETURN
;INITIALIZE BANK INDICATOR TO BANK 1
;INITIALIZE MEMORY STARTING ADDRESS
;DOES THIS 4K HAVE PARITY?
;YES, TEST IT
;NO- UPDATE MEMORY ADDRESS
;UPDATE BIT POINTER
;THIS 28K DONE?
;NO BRANCH TO SEE IF NEXT 4K
;SHOULD BE TESTED
;YES, EXIT
;INITIALIZE PATTERN POINTER
;SET UPPER LIMIT FOR THIS 4K
;INITIALLY CLEAR CORE BLOCK UNDER TEST
;INITIALIZE TO SET AE IN ALL REGISTERS
;SET ACTION ENABLE IF REGISTER IS PRESENT
;GO TO ROUTINE TO EXERCISE THIS 4K
;WITH THE CURRENT PATTERN
;UPDATE PATTERN
;LAST PATTERN?
;NO LOOP
;YES, CLEAR ALL PARITY REGISTERS
;UPDATE AND CHECK NEXT 4K
;RESTORE TRAP CATCHER
;GO TO NEXT TEST

```



```

1690
1691
1692
1693 005450 016705 173042
1694
1695 005454 011415
1696 005456 011567 173054
1697 005462 021467 173050
1698 005466 001401
1699 005470 104002
1700
1701
1702 005472 005725
1703 005474 020567 173030
1704 005500 103765
1705 005502 005067 173262
1706
1707 005506 012701 000570
1708 005512 032711 000001
1709 005516 001003
1710 005520 005771 000000
1711 005524 100406
1712 005526 062701 000010
1713 005532 020127 000770
1714 005536 103765
1715 005540 000207
1716 005542 011167 173222
1717 005546 104004
1718
1719
1720
1721 005550 004767 006436
1722
1723
1724 005554 000207
1725
1726
1727 005556 005067 173206
1728 005562 012701 000570
1729 005566 032711 000001
1730 005572 001003
1731 005574 005771 000000
1732 005600 100407
1733 005602 062701 000010
1734 005606 020127 000770
1735 005612 103765
1736 005614 104002
1737 005616 000406
1738
1739 005620 011167 173144
1740 005624 104004
1741
1742
1743
1744
1745 005626 004767 006360

```

```

;ROUTINE TO WRITE AND CHECK EACH LOCATION IN 4K (STARTING AT ADDRESS
;IN ADRPT) WITH VALUE POINTED TO BY R4
TPCORE: MOV ADRPT,R5
;SETUP R5 TO ADDRESS MEMORY
;LOCATION BEING CHECKED
;WRITE PATTERN INTO MEMORY
;READ TEST LOCATION
;DATA OK?
;YES- BRANCH
;DATA INCORRECT IN LOCATION WHOSE
;ADDRESS IS IN R5. R4 POINTS TO THE
;DATA WRITTEN.
;UPDATE ADDRESS POINTER
;THIS 4K DONE?
;NO, BRANCH TO TEST NEXT LOCATION
;YES, DID ANY PARITY ERRORS OCCUR
;WITHOUT TRAPPING?

1S: MOV (4),(5)
MOV (5),R5
CMP (4),R5
BEQ .+4
ERROR

TST (5)+
CMP R5,MIADR
BLO 1S
CLR TREG

2S: MOV #MPRO,R1
BIT #1,(1)
BNE .+10
TST @R1
BMI 3S ;YES- BRANCH
ADD #10,R1
CMP R1,@TREG
BLO 2S
RTS %7

3S: MOV @R1,TREG
ERRORS

JSR %7,PSCAN

RTS %7

;PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
TRP14: CLR TREG
MOV #MPRO,R1
;FIND PARITY REGISTER INDICATING PARITY ERROR

1S: BIT #1,(R1)
BNE .+10
TST @R1
BMI 2S ;BRANCH IF PARITY ERROR SET
ADD #10,R1
CMP R1,@TREG
BLO 1S
ERROR
BR 3S

2S: MOV @R1,TREG
ERRORS

JSR %7,PSCAN

```

H03

DCMFAC.MACY11 27(732) 10-SEP-76 09:52 PAGE 33
DCMFAC.P11

```
1746                                     :OF LOCATIONS FOUND BAD
1747 005632 022626                       3S:  CMP      (SP)+,(SP)+      :RESTORE STACK POINTER
1748 005634 000207                       RTS      X7                :RETURN (FROM JSR TO TPCORE) TO
1749                                     :CHECK NEXT PATTERN
1750
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 005624                  JSR      PC,CLAPAR       :CLEAR ALL PARITY REGISTERS
1763 005664 004767 005520                  JSR      X7,NRALL        :MAP KERNEL 0 TO BANK 0,RW
1764 005670 004767 005664                  JSR      X7,MAP1         :MAP KERNEL 7 TO THE EXTERNAL BANK
1765                                     :SET KERNEL 1 RW AND TURN ON KT11
1766 005674 012777 001600 173344          MOV      #1600,AKPAR1    :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      PHEML,PHEMX
1769 005714 012737 006224 000114          MOV      #TRP15,PARVEC  :SETUP PARITY TRAP RETURN
1770 005722 012767 000200 172570          MOV      #200,BITPT     :INITIALIZE BIT POINTER
1771 005730 036767 172564 173140          LOOP15: BIT BITPT,PHEMX :DOES THIS 4K HAVE PARITY?
1772 005736 001022                          BNE      TST15           :YES, BRANCH TO TEST IT
1773 005740 062777 000200 173300          LOP15:  ADD #200,AKPAR1  :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      PHEMH,PHEMX    :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                          CLR      (5)+           :INITIALLY CLEAR CORE BLOCK UNDER TEST
1787 006024 020527 040000                  CMP      R5,#40000
1788 006030 103774                          BLO     2S
1789 006032 012701 000570                  MOV      #MPRO,R1
1790                                     :INITIALIZE TO SET ACTION ENABLE IN ALL
1791                                     :PARITY REGISTERS
1791 006036 032711 000001                  3S:  BIT      #1,R1
1792 006042 001003                          BNE     +10
1793 006044 012771 000001 000000          MOV      #AE,2(R1)
1794                                     :SET ACTION ENABLE IF THIS REGISTER
1795                                     :IS PRESENT
1795 006052 062701 000010                  ADD      #10,R1
1796 006056 020127 000770                  CMP      R1,#TREG
1797 006062 103765                          BLO     3S
1798 006064 004767 000030                  4S:  JSR      X7,TPCORX
1799 006070 005724                          TST     (4)+
1800 006072 005714                          TST     (4)
1801 006074 001373                          BNE     4S
1801                                     :EXERCISE THIS 4K
1801                                     :UPDATE PATTERN
1801                                     :LAST PATTERN?
1801                                     :NO, LOOP
```

```

1802 006076 004767 005406 JSR PC,CLPAR ;YES, CLEAR ALL PARITY REGISTERS
1803 006102 000716 BR LOP15 ;UPDATE AND CHECK NEXT 4K
1804 006104 005037 177572 DONE15: CLR 2#SRD ;TURN OFF KT11 WHEN DONE
1805 006110 012737 000116 000114 MOV 2#PARVEC+2,2#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 1S: 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 1S ;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 2#PRO,R1
1827 006164 032711 000001 2S: BIT 21,(1) ;IS THIS PARITY REGISTER PRESENT?
1828 006170 001003 BNE .+10 ;NO, GET NEXT ONE
1829 006172 005771 000000 TST 2(R1) ;YES- DID ERROR SET?
1830 006176 100406 BMI 3S ;YES- BRANCH
1831 006200 062701 000010 ADD 210,R1 ;NO- GET NEXT REGISTER
1832 006204 020127 000770 CMP R1,2TREG
1833 006210 103765 BLO 2S
1834 006212 000207 RTS 27
1835 006214 011167 172550 3S: MOV 2R1,TREG ;NO ERRORS- EXIT
1836 006220 104004 ERRORS ;STORE ADDRESS OF REGISTER GETTING ERROR
1837 ;PARITY ERROR SET (AE ALREADY SET)
1838 ;AND NO TRAP OR TIMEOUT OCCURRED
1839 ;R1 POINTS TO THE ADDRESS OF THE
1840 ;PARITY REGISTER
1840 006222 000207 RTS 27
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 2#PRO,R1 ;LOCATE PARITY REGISTER INDICATING ERROR
1845 006234 032711 000001 1S: BIT 21,(R1)
1846 006240 001003 BNE .+10
1847 006242 005771 000000 TST 2(R1)
1848 006246 100407 BMI 2S ;BRANCH IF PARITY ERROR IS SET
1849 006250 062701 000010 ADD 210,R1
1850 006254 020127 000770 CMP R1,2TREG
1851 006260 103765 BLO 1S
1852 006262 104002 ERROR ;TRAP TO 114 OCCURRED DURING TEST 15 BUT
1853 ;NO PARITY REGISTERS HAVE PARITY ERROR SET
1854 006264 000405 BR 3S
1855 006266 011167 172476 2S: MOV 2R1,TREG ;STORE ADDRESS OF REGISTER GETTING ERROR
1856 006272 104004 ERRORS ;PARITY TRAP TO 114 OCCURRED DUE TO
1857 ;PARITY ERROR WHILE EXERCISING MEMORY

```



```

1970 006744 005767 172020      TST      TREG
1971 006750 001401              BEQ      .+4
1972 006752 104002              ERROR
1973
1974
1975 006754 036761 171540 000002  BIT      BITPT,2(R1)
1976
1977 006762 001001              BNE      .+4
1978 006764 104002              ERROR
1979
1980
1981
1982
1983 006766 010304              MOV      R3,R4
1984 006770 011167 171774              MOV      @R1,TREG
1985 006774 062701 000010      2$:     ADD      @10,R1
1986 007000 005723              TST      (R3)+
1987 007002 020127 000770      CMP      R1,@TREG
1988 007006 103750              BLO      1$
1989
1990 007010 011567 171522      MOV      (5),WAS
1991 007014 022715 125253      CMP      @125253,@R5
1992 007020 001401              BEQ      .+4
1993 007022 104003              ERRORP
1994
1995
1996
1997
1998
1999
2000 007024 005767 171740      TST      TREG
2001 007030 001002              BNE      3$
2002 007032 104002              ERROR
2003
2004
2005
2006
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
2011 007050 042701 170037      BIC      @170037,R1
2012 007054 010502              MOV      R5,R2
2013 007056 042702 003777      BIC      @3777,R2
2014 007062 000302              SWAB    R2
2015 007064 006302              ASL     R2
2016 007066 006302              ASL     R2
2017 007070 020102              CMP      R1,R2
2018 007072 001401              BEQ      .+4
2019 007074 104003              ERRORP
2020
2021
2022
2023
2024 007076 022626      4$:     CMP      (SP)+,(SP)+
2025 007100 011515      CNT16:  MOV      (5),(5)

```

```

;YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
;NO- BRANCH
;ERROR SET IN MORE THAN ONE PARITY REGISTER
;AFTER WRITING WRONG PARITY IN LOCATION
;WHOSE ADDRESS IS IN R5
;DOES MAP INDICATE THIS PARITY REGISTER
;CONTROLS THIS MEMORY?
;YES- BRANCH
;PARITY REGISTER RESPONDED TO MEMORY
;NOT INCLUDED IN ITS MAP
;PARITY REGISTER'S ADDRESS IS POINTED
;TO BY R1. ADDRESS OF LOCATION CAUSING
;PARITY ERROR IS IN R5
;STORE REGISTER ADDRESS
;BRANCH UNTIL ALL THE PARITY
;REGISTERS HAVE BEEN CHECKED
;SAVE DATA FROM LOCATION UNDER TEST
;DID BICB CHANGE DATA?
;NO. CONTINUE
;DATA WAS MODIFIED BY THE BICB WHICH
;GOT A PARITY ERROR TRAP- SINCE PARITY ERROR
;TRAP OCCURRED, CONTENTS SHOULD NOT HAVE
;BEEN MODIFIED. R5 CONTAINS ADDRESS
;OF TEST LOCATION. "TREG" CONTAINS
;ADDRESS OF PARITY REGISTER SENSING
;ERROR
;WAS PARITY ERROR SET IN ANY REGISTERS?
;YES- BRANCH
;PARITY TRAP OCCURRED ON READING
;WRONG PARITY (WITH AE SET) BUT NO
;REGISTERS HAD PARITY ERROR BIT SET.
;R5 CONTAINS THE ADDRESS OF THE
;TEST LOCATION.
;GET PARITY REGISTER CONTENTS
;MASK OFF ALL BUT ERROR ADDRESS BITS
;GET ADDRESS OF LOCATION UNDER TEST
;POSITION BITS IN R2
;PARITY ERROR ADDRESS BITS CORRECT?
;ERROR ADDRESS BITS (PARITY REGISTER
;BITS 11-5) ARE INCORRECT. R5 CONTAINS
;THE ADDRESS OF THE TEST LOCATION.
;"TREG" CONTAINS THE ADDRESS OF THE
;PARITY REGISTER DETECTING THE ERROR
;RESTORE STACK POINTER
;RESTORE TEST LOCATION TO FIX BAD PARITY

```

M03

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 38
DCMFAC.P11

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	WMP16A	
2043							
2044							
2045							
2046							
2047							
2048							
2049							
2050							
2051							
2052							
2053	007144	104001					
2054	007146	012777	000017	171726	TEST17: SCOPE	MOV	#17,@DISPLAY
2055	007154	005767	171372			TST	NOKT
2056	007160	001402				BEQ	+.6
2057	007162	000167	000636			JMP	XFR1
2058	007166	004767	004316			JSR	PC,CLPAR
2059	007172	012737	007534	000114		MOV	#TRP17,@PARVEC
2060	007200	012767	000200	171312		MOV	#200,BITPT
2061	007206	004767	004176			JSR	%7,MRALL
2062	007212	004767	004342			JSR	%7,MAP1
2063							
2064							
2065	007216	012777	001600	172022		MOV	#1600,@KPAR1
2066	007224	005067	171326			CLR	LOWFLG
2067	007230	016767	171636	171640		MOV	PMEML,PMEMX
2068	007236	012767	000002	000366		MOV	#2,INDX17
2069	007244	036767	171250	171624	1\$:	BIT	BITPT,PMEMX
2070	007252	001025				BNE	3\$
2071	007254	062777	000200	171764	2\$:	ADD	#200,@KPAR1
2072	007262	006367	171232			ASL	BITPT
2073	007266	103366				BCC	1\$
2074	007270	005767	171262			TST	LOWFLG
2075	007274	001025				BNE	DONE17
2076	007276	005267	171254			INC	LOWFLG
2077	007302	016767	171566	171566		MOV	PMEMH,PMEMX
2078	007310	012767	000001	171202		MOV	#1,BITPT
2079	007316	012767	000004	000306		MOV	#4,INDX17
2080	007324	000747				BR	1\$
2081	007326	012705	020000		3\$:	MOV	#20000,R5

```

*****
;FORCE WRONG PARITY IN EACH BYTE OF PARITY MEMORY ABOVE 28K
;WRITE WRONG PARITY AND READ IT WITH ACTION ENABLE SET, MAKING SURE
;THAT A TRAP OCCURS. THEN WRITE AND READ THE SAME LOCATION WITH GOOD PARITY
;(USING SAME DATA) TO SHOW THAT THE PARITY BIT TOGGLES. MAKE SURE THAT
;THE ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11) ARE CORRECT.
*****

```

```

TEST17: SCOPE
;LOAD THE TEST NUMBER INTO THE DISPLAY
;KT11 PRESENT?
;YES, BRANCH
;NO, SKIP TO NEXT TEST
;CLEAR ALL PARITY REGISTERS
;SETUP FOR PARITY TRAP
;INITIALIZE 4K BIT POINTER
;INITIALIZE ALL PAGES TO NON-RESIDENT
;MAP KERNEL 0 TO BANK 0, RW; KERNEL 7
;TO THE EXTERNAL BANK, RW. TURN ON
;THE KT11 AND MAKE KERNEL 1 RW
;INITIALIZE KERNEL PAGE 1 TO 28K
;CLEAR FLAG TO INDICATE TESTING LOWER 64K

```

```

;SETUP OFFSET TO CHECK LOWER 64K OF MAPS
;DOES THIS 4K HAVE PARITY?
;YES, BRANCH TO TEST IT
;NO, MAP TO NEXT 4K
;UPDATE BIT POINTER
;GO CHECK TO SEE IF THIS 4K HAS PARITY
;END OF 128K?
;YES, EXIT
;NO, SET FLAG TO INDICATE SHIFT TO
;HIGH 64K AND CHANGE MAPS

```

```

;SETUP OFFSET TO CHECK UPPER 64K OF MAPS

```

N03

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 39
DCMFAC.P11

```

2082 007332 005025          5S:   CLR      (5)+          ;CLEAR BANK UNDER TEST
2083 007334 020527 040000      CMP      R5,#40000
2084 007340 103774          BLO      5S
2085 007342 004767 000020      6S:   JSR      %7,WMP17      ;GO WRITE WRONG PARITY AND CHECK IT
2086 007346 000742          BR       2S                  ;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      WMP17:  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      WMP17A: MOV     @125253,@RS     ;INITIALIZE LOCATION
2096 007410 012701 000570      MOV     @MPRO,R1          ;INITIALIZE REGISTER ADDRESS POINTER
2097 007414 032711 000001      1S:   BIT     @1,(1)        ;DOES THIS CONTROL EXIST?
2098 007420 001003          BNE     .+10              ;NO, GET NEXT
2099 007422 012771 000005 000000  MOV     @WMP+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     1S                ;NO- LOOP
2104 007442 005767 171112      TST     ODDFLG            ;YES- TESTING HIGH BYTE?
2105 007446 100405          BMI     2S                ;YES, BRANCH
2106 007450 112715 000253      MOVB   @253,@RS          ;NO, WRITE WRONG PARITY IN LOW BYTE
2107 007454 142715 000377      BICB   @377,@RS          ;DETECT WRONG PARITY WITH DATIP-
2108          ;SHOULD TRAP TO TRP17 BEFORE DOING THE DATOB
2109 007460 000406          BR      3S
2110 007462 112765 000252 000001 2S:   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      3S:   MOV     @MPRO,R1      ;IF NO TRAP, CLEAR AE AND WMP IN ALL
2114 007502 032711 000001      4S:   BIT     @1,@R1          ;PARITY REGISTERS
2115 007506 001003          BNE     .+10
2116 007510 042771 000005 000000  BIC     @AE+WMP,@(R1)
2117 007516 062701 000010      ADD     @10,R1
2118 007522 020127 000770      CMP     R1,@TREG
2119 007526 103765          BLO     4S
2120 007530 104002          ERROR
2121          ;NO TRAP AFTER WRITING AND READING
2122          ;WRONG PARITY WITH AE SET- VIRTUAL
2123          ;ADDRESS OF LOCATION IS IN R5
2124          ;(MAPPED THRU KERNEL PAGE 1)
2125          ;WROTE LOW BYTE IF ODDFLG IS POSITIVE
2126          ;WROTE HIGH BYTE IF IT IS NEGATIVE
2127          ;NOTE THAT WMP AND AE WERE CLEARED
2128          ;BEFORE ERROR PRINTOUT
2128 007532 000512          BR      CNT17
2129
2130          ;WHEN WRONG PARITY DATA IS READ BACK, SHOULD ENTER HERE VIA TRAP TO 114
2131 007534 012701 000570      TRP17: MOV     @MPRO,R1    ;PARITY TRAP OCCURRED- BEFORE CHECKING
2132 007540 032711 000001      TRP17A: BIT     @1,@R1    ;IT, CLEAR WMP AND AE IN ALL REGISTERS
2133 007544 001003          BNE     .+10
2134 007546 042771 000005 000000  BIC     @AE+WMP,@(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	
2139	007572	012703	000772	MOV	#INDCO,R3	
2140	007576	005067	171166	CLR	TREG	
2141	007602	032711	000001	18: BIT	#1,R1	
2142	007606	001017		BNE	25	
2143	007610	005771	000000	TST	2(R1)	
2144	007614	100014		BPL	25	
2145	007616	005767	171146	TST	TREG	
2146	007622	001401		BEQ	.+4	
2147	007624	104002		ERROR		
2148						
2149						
2150	007626	036761	170666	000002	BIT	BITPT,2(R1)
2151						
2152		007632		INDX17=.	-2	
2153	007634	001001		BNE	.+4	
2154	007636	104002		ERROR		
2155						
2156						
2157						
2158						
2159						
2160	007640	011167	171124	MOV	2R1,TREG	
2161	007644	010304		MOV	R3,R4	
2162	007646	062701	000010	25: ADD	#10,R1	
2163	007652	005723		TST	(R3)+	
2164	007654	020127	000770	CMP	R1,#TREG	
2165	007660	103750		BLO	15	
2166						
2167	007662	011567	170650	MOV	(5),R5	
2168	007666	022715	125253	CMP	#125253,2R5	
2169	007672	001401		BEQ	.+4	
2170	007674	104002		ERROR		
2171						
2172						
2173						
2174						
2175						
2176						
2177	007676	005767	171066	TST	TREG	
2178	007702	001003		BNE	35	
2179	007704	104002		ERROR		
2180						
2181						
2182						
2183						
2184						
2185	007706	000423		BR	45	
2186	007710	001022		BNE	45	
2187	007712	022714	000001	35: CMP	#1,(R4)	
2188	007716	001017		BNE	45	
2189						
2190						
2191						
2192						
2193	007720	017701	171044	MOV	2TREG,R1	

```

:FIND REGISTER THAT SENSED THE ERROR
:SETUP PTR TO INDICATOR

:DOES THIS CONTROL EXIST?
:NO BRANCH
:YES- IS ERROR SET?
:NO BRANCH
:YES- WAS IT SET IN ANY OTHER REGISTER ALSO?
:NO BRANCH
:ERROR SET IN MORE THAN ONE PARITY REGISTER
:AFTER READING WRONG PARITY IN LOCATION
:WHOSE VIRTUAL ADDRESS IS IN R5
:DOES MAP INDICATE THAT THIS REGISTER
:CONTROLS THIS MEMORY?

:YES- BRANCH
:PARITY REGISTER RESPONDED TO MEMORY
:NOT INCLUDED IN ITS MAP. R1 POINTS TO
:THE PARITY REGISTER'S ADDRESS. R5
:CONTAINS VIRTUAL ADDRESS OF THE LOCATION
:BEING TESTED (MAPPED THRU KERNEL
:PAGE 1)
:STORE REGISTER ADDRESS
:STORE PTR TO INDICATOR

:INCREMENT PTR TO INDICATOR

:LOOP UNTIL ALL THE PARITY REGISTERS
:HAVE BEEN CHECKED
:SAVE CONTENTS OF LOCATION UNDER TEST
:DID BICB CHANGE DATA?
:NO CONTINUE
:DATA WAS MODIFIED BY THE BICB WHICH
:GOT A PARITY ERROR TRAP- SINCE PARITY
:TRAP OCCURRED, CONTENTS SHOULD NOT
:HAVE BEEN MODIFIED. R5 CONTAINS TEST
:LOCATION ADDRESS (VIRTUAL, MAPPED THRU
:KERNEL PAGE 1). "TREG" CONTAINS ADDRESS OF
:PARITY REGISTER DETECTING PARITY ERROR
:WAS PARITY ERROR SET IN ANY REGISTER?
:YES- BRANCH
:PARITY TRAP OCCURRED ON READING
:WRONG PARITY WITH AE SET BUT NO
:REGISTER HAS THE PARITY ERROR BIT
:SET. R5 CONTAINS VIRTUAL ADDRESS
:OF THE TEST LOCATION (MAPPED THRU
:KERNEL PAGE 1)

:NO, BRANCH. DO NOT CHECK THE
:IS THIS A CORE PARITY REGISTER?
:NO, BRANCH. DO NOT CHECK THE
:ADDRESS BITS OF THE LAST PARITY
:ERROR
:IF CORE REG. CHECK IF ADDRESS
:BITS OF LAST PAR ERR WERE STORED
:GET CONTENTS OF REGISTER DETECTING PARITY ERROR

```

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	067702	171276		ADD	#KPAR1,R2		
2201	007750	020102			CMF	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		4S:	CMF	(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	00DFLG		: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		CMF	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	1S:	JMP	IMP17A		: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,CLPAR		: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	1S:	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- BRANCH
2251 010120 062706 020000  ADD      @20000,SP      ;YES, MOVE STACK POINTER TO POINT TO BANK 1
2252 010124 012767 030222 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,ERRA4+20000
2257 010162 062767 020000 023174  ADD      @20000,ERRA5+20000
2258 010170 022767 177570 170702  CMP      @177570,SNR      ;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,SNR+20000 ;OTHERWISE THERE IS A SOFTWARE SWITCH LOCATION
2261                                ;AND ITS ADDRESS SHOULD BE UPDATED
2262 010206 000167 020010      2$:      JMP      TEST20+20010 ;GO TO TEST 20 IN BANK 1
2263
2264
2265
2266                                ;*****
2267                                ;IF FIRST 4K IS PARITY MEMORY, CHECK IT WITH A SERIES OF PATTERNS
2268                                ;THIS SUBTEST IS RUN IN BANK 1 (20000 ABOVE THE ADDRESSES IN THE LISTING)
2269                                ;*****
2270 010212 013746 177776      TEST20: MOV    @PS,-(SP)      ;THESE 2 LINES DO THE SAME AS A SCOPE WITHOUT
2271 010216 004767 005632      JSR    PC,SCOPEC      ;USING AN ENT
2272 010222 012777 000020 170652  MOV    @20,@DISPLAY   ;LOAD THE TEST NUMBER INTO THE DISPLAY
2273 010230 004767 003254      JSR    X7,CLPAR      ;CLEAR ALL PARITY REGISTERS
2274 010234 012704 021040      MOV    @PARPAT+20000,R4 ;INITIALIZE PATTERN POINTER
2275 010240 005005          CLR    R5
2276 010242 005025          1$:      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    @MPRO+20000,R1 ;SETUP TO SET ACTION ENABLE IN ALL
2281                                ;PARITY REGISTERS PRESENT
2282 010264 032711 000001      2$:      BIT    @R1
2283 010270 001003          BNE   .+10
2284 010272 012771 000001 000000  MOV    @AE,@(R1)      ;SET ACTION ENABLE IF REGISTER IS PRESENT
2285 010300 062701 000010      ADD    @10,R1
2286 010304 020127 020770      CMP    R1,@TREG+20000
2287 010310 103765          BLO   2$          ;BRANCH UNTIL ALL REGISTER ADDRESSES
2288                                ;HAVE BEEN CHECKED
2289 010312 004767 000022      3$:      JSR    X7,CKBKD      ;EXERCISE THE 5 4K
2290 010316 005724          TST   (4)+          ;UPDATE PATTERN
2291 010320 005714          TST   (4)           ;LAST PATTERN?
2292 010322 001373          BNE   3$          ;NO LOOP
2293 010324 004767 003160      DONE20: JSR   PC,CLPAR      ;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
2298
2299
2300 010340 005005
2301 010342 011415
2302 010344 021415
2303 010346 001404
2304 010350 013746 177776
2305 010354 004767 002710
2306
2307
2308 010360 005725
2309 010362 020527 000114
2310 010366 001002
2311 010370 062705 000004
2312 010374 020527 020000
2313
2314 010400 103760
2315 010402 005067 170362
2316 010406 012701 020570
2317 010412 032711 000001
2318 010416 001003
2319 010420 005771 000000
2320 010424 100406
2321 010426 062701 000010
2322 010432 020127 020770
2323 010436 103765
2324 010440 000207
2325 010442 013746 177776
2326 010446 004767 002616
2327
2328
2329
2330 010452 022626
2331 010454 000167 177644
2332
2333
2334 010460 005067 170304
2335 010464 012701 020570
2336 010470 032711 000001
2337 010474 001003
2338 010476 005771 000000
2339 010502 100412
2340 010504 062701 000010
2341 010510 020127 020770
2342 010514 103765
2343 010516 013746 177776
2344 010522 004767 002542
2345
2346 010526 000430
2347 010530 017102 000000
2348 010534 005003
2349 010536 005071 000000
2350 010542 005713
2351 010544 005771 000000
2352 010550 100413

```

```

:PARITY MEMORY TEST ROUTINE
:WRITES AND CHECKS EACH LOCATION IN BANK 0 (EXCEPT 114 AND 116)
:WITH VALUE POINTED TO BY R4
CKBK0: CLR R5
1S: MOV (4),(5)
CMP (4),(5)
BEQ +12
MOV @R5, -(SP)
JSR PC,ERR
:SET ADDRESS POINTER
:WRITE PATTERN
:DATA OK?
:YES- BRANCH
:SETUP TO DO ERROR CALL VIA JSR
:ERROR- DATA INCORRECT IN LOCATION
:WHOSE ADDRESS IS IN R5. R4 POINTS
:TO THE VALUE WRITTEN.
:UPDATE ADDRESS POINTER
:DON'T CHANGE CONTENTS OF 114 AND 116

TST (5)+
CMP R5, #114
BNE +6
ADD #4, R5
CMP R5, #20000
:HAS THE WHOLE BANK BEEN TESTED WITH
:THIS PATTERN?
:NO, BRANCH TO TEST NEXT LOCATION
:YES, DID ANY PARITY ERROR BITS SET?

BLO 1S
CLR TREG
MOV @MPRO+20000, R1
2S: BIT #1, (R1)
BNE +10
TST @R1
BHI 3S
ADD #10, R1
CMP R1, #TREG+20000
BLO 2S
RTS #7
3S: MOV @R5, -(SP)
JSR PC,ERR
:NO- RETURN
:SETUP TO DO ERROR CALL VIA JSR
:ERROR- PARITY ERROR BIT SET AND NO
:PARITY TRAP OCCURRED
:(AE WAS SET) - R1 POINTS TO ADDRESS
:OF PARITY REGISTER
:RESTORE STACK POINTER

CMP (SP)+, (SP)+
JMP DONE20

:PARITY TRAP SERVICE (NO TRAPS TO 114 SHOULD OCCUR IN THIS SUBTEST)
TRP20: CLR TREG
MOV @MPRO+20000, R1
1S: BIT #1, (R1)
BNE +10
TST @R1
BHI 2S
ADD #10, R1
CMP R1, #TREG+20000
BLO 1S
MOV @R5, -(SP)
JSR PC,ERR
:SETUP TO DO ERROR CALL VIA A JSR
:ERROR- PARITY TRAP TO 114 OCCURRED DURING
:TEST 20 BUT NO REGISTERS HAVE PARITY
:ERROR SET
:SAVE CONTENTS OF PARITY REGISTER
:INITIALIZE ADDRESS POINTER
:CLEAR PARITY REGISTER
:READ LOCATION
:PARITY ERROR SET?
:YES- BRANCH

BR 5S
2S: MOV @R1, R2
CLR R3
3S: CLR @R1
TST @R3
TST @R1
BMI 4S

```

```

2353 010552 005723          TST      (R3)+          ;MOVE POINTER
2354 010554 020327 020000    CMP      R3,#20000
2355 010556 103766          BLO     3$
2356 010559 010271 000000    MOV      R2,R1
2357 010561 013746 177776    MOV      @#PS,-(SP)
2358 010572 004767 002472    JSR     PC,ERR
                ;RESTORE PARITY REGISTER CONTENTS
                ;SETUP TO CALL ERROR VIA JSR
                ;PARITY ERROR OCCURRED WHILE TESTING
                ;BANK 0 BUT NO BAD PARITY WAS FOUND
                ;DURING SCAN OF BANK 0. R1 POINTS
                ;TO THE ADDRESS OF THE PARITY REGISTER
                ;DETECTING THE ERROR
2363 010576 000404          BR      5$
2364 010600 013746 177776    4$:     MOV      @#PS,-(SP)
2365 010604 004767 002460    JSR     PC,ERR
                ;SETUP TO DO ERROR CALL VIA JSR
                ;PARITY ERROR WHILE EXERCISING MEMORY
                ;BANK 0- R1 POINTS TO ADDRESS OF PARITY
                ;REGISTER DETECTING THE ERROR. R3 CONTAINS
                ;THE ADDRESS OF THE LOCATION HAVING
                ;BAD PARITY
2370 010610 022626          5$:     CMP      (SP)+,(SP)+
2371 010612 000207          RTS     %7
                ;RETURN TO CHECK USING THE NEXT PATTERN
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382 010614 013746 177776
2383 010620 004767 005230
2384 010624 012777 000021 170250
2385 010632 004767 002652
2386 010636 005005
2387 010640 005025
2388 010642 020527 020000
2389 010646 103774
                ;*****
                ;FORCE WRONG PARITY IN EACH LOCATION IN BANK 0
                ;NOTE THAT THIS SUBTEST IS EXECUTED IN BANK 1 (20000 ABOVE ADDRESSES
                ;IN THE LISTING). MAKE SURE THAT WRONG PARITY IN EACH BYTE CAN BE DETECTED,
                ;AND THAT WHEN GOOD PARITY IS WRITTEN AND READ NO PARITY ERROR IS DETECTED.
                ;CHECK ERROR ADDRESS BITS (PARITY REGISTER BITS 5-11)
                ;*****
2392 010614 013746 177776    TEST21: MOV      @#PS,-(SP)          ;SAVE AS SCOPE WITHOUT DOING ENT
2393 010620 004767 005230    JSR     PC,SCOPEC
2394 010624 012777 000021    MOV      #21,@DISPLAY          ;LOAD THE TEST NUMBER INTO THE DISPLAY
2395 010632 004767 002652    JSR     PC,CLRPAR              ;CLEAR ALL PARITY REGISTERS
2396 010636 005005
2397 010640 005025
2398 010642 020527 020000    1$:     CLR      (R5)+
2399 010646 103774          CMP      R5,#20000
2400          BLO     1$
                ;CLEAR PARITY ERRORS IN BANK 0
                ;WRITE WRONG PARITY TEST ROUTINE
                ;USING SAME DATA VALUE, WRITES AND CHECKS PARITY IN WRONG STATE
                ;AND THEN IN CORRECT STATE TO PROVE THAT PARITY BITS TOGGLE
2401 010650 005005
2402 010652 005067 167702
2403 010656 012767 125253 167650
2404 010664 012715 125253
2405 010670 012701 020570
                WMP21: CLR      R5          ;SET TEST ADDRESS POINTER
                CLR      000FLG        ;CLEAR FLAG TO INDICATE TESTING LOW BYTE
                MOV      #125253,@SHDBE ;STORE DATA FOR USE BY ERROR TYPEOUT ROUTINE
                WMP21A: MOV      #125253,@R5 ;INITIALIZE LOCATION
                MOV      @#PRD+20000,R1 ;SETUP TO SET WRITE WRONG PARITY
                ;IN ALL REGISTERS
2406 010674 032711 000001    1$:     BIT      #1(1)
2407 010700 001003          BNE     .+10
2408 010702 012771 000004 000000    MOV      @#WMP,R1          ;SET WRITE WRONG PARITY
                ADD      #10,R1
                CMP      R1,@TREG+20000
                BLO     1$
                TST      000FLG        ;TESTING HIGH BYTE?
                BMI     2$          ;YES, BRANCH
                MOV      #253,@R5     ;NO, WRITE WRONG PARITY IN LOW BYTE

```

409	010734	005715			TST	(R5)					: DETECT WRONG PARITY WITH DATI
410	010736	000405			BR	3S					: WRITE WRONG PARITY IN HIGH BYTE
411	010740	112765	000252	000001	2S: MOVB	8252,1(R5)					: DETECT WRONG PARITY WITH DATI
412	010746	105765	000001		TSTB	1(R5)					: SETUP TO CLEAR WMP IN ALL PARITY REGISTERS
413	010752	012701	020570		3S: MOV	8MPRO+20000,R1					: POINTER TO INDICATOR
414	010756	012703	000772		NOV	8INDCO,R3					
415	010762	032711	000001		4S: BIT	81,2R1					
416	010766	001003			BNE	.+10					
417	010770	042771	000004	000000	BIC	8WMP,2(R1)					: CLEAR WMP IN ALL PARITY REGISTERS
418	010776	062701	000010		ADD	810,R1					
419	011002	020127	020770		CMP	R1,8TREG+20000					
420	011006	103765			BLO	4S					
421	011010	012701	020570		NOV	8MPRO+20000,R1					: CHECK TO SEE IF ANY REGISTER DETECTED
422	011014	005067	167750		CLR	TREG					: THE PARITY ERROR
423	011020	032711	000001		LOOP21: BIT	81,2R1					
424	011024	001024			BNE	5S					
425	011026	005771	000000		TST	2(R1)					: CHECK PARITY REGISTER
426	011032	100021			BPL	5S					: BRANCH IF PARITY ERROR IS NOT SET
427	011034	005767	167730		TST	TREG					: WAS PARITY ERROR SET IN ANY OTHER REGISTER?
428	011040	001404			BEQ	.+12					: NO- BRANCH
429	011042	013746	177776		NOV	28PS,-(SP)					: SETUP TO DO ERROR CALL VIA A JSR
430	011046	004767	002216		JSR	PC,ERR					: ERROR- PARITY ERROR SET IN MORE THAN
431											: ONE REGISTER AFTER WRITING WRONG PARITY
432											: IN LOCATION WHOSE ADDRESS IS IN R5
433	011052	032761	000001	000002	BIT	81,2(R1)					: DOES WMP INDICATE THAT THIS REGISTER
434											: CONTROLS THIS MEMORY?
435	011060	001004			BNE	.+12					: YES, BRANCH
436	011062	013746	177776		NOV	28PS,-(SP)					: NO- SETUP TO DO ERROR CALL VIA A JSR
437	011066	004767	002176		JSR	PC,ERR					: ERROR- PARITY REGISTER RESPONDED
438											: TO MEMORY WHICH IS NOT IN ITS MAP.
439											: R1 POINTS TO PARITY REGISTER'S ADDRESS.
440											: R5 CONTAINS ADDRESS OF LOCATION BEING
441											: TESTED
442											: STORE REGISTER ADDRESS
443	011072	011167	167672		MOV	2R1,TREG					
444	011076	062701	000010		5S: ADD	810,R1					
445	011102	005723			TST	(R3)+					
446	011104	020127	020770		CMP	R1,8TREG+20000					
447	011110	103743			BLO	LOOP21					: BRANCH UNTIL ALL REGISTERS HAVE BEEN
448											: CHECKED
449	011112	005767	167652		TST	TREG					: WAS PARITY ERROR SET IN ANY REGISTER?
450	011116	001005			BNE	6S					: YES- BRANCH
451	011120	013746	177776		NOV	28PS,-(SP)					: NO- SETUP TO DO ERROR CALL VIA A JSR
452	011124	004767	002140		JSR	PC,ERR					: ERROR- NO REGISTER HAS PARITY ERROR
453											: SET AFTER READING WRONG PARITY IN LOCATION
454											: WHOSE ADDRESS IS IN R5
455	011130	000423			BR	7S					
456	011132	022713	000001		6S: CMP	81,(R3)					: IS THIS A CORE PAR REG?
457	011136	001020			BNE	7S					: NO, BRANCH(MOS PAR REG DOES NOT
458											: STORE ADDR BITS OF PAR ERROR)
459	011140	017701	167624		MOV	2TREG,R1					: GET PARITY REGISTER CONTENTS
460	011144	042701	170037		BIC	8170037,R1					: MASK ALL BUT ERROR ADDRESS BITS
461	011150	010502			MOV	R5,R2					: GET ADDRESS OF LOCATION UNDER TEST
462	011152	042702	003777		BIC	83777,R2					: POSITION BITS IN R2
463	011156	000302			SWAB	R2					
464	011160	006302			ASL	R2					
465	011162	006302			ASL	R2					

H04

```

2465 011164 000102          CMP      R1,R2          ;PARITY ERROR ADDRESS BITS CORRECT?
2466 011166 001474          BEQ      7S             ;BRANCH IF YES
2467 011170 013745 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 RS
2472 011200 011515          7S:     MOV      @RS,@RS  ;RESTORE TEST LOCATION TO FIX BAD PARITY
2473 011202 005077 167562    CLR      @TREG        ;CLEAR ERROR BIT IN PARITY REGISTER
2474 011206 005715          TST      @RS          ;READ LOCATION TO SET IF PARITY IS GOOD
2475 011210 005777 167554    TST      @TREG        ;CHECK PARITY ERROR BIT
2476 011214 100004          BPL      +12          ;BRANCH IF NOT SET
2477 011216 013746 177776    MOV      @#PS,-(SP)    ;SETUP TO DO ERROR CALL VIA A JSR
2478 011222 004767 002042    JSR     PC,ERR        ;ERROR- WRITING LOCATION WITH WRITE
2479                                     ;WRONG PARITY CLEAR DIDN'T CLEAR BAD
2480                                     ;PARITY (ADDRESS OF LOCATION IS IN RS)
2481                                     ;"TREG" +20000 CONTAINS THE ADDRESS
2482                                     ;OF THE PARITY REGISTER
2483 011226 005167 167326    COM      00DFLG       ;TOGGLE BYTE INDICATOR
2484 011232 100401          BMI      +4           ;BRANCH IF HIGH BYTE NOT YET TESTED
2485 011234 005725          TST      (RS)+        ;UPDATE ADDRESS POINTER
2486 011236 020527 020000    CMP      RS,#20000    ;THIS 4K DONE?
2487 011242 103610          BLO      WWP21A       ;LOOP TILL ALL 4K HAS BEEN TESTED
2488 011244 004767 002240    JSR     PC,CLPAR      ;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    1S:     MOV      20000(R1),@R1 ;COPY BANK 1 TO BANK 0
2496 011272 005721          TST      (R1)+
2497 011274 005300          DEC      R0
2498 011276 001373          BNE      1S
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      2S           ;IF SO THEN GO TO 2S
2507 011352 162737 020000 001100    SUB      @20000,@#SWR ;OTHERWISE RESTORE THE ADDRESS OF SOFTWARE
2508                                     ;SWITCH REGISTER
2509 011360 000137 011364    2S:     JMP      @#DONE       ;RETURN TO BANK 0
2510
2511                                     ;*****
2512                                     ;AT THIS POINT EXECUTION RETURNS TO BANK 0
2513                                     ;*****
2514
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,CLPAR     ;CLEAR ALL PARITY REGISTERS
2519 011402 005267 167136          INC      PASCNT       ;KEEP TRACK OF PASSES COMPLETED
2520 011406 004567 003502          JSR      RS,OACNV

```

```

011412 000544 PASCNT
011414 017057 MPCNT
011416 000006 6
011420 104000 TYPE ;TYPE BELL, "END PASS=" AND PASS COUNT
011422 017041 MPGEND
011424 013705 000042 LOGICAL: MOV 2042,R5 ;LOADED BY MONITOR?
011430 001405 BEQ CONT ;BRANCH IF NO
011432 000005 RESET ;SETUP FOR MONITOR EXIT
011434 004715 SENDAD: JSR 7,(5)
011436 000240 NOP
011440 000240 NOP
011442 000240 NOP
011444 032777 000400 167426 CONT: BIT 8BITB,204R ;SWITCH 8 SET?
011452 001401 BEQ .+4 ;HALT AT END OF PASS SET
011454 000000 HALT
011456 105767 167430 TSTB STPFLG
011462 001006 BNE 15 ;IF NO TERMINAL, SKIP
011464 105777 167414 TSTB 2TPS ;WAIT FOR TTY TO FINISH SO THAT RESET
011470 100375 BPL .-4 ;DON'T CLOBBER THE BELL
011472 112777 000000 167406 IS: MOVB 80,2TPB ;OUTPUT A NULL
011500 000167 170336 JMP BEGIN

;*****
;CREATE MAP INDICATING WHERE 4K BLOCKS OF MEMORY ARE PRESENT
;*****
011504 012737 011710 000004 MAPMEM: MOV 8MAPMB,204 ;SET NO MEM MANAGEMENT TRAP
011512 005737 177572 TST 20SRD ;IS KT PRESENT? (TIMEOUT IF NO)

;MAP MEMORY USING KT11 - MAX OF 124K POSSIBLE
011516 005067 167030 MAPMA: CLR NOKT ;INDICATE KT11 PRESENT
011522 004767 001662 JSR %7,NRALL ;INITIALLY SET ALL PAGES NONRESIDENT, BANK 0
011526 004767 002026 JSR %7,MAP1 ;MAP KERNEL 0 TO BANK 0, RW
;MAP KERNEL 7 TO THE EXTERNAL BANK, RW
;MAP KERNEL 1 RW, AND TURN ON KT11

011532 005067 167010 CLR TBANK
011536 012767 177777 167322 MOV 8177777,MEML ;SET UP CORE MAPS
011544 012767 077777 167316 MOV 8777777,MEMH
011552 012767 000001 166740 MOV 81,BITPT ;SET UP 4K POINTER
011560 012767 001066 166762 MOV 8MEML,MEMUT
011566 012737 011676 000004 MOV 855,204 ;SET UP FOR TIME OUTS
011574 016777 166746 167444 2S: MOV TBANK,2KPAR1 ;MAP KERNEL PAGE 1 TO BANK BEING TESTED
011602 005737 021000 TST 2021000 ;1ST K PRESENT?
011606 005737 025000 TST 2025000 ;2ND K PRESENT?
011612 005737 031000 TST 2031000 ;3RD K PRESENT
011616 005737 035000 TST 2035000 ;4TH K PRESENT?
011622 062767 000200 166716 3S: ADD 8200,TBANK ;UPDATE TEST ADDRESS
011630 006367 166664 ASL BITPT ;UPDATE BANK POINTER
011634 103006 BCC 4S ;BRANCH IF NOT DONE WITH 64K SECTION
011636 012767 000001 166654 MOV 81,BITPT ;YES, DO MEMH(64-124K)
011644 012767 001070 166676 MOV 8MEMH,MEMUT
011652 022767 007600 166666 4S: CMP 87600,TBANK ;EXTERNAL BANK YET?
011660 003345 BGT 2S ;NO, NOT YET
011662 005037 177572 CLR 20SRD ;YES- DISABLE KT11

```



```

2577 011666 012737 000006 000004      MOV      #6,284      ;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$,%284    ;SET UP TIMEOUT RETURN
2588 011726 012767 000177 167132      MOV      #177,MEML   ;INITIALIZE MAP
2589 011734 005067 167130      CLR      MEMH
2590 011740 012767 000001 166552      MOV      #1,BITPT    ;SETUP 4K POINTER
2591 011746 005001      CLR      R1          ;INITIALIZE BANK ADDRESS
2592 011750 005761 001000 1$:      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,%284
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,MEML   ;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 DAT0 AND READ TEST LOCATION VIA DAT1
2623                                ;THEN CLEAR WRITE WRONG PARITY IN ALL REGISTERS
2624 012074 012702 000570      MAPRB:  MOV      #MPRO,R2 ;LOAD ADDRESS OF TABLE
2625 012100 032712 000001 1$:      BIT      #1,(2)      ;IS THIS REGISTER PRESENT?
2626 012104 001003      BNE      .+10        ;NO, GET NEXT ONE
2627 012106 012772 000004 000000      MOV      #MWP,%2(2) ;YES, SET WRITE WRONG PARITY AND CLEAR REST
2628 012114 062702 000010      ADD      #10,R2
2629 012120 020227 000770      CMP      R2,%TREG    ;DONE WITH TABLE?
2630 012124 103765      BLO      1$          ;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      BIT      @1,(2)
2637 012152 001003      BNE      .+10
2638 012154 042772 000004 000000      BIC      @MWP,@(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 000560      MAPRC:  MOV      @MPRO-10,R2      ;INIT FOR ERROR CHECKS
2644 012200 062702 000010      1$:    ADD      @10,R2
2645 012204 020227 000770      CMP      R2,@TREG
2646 012210 002031      BGE      MAPRD
2647 012212 032712 000001      BIT      @1,(2)
2648 012216 001370      BNE      1$
2649 012220 005772 000000      TST      @(R2)
2650                                     ;BRANCH IF DONE WITH TABLE
2651 012224 100365      BPL      1$
2652 012226 005767 166322      TST      HIWORD
2653 012232 001004      BNE      2$
2654 012234 056762 166260 000002      BIS      BITPT,2(2)
2655 012242 000403      BR       3$
2656 012244 056762 166250 000004      2$:    BIS      BITPT,4(2)
2657 012252 105762 000007      3$:    TSTB     7(2)
2658                                     ;SET BIT IN MAP FOR THIS PARITY REGISTER
2659 012256 001005      BNE      4$
2660 012260 116762 166740 000006      MOVB    ADRTYP,6(2)
2661 012266 105262 000007      INCB    7(2)
2662                                     ;IS THIS THE FIRST ADDRESS FOUND FOR
2663                                     ;THIS PARITY REGISTER?
2664 012272 000742      4$:    BR       1$
2665 012274 011313      MAPRD:  MOV      @R3,@R3
2666 012276 062767 000002 166720      ADD      @2,ADRTYP
2667 012304 026727 166714 000010      CMP      ADRTYP,@10
2668 012312 001402      BEQ     1$
2669 012314 000167 177554      JMP     MAPRB
2670 012320 005767 166230      1$:    TST      HIWORD
2671 012324 001021      BNE     MAPRE
2672 012326 026727 166166 000100      CMP      BITPT,@100
2673 012334 103015      BHIS   MAPRE
2674 012336 062767 020000 166202      ADD      @20000,TBANK
2675 012344 006367 166150      ASL     BITPT
2676 012350 005067 166650      CLR     ADRTYP
2677 012354 036767 166140 166166      BIT     BITPT,MEMUT
2678 012362 001756      BEQ     1$
2679 012364 000167 177504      JMP     MAPRB
2680 012370 005767 166156      MAPRE:  TST      NOKT
2681 012374 001401      BEQ     .+4
2682 012376 000207      RTS     %7
2683 012400 005767 166616      TST     KTSTART
2684 012404 001417      BEQ     1$
2685 012406 012767 020000 166132      MOV     @20000,TBANK
2686 012414 012767 001400 166144      MOV     @1400,RELOC
2687 012422 004767 000762      JSR     %7,NRALL
2688 012426 004767 001126      JSR     %7,MAP1
                                     ;NO, INIT TBANK TO SELECT KERNEL PAGE 1
                                     ;SET UP FOR ACCESS TO 28K BANK
                                     ;INITIALLY MAP ALL PAGES NR, BANK 0
                                     ;MAP KERNEL 0 TO BANK 0, RW
                                     ;MAP KERNEL 7 TO THE EXTERNAL BANK, RW

```

```

2689                                     ;SET KERNEL 1 RW AND TURN ON KT11
2690 012432 005067 166564 CLR          KTSTART ;INDICATE KT11 NOW IN USE
2691 012436 012737 000001 177572 MOV      #1, #SR0 ;TURN ON KT11
2692 012444 006367 166050 1S: ASL      BITPT ;SHIFT BANK INDICATOR
2693 012450 103011 BCC      2S ;BRANCH IF FIRST 64K NOT DONE
2694 012452 012767 000001 166040 MOV      #1, BITPT ;IF FIRST 64K DONE, SETUP FOR
2695 012460 013767 001070 166062 MOV      #MEMH, MEMUT ;SECOND 64K
2696 012466 012767 000001 166060 MOV      #1, HIWORD ;INDICATE NOW TESTING HIGH 64K
2697 012474 062767 000200 166064 2S: ADD      #200, RELOC
2698 012502 022767 007600 166056 CMP      #7600, RELOC ;UP TO EXTERNAL BANK YET?
2699 012510 003003 BGT      3S ;NO, CONTINUE
2700 012512 005037 177572 CLR      #SR0 ;YES, TURN OFF KT11 AND EXIT
2701 012516 000207 RTS      %7
2702 012520 036767 165774 166022 3S: BIT      BITPT, MEMUT ;IS THIS 4K PRESENT?
2703 012526 001746 BEQ      1S ;NO- BRANCH
2704 012530 016777 166032 166510 MOV      RELOC, #KPAR1 ;YES, MAP PAGE 1 TO THIS BANK
2705 012536 005067 166462 CLR      ADRTYP
2706 012542 000167 177326 JMP      MAPRB ;GO TEST FOR PARITY MEMORY
2707
2708
2709
2710 ;*****
2711 ;ROUTINE TO TYPE MAP OF WHERE PARITY MEMORY IS PRESENT
2712 ;AND WHICH CONTROL REGISTERS CONTROL WHICH MEMORY
2713 ;*****
2714 012546 004767 000736 TMAP: JSR      %7, CLPAR ;CLEAR ALL PARITY REGISTERS PRESENT
2715 012552 104000 TYPE ;TYPE "THE PARITY REGISTERS CONTROL MEMORY
2716 012554 016564 MTMAP ;AS FOLLOWS:"
2717 012556 012701 000560 MOV      #MPRO-10, R1 ;SET UP POINTER
2718 012562 062701 000010 TMAPA: ADD     #10, R1
2719 012566 020127 000770 CMP      R1, #TREG ;DONE WITH MAP?
2720 012572 002136 BGE      TMAPEX ;YES, BRANCH
2721 012574 005067 165754 CLR      HIWORD
2722 012600 005067 165716 CLR      TRFLG
2723 ;INITIALIZE TRANSITION FLAG (USED TO
2724 012604 005067 165714 CLR      TYFLG ;FIGURE MEMORY LIMITS)
2725 ;INITIALIZE TO INDICATE NOTHING TYPED FOR
2726 012610 012767 177774 165710 MOV      #-4, TYCOR ;THIS REGISTER YET
2727 012616 016167 000002 165672 MOV      2(1), ADRTPT
2728 012624 012767 000001 165666 MOV      #1, BITPT
2729 012632 032711 000001 BIT      #1, (1) ;GET LOW 64K MEMORY MAP WORD
2730 012636 001351 BNE      TMAPA ;INITIALIZE 4K POINTER
2731 012640 011167 165650 MOV      (1), TEMPX ;DOES THIS CONTROL EXIST?
2732 012644 004567 002244 JSR      R5, #ACNV ;NO, GET ADDRESS OF NEXT ONE
2733 ;YES, PRINT ITS ADDRESS
2734 012650 000514 TEMPX
2735 012652 016733 MPRAD
2736 012654 000006 6
2737 012656 104000 TYPE
2738 012660 017125 MX1
2739 012662 104000 TYPE
2740 012664 016733 MPRAD
2741 012666 062767 000004 165632 TMAPB: ADD     #4, TYCOR ;KEEP TRACK OF # OF K OF CORE
2742 012674 036767 165620 165614 BIT      BITPT, ADRTPT ;DOES THIS PARITY REGISTER CONTROL THIS 4K?
2743 012702 001424 BEQ      TMAPC ;NO- BRANCH
2744 012704 005767 165612 TST      TRFLG ;YES, DOES IT CONTROL PREVIOUS 4K?
2745 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     15       ;NO, BRANCH
2776 013034 005767 165464      TST     TYFLG   ;YES, WAS ANY PARITY MEMORY
2777                                     ;FOUND FOR THIS REGISTER?
2778
2779 013040 001250      BNE     TMAPA     ;NO PARITY MEMORY WAS FOUND FOR THIS
2780 013042 104002      ERROR
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
2790 013070 000207      TMAPEX: RTS      %7   ;LOOP
2791                                     ;RETURN WHEN DONE
2792
2793                                     ;*****
2794                                     ;ERROR HANDLER
2795                                     ;*****
2796                                     ;ERRORS CALL ENTERS HERE
2797                                     ;TYPES PC, ICNT, MPR ADDRESS, AND MPR CONTENTS
2798                                     ;TREG SHOULD CONTAIN ADDRESS OF PARITY REGISTER
2799                                     ;ERRST: MOV     #MSTR,ERRB ;SETUP TO TYPE MPR ADDRESS AND CONTENTS
2800 013072 012767 016207 000266

```

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	RS,OACNV	;CONVERT TO ASCII
2805	013126	000770			TREG		
2806	013130	016216			MTREG		
2807	013132	000006			6		
2808	013134	004567	001754		JSR	RS,OACNV	;CONVERT TO ASCII
2809	013140	000540			TRDATA		
2810	013142	016240			MDATA		
2811	013144	000006			6		
2812	013146	000461			BR	ERRA	
2813							
2814							
2815							
2816							
2817							
2818							
2819							
2820							
2821							
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	RS,TSTLOC	;STORE ADDRESS BEING TESTED
2827	013176	017767	165566	165334	MOV	@TREG,TRDATA	
2828	013204	004567	001704		JSR	RS,OACNV	
2829	013210	000770			TREG		
2830	013212	016216			MTREG		
2831	013214	000006			6		
2832	013216	004567	001672		JSR	RS,OACNV	
2833	013222	000540			TRDATA		
2834	013224	016240			MDATA		
2835	013226	000006			6		
2836	013230	004567	001660		JSR	RS,OACNV	
2837	013234	000532			TSTLOC		
2838	013236	016275			MSTRX1		
2839	013240	000006			6		
2840	013242	004567	001646		JSR	RS,OACNV	
2841	013246	000534			SHDBE		
2842	013250	016311			MSTRX3		
2843	013252	000006			6		
2844	013254	004567	001634		JSR	RS,OACNV	
2845	013260	000536			WAS		
2846	013262	016325			MSTRX5		
2847	013264	000006			6		
2848	013266	000411			BR	ERRA	
2849							
2850							
2851							
2852							
2853							
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	

```

;ERRORP CALL ENTERS HERE
;TYPES PC ICNT MPR ADDRESS, MPR CONTENTS
;TEST LOCATION ADDRESS, VALUE EXPECTED, VALUE FOUND
;RS MUST CONTAIN ADDRESS OF TEST LOCATION
;SHDBE MUST CONTAIN EXPECTED VALUE
;WAS MUST CONTAIN ACTUAL DATA
;TREG MUST CONTAIN ADDRESS OF PARITY REGISTER

```

2857	013312	032777	020000	165560	ERRA:	BIT	#BIT13,JSWR		: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	RS,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	RS,OACNV		:CONVERT ICNT TO ASCII
2866	013352	016150			ERRA3:	ICNT			
2867	013354	016200			ERRA4:	MICNT			
2868	013356	000006				6			
2869	013360	004567	001476			JSR	RS,TYPSX		:TYPE MESSAGE
2870	013364	016154			ERRA5:	MED			: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	JSWR		: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									
2881									
2882									
2883	013410	013746	000004						
2884	013414	013746	000006		NRALL:	MOV	#4,-(SP)		:MAP ALL PAGES NON-RESIDENT, BANK 0
2885	013420	012737	000006	000004		MOV	#6,-(SP)		
2886	013426	012737	000002	000006		MOV	#6,#4		
2887	013434	010146				MOV	#RTI,#6		
2888	013436	010246				MOV	R1,-(SP)		
2889	013440	010346				MOV	R2,-(SP)		
2890	013442	012701	001226			MOV	R3,-(SP)		
2891	013446	012703	000040		1\$:	MOV	#PORTAB,R1		
2892	013452	012102				MOV	#32,R3		
2893	013454	005022			2\$:	(R1)+,R2			
2894	013456	005303				CLR	(R2)+		
2895	013460	001375				DEC	R3		
2896	013462	020127	001232			BNE	2\$		
2897	013466	003767				CMP	R1,#PDREND		
2898	013470	012603				BLE	1\$		
2899	013472	012602				MOV	(SP)+,R3		
2900	013474	012601				MOV	(SP)+,R2		
2901	013476	012637	000006			MOV	(SP)+,R1		
2902	013502	012637	000004			MOV	(SP)+,#6		
2903	013506	000207				MOV	(SP)+,#4		
2904						RTS	%7		
2905									
2906									
2907									
2908	013510	010146							
2909	013512	010246							
2910	013514	010701							
2911	013516	062701	165052						
2912	013522	010702							

:ROUTINE TO CLEAR ALL PARITY REGISTERS PRESENT

```
CLRPAR: MOV R1,-(SP)
        MOV R2,-(SP)
        MOV PC,R1
        ADD #MPRO--,R1
        MOV PC,R2
```

```

2913 013524 062702 165244
2914 013530 032711 000001
2915 013534 001002
2916 013536 005071 000000
2917 013543 062701 000010
2918 013546 020102
2919 013550 103767
2920 013552 012602
2921 013554 012601
2922 013556 000207
    
```

```

1S: ADD @TREG--,R2
    BIT @1,@R1 ;IS THIS REGISTER PRESENT?
    BNE +6
    CLR @R1 ;CLEAR ALL PARITY REGISTERS
    ADD @10,R1
    CMP R1,R2
    BLO 1$
    MOV (SP)+,R2
    MOV (SP)+,R1
    RTS X7
    
```

;ROUTINE TO MAP KERNEL 0 TO BANK 0 READ/WRITE.
;KERNEL 1 READ/WRITE BUT BANK MAPPED BY CALLING ROUTINE,
;AND KERNEL 7 TO EXTERNAL BANK, READ/WRITE

```

013560 012777 077406 165446
013566 012777 077406 165442
013574 012777 077406 165440
013602 012777 007600 165442
013610 005077 165430
013614 012737 000001 177572
013622 000207
    
```

```

MAP1: MOV @77406,@KPOR0
      MOV @77406,@KPOR1
      MOV @77406,@KPOR7
      MOV @7600,@KPAR7
      CLR @KPAR0
      MOV @1,@SRO
      RTS X7
    
```

;ROUTINE TO LOCATE THE FIRST PARITY MEMORY ADDRESS (ABOVE BANK 0)
;CORRESPONDING TO A GIVEN PARITY REGISTER-REQUIRES THAT THE ROUTINES
;MAPMEM AND MAPREG HAVE ALREADY BEEN RUN
;TO USE, PUT THE ADDRESS OF THE REGISTER IN R0 (I.E. POINT TO
;TO MAP TABLE)-THE DESIRED ADDRESS IS RETURNED IN R1, USING KERNEL PAGE 1 IF
;KT11 IS PRESENT

```

2945 013624 010246
2946 013626 012702 000200
2947 013632 012701 000002
2948 013636 005767 164710
2949 013642 001403
2950 013644 022701 000200
2951 013650 101420
2952 013652 030160 000002
2953 013656 001021
2954 013660 062702 000200
2955 013664 006301
2956 013666 103363
2957 013670 012701 000001
2958 013674 030160 000004
2959 013700 001010
2960 013702 062702 000200
2961 013706 006301
2962 013710 103371
2963 013712 012701 000001
2964 013716 012602
2965 013720 000207
2966 013722 005767 164624
2967 013726 001005
    
```

```

LOCATM: MOV R2,-(SP)
        MOV @200,R2
        MOV @2,R1
1S:     TST NOKT ;SKIP USE OF BANK 0
        BEQ 3$ ;KT11 PRESENT?
        CMP @200,R1 ;YES, BRANCH
        BLOS 4$ ;NO, CHECK ONLY FOR MEMORY IN FIRST 28K
        ;IF NO MEMORY FOUND IN 1ST 28K, GIVE
        ;ERROR RETURN
        ;DOES THIS 4K CORRESPOND TO THIS REGISTER?
        BIT R1,2(R0) ;YES, BRANCH
        BNE LOCAT1 ;NO, CHECK TO SEE IF NEXT 4K CORRESPONDS
        ADD @200,R2
        ASL R1
        BCC 1$
        MOV @1,R1
2S:     BIT R1,4(R0) ;CHECK HIGH 64K
        BNE LOCAT1
        ADD @200,R2
        ASL R1
        BCC 2$
        MOV @1,R1
4S:     MOV (SP)+,R2
        RTS X7
LOCAT1: TST NOKT
        BNE 1$
    
```

```

2969 013730 010277 165312      MOV      R2,2KPAR1      ;YES- SETUP R1 TO REFERENCE 4K
2970 013734 012701 020000      MOV      #20000,R1     ;BANK USING KERNEL PAGE 1
2971 013740 000407      BR       25
2972 013742 010201      1S:     MOV      R2,R1      ;SETUP R1 TO REFERENCE 4K BANK
2973 013744 006301      ASL     R1              ;WITHOUT KTI1
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 2S:     MOV      6(R0),LSAV
2980 013766 066701 000004      ADD     LSAV,R1
2981 013772 012602      MOV     (SP)+,R2      ;RESTORE R2
2982 013774 000207      RTS
2983 013776 000000      LSAV:   0              ;AND RETURN

2984
2985
2986
2987
2988 014000 013746 000004      ;SAVE LOADER IN TOP OF FIRST 4K
2989 014004 013746 000006      SAVLDR: MOV     #24,-(SP)      ;SAVE CONTENTS OF TIMEOUT VECTOR
2990 014010 012737 000006 000004      MOV     #26,-(SP)
2991 014016 012737 000002 000006      MOV     #24,#24
2992 014024 010146      MOV     #RTI,#26      ;SETUP TO RTI ON TIMEOUT
2993 014026 010246      MOV     R1,-(SP)      ;SAVE REGISTERS
2994 014030 012701 157500      MOV     R2,-(SP)
2995 014034 000261      1S:     MOV     #157500,R1
2996 014036 005711      SEC
2997 014040 103006      TST     #R1
2998 014042 162701 020000      BCC     25
2999 014046 020127 020000      SUB     #20000,R1
3000 014052 101370      CMP     R1,#20000
3001 014054 000410      BHI     15
3002 014056 012702 017500      BR      SAVLDR
3003 014062 012122 3S:     MOV     #17500,R2
3004 014064 020227 020000 3S:     MOV     (R1)+(R2)+
3005 014070 103774      CMP     R2,#20000
3006 014072 005267 000016      BLO     3S
3007      INC     LDRSVD
3008      ;INDICATE LOADER HAS BEEN MOVED TO
3009      ;THE TOP OF THE FIRST 4K
3008 014076 012602      SAVLDR: MOV     (SP)+,R2
3009 014100 012601      MOV     (SP)+,R1
3010 014102 012637 000006      MOV     (SP)+,#26
3011 014106 012637 000004      MOV     (SP)+,#24
3012 014112 000207      RTS
3013 014114 000000      LDRSVD: 0

3014
3015
3016
3017
3018      ;ROUTINE TO RESTORE THE LOADER FROM BANK 0 (WHERE IT WAS SAVED) TO THE
3019      ;HIGHEST BANK IN THE FIRST 28K OF MEMORY
3019 014116 005767 177772      RSTLDR: TST     LDRSVD
3020 014122 001431      BEQ     RSTLDR
3021 014124 012737 000006 000004      MOV     #24,#24
3022 014132 012737 000002 000006      MOV     #RTI,#26
3023 014140 012701 157500      MOV     #157500,R1
3024 014144 000261      1S:     SEC

```



```

3025 014146 005711          TST      2R1          ; 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 014154 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
3036 014204 017156          LDRMSG ;TYPE MESSAGE "LOADER RESTORED"
3037 014206 000000  RSTLDX: HALT
3038 014210 000776          BR      .-2        ;LOADER HAS BEEN RESTORED
                                     ;TO HIGHEST BANK IN FIRST 28K

                                     ;SCAN ALL MEMORY FOR BAD PARITY, TYPE 18 BIT ADDRESSES OF
                                     ;LOCATIONS FOUND TO BE BAD, AND WRITE INTO LOCATIONS WITH GOOD PARITY
PSCAN: MOV     R1,-(SP)  ;STORE REGISTERS AND LOCATIONS TO BE
        MOV     R2,-(SP)  ;ALTERED
        MOV     R3,-(SP)
        MOV     R4,-(SP)
        MOV     284,-(SP)
        MOV     286,-(SP)
        MOV     28114,-(SP)
        MOV     28116,-(SP)
        MOV     86,284    ;SETUP TIMEOUT TRAPCATCHER
        CLR     286
        MOV     8116,28114 ;SETUP PARITY TRAP TRAPCATCHER
        CLR     28116
        TST     NOKT      ;KT11 PRESENT?
        BEQ     PSCAN1    ;YES, BRANCH
        CLR     R2        ;R2 CONTAINS TEST ADDRESS
        MOV     81,R3     ;R3 USED AS A BIT POINTER
        JSR     X7,CLAPAR ;CLEAR ALL PARITY REGISTERS
        TST     2R2      ;READ LOCATION TO CHECK FOR BAD PARITY
        NOP
        MOV     8MPRO,R1 ;SETUP TO SCAN REGISTERS FOR PARITY
                                     ;ERROR SET
3044 014212 010146          3$:    BIT     81,2R1
3045 014214 010246          BNE     .+10
3046 014216 010346          TST     2(R1)
3047 014220 010446          BHI     6$
3048 014222 013746 000004  ADD     810,R1
3049 014226 013746 000006  CMP     R1,8TREG
3050 014232 013746 000114  BLO     3$
3051 014236 013746 000116  4$:    ADD     82,R2
3052 014242 012737 000006 000004 BIT     817777,R2
3053 014250 005037 000006  BNE     1$
3054 014254 012737 000116 000114 ASL     R3
3055 014258 005037 000116 000114 CMP     R3,#200
3056 014262 005767 164260  BHIS   PSCANX
3057 014272 001513          BIT     R3,MEML
3058 014274 005002          BNE     1$
3059 014276 012703 000001  ADD     82000,R2
3060 014302 004767 177202  1$:    ;EXIT IF DONE WITH 28K
3061 014306 005712          ;IS THIS MEMORY PRESENT?
3062 014310 000240          ;YES, GO TEST IT
3063 014312 012701 000570  5$:    ;NO, UPDATE ADDRESS
3064
3065 014316 032711 000001  3$:    BIT     81,2R1
3066 014322 001003          BNE     .+10
3067 014324 005771 000000  TST     2(R1)
3068 014330 100424          BHI     6$
3069 014332 062701 000010  ADD     810,R1
3070 014336 020127 000770  CMP     R1,8TREG
3071 014342 103765          BLO     3$
3072 014344 062702 000002  4$:    ADD     82,R2
3073 014350 032702 017777  BIT     817777,R2
3074 014354 001352          BNE     1$
3075 014356 006303 5$:    ASL     R3
3076 014360 020327 000200  CMP     R3,#200
3077 014364 103035          BHIS   PSCANX
3078 014366 030367 164474  BIT     R3,MEML
3079 014372 001343          BNE     1$
3080 014374 062702 020000  ADD     82000,R2

```

F05

```

3081 014400 000766          BR      55
3082 014402 010267 000110 6S:  MOV    R2,PSADRS
3083 014406 004567 000502  JSR    R5,OACNV
3084 014412 014516          PSADRS
3085 014414 017116          MPSERI
3086 014416 000006          6
3087 014420 104000          TYPE
3088 014422 017066          MPSER
3089 014424 032777 002000 164446 BIT    #2000,ASWR
3090 014432 001401          .+4
3091 014434 000000          HALT
3092 014436 011212          MOV    R2,R2
3093
3094 014440 005071 000000          CLR    R(R1)
3095 014444 005712          TST   R2
3096 014446 005771 000000          TST   R(R1)
3097 014452 100001          BPL   .+4
3098 014454 104002          ERROR
3099
3100 014456 000732          BR      45
3101 014460 004767 177024  PSCANX: JSR   PC,CLPAR
3102 014464 012637 000116  MOV    (SP)+,R16
3103 014470 012637 000114  MOV    (SP)+,R14
3104 014474 012637 000006  MOV    (SP)+,R6
3105 014500 012637 000004  MOV    (SP)+,R4
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
3111 014516 000000          X7
3112 014520 000000          ;RETURN
3113
3114
3115
3116
3117          ;SCAN ALL MEMORY FOR BAD PARITY USING KT11
3118          ;TYPE 18 BIT ADDRESSES OF LOCATIONS FOUND BAD, AND WRITE GOOD PARITY BACK IN
3119 014522 017746 164520  PSCAN1: MOV   2KPAR1,-(SP) ;SAVE CONTENTS OF KERNEL PAR1
3120 014526 032737 000001 177572 BIT    #1,ASRO ;SKIP IF KT11 IS ALREADY ON
3121 014534 001004          BNE
3122 014536 004767 176646  JSR    PC,NRALL
3123 014542 004767 177012  JSR    PC,MAP1
3124
3125 014546 005067 177746  1S:  CLR    PSCANH
3126 014552 005077 164470  CLR    2KPAR1
3127 014556 012703 000001  PSL00P: MOV   #1,R3
3128 014562 005767 177732  PSLUP: TST   PSCANH
3129 014566 001004          BNE    2S
3130 014570 030367 164272  BIT    R3,MEML
3131 014574 001022          BNE    PSXTST
3132 014600 030367 164264  2S:  BR     PSNXT
3133 014604 001016          BIT    R3,MEMH
3134 014606 062777 000200 164432  PSNXT: ADD   #200,2KPAR1
3135 014614 006303          ASL   R3
3136 014616 103361          BCC   PSLUP

```



```

3193
3194
3195
3196
3197 015062 012567 000022
3198 015066 022767 177777 000014
3199 015074 001001
3200 015076 000205
3201 015100 013746 177776
3202 015104 004767 164004
3203 015110 000000
3204 015112 000763
3205
3206
3207
3208
3209 015114 013567 000074
3210 015120 012567 000072
3211 015124 012567 000070
3212 015130 066767 000064 000060
3213 015136 016746 000052
3214 015142 042716 177770
3215 015146 062716 000060
3216 015152 005367 000040
3217 015156 112677 000034
3218 015162 042767 000007 000024
3219 015170 006067 000020
3220 015174 006067 000014
3221 015200 006067 000010
3222 015204 005367 000010
3223 015210 001352
3224 015212 000205
3225 015214 000000
3226 015216 000000
3227 015220 000000
3228
3229
3230
3231
3232 015222 104005
3233 015224 012700 015400
3234 015230 013501
3235 015232 012567 000052
3236 015236 012567 000050
3237 015242 012702 015366
3238 015246 012767 000005 000104
3239 015254 012267 000104
3240 015260 004767 000034
3241 015264 005367 000070
3242 015270 001371
3243 015272 166700 000014
3244 015276 010067 000004
3245 015302 004567 000100
3246 015306 000000
3247 015310 000000
3248 015312 000000

:PIC ROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES (CALLED VIA JSR RS)
TYP5X: MOV (RS)+,TYP5BX ;GET ADDRESS OF MESSAGE
        CMP #1,TYP5BX ;TERMINATOR?
        BNE TYP5AX ;NO BRANCH
        RTS #5 ;YES, RETURN
TYP5AX: MOV #PS,-(SP) ;SETUP TO CALL TYPE ROUTINE VIA JSR
        JSR PC,STYPE ;TYPE ASCII MESSAGE
TYP5BX: OPEN
        BR TYP5X

:SUBROUTINE FOR OCTAL TO ASCII CONVERSION
OACNV: MOV @5+,OACNVX ;GET OCTAL VALUE
        MOV (5)+,OACDST ;GET DESTINATION ADDRESS
        MOV (5)+,OACNT ;GET CONVERT COUNT
OACNVA: ADD OACNT,OACDST ;DEVELOP ADDRESS TO STORE 1ST CHAR.
        MOV OACNVX,-(SP)
        BIC #177770,@SP ;ISOLATE LEAST SIGNIFICANT DIGIT
        ADD #60,@SP ;CONVERT DIGIT TO ASCII
        DEC OACDST
        MOVB (SP)+,@OACDST ;STORE ASCII CHARACTER
        BIC #7,OACNVX
        ROR OACNVX
        ROR OACNVX
        ROR OACNVX
        DEC OACNT ;DONE ALL DIGITS?
        BNE OACNVA ;BRANCH IF NOT DONE
        RTS RS ;DONE, EXIT
OACNVX: OPEN
OACDST: 0
OACNT: 0

:SUBROUTINE FOR BINARY TO DECIMAL ASCII CONVERSION
BDCNV: SAVO4 ;SAVE REGS
        MOV #DECVAL,%0 ;SET UP ADDR TO STORE DECIMAL ASCII
        MOV @5+,R1 ;BINARY VALUE TO R1
        MOV (5)+,BDCNVC ;DESTINATION ADDR TO BDCNVC
        MOV (5)+,BDCNVD ;CHARACTER COUNT TO BDCNVD
        MOV #ADTEMP,R2 ;ADDR OF TEN POWER STRING
        MOV #5,CNVCTR ;SET UP FOR 5 POWER CONVERSIONS
BDCNVA: MOV (2)+,TENPWR ;MOVE POWER OF TEN VALUE
        JSR PC,SUBTEN ;PERFORM CONVERSION
        DEC CNVCTR ;DONE 5 CONVERSIONS?
        BNE BDCNVA ;BRANCH IF NOT YET 5.
        SUB BDCNVD,%0
        MOV %0,BDCNVB
        JSR RS,BMOVE
BDCNVB: OPEN
BDCNVC: OPEN
BDCNVD: OPEN

```

```

3249 015314 104006 RSTO4 ;RESTORE REGS AND EXIT
3250 015316 000205 RTS RS
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 UNSUCCESSFUL 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
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 RS
3283
3284
3285
3286 ;UNEXPECTED POWER FAIL SERVICE
3287 ;BECAUSE WMP 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,2824 ;SET UP FOR POWER UP
3291 015436 012701 000570 MOV #MPRO,R1
3292 015442 032711 000001 IS: BIT #1,R1
3293 015446 001002 BNE .+6
3294 015450 005071 000000 CLR 2(R1) ;CLEAR PARITY REGISTERS IN CASE
3295 015454 062701 000010 ADD #10,R1 ;WMP IS SET
3296 015460 020127 000770 CMP R1,#TREG

```

```

3297 015464 103766          BLO      15
3298 015466 010667 163562    MOV     SP,SPSAV
3299 015472 000000          HALT
3300 015474 012737 015430 000024 PWRUP:  MOV     #PWRDN, @#24      ;POWER DOWN HALT
3301 015502 016706 163546      MOV     SPSAV, SP      ;SET UP FOR POWER DOWN
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
3306 015522 017022          MPMRF
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
3318 015552 000776          BR     :-2              ;CALL IS NOT WITHIN LIMITS
3319 015554 006116          EMTA:  ROL     (SP)
3320 015556 042716 177001    BIC     #177001, (SP)  ;EMT ARG X 2
3321 015562 062716 015574    ADD     #EMTTAB, (SP)  ;REMOVE 7 MSB
3322 015566 017616 000000      MOV     @ (SP), (SP)  ;FORM EMT RTN ADDRESS
3323 015572 000136          JMP     @ (SP)+        ;GO TO EMT RETURN
3324
3325
3326          ;EMT DEFINITIONS AND ASSIGNMENTS
3327 015574          EMTTAB:
3328          TYPE=EMT+EMTX
3329 015574 001114          $TYPE
3330          SCOPE=EMT+EMTX
3331 015576 016054          SCOPEC
3332          ERROR=EMT+EMTX
3333 015600 013270          ERR
3334          ERRORP=EMT+EMTX
3335 015602 013150          ERRP
3336          ERRORS=EMT+EMTX
3337 015604 013072          ERRST
3338          SAVO4=EMT+EMTX
3339 015606 015620          SVO4
3340          RSTO4=EMT+EMTX
3341 015610 015706          RSO4
3342          RSTO5=EMT+EMTX
3343 015612 015734          RSO5
3344          SAVO5=EMT+EMTX
3345 015614 015640          SVO5
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, SVO5C

```

K05

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 62
DCMFAC.P11

```

3353 015636 000411          BR      SV05B
3354
3355
3356
3357          ;SUBROUTINE TO SAVE REGS 0-5 + PLACE EMT PC IN R5
3358 015640 012767 000240 000030 SV05S: MOV      #NOP,SV05C
3359 015646 000400          BR      SV05A
3360
3361          ;SUBROUTINE TO SAVE REGS 0-5
3362 015650 012666 177762 SV05A: MOV      (SP)+,-14.(SP)
3363 015654 012666 177762      MOV      (SP)+,-14.(SP)
3364 015660 010546          MOV      R5,-(SP)
3365 015662 010446 SV05B: 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 SV05C: CMP      -(SP),-(SP)
3371 015676 000002          RTI
3372 015700 016505 000020 SV05C: MOV      16.(SP),R5          ;RTI OR NOP
3373 015704 000002          RTI          ;EMT PC TO R5
3374
3375
3376
3377          ;SUBROUTINE TO RESTORE REGS 0-4
3378 015706 022626 RS04:  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 RS05S: 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

```

L05

```

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
3413 015776 017767 163076 000146 MOV     @SWR, RETURN
3414 016004 062767 000002 000140 ADD     @2, RETURN
3415 016012 000000          HALT
3416 016014 012767 177777 162466 MOV     @-1, TSTX
3417 016022 032777 010000 163050 BIT     @10000, @SWR
3418 016030 001404          BEQ     .+12
3419 016032 042737 000020 177776 BIC     @20, @#PS
3420 016040 000403          BR      .+10
3421 016042 052737 000020 177776 BIS     @20, @#PS
3422 016050 000177 000076          JMP     @RETURN
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
3429 016062 001015          BNE     SCOPEB
3430 016064 032777 004000 163006 BIT     @4000, @SWR
3431 016072 001016          BNE     SCOPEG
3432 016074 005767 162410          TST     TSTX
3433 016100 001006          BNE     SCOPEB
3434 016102 026767 000042 000036 CMP     ICNT, IMAX
3435 016110 100007          BPL     SCOPEG
3436 016112 005267 000032          INC     ICNT
3437 016116 022606          SCOPEB: CMP     (6)+, %6
3438 016120 012677 161652          MOV     (6)+, @PS
3439 016124 000177 000022          JMP     @RETURN
3440 016130 005067 162354          SCOPEG: CLR     TSTX
3441 016134 005067 000010          CLR     ICNT
3442 016140 011667 000006          MOV     @%6, RETURN
3443 016144 000002          RTI
3444 016146 000100          IMAX: 100
3445 016150 000000          ICNT: 0
3446 016152 000000          RETURN: 0
3447
3448
3449
3450
3451                                     ;ASCII MESSAGES
3452 016154          MED:
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 '          '

```


M05

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 64
DCMFAC.P11

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			

N05

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 65
 DCMFAC.P11

3521	016664	047515	054522	043040		
3522	016672	052517	042116	005015		
3523	016700	000				
3524	016701	116	020117	040520	MTR:	.ASCIZ 'NO PARITY REGISTER 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 .ASCII <15><12>'END PASS = '
3543	017042	005015	047105	020104		
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 USE 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	MPCOR: .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	MPMOS: .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				
3592		000001				

.EVEN
.END

H06

DCMFA.C MACY11 27(732) 10-SEP-76 09:52 PAGE 73
 DCMFAC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

RSTLDR 014116
 RSTLDX 014206
 RSTO4 = 104006
 RSTOSS= 104007
 RS04 015706
 RS05S 015734
 RO =X000000

417	3019#													
3020	3030	3037#												
3249	3281	3340#												
3342#														
3341	3378#													
3343	3391#													
381#	635	636#	637	640#	828#	830	832	866#	869	881#	883	887#		
889#	890#	893	901#	903	907	908#	919	920#	929#	930#	933	998#		
1000	1002#	1025#	1028#	1032#	1044#	1046#	1064	1089#	1090	1092#	1093	1108#		
1112#	1115	1121#	1142#	1143	1145#	1146	1166#	1171	1177#	1185	1198#	1200#		
1203	1217#	1220	1225#	1227#	1230	1245#	1248	1268#	1269	1271#	1272	1275#		
1276#	1278#	1288#	1307#	1309	1311#	1313	1324#	1327#	1348	1363	1374#	1377#		
1400#	1402	1407#	1409	1422#	1425#	1428	1433	1442#	1445#	1466#	1468	1473#		
1475	1488#	1490#	1503#	1512	1519#	1522#	1526#	1535	1544#	1546#	1569#	1570		
1573#	1574	1589#	1591#	1602#	1612	1621#	1623#	1627#	1636#	1638#	2245#	2249#		
2493#	2497#	2953	2959	2979										

ROSAV 001256
 R1 =X000001

679#														
382#	832#	841#	842	852#	853	854#	965#	966	968	969	970#	971		
1017	1026#	1029	1035	1045#	1101	1109#	1111	1113#	1114	1155	1165#	1169#		
1170	1178#	1183#	1184	1196#	1199#	1202	1216#	1218	1226#	1229	1244#	1246		
1317	1325#	1326#	1331	1337	1375#	1376#	1414	1420	1423#	1426	1443#	1481		
1489#	1492#	1504	1521#	1524#	1527	1545#	1581	1590#	1593	1603	1622#	1625		
1628	1637#	1674#	1675	1677#	1678#	1679	1707#	1710	1712#	1713	1716	1728#		
1729	1731	1733#	1734	1739	1789#	1791	1793#	1795#	1796	1826#	1829	1831#		
1832	1835	1844#	1845	1847	1849#	1850	1855	1909#	1912#	1915#	1916	1921#		
1924#	1925#	1926	1937#	1939	1941#	1942#	1943	1956#	1957	1959#	1960#	1961		
1963#	1966	1968	1975	1984	1985#	1987	2010#	2011#	2017	2096#	2099#	2101#		
2102	2113#	2114	2116#	2117#	2118	2131#	2132	2134#	2135#	2136	2138#	2141		
2143	2150	2160	2162#	2164	2193#	2194#	2201	2246#	2247#	2248	2280#	2282		
2284#	2285#	2286	2316#	2319	2321#	2322	2335#	2336	2338	2340#	2341	2347		
2349#	2351	2356#	2398#	2402#	2403#	2404	2413#	2415	2417#	2418#	2419	2421#		
2423	2425	2433	2442	2443#	2445	2458#	2459#	2465	2494#	2495#	2496	2591#		
2595#	2717#	2718#	2719	2887	2890#	2892	2896	2900#	2908	2910#	2911#	2914		
2916#	2917#	2918	2921#	2947#	2950	2953	2956#	2958#	2959	2962#	2964#	2970#		
2972#	2973#	2974#	2975#	2976#	2977#	2978#	2980#	2992	2994#	2996	2998#	2999		
3003	3009#	3023#	3025	3027#	3028	3032#	3044	3063#	3065	3067	3069#	3070		
3094#	3096	3109#	3147#	3148	3150	3152#	3153	3186#	3188	3234#	3252#	3256#		
3275#	3291#	3292	3294#	3295#	3296	3368	3380#	3394#						

RISAV 001260
 R2 =%000002

680#														
383#	762#	767#	768#	769#	794#	795#	797	839#	849	852	857	863#		
907#	909#	912	919#	921#	922	999#	1003	1167#	1169	1179#	1181#	1183		
1191#	1197#	1199	1214#	1226	1242#	1308#	1312	1345	1361	1420#	1421#	1424#		
1426	1444#	1467#	1470	1474	2012#	2013#	2014#	2015#	2016#	2017	2195#	2196#		
2197#	2198#	2199#	2200#	2201	2347#	2356	2460#	2461#	2462#	2463#	2464#	2465		
2624#	2628#	2629	2635#	2640#	2641	2643#	2644#	2645	2649	2888	2892#	2893#		
2899#	2909	2912#	2913#	2918	2920#	2945	2946#	2955#	2961#	2965#	2969	2972		
2981#	2993	3002#	3003#	3004	3008#	3031#	3032	3033	3045	3058#	3061	3072#		
3073	3080#	3082	3092#	3095	3108#	3144#	3146	3155#	3156	3159	3185#	3187		
3237#	3276#	3367	3381#	3395#										

R2SAV 001262
 R3 =%000003

681#														
384#	763#	781#	790#	796	853#	855#	857	882#	885	891	902#	905		
932	1401#	1404	1408	1964#	1983	1986	2139#	2161	2163	2348#	2350	2353		
2354	2414#	2444	2455	2631#	2632#	2664#	2889	2891#	2894#	2898#	3046	3059#		
3075#	3076	3078	3107#	3126#	3129	3132	3135#	3277#	3279#	3366	3382#	3396#		

R3SAV 001264
 R4 =%000004

682#														
385#	829#	834	867	1667#	1782#	1983#	2008	2161#	2187	2274#	3047	3106#		

R4SAV 001266	3177*	3178*	3179	3365	3383*	3397*									
RS =%000005	683#														
	386#	775*	784*	1670*	1672	1693*	1703	1785*	1787	1811*	1822	1892*	1894		
	1905*	1908*	1920*	1930*	1933*	1935*	1991	2012	2027	2039	2081*	2083	2092*		
	2095*	2106*	2107*	2110*	2111*	2168	2195	2212	2224	2275*	2277	2300*	2309		
	2311*	2312	2386*	2387*	2388	2394*	2397*	2408*	2409	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 =%000006	387#														
SAVLDR 014000	745	2988#													
SAVLDX 014076	3001	3008#													
SAV04 = 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#												
SHDBE 000534	440#	1907*	2094*	2396*	2841										
SP =%000006	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		
	2586	2604	2859	2883*	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												
SRO = 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#													
SV04 015620	3339	3350#													
SV05A 015650	3359	3362#													
SV05B 015662	3353	3365#													
SV05C 015676	3352*	3358#	3371#												
SV05S 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													
TBANK 000546	445#	2558*	2564	2569*	2574	2616*	2631	2673*	2684*						

ADC	1045	1375	1376	1443	1444										
ADD	641	795	866	890	930	970	1002	1092	1145	1271	1311	1407	1421	1473	1573
	1661	1669	1678	1712	1733	1773	1795	1831	1849	1884	1890	1915	1925	1942	1960
	1985	2071	2101	2117	2135	2162	2200	2251	2253	2254	2255	2256	2257	2260	2285
	2311	2321	2340	2403	2418	2443	2569	2596	2628	2632	2640	2644	2665	2673	2697
	2718	2740	2911	2913	2917	2955	2961	2980	3069	3072	3080	3134	3152	3155	3172
	3178	3212	3215	3256	3257	3295	3321	3414							
ASL	863	1662	1774	1885	2015	2016	2072	2198	2199	2463	2464	2570	2597	2674	2692
	2772	2956	2962	2973	2974	2975	2976	2977	2978	3075	3135	3163	3164	3165	3166
	3167	3169	3171												
BCC	864	1775	2073	2571	2693	2773	2957	2963	2997	3026	3136				
BCS	3253														
BEQ	633	773	802	810	844	858	886	913	923	976	1001	1018	1091	1102	1144
	1156	1270	1310	1318	1344	1350	1405	1415	1471	1482	1505	1528	1571	1582	1604
	1629	1698	1760	1816	1971	1992	2018	2056	2146	2169	2202	2234	2259	2303	2428
	2466	2506	2527	2534	2667	2677	2680	2683	2703	2742	2759	2775	2949	3020	3057
	3090	3183	3418												
BGE	2646	2720													
BGT	1664	1887	2575	2599	2699										
BHI	3000	3029													
BHIS	2672	3077													
BIC	766	843	855	909	921	1112	1924	1941	1959	2011	2013	2116	2134	2194	2196
	2417	2459	2461	2579	2602	2638	3160	3214	3218	3320	3419				
BICB	1930	1935	2107	2111											
BIS	771	794	968	969	1032	1276	2654	2656	3421						
BIT	772	830	849	883	903	966	975	1000	1017	1090	1101	1143	1155	1269	1309
	1317	1348	1363	1402	1414	1433	1468	1481	1570	1581	1659	1675	1708	1729	1771
	1791	1827	1845	1882	1910	1922	1939	1957	1966	1975	2069	2097	2114	2132	2141
	2150	2237	2240	2282	2317	2336	2400	2415	2423	2433	2533	2625	2636	2647	2676
	2702	2729	2741	2857	2914	2953	2959	3065	3073	3078	3089	3119	3129	3132	3148
	3182	3292	3417	3428	3430										
BLE	2897														
BLO	972	1004	1094	1147	1273	1314	1410	1476	1575	1673	1680	1704	1714	1735	1788
	1797	1823	1833	1851	1895	1917	1927	1944	1962	1988	2040	2084	2103	2119	2137
	2165	2225	2278	2287	2314	2323	2342	2355	2389	2405	2420	2446	2487	2630	2642
	2919	3005	3034	3071	3154	3157	3297								
BLOS	2951	3316													
BLT	650	798	870	895	934										
BMI	1065	1116	1204	1231	1429	1513	1613	1711	1732	1830	1848	1919	1929	2037	2105
	2222	2320	2339	2352	2407	2484	3068	3151							
BNE	638	646	709	718	744	806	831	835	850	884	897	904	906	957	967
	993	1073	1084	1124	1138	1182	1192	1215	1243	1255	1302	1346	1362	1365	1380
	1396	1403	1436	1448	1462	1469	1551	1565	1643	1660	1676	1685	1709	1730	1772
	1777	1792	1801	1828	1846	1883	1911	1923	1940	1958	1967	1977	2001	2009	2070
	2075	2098	2115	2133	2142	2153	2178	2186	2188	2238	2241	2250	2283	2292	2310
	2318	2337	2401	2416	2424	2435	2449	2456	2498	2537	2626	2637	2648	2653	2659
	2670	2730	2744	2779	2858	2895	2915	2954	2960	2968	3066	3074	3079	3120	3128
	3130	3133	3138	3149	3199	3223	3242	3280	3293	3304	3429	3431	3433		
BPL	654	899	1172	1186	1221	1249	1536	1969	2029	2144	2214	2426	2476	2539	2651
	2875	3097	3189	3435											
BR	652	721	736	782	792	837	888	918	979	1005	1047	1054	1063	1070	1095
	1122	1148	1180	1190	1213	1241	1253	1274	1279	1281	1289	1315	1336	1342	1378
	1382	1411	1446	1477	1502	1547	1576	1601	1639	1666	1687	1689	1737	1781	1803
	1806	1854	1888	1897	1932	1934	1953	2007	2080	2086	2109	2128	2185	2295	2346
	2363	2410	2454	2582	2605	2655	2663	2757	2785	2789	2812	2848	2971	3001	3030
	3038	3081	3100	3131	3141	3143	3158	3191	3204	3255	3318	3353	3359	3420	

CLR	691	694	695	697	712	719	728	729	742	761	765	767	768	769	841
	854	871	880	908	920	929	963	964	1028	1044	1046	1074	1121	1125	1165
	1166	1167	1177	1178	1179	1196	1197	1200	1216	1217	1227	1244	1245	1256	1267
	1278	1288	1374	1377	1381	1425	1442	1445	1449	1503	1522	1526	1544	1545	1546
	1552	1602	1623	1627	1636	1637	1638	1644	1654	1671	1705	1727	1767	1786	1804
	1824	1843	1893	1906	1965	2026	2066	2082	2087	2093	2140	2211	2243	2246	2275
	2276	2300	2315	2334	2348	2349	2386	2387	2394	2395	2422	2473	2494	2553	2558
	2576	2589	2591	2615	2616	2617	2619	2675	2690	2700	2705	2721	2722	2724	2760
	2893	2916	2933	3053	3055	3058	3094	3124	3125	3161	3186	3251	3294	3302	3411
	3440	3441													
CMP	739	793	797	834	857	869	885	893	905	933	971	1053	1062	1069	1093
	1146	1272	1287	1313	1345	1361	1372	1404	1409	1426	1470	1475	1504	1518	1527
	1574	1603	1620	1628	1663	1672	1679	1697	1703	1713	1734	1747	1787	1796	1815
	1822	1832	1850	1864	1886	1894	1916	1926	1943	1961	1987	1991	2008	2017	2024
	2039	2083	2102	2118	2136	2164	2168	2187	2201	2209	2224	2258	2277	2286	2302
	2309	2312	2322	2330	2341	2354	2370	2388	2404	2419	2445	2455	2465	2486	2505
	2574	2581	2586	2598	2604	2629	2641	2645	2666	2671	2698	2719	2896	2918	2950
	2999	3004	3028	3033	3070	3076	3153	3156	3198	3296	3370	3378	3392	3434	3437
CMPB	644	3315													
COM	2036	2221	2483												
DEC	1343	2249	2497	2894	3216	3222	3241	3279	3303						
DECB	649														
EMT	3328	3330	3332	3334	3336	3338	3340	3342	3344						
HALT	401	634	720	812	977	2535	2876	3037	3091	3184	3299	3317	3412	3415	
INC	710	746	791	974	1492	1524	1778	2076	2519	2756	2771	2788	3006	3139	3254
	3436														
INCB	1181	1191	1214	1242	2661										
JMP	415	417	419	421	698	711	803	811	813	1023	1107	1161	1323	1419	1486
	1587	1761	1900	2042	2057	2089	2228	2235	2239	2242	2262	2331	2509	2541	2668
	2678	2706	3307	3323	3422	3439									
JSR	643	651	713	714	745	775	784	958	959	960	973	991	994	995	1006
	1082	1085	1086	1096	1136	1139	1140	1150	1265	1300	1303	1304	1316	1394	1397
	1398	1412	1460	1463	1464	1479	1563	1566	1567	1578	1655	1681	1686	1721	1745
	1762	1763	1764	1798	1802	1860	1891	1896	1898	2058	2061	2062	2085	2244	2271
	2273	2289	2293	2305	2326	2344	2358	2365	2383	2385	2430	2437	2451	2468	2478
	2488	2490	2518	2520	2529	2554	2555	2686	2687	2714	2732	2746	2761	2804	2808
	2828	2832	2836	2840	2844	2861	2865	2869	3060	3083	3101	3121	3122	3145	3173
	3202	3240	3245												
MOV	635	636	640	647	689	690	692	693	696	705	727	730	731	732	733
	734	737	738	740	741	762	763	764	774	781	783	790	799	815	816
	817	827	828	829	832	836	838	839	842	852	853	879	881	882	887
	889	901	902	907	919	955	965	989	990	998	999	1024	1025	1026	1031
	1034	1071	1081	1089	1108	1109	1113	1135	1142	1198	1225	1264	1266	1268	1275
	1280	1299	1307	1308	1324	1325	1326	1327	1328	1329	1356	1358	1359	1373	1393
	1400	1401	1420	1422	1423	1424	1459	1466	1467	1487	1488	1489	1490	1519	1521
	1523	1549	1562	1569	1588	1589	1590	1591	1621	1622	1624	1641	1653	1656	1657
	1658	1667	1668	1670	1674	1677	1688	1693	1695	1696	1707	1716	1728	1739	1758
	1766	1768	1769	1770	1779	1780	1782	1783	1785	1789	1793	1805	1811	1813	1814
	1826	1835	1844	1855	1878	1879	1880	1881	1889	1892	1899	1905	1907	1908	1909
	1912	1921	1937	1956	1963	1964	1983	1984	1990	2010	2012	2025	2054	2059	2060
	2065	2067	2068	2077	2078	2079	2081	2088	2092	2094	2095	2096	2099	2113	2131
	2138	2139	2160	2161	2167	2193	2195	2210	2245	2247	2252	2270	2272	2274	2279
	2280	2284	2294	2301	2304	2316	2325	2335	2343	2347	2356	2357	2364	2382	2384
	2396	2397	2398	2402	2413	2414	2421	2429	2436	2442	2450	2458	2460	2467	2472
	2477	2489	2493	2495	2516	2517	2526	2549	2559	2560	2561	2562	2563	2564	2572
	2573	2577	2585	2587	2588	2590	2600	2612	2613	2614	2624	2627	2631	2633	2635

C07

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.SEO/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)

