

RH70/RS03

BASIC FUNCTIONAL DIAG
MD-11-DERSA-A

EP-DERSA-A-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN US



B01

-----DERSA-A

.REM.

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DERSA-A-D
PRODUCT NAME:	RS11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
DATE CREATED:	MARCH-1975
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	STANLEY HARACKIEWICZ

MAINDEC-11-DERSA-A-D
RS11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
MARCH-1975
DIAGNOSTIC GROUP
STANLEY HARACKIEWICZ

73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT COPR., MAYNARD, MASS.

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.3 PRELIMINARY PROGRAMS
- 3. LOADING PROCEDURE
- 4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND/OR OPERATING PROCEDURE
- 5. OPERATIONAL SWITCH SETTINGS
 - 5.2 SUBROUTINE ABSTRACT
- 6. ERRORS
- 7. RESTRICTIONS
- 8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
- 9. WRITE LOCK TEST
- 10. TEST DESCRIPTION

74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 4
DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03 AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03 OR RS04) DISKS ARE OPERATING CORRECTLY. THIS IS NOT A RELIABILITY DIAGNOSTIC AND THEREFORE SHOULD NOT BE USED AS ONE. THIS PROGRAM CAN TEST UP TO 8 DRIVES. THE DRIVES CAN BE INTERMIXED AND IN ANY ORDER.

IF THE OPERATOR WOULD LIKE TO CHECK THE DISK REGISTERS PRIOR TO ENTERING THIS DIAGNOSTIC, THERE ARE SOME ROUTINES IN THE BACK OF THE DIAGNOSTIC WHICH CAN BE USED. THESE ROUTINES WILL ALLOW THE OPERATOR TO LOAD THE REGISTERS THROUGH THE SWITCHES. PLEASE REFERENCE THE STARTING ADDRESSES THAT WILL TEST THE REGISTERS YOU DESIRE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OF 8K OF MEMORY, AND AN RH11 CONTROLLER WITH A RS03 OR RS04 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

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
164
165
166
167
168
169
170
171
172
173
174

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 5
DESCRIPTION

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

1. STARTING ADDRESS 200.

- A. SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
- B. PRESS START.
- C. THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
- D. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
- E. THE PROGRAM WILL TEST ALL RS03 AND RS04 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/04 REGISTERS USING THE SWITCH REGISTER.

A.	250	WORD COUNT REGISTER TEST
B.	254	BUS ADDRESS REG. TEST
C.	260	DISK ADDRESS REG. TEST
D.	264	DRIVE STATUS REG. TEST
E.	270	ERROR REG. TEST
F.	274	LOOK AHEAD REG. TEST
G.	300	RSCS2 REG. TEST
H.	304	ATTENTION SUMMARY REG. TEST
I.	310	MAINTENANCE REG. TEST
J.	314	RSCS1 REG TEST

5. OPERATIONAL SWITCH SETTINGS

SWITCH SETTINGS ARE:

- SW<15> = 1 HALT ON ERROR
- SW<14> = 1 LOOP ON TEST
- SW<13> = 1 INHIBIT TYPEOUTS
- SW<11> = 1 INHIBIT ITERATIONS OF SUBTEST
- SW<10> = 1 BELL ON ERROR
- SW<10> = 0 BELL ON PASS COMPLETE
- SW<09> = 1 LOOP ON ERROR
- SW<08> = 1 LOOP ON TEST IN SW<7:0>

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION

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
220
221
222
223
224
225
226
227
228
229
230

GO1

MAINDEC-11-DERSA-A
DERSAA.P11

RS11-REC3-RSD4 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 7

231
232

SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS
BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 6
DESCRIPTION

CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR CS1 = ----- CS2 = ----- ER = -----
GOOD = ----- BAD = -----

WHERE:

CS1, CS2, ER ETC. = RS11 DISK REGISTERS.
GOOD = EXPECTED DATA.
BAD = DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

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
276
277
278
279
280
281
282
283
284
285
286
287

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 7
DESCRIPTION

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 1 MINUTE WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500

9. WRITE LOCK TEST

THE WRITE LOCK TEST REQUIRES OPERATOR INTERVENTION. THE STARTING ADDRESS FOR THIS TEST IS 220. THE PROGRAM WILL TELL THE OPERATOR WHICH SWITCHES HAVE TO BE SET.

10. TEST DESCRIPTION

1. TEST RSCS2

CLEAR ALL READ/WRITE BITS AND CHECK. SET ALL R/W BITS AND CHECK. NOW CLEAR AND RECHECK.

2. TEST FOR ONLINE DRIVES

SET ERROR BITS IN RSER. THIS CAUSES ATTENTION SUMMARY BITS TO SET IN RSAS. DO FOR ALL DRIVES. RSAS HAS NOT YET BEEN TESTED. SO IN THE CASE OF NO BITS IN RSAS SETTING, DRIVE 0 IS TESTED.

3. RESET TEST FOR REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSOB, AND RSMR. DO A RESET AND TEST ALL R/W BITS TO BE CLEARED.

4. SET AND CLEAR ALL REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSOB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.

5. RANDOM NUMBER TEST FOR RSWC AND RSDA

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
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343

344
345

THIS TEST GENERATES RANDOM NUMBERS AND LOADS THEM INTO RSWC,

21-201

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 8
DESCRIPTION

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
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401

RSDA AND RSBA.

6. TEST "CLEAR BIT" IN RSCS2

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. SET CLEAR BIT IN RSCS2. NOW TEST ALL R/W BITS FOR 0 IN ALL THE ABOVE REGISTERS.

7. LOAD RSDB WITH ALL ONES AND ALL ZEROS

LOAD RSDB WITH A WORD OF ZEROS AND A WORD OF ONES. WAIT FOR "OR" TO SET AND THEN CHECK OUTPUT OF SILO. IF OR DID NOT SET ERROR MESSAGE APPEARS.

8. TEST NO-OP FUNCTION

THE NO-OP FUNCTION IS TESTED WITH AND WITHOUT ERROR BITS SET. ALL THE REGISTERS ARE CHECKED AFTER BOTH CASES.

9. TEST DRIVE CLEAR FUNCTION

FIRST SET ALL R/W BITS IN RSDA, RSWC, RSER, AND RSMR. DO A DRIVE CLEAR FUNCTION. NOW TEST ALL REGISTERS FOR CORRECT DATA.

10. EXECUTE A ONE WORD WRITE FUNCTION

SET RSWC TO -1. MOV -1 INTO OUTBUF. LOAD RSBA WITH OUTBUF. DO A WRITE TEST RDY BIT FOR 0 THEN WAIT FOR IT TO SET. TIME OUT TO ERROR IF RDY BIT DOESN'T SET AND CHECK FOR ERROR CONDITIONS. TEST RSDA FOR CORRECT ADDRESS. TEST WORD COUNT FOR 0.

11. EXECUTE A ONE WORD WRITE CHECK

SET UP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION TEST. DO A WRITE CHECK FUNCTION. TEST RDY AS DONE IN THE WRITE TEST. CHECK FOR WRITE CHECK ERROR. THEN TEST RSDA, RSWC AND RSBA FOR CORRECT DATA.

12. TEST READ FUNCTION

SETUP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION DO A READ FUNCTION. TEST RDY BIT AS DONE IN THE WRITE FUNCTION. TEST FOR ERRORS. ALSO TEST RSDA, RSWC AND RSBA FOR CORRECT DATA.

13. TEST BLOCK SEARCH FUNCTION, PIP AND DRY BITS AND ADDR. CONF. BIT

DO A BLOCK SEARCH FOR SECTOR 32, LOOP ON ADDR. CONF. BIT IN RSMR. IF IT DOESN'T SET, TIMEOUT. WHEN YOU GET THERE DO A

L01

MAINDEC-11-DERSA-A
DERSAA.P11

RS11-R503-R504 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 12

BLOCK SEARCH FOR SECTOR 0. NOW WE KNOW THAT WE HAVE TIME TO
TEST FOR DRY AND PIP BITS BEFORE FINDING SECTOR 0. FOR PIP

402
403

404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459

SHOULD SET AND DRY SHOULD CLEAR BEFORE FINDING SECTOR 0.
ONCE SECTOR 0 IS FOUND PIP SHOULD CLEAR AND DRY SHOULD SET.
IF DRY DOES NOT SET A TIME OUT ERROR WILL OCCUR INDICATING
SECTOR 0 WAS NOT FOUND. SC IN RSCS1 SHOULD ALSO SET. RSBA
AND RSWC SHOULD NOT MOVE, THIS IS ALSO TESTED.

14. ILLEGAL FUNCTION CODE TEST

IN THIS TEST RSBA, RSWC AND RSDA ARE SET UP AS IF TO DO A
LEGAL FUNCTION. AN ILLEGAL FUNCTION IS THEN EXECUTED. THE
PROGRAM TEST FOR ILF AND ERR BITS TO SET. RSBA, RSWC AND
RSDA ARE ALSO TESTED FOR CORRECT DATA. THIS IS DONE FOR ALL
THE ILLEGAL FUNCTIONS.

FOR AN AID IN TROUBLE SHOOTING THE ILLEGAL FUNCTION CODE CAN
BE LOADED INTO LOCATION ILLTAB OR ILFTB2, DEPENDING ON WHICH
ILLEGAL FUNCTION TEST YOU WISH TO LOOP ON. IN THE NEXT
LOCATION, FOLLOWING THE ILLEGAL FUNCTION, A 0 MUST BE LOADED.
NOW BY SETTING SWITCH 14 (LOOP ON TEST), YOU WILL LOOP ON THE
ILLEGAL FUNCTION.

15. TEST PAR IN RSER

SET PAR IN RSER AND CHECK. ALSO TEST ERR IN RSDS TO SET
BECAUSE OF THE PAR SETTING.

16. TEST DPR AND MOL IN RSDS

BOTH THESE BITS SHOULD BE SET IN RSDS IF THE DRIVE IS ON LINE
AND UP TO SPEED.

17. LOOK AHEAD TEST

FIRST CHECK TO SEE IF SECTOR FRACTION BITS ARE MOVING. NOW
SET RSDA TO 0 AND INCREMENT IT EVERY TIME THE ADDR.CONF BIT
SETS. IF THE ADDR.CONF BIT DOES NOT SET IN A CERTAIN LENGTH
OF TIME, A TIME OUT ERROR OCCURS.

18. PARITY TEST

THIS WILL TEST THE PARITY LOGIC ONLY IF THERE IS PARITY
MEMORY ON THE SYSTEM IN LESS THAN 2BK. IT WILL WRITE BAD
PARITY IN A MEMORY LOCATION THEN TRY TO DO A WRITE TO THE
DRIVE FROM THAT LOCATION. THIS SHOULD CAUSE A PARITY ERROR.

19. TEST WRITE CHECK ERROR

IN THIS TEST THE PROGRAM WRITES A -1 ON TO THE DISK. A 0 IS
NOW FLOATED THROUGH THE WORD IN THE BUS ADDRESS LOCATION, AND
A WRITE CHECK FUNCTION IS DONE. THE WCE BIT IN RSCS2 SHOULD
SET AND SHOULD CAUSE THE TRE BIT IN RSCS1 TO SET. THESE BITS
ARE THEN CLEARED. A WORD OF 0 IS NOW WRITTEN ON THE DISK AND

NO1

MAINDEC-11-DERSA-A
DERSAA.P11

RS11-RSC3-RS04 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 14

A 1 IS FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE
WRITE CHECK FUNCTION TEST IS REPEATED.

460
461

MAINDEC-11-DERSA-A
MAY11 27(732) 06-OCT-76 08:51

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 10
DESCRIPTION

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17

20. TEST PROGRAM ERROR BIT IN R5C52

HERE THE PROGRAM ATTEMPTS TO INITIATE A DATA TRANSFER OPERATION WHILE THE CONTROL IS CURRENTLY PERFORMING ONE. THIS SHOULD CAUSE PGE TO SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED. R5WC IS ALSO TESTED FOR IT SHOULD NOT BE 0 FOR THE CURRENT OPERATION SHOULD HAVE BEEN ABORTED DUE TO THE PGE ERROR.

21. TEST RMR IN R5ER

HERE A WRITE COMMAND IS GIVEN AND DURING ITS EXECUTION THE PROGRAM TRYS TO MODIFY THE R5DA REG. THIS SHOULD CAUSE THE RMR BIT TO SET WHICH CAUSES THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

22. TEST DCK IN R5ER

HERE A WRITE COMMAND IS GIVEN THEN DURING THIS FUNCTION A DRIVE CLEAR COMMAND IS GIVEN. THIS SHOULD CAUSE THE DCK BIT TO SET WHICH SHOULD CAUSE THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

23. TEST DISK ADDRESS REGISTER

LOAD THE LAST DISK ADDRESS (7777) INTO R5DA. DO A ONE WORD WRITE AND CHECK THAT R5DA INCREMENTED TO 10000.

24. TEST IAE ERROR

DO A ONE WORD WRITE BUT FIRST SET R5DA TO AN INVALID ADDRESS SUCH AS 10000. THIS SHOULD CAUSE A IAE ERROR WHICH WILL CAUSE ERR, ATA AND SC BITS TO SET. THESE BITS ARE THEN CLEARED BY LOADING A 1 INTO ATA IN R5AS.

25. TEST FOR NON-EXISTENT DISK ERROR

FIRST FIND A DRIVE THAT IS NOT ON THE SYSTEM OR OFF LINE. NOW TRY TO DO A ONE WORD WRITE TO THAT DRIVE. NED IN R5C52 SHOULD SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED BY MOVING A 1 INTO TRE.

26. TEST DAO IN R5ER AND LBT IN R5DS

SET R5DA TO ITS LAST ADDRESS. NOW WRITE ONE SECTOR PLUS ONE WORD. DAO SHOULD SET AND LBT SHOULD SET. THESE SHOULD CAUSE ERR, ATA, TRE AND SC TO SET. THESE ARE CLEARED BY DOING A CLEAR.

27. TEST BAI IN R5C52

SET BAI IN R5C52. DO A ONE WORD WRITE AND CHECK R5BA TO SEE

C02

MAINDEC-11-DERSA-A
DERSAA.P11

RS:1-RSC3-R504 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 16

518

IF IT INCREMENTED.

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 11
DESCRIPTION

28. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS 0 THROUGH 5 IN RHBAE FOR AN 22 BIT ADDRESS. MOV 173000 INTO RSBA. MOV -1000 INTO RSMC AND DO A WRITE FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.

29. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR

FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.

30. PRIORITY INTERRUPT TEST

HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE PROCESSOR IS DROPPED TO PRIORITY 4.

31. DYNAMIC FUNCTION TEST

WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED. IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS STILL READING.

%
.REM%

IDENTIFICATION

519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574

E02

MAINDEC-11-DERSA-A
DERSAA.P11

RS.1-RS03-RS04 BASIC FUNCTION DIAGNOSTIC

MAY11 27(732) 06-OCT-76 08:51 PAGE 18

5
5
5
5
5
5
5
5
5
5
5
5
5
5
5

PRODUCT CODE:	MAINDEC-11-DERSA-A-D
PRODUCT NAME:	RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
DATE CREATED:	MARCH-1975
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	STANLEY HARACKIEWICZ

595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT COPR., MAYNARD, MASS.

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 3
TABLE OF CONTENTS

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATING PROCEDURE
5.	OPERATIONAL SWITCH SETTINGS
5.2	SUBROUTINE ABSTRACT
6.	ERRORS
7.	RESTRICTIONS
8.	MISCELLANEOUS
8.1	EXECUTION TIME
8.2	STACK POINTER
9.	WRITE LOCK TEST
10.	TEST DESCRIPTION

622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 4
DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST R503 AND R504 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (R503 OR R504) DISKS ARE OPERATING CORRECTLY. THIS IS NOT A RELIABILITY DIAGNOSTIC AND THEREFORE SHOULD NOT BE USED AS ONE. THIS PROGRAM CAN TEST UP TO 8 DRIVES. THE DRIVES CAN BE INTERMIXED AND IN ANY ORDER.

IF THE OPERATOR WOULD LIKE TO CHECK THE DISK REGISTERS PRIOR TO ENTERING THIS DIAGNOSTIC, THERE ARE SOME ROUTINES IN THE BACK OF THE DIAGNOSTIC WHICH CAN BE USED. THESE ROUTINES WILL ALLOW THE OPERATOR TO LOAD THE REGISTERS THROUGH THE SWITCHES. PLEASE REFERENCE THE STARTING ADDRESSES THAT WILL TEST THE REGISTERS YOU DESIRE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM 64K BK OF MEMORY, AND AN RH11 CONTROLLER WITH A R503 OR R504 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 5
DESCRIPTION

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

1. STARTING ADDRESS 200.

- A. SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
- B. PRESS START.
- C. THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
- D. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
- E. THE PROGRAM WILL TEST ALL R503 AND R504 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-R503/04 REGISTERS USING THE SWITCH REGISTER.

- A. 250 WORD COUNT REGISTER TEST
- B. 254 BUS ADDRESS REG. TEST
- C. 260 DISK ADDRESS REG. TEST
- D. 264 DRIVE STATUS REG. TEST
- E. 270 ERROR REG. TEST
- F. 274 LOOK AHEAD REG. TEST
- G. 300 RSCS2 REG. TEST
- H. 304 ATTENTION SUMMARY REG. TEST
- I. 310 MAINTENANCE REG. TEST
- J. 314 RSCS1 REG TEST

5. OPERATIONAL SWITCH SETTINGS

SWITCH SETTINGS ARE:

- SW<15> = 1 HALT ON ERROR
- SW<14> = 1 LOOP ON TEST
- SW<13> = 1 INHIBIT TYPEOUTS
- SW<11> = 1 INHIBIT ITERATIONS OF SUBTEST
- SW<10> = 1 BELL ON ERROR
- 0 BELL ON PASS COMPLETE
- SW<09> = 1 LOOP ON ERROR
- SW<08> = 1 LOOP ON TEST IN SW<7:0>

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION

723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778

779
780

SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS
BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 6
DESCRIPTION

CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR CS1 = ----- CS2 = ----- ER = -----
GOOD = ----- BAD = -----

WHERE:

CS1, CS2, ER ETC. = RS11 DISK REGISTERS.
GOOD = EXPECTED DATA.
BAD = DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 7
DESCRIPTION

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 1 MINUTE WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500

9. WRITE LOCK TEST

THE WRITE LOCK TEST REQUIRES OPERATOR INTERVENTION. THE STARTING ADDRESS FOR THIS TEST IS 220. THE PROGRAM WILL TELL THE OPERATOR WHICH SWITCHES HAVE TO BE SET.

10. TEST DESCRIPTION

1. TEST RSCS2

CLEAR ALL READ/WRITE BITS AND CHECK. SET ALL R/W BITS AND CHECK. NOW CLEAR AND RECHECK.

2. TEST FOR ONLINE DRIVES

SET ERRJR BITS IN RSER. THIS CAUSES ATTENTION SUMMARY BITS TO SET IN RSAS. DO FOR ALL DRIVES. RSAS HAS NOT YET BEEN TESTED. SO IN THE CASE OF NO BITS IN RSAS SETTING, DRIVE 0 IS TESTED.

3. RESET TEST FOR REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. DO A 'RESET' AND TEST ALL R/W BITS TO BE CLEARED.

4. SET AND CLEAR ALL REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.

5. RANDOM NUMBER TEST FOR RSWC AND RSDA

836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891

892
893

THIS TEST GENERATES RANDOM NUMBERS AND LOADS THEM INTO RSWC,

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 8
DESCRIPTION

894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949

- RSDA AND RSBA.
6. TEST "CLEAR BIT" IN RSC52
SET ALL R/W BITS IN RSC51, RSC52, RSBA, RSDB, RSER, RSWC, RSDB, AND RSMR. SET CLEAR BIT IN RSC52. NOW TEST ALL R/W BITS FOR 0 IN ALL THE ABOVE REGISTERS.
 7. LOAD RSDB WITH ALL ONES AND ALL ZEROS
LOAD RSDB WITH A WORD OF ZEROS AND A WORD OF ONES. WAIT FOR "OR" TO SET AND THEN CHECK OUTPUT OF SILO. IF OR DID NOT SET ERROR MESSAGE APPEARS.
 8. TEST NO-OP FUNCTION
THE NO-OP FUNCTION IS TESTED WITH AND WITHOUT ERROR BITS SET. ALL THE REGISTERS ARE CHECKED AFTER BOTH CASES.
 9. TEST DRIVE CLEAR FUNCTION
FIRST SET ALL R/W BITS IN RSDB, RSWC, RSER, AND RSMR. DO A DRIVE CLEAR FUNCTION. NOW TEST ALL REGISTERS FOR CORRECT DATA.
 10. EXECUTE A ONE WORD WRITE FUNCTION
SET RSWC TO -1. MOV -1 INTO OUTBUF. LOAD RSBA WITH OUTBUF. DO A WRITE TEST RDY BIT FOR 0 THEN WAIT FOR IT TO SET. TIME OUT TO ERROR IF RDY BIT DOESN'T SET AND CHECK FOR ERROR CONDITIONS. TEST RSDB FOR CORRECT ADDRESS. TEST WORD COUNT FOR 0.
 11. EXECUTE A ONE WORD WRITE CHECK
SET UP RSDB, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION TEST. DO A WRITE CHECK FUNCTION. TEST RDY AS DONE IN THE WRITE TEST. CHECK FOR WRITE CHECK ERROR. THEN TEST RSDB, RSWC AND RSBA FOR CORRECT DATA.
 12. TEST READ FUNCTION
SETUP RSDB, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION DO A READ FUNCTION. TEST RDY BIT AS DONE IN THE WRITE FUNCTION. TEST FOR ERRORS. ALSO TEST RSDB, RSWC AND RSBA FOR CORRECT DATA.
 13. TEST BLOCK SEARCH FUNCTION, PIP AND DRY BITS AND ADDR. CONF. BIT
DO A BLOCK SEARCH FOR SECTOR 32, LOOP ON ADDR. CONF. BIT IN RSMR. IF IT DOESN'T SET, TIMEOUT. WHEN YOU GET THERE DO A

B03

MA:NDCC-11-DERSA-A
DERSA.P:1

RS:1-RSC3-RSD4 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 28

950
951

BLOCK SEARCH FOR SECTOR D. NOW WE KNOW THAT WE HAVE TIME TO
TEST FOR DRY AND PIP BITS BEFORE FINDING SECTOR C. FOR PIP

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 9
DESCRIPTION

SHOULD SET AND DRY SHOULD CLEAR BEFORE FINDING SECTOR 0.
ONCE SECTOR 0 IS FOUND PIP SHOULD CLEAR AND DRY SHOULD SET.
IF DRY DOES NOT SET A TIME OUT ERROR WILL OCCUR INDICATING
SECTOR 0 WAS NOT FOUND. SC IN RSCS1 SHOULD ALSO SET. RSBA
AND RSWC SHOULD NOT MOVE, THIS IS ALSO TESTED.

14. ILLEGAL FUNCTION CODE TEST

IN THIS TEST RSBA, RSWC AND RSDA ARE SET UP AS IF TO DO A
LEGAL FUNCTION. AN ILLEGAL FUNCTION IS THEN EXECUTED. THE
PROGRAM TEST FOR ILF AND ERR BITS TO SET. RSBA, RSWC AND
RSDA ARE ALSO TESTED FOR CORRECT DATA. THIS IS DONE FOR ALL
THE ILLEGAL FUNCTIONS.

FOR AN AID IN TROUBLE SHOOTING THE ILLEGAL FUNCTION CODE CAN
BE LOADED INTO LOCATION ILLTAB OR ILFTB2, DEPENDING ON WHICH
ILLEGAL FUNCTION TEST YOU WISH TO LOOP ON. IN THE NEXT
LOCATION, FOLLOWING THE ILLEGAL FUNCTION, A 0 MUST BE LOADED.
NOW BY SETTING SWITCH 14 (LOOP ON TEST), YOU WILL LOOP ON THE
ILLEGAL FUNCTION.

15. TEST PAR IN RSER

SET PAR IN RSER AND CHECK. ALSO TEST ERR IN RSDS TO SET
BECAUSE OF THE PAR SETTING.

16. TEST DPR AND MOL IN RSDS

BOTH THESE BITS SHOULD BE SET IN RSDS IF THE DRIVE IS ON LINE
AND UP TO SPEED.

17. LOOK AHEAD TEST

FIRST CHECK TO SEE IF SECTOR FRACTION BITS ARE MOVING. NOW
SET RSDA TO 0 AND INCREMENT IT EVERY TIME THE ADDR.CONF BIT
SETS. IF THE ADDR.CONF BIT DOES NOT SET IN A CERTAIN LENGTH
OF TIME, A TIME OUT ERROR OCCURS.

18. PARITY TEST

THIS WILL TEST THE PARITY LOGIC ONLY IF THERE IS PARITY
MEMORY ON THE SYSTEM IN LESS THAN 28K. IT WILL WRITE BAD
PARITY IN A MEMORY LOCATION THEN TRY TO DO A WRITE TO THE
DRIVE FROM THAT LOCATION. THIS SHOULD CAUSE A PARITY ERROR.

19. TEST WRITE CHECK ERROR

IN THIS TEST THE PROGRAM WRITES A -1 ON TO THE DISK. A 0 IS
NOW FLOATED THROUGH THE WORD IN THE BUS ADDRESS LOCATION, AND
A WRITE CHECK FUNCTION IS DONE. THE WCE BIT IN RSCS2 SHOULD
SET AND SHOULD CAUSE THE TRE BIT IN RSCS1 TO SET. THESE BITS
ARE THEN CLEARED. A WORD OF 0 IS NOW WRITTEN ON THE DISK AND

99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

:008
:009

N : IS FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE
WRITE CHECK FUNCTION TEST IS REPEATED.

MAINDEC-11-DERSA-A RH11-RSD3-RSD4 BASIC FUNCTION DIAGNOSTIC PAGE 10
DESCRIPTION

1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065

20. TEST PROGRAM ERROR BIT IN RSCS2

HERE THE PROGRAM ATTEMPTS TO INITIATE A DATA TRANSFER OPERATION WHILE THE CONTROL IS CURRENTLY PERFORMING ONE. THIS SHOULD CAUSE PGE TO SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED. RSWC IS ALSO TESTED FOR IT SHOULD NOT BE 0 FOR THE CURRENT OPERATION SHOULD HAVE BEEN ABORTED DUE TO THE PGE ERROR.

21. TEST RMR IN RSER

HERE A WRITE COMMAND IS GIVEN AND DURING ITS EXECUTION THE PROGRAM TRYS TO MODIFY THE RSDA REG. THIS SHOULD CAUSE THE RMR BIT TO SET WHICH CAUSES THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

22. TEST DCK IN RSER

HERE A WRITE COMMAND IS GIVEN THEN DURING THIS FUNCTION A DRIVE CLEAR COMMAND IS GIVEN. THIS SHOULD CAUSE THE DCK BIT TO SET WHICH SHOULD CAUSE THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

23. TEST DISK ADDRESS REGISTER

LOAD THE LAST DISK ADDRESS (7777) INTO RSDA. DO A ONE WORD WRITE AND CHECK THAT RSDA INCREMENTED TO 10000.

24. TEST IAE ERROR

DO A ONE WORD WRITE BUT FIRST SET RSDA TO AN INVALID ADDRESS SUCH AS 10000. THIS SHOULD CAUSE A IAE ERROR WHICH WILL CAUSE ERR, ATA AND SC BITS TO SET. THESE BITS ARE THEN CLEARED BY LOADING A 1 INTO ATA IN RSAS.

25. TEST FOR NON-EXISTENT DISK ERROR

FIRST FIND A DRIVE THAT IS NOT ON THE SYSTEM OR OFF LINE. NOW TRY TO DO A ONE WORD WRITE TO THAT DRIVE. NED IN RSCS2 SHOULD SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED BY MOVING A 1 INTO TRE.

26. TEST DAO IN RSER AND LBT IN RSDS

SET RSDA TO ITS LAST ADDRESS. NOW WRITE ONE SECTOR PLUS ONE WORD. DAO SHOULD SET AND LBT SHOULD SET. THESE SHOULD CAUSE ERR, ATA, TRE AND SC TO SET. THESE ARE CLEARED BY DOING A CLEAR.

27. TEST BAI IN RSCS2

SET BAI IN RSCS2. DO A ONE WORD WRITE AND CHECK RSBA TO SEE

1066

IF IT INCREMENTED.

V
•

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 11
DESCRIPTION

1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122

28. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS 0 THROUGH 5 IN RMBAE FOR AN 22 BIT ADDRESS. MOV
173000 INTO RSBA. MOV -1000 INTO RSWC AND DO A WRITE
FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO
SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.

29. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR

FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE
WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS
ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.

30. PRIORITY INTERRUPT TEST

HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD
WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE
PROCESSOR IS DROPPED TO PRIORITY 4.

31. DYNAMIC FUNCTION TEST

WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED.
IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS
PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS
STILL READING.

%
.REM%

IDENTIFICATION

MAINDEC-11-DERSA-A
DERSAA.P11

RS11-R503-R504 BASIC FUNCTION DIAGNOSTIC

H03

MACY11 27(732) 06-OCT-76 08:51 PAGE 34

1123
1124
1125
1126
1127
1128
1129
1130
1131
1132

PRODUCT CODE:	MAINDEC-11-DERSA-A-D
PRODUCT NAME:	RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC
DATE CREATED:	MARCH-1975
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	STANLEY HARACKIEWICZ

1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT COPR., MAYNARD, MASS.

MAINDEC-11-DERSA-A
TABLE OF CONTENTS

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATING PROCEDURE
5	OPERATIONAL SWITCH SETTINGS
5.2	SUBROUTINE ABSTRACT
6.	ERRORS
7.	RESTRICTIONS
8.	MISCELLANEOUS
8.1	EXECUTION TIME
8.2	STACK POINTER
9.	WRITE LOCK TEST
10.	TEST DESCRIPTION

1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 4
DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03 AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03 OR RS04) DISKS ARE OPERATING CORRECTLY. THIS IS NOT A RELIABILITY DIAGNOSTIC AND THEREFORE SHOULD NOT BE USED AS ONE. THIS PROGRAM CAN TEST UP TO 9 DRIVES. THE DRIVES CAN BE INTERMIXED AND IN ANY ORDER.

IF THE OPERATOR WOULD LIKE TO CHECK THE DISK REGISTERS PRIOR TO ENTERING THIS DIAGNOSTIC, THERE ARE SOME ROUTINES IN THE BACK OF THE DIAGNOSTIC WHICH CAN BE USED. THESE ROUTINES WILL ALLOW THE OPERATOR TO LOAD THE REGISTERS THROUGH THE SWITCHES. PLEASE REFERENCE THE STARTING ADDRESSES THAT WILL TEST THE REGISTERS YOU DESIRE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OK BK OF MEMORY, AND AN RH11 CONTROLLER WITH A RS03 OR RS04 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR .ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270

MAINDEC-11-DERSA-A RH11-R503-R504 BASIC FUNCTION DIAGNOSTIC PAGE 5
DESCRIPTION

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

1. STARTING ADDRESS 200.

- A. SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
- B. PRESS START.
- C. THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
- D. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
- E. THE PROGRAM WILL TEST ALL R503 AND R504 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-R503/04 REGISTERS USING THE SWITCH REGISTER.

A.	250	WORD COUNT REGISTER TEST
B.	254	BUS ADDRESS REG. TEST
C.	260	DISK ADDRESS REG. TEST
D.	264	DRIVE STATUS REG. TEST
E.	270	ERROR REG. TEST
F.	274	LOOK AHEAD REG. TEST
G.	300	RSCS2 REG. TEST
H.	304	ATTENTION SUMMARY REG. TEST
I.	310	MAINTENANCE REG. TEST
J.	314	RSCS1 REG TEST

5. OPERATIONAL SWITCH SETTINGS

SWITCH SETTINGS ARE:

- SW<15> = 1 HALT ON ERROR
- SW<14> = 1 LOOP ON TEST
- SW<13> = 1 INHIBIT TYPEOUTS
- SW<11> = 1 INHIBIT ITERATIONS OF SUBTEST
- SW<10> = 1 BELL ON ERROR
- 0 BELL ON PASS COMPLETE
- SW<09> = 1 LOOP ON ERROR
- SW<08> = 1 LOOP ON TEST IN SW<7:0>

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION

1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326

SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS
BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE

1327
1328

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 6
DESCRIPTION

CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION
OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST
SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT
TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY
UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT
THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR CS1 = ----- CS2 = ----- ER = -----
GOOD = ----- BAD = -----

WHERE:

CS1,CS2,ER ETC. = RS11 DISK REGISTERS.
GOOD = EXPECTED DATA.
BAD = DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 7
DESCRIPTION

B. MISCELLANEOUS

B.1 EXECUTION TIME

A BELL WILL RING WITHIN 1 MINUTE WITH ALL SWITCHES DOWN.

B.2 STACK POINTER

STACK IS INITIALLY SET TO 500

9. WRITE LOCK TEST

THE WRITE LOCK TEST REQUIRES OPERATOR INTERVENTION. THE STARTING ADDRESS FOR THIS TEST IS 220. THE PROGRAM WILL TELL THE OPERATOR WHICH SWITCHES HAVE TO BE SET.

10. TEST DESCRIPTION

1. TEST RSCS2

CLEAR ALL READ/WRITE BITS AND CHECK. SET ALL R/W BITS AND CHECK. NOW CLEAR AND RECHECK.

2. TEST FOR ONLINE DRIVES

SET ERROR BITS IN RSER. THIS CAUSES ATTENTION SUMMARY BITS TO SET IN RSAS. DO FOR ALL DRIVES. RSAS HAS NOT YET BEEN TESTED. SO IN THE CASE OF NO BITS IN RSAS SETTING, DRIVE 0 IS TESTED.

3. RESET TEST FOR REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. DO A RESET AND TEST ALL R/W BITS TO BE CLEARED.

4. SET AND CLEAR ALL REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.

5. RANDOM NUMBER TEST FOR RSWC AND RSDA

C04

MAINDEC-11-DERSA-A
DERSAA.P11

RS:1-REC3-RS04 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 42

1440
1441

THIS TEST GENERATES RANDOM NUMBERS AND LOADS THEM INTO RSWC.

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 8
DESCRIPTION

1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497

RSDA AND RSBA.

6. TEST "CLEAR BIT" IN RSCS2

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSD8, AND RSMR. SET CLEAR BIT IN RSCS2. NOW TEST ALL R/W BITS FOR 0 IN ALL THE ABOVE REGISTERS.

7. LOAD RSD8 WITH ALL ONES AND ALL ZEROS

LOAD RSD8 WITH A WORD OF ZEROS AND A WORD OF ONES. WAIT FOR "OR" TO SET AND THEN CHECK OUTPUT OF SILO. IF OR DID NOT SET ERROR MESSAGE APPEARS.

8. TEST NO-OP FUNCTION

THE NO-OP FUNCTION IS TESTED WITH AND WITHOUT ERROR BITS SET. ALL THE REGISTERS ARE CHECKED AFTER BOTH CASES.

9. TEST DRIVE CLEAR FUNCTION

FIRST SET ALL R/W BITS IN RSDA, RSWC, RSER, AND RSMR. DO A DRIVE CLEAR FUNCTION. NOW TEST ALL REGISTERS FOR CORRECT DATA.

10. EXECUTE A ONE WORD WRITE FUNCTION

SET RSWC TO -1. MOV -1 INTO OUTBUF. LOAD RSBA WITH OUTBUF. DO A WRITE TEST RDY BIT FOR 0 THEN WAIT FOR IT TO SET. TIME OUT TO ERROR IF RDY BIT DOESN'T SET AND CHECK FOR ERROR CONDITIONS. TEST RSDA FOR CORRECT ADDRESS. TEST WORD COUNT FOR 0.

11. EXECUTE A ONE WORD WRITE CHECK

SET UP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION TEST. DO A WRITE CHECK FUNCTION. TEST RDY AS DONE IN THE WRITE TEST. CHECK FOR WRITE CHECK ERROR. THEN TEST RSDA, RSWC AND RSBA FOR CORRECT DATA.

12. TEST READ FUNCTION

SETUP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION DO A READ FUNCTION. TEST RDY BIT AS DONE IN THE WRITE FUNCTION. TEST FOR ERRORS. ALSO TEST RSDA, RSWC AND RSBA FOR CORRECT DATA.

13. TEST BLOCK SEARCH FUNCTION, PIP AND DRY BITS AND ADDR. CONF. BIT

DO A BLOCK SEARCH FOR SECTOR 32, LOOP ON ADDR. CONF. BIT IN RSMR. IF IT DOESN'T SET, TIMEOUT. WHEN YOU GET THERE DO A

E04

MAINDEC-11-DERSA-A
DERSAA.P11

RS11-REC3-RSD4 BASIC FUNCTION DIAGNOSTIC

MACY11 27(732) 06-OCT-76 08:51 PAGE 44

1498
1499

BLOCK SEARCH FOR SECTOR D. NOW WE KNOW THAT WE HAVE TIME TO
TEST FOR DRY AND PIP BITS BEFORE FINDING SECTOR C. FOR PIP

MAINDEC-11-DERSA-A RH11-RSD3-RSD4 BASIC FUNCTION DIAGNOSTIC PAGE 9
DESCRIPTION

1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555

SHOULD SET AND DRY SHOULD CLEAR BEFORE FINDING SECTOR 0.
ONCE SECTOR 0 IS FOUND PIP SHOULD CLEAR AND DRY SHOULD SET.
IF DRY DOES NOT SET A TIME OUT ERROR WILL OCCUR INDICATING
SECTOR 0 WAS NOT FOUND. SC IN RSCS1 SHOULD ALSO SET. RSBA
AND RSWC SHOULD NOT MOVE, THIS IS ALSO TESTED.

14. ILLEGAL FUNCTION CODE TEST

IN THIS TEST RSBA, RSWC AND RSDA ARE SET UP AS IF TO DO A
LEGAL FUNCTION. AN ILLEGAL FUNCTION IS THEN EXECUTED. THE
PROGRAM TEST FOR ILF AND ERR BITS TO SET. RSBA, RSWC AND
RSDA ARE ALSO TESTED FOR CORRECT DATA. THIS IS DONE FOR ALL
THE ILLEGAL FUNCTIONS.

FOR AN AID IN TROUBLE SHOOTING THE ILLEGAL FUNCTION CODE CAN
BE LOADED INTO LOCATION ILLTAB OR ILFTB2, DEPENDING ON WHICH
ILLEGAL FUNCTION TEST YOU WISH TO LOOP ON. IN THE NEXT
LOCATION, FOLLOWING THE ILLEGAL FUNCTION, A 0 MUST BE LOADED.
NOW BY SETTING SWITCH 14 (LOOP ON TEST), YOU WILL LOOP ON THE
ILLEGAL FUNCTION.

15. TEST PAR IN RSER

SET PAR IN RSER AND CHECK. ALSO TEST ERR IN RSDS TO SET
BECAUSE OF THE PAR SETTING.

16. TEST DPR AND MOL IN RSDS

BOTH THESE BITS SHOULD BE SET IN RSDS IF THE DRIVE IS ON LINE
AND UP TO SPEED.

17. LOOK AHEAD TEST

FIRST CHECK TO SEE IF SECTOR FRACTION BITS ARE MOVING. NOW
SET RSDA TO 0 AND INCREMENT IT EVERY TIME THE ADDR.CONF BIT
SETS. IF THE ADDR.CONF BIT DOES NOT SET IN A CERTAIN LENGTH
OF TIME, A TIME OUT ERROR OCCURS.

18. PARITY TEST

THIS WILL TEST THE PARITY LOGIC ONLY IF THERE IS PARITY
MEMORY ON THE SYSTEM IN LESS THAN 28K. IT WILL WRITE BAD
PARITY IN A MEMORY LOCATION THEN TRY TO DO A WRITE TO THE
DRIVE FROM THAT LOCATION. THIS SHOULD CAUSE A PARITY ERROR.

19. TEST WRITE CHECK ERROR

IN THIS TEST THE PROGRAM WRITES A -1 ON TO THE DISK. A 0 IS
NOW FLOATED THROUGH THE WORD IN THE BUS ADDRESS LOCATION, AND
A WRITE CHECK FUNCTION IS DONE. THE WCE BIT IN RSCS2 SHOULD
SET AND SHOULD CAUSE THE TRE BIT IN RSCS1 TO SET. THESE BITS
ARE THEN CLEARED. A WORD OF 0 IS NOW WRITTEN ON THE DISK AND

MAINDEC-11-DERSA-A
DERSAA.P11

RS11-REC3-RS04 BASIC FUNCTION DIAGNOSTIC

G04

MACY11 27(732) 06-OCT-76 08:51 PAGE 46

1556
1557

H 1 IS FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE
WRITE CHECK FUNCTION TEST IS REPEATED.

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 10
DESCRIPTION

1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
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

20. TEST PROGRAM ERROR BIT IN RSCS2

HERE THE PROGRAM ATTEMPTS TO INITIATE A DATA TRANSFER OPERATION WHILE THE CONTROL IS CURRENTLY PERFORMING ONE. THIS SHOULD CAUSE PGE TO SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED. RSWC IS ALSO TESTED FOR IT SHOULD NOT BE 0 FOR THE CURRENT OPERATION SHOULD HAVE BEEN ABORTED DUE TO THE PGE ERROR.

21. TEST RMR IN RSER

HERE A WRITE COMMAND IS GIVEN AND DURING ITS EXECUTION THE PROGRAM TRYS TO MODIFY THE RSDA REG. THIS SHOULD CAUSE THE RMR BIT TO SET WHICH CAUSES THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

22. TEST DCK IN RSER

HERE A WRITE COMMAND IS GIVEN THEN DURING THIS FUNCTION A DRIVE CLEAR COMMAND IS GIVEN. THIS SHOULD CAUSE THE DCK BIT TO SET WHICH SHOULD CAUSE THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

23. TEST DISK ADDRESS REGISTER

LOAD THE LAST DISK ADDRESS (7777) INTO RSDA. DO A ONE WORD WRITE AND CHECK THAT RSDA INCREMENTED TO 10000.

24. TEST IAE ERROR

DO A ONE WORD WRITE BUT FIRST SET RSDA TO AN INVALID ADDRESS SUCH AS 10000. THIS SHOULD CAUSE A IAE ERROR WHICH WILL CAUSE ERR, ATA AND SC BITS TO SET. THESE BITS ARE THEN CLEARED BY LOADING A 1 INTO ATA IN RSAS.

25. TEST FOR NON-EXISTENT DISK ERROR

FIRST FIND A DRIVE THAT IS NOT ON THE SYSTEM OR OFF LINE. NOW TRY TO DO A ONE WORD WRITE TO THAT DRIVE. NED IN RSCS2 SHOULD SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED BY MOVING A 1 INTO TRE.

26. TEST DAO IN RSER AND LBT IN RSDS

SET RSDA TO ITS LAST ADDRESS. NOW WRITE ONE SECTOR PLUS ONE WORD. DAO SHOULD SET AND LBT SHOULD SET. THESE SHOULD CAUSE ERR, ATA, TRE AND SC TO SET. THESE ARE CLEARED BY DOING A CLEAR.

27. TEST BAI IN RSCS2

SET BAI IN RSCS2. DO A ONE WORD WRITE AND CHECK RSBA TO SEE

1614

IF IT INCREMENTED.

MAINDEC-11-DERSA-A RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 11
DESCRIPTION

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

- 28. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2
SET BITS 0 THROUGH 5 IN RHBAE FOR AN 22 BIT ADDRESS. MOV 173000 INTO RSBA. MOV -1000 INTO RSMC AND DO A WRITE FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.
- 29. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR
FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.
- 30. PRIORITY INTERRUPT TEST
HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE PROCESSOR IS DROPPED TO PRIORITY 4.
- 31. DYNAMIC FUNCTION TEST
WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED. IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS STILL READING.

```
%
;TITLE MAINDEC-11-DERSA-A RS11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
;COPYRIGHT 1973,1974 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
;PROGRAM BY STANLEY HARACKIEWICZ
```

	SWITCH	USE
	SW15= 100000	;HALT ON ERROR
	SW14= 40000	;LOOP ON TEST
	SW13= 20000	;INHIBIT ERROR TYPEOUTS
	SW12= 10000	
	SW11= 4000	;INHIBIT ITERATIONS
	SW10= 2000	;0 - BELL ON PASS COMPLETE
		;1 - BELL ON ERROR
	SW9= 1000	;LOOP ON ERROR
	SW8= 400	;LOOP ON TEST IN SW<7:0>
	0	;TRAP CATCHER FROM 0 - 776
	46	;HOOKS FOR ACT 11
SENDAD	52	
BIT14	200	
	JMP @#STRT	
	220	
	JMP @#WRTLCK	;WRITE LOCK TEST

```
100000
040000
020000
010000
004000
002000
001000
000400
000000
000046
000046
000052
000052
000200
000137 000230
000220
000137 022264
```


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

000001
104000
177776
177776
177570
177570
000007
000000
000001
000002
000003
000004
000005
000006
000007
000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000
000001
000000

GOOD=
BAD=

N= 1
HLT= EMT
PS= 177776
PSW= PS
SWR= 177570
DISPLAY=SWR
BELL= 7
R0= %0
R1= %1
R2= %2
R3= %3
R4= %4
R5= %5
SP= %6
PC= %7
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
%1
%0

```

;INITALIZE FOR NEWTST
;SET HLT TO EMT FOR ERROR TYPEOUTS
;PROCESSOR STATUS
;PROCESSOR STATUS WORD
;SWITCH REGISTER
;DISPLAY REGISTER
;BELL
;R0 - DEFINE REGISTERS
;R1
;R2
;R3
;R4
;R5
;R6 - STACK POINTER
;R7 - PROGRAM COUNTER
;BIT EQUATES

;FOR GOOD DATA
;FOR BAD DATA

```

1731 001000 001000
 1732
 1733 001000 000000
 1734 001002 000000
 1735 001004 000000 000000
 1736 001010 000000
 1737 001012 000000
 1738 001014 001000
 1739 001016 177564
 1740 001020 177566
 1741
 1742 001100 001100
 1743 001102 000000
 1744 001102 000000
 1745
 1746
 1747
 1748 001104 172040
 1749 001106 172050
 1750 001110 172042
 1751 001112 172044
 1752 001114 172046
 1753 001116 172052
 1754 001120 172054
 1755 001122 172056
 1756 001124 172060
 1757 001126 172062
 1758 001130 172064
 1759 001132 172066
 1760 001134 172070
 1761 001136 172072
 1762 001140 000204
 1763 001142 000206
 1764 001144 172041
 1765 001146 172051
 1766 001150 172043
 1767 001152 172045
 1768
 1769
 1770
 1771 177572
 1772 172340
 1773 172342
 1774 172344
 1775 172356
 1776 172300
 1777 172302
 1778 172304
 1779 172316
 1780 000006
 1781 000000
 1782

000000

. = 1000

ICNT: 0
 ERRORS: 0
 PCNT: 0,0
 LAD: 0
 HLTADR: 0
 FILCHR: 1000
 TPS: 177564
 TPB: 177566

; LH = ITERATION COUNT ; RH = TEST NO.
 ; ERROR COUNT
 ; 2 WORD PASS COUNT
 ; LOOP ADDRESS FOR SCOPE
 ; ADDRESS OF LAST HLT INSTRUCTION EXECUTED
 ; FILCHR=0 (CHAR) ; FILCHR+1=2 (COUNT)
 ; OUTPUT STATUS REGISTER
 ; OUTPUT BUFFER

. = 1100

SAVBAD: 0
 OBUFSV: 0

; LOC FOR ILLEGAL FUNCTION CODE
 ; LOC OF OUTBUF

;DISK I/O REGISTERS

RSCS1: 172040
 RSCS2: 172050
 RSWC: 172042
 RSBA: 172044
 RSDA: 172046
 RSDS: 172052
 RSER: 172054
 RSAS: 172056
 RSLA: 172060
 RSOB: 172062
 RSMR: 172064
 RSDT: 172066
 RSBAB: 172070
 RSCS3: 172072
 RSVEC: 204
 RSVCP5: 206
 RSCS1B: 172041
 RSCS2B: 172051
 RSWCB: 172043
 RSBAB: 172045

; DISK CONTROL + STATUS REGISTER
 ; DISK CONTROL + STATUS REGISTER
 ; WORD COUNT REGISTER
 ; BUS ADDRESS
 ; DISK ADDRESS (DESIRED ADDRESS)
 ; DRIVE STATUS
 ; ERROR REG.
 ; ATTENTION SUMMARY
 ; LOOK AHEAD
 ; DATA BUFFER REGISTER
 ; MAINTENANCE REGISTER
 ; DRIVE TYPE REGISTER

; INTERRUPT VECTOR
 ; INTERRUPT PRIO. VECTOR
 ; ODD BYTE ADD FOR CS1
 ; ODD BYTE ADD FOR CS2
 ; ODD BYTE ADD FOR CW
 ; ODD BYTE ADD FOR BA

;MEMORY MANAGEMENT REGISTER ASSIGNMENTS

SRO=177572
 KIPAR0=172340
 KIPAR1=172342
 KIPAR2=172344
 KIPAR7=172356
 KIPDR0=172300
 KIPDR1=172302
 KIPDR2=172304
 KIPDR7=172316
 RW=6
 UP=00

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

000001
000002
000004
000010
000020
000040
000100
000200
000204
000210
000220
000240

040000
100000
000100
000200
002000
010000
040000
100000
000200
020000
002000
040000
100000
001000
100000
000010
000100

```

;BIT ASSIGNMENTS FOR ERROR TYPEOUTS
;THE RS REGISTERS ARE DIVIDED INTO 3 GROUPS.
;CS1 CS2 AND ER ARE IN THE FIRST GROUP. THIS GROUP IS ALWAYS
;TYPED WITH EITHER OF THE OTHER GROUPS. AS,BA,DA, WC AND DS
;ARE IN THE SECOND GROUP. DT, DB, MR, AND LA ARE IN THE 3RD
;GROUP. YOU CAN NOT INTERMIX GROUP 2 OR 3. THEY HAVE
;TO BE TYPED SEPERATELY.
;EXAMPLE:  HLT !CS1 AS BA
           HLT !CS1!DT!DB
    
```

```

CS1=1      ;CONTROL AND STATUS 1
ER=2       ;CONTROL AND STATUS 2
DA=4       ;DESIRED ADD
WC=10      ;WORD COUNT
BA=20      ;BUS ADDRESS
DS=40      ;DRIVE STATUS
AS=100     ;ATTENTION SUMMARY
CS2=200    ;CONTROL AND STATUS REG
LA=204     ;LOOK AHEAD
DB=210     ;DATA BUFFER
MR=220     ;MAINTENANCE
DT=240     ;DRIVE TYPE
    
```

;BIT ASSIGNMENTS FOR THE REGISTER BITS

```

TRE=40000  ;TRANSFER ERROR CS1
SC=100000  ;SPECIAL CONDITIONS CS1
IR=100     ;INPUT READY CS2
OR=200     ;OUTPUT READY CS2
PGE=2000   ;PROGRAM ERROR-CS2
NED=10000  ;NON-EXISTENT DRIVE CS2
WCE=40000  ;WRITE CHECK ERROR-CS2
DLT=100000 ;DATA LATE ERROR CS2
DRY=200    ;DRIVE READY DS
PIP=20000  ;POSITIONING IN PROGRESS DS
LBT=2000   ;LAST BLOCK TRANSFER-DS
ERR=40000  ;ERROR DS
ATA=100000 ;ATTENTION ACTIVE-DS
DAO=1000   ;DISK OVERFLOW ERROR-ER
DCK=100000 ;DATA CHECK ERROR-ER
BAI=10     ;BUS ADDR INCREMENT INHIBIT
IE=100     ;INTERRUPT INABLE CS1
    
```

:WORKING LOCATIONS

```

1828 001154 146723
1829 001156 000000
1830 001160 000000
1831 001162 000000
1832 001164 000000
1833 001166 000000
1834 001170 000000
1835 001172 000000
1836 001174 000000
1837 001176 000000

```

```

RANMU: 146723
UNNUM: 0
UNITSV: 0
UNCMP: 0
ONCEE: 0
RSC4DT: 0
TIMSV: 0
AOS1: 0
MWP=4
MPRO=172100
BPORTT: 0
SAVEE: 0

```

```

:RANDOM NUMBER PRIME
:UNIT CURRENTLY BEING TESTED
:SET BIT=UNIT ON BUS
:FOR COMPARING FOR # OF DEVICE
:DID WE TEST ANY DRIVES
:CLR IF R503 SET IF R504
:SAVE LOC FOR TIME
:PORT SWITCH
:WRITE WRONG PARITY
:PARITY REG
:BUFFER ADDR FOR -B- PORT
:WORK LOC

```

:DISCRIPTION OF ONCEE BITS

```

:BIT0 MEANS FOUND DRIVE
:BIT11 DO TKSEL TEST
:BIT12 TYPE COULD NOT FIND NED ONLY ONCE
:BIT13 TYPE NO MEM ON B PORT ONLY ONCE
:BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
:BIT15 MEANS ERROR FOUND

```

:RH11 WORK REGISTERS

:(CAN BE CHANGED IN ANY ROUTINE)

```

1850 001200 000000
1851 001202 000000
1852 001204 000000

```

```

WORK: 0
WORK1: 0
WORK2: 0

```

```

1853 001206 012706 000500
1854 001212 012737 021040 000024
1855 001220 012737 000340 000026
1856 001226 012737 020470 000030
1857 001234 012737 000340 000032
1858 001242 012737 021440 000034
1859 001250 012737 000340 000036
1860 001256 005067 177516
1861 001262 005067 177522
1862 001266 042767 143777 177570
1863 001274 012767 024550 177600
1864 001302 232767 004000 177654
1865 001310 001402
1866 001312 030137 021736

```

```

BEGIN: MOV #500,SP ;SET STACK TO *** 500 ***
MOV #.POWER,2#24 ;SET UP PF VECTOR
MOV #340,2#26 ;LOCK OUT THE WORLD
MOV #.HLT,2#30 ;SET ENT VECTOR
MOV #340,2#32 ;LOCK UP
MOV #.TRAP,2#34 ;SET TRAP VECTOR
MOV #340,2#36 ;LOCK UP
CLR ICNT ;INIT ICNT
CLR LAD ;INIT LAD
BIC #143777,ONCEE ;CLEAR ONCEE
MOV #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
BIT #BIT11,ONCEE ;DO TKSEL TEST?
BEQ +6 ;NO
JMP 2#TKSEL ;YES

```

```

1868 :NOW TEST FOR DRIVES
1869 :*****
1870 :TEST 1 TEST FOR DRIVES
1871 :*****
1872 001316 104400 †ST1: SCOPE
1873
1874 001320 012701 000010 MULTII: MOV #8, R1 ;PUT 8 INTO R1 FOR COUNT
1875 001324 005077 177556 CLR #RSCS2 ;SET DEVICE TO ZERO
1876 001330 012777 000007 177562 TRY: MOV #7, RSRER ;CAUSE AN ERROR +SETS BIT IN RSAS REG
1877 001336 005301 DEC R1 ;DO A MAXIMUM OF 8 TIMES
1878 001340 001403 BEQ DVNUM ;TESTED FOR ALL DRIVES GET OUT
1879 001342 005277 177540 INC #RSCS2 ;INCREMENT DRIVE UNIT
1880 001346 000770 BR TRY ;REPEAT FOR NEXT DRIVE
1881 001350 017767 177546 177602 DVNUM: MOV #RSAS, UNITSV ;SAVE
1882 001356 012767 000401 177576 MOV #401, UNCMP ;SETUP TO CMP WITH UNITSV
1883 001364 012767 000000 177564 MOV #0, UNNUM ;PUT 0 INTO UNIT NO.
1884 001372 032767 020000 176170 BIT #BIT13, SWR ;INHIBIT TYPE OUT?
1885 001400 001015 BNE STTEST ;YES
1886 001402 104402 001406 TYPE ;ASCIZ <15><12>"TESTING UNIT "
1887 001426 042767 100000 177530 BIC #BIT15, ONCEE ;CLEAR ERROR FLAG
1888 001434 036767 177522 177516 STTEST: BIT UNCMP, UNITSV ;IS THIS DRIVE ON THE SYSTEM
1889 001442 001461 BEQ TRYNX ;NO
1890 001444 016777 177506 177434 MOV UNNUM, #RSCS2 ;YES PUT UNIT # INTO CS2
1891 001452 022777 000000 177452 CMP #0, #RSOT ;IS THIS A RS03?
1892 001460 001003 BNE 2$ ;NO
1893 001462 005067 177500 CLR #RS04DT ;YES
1894 001466 000422 BR 1$ ;GET OUT
1895 001470 022777 000001 177434 2$: CMP #1, #RSOT ;IS THIS A RS03 4US?
1896 001476 001003 BNE 3$ ;NO
1897 001500 005067 177462 CLR #RS04DT ;RS03
1898 001504 000413 BR 1$ ;GET OUT
1899 001506 022777 000002 177416 3$: CMP #2, #RSOT ;IS THIS A RS04?
1900 001514 001404 BEQ 6$ ;YES
1901 001516 022777 000003 177406 CMP #3, #RSOT ;IS IT A RS04?
1902 001524 001030 BNE TRYNX ;GET A NEW NUMBER
1903 001526 052767 177777 177432 6$: BIS #-1, #RS04DT ;YES RS04
1904 001534 032767 020000 176026 1$: BIT #BIT13, SWR ;INHIBIT TYPE OUT?
1905 001542 001020 BNE 4$ ;YES
1906 001544 032767 100000 177412 BIT #BIT15, ONCEE ;ANY ERRORS?
1907 001552 001404 BEQ 5$ ;NO
1908 001554 104402 001560 TYPE ;ASCIZ <15><12><12>
1909 001564 5$:
1910 001564 016746 177366 MOV UNNUM, -(6) ;PUT UNNUM ON STACK
1911 001570 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
1912 001572 104402 000040 TYPE ;TYPE SPACE
1913 001576 042767 100000 177360 BIC #BIT15, ONCEE ;CLEAR ERROR FLAG
1914 001604 000473 4$: BR NOWGO ;NOW TEST

```

```

1915 001606 006367 177350 TRYNY: ASL UNCMP ;CHECK NEXT BIT FOR DRIVE
1916 001612 103403 BCS CHCKDV ;DID WE TEST ANY REG?
1917 001614 005267 177336 INC UNNUM ;INC UNIT #
1918 001620 000705 BR STTEST ;CHECK FOR NEXT DRIVE
1919
1920 001622 032767 000001 177334 CHCKDV: BIT #BIT0,ONCEE ;DID WE TEST ANY DRIVES?
1921 001630 001057 BNE DONEE ;YES WE DID TEST A DRIVE
1922 001632 012767 100000 177322 MOV #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
1923 001640 005067 177312 CLR UNNUM ;ANY AS BITS, THUS DEFAULTS TO
1924 001644 032767 020000 175716 BIT #BIT13,SWR ;INHIBIT TYPE OUT?
1925 001652 001045 BNE 4$ ;YES
1926 001654 016746 177276 MOV UNNUM,-(6) ;PUT UNNUM ON STACK
1927 001660 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
1928 001662 104402 000040 TYPE ,40 ;TYPE SPACE
1929 001666 104402 001672 TYPE ,.+2 ;ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
1930 001764 000000 HALT ;WAIT
1931 001766 000402 4$: BR NOWGO ;TEST DRIVE 0
1932 001770 000167 016042 DONEE: JMP DONE ;GET OUT
1933
1934 ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
1935 ;TO CLEAR ALL THE RH AND RS REGISTERS
1936
1937 001774 012767 024550 177100 NOWGO: MOV #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
1938 002002 052767 000001 177154 BIS #BIT0,ONCEE ;SET FOUND DRIVE FLAG
1939 002010 016767 016452 177152 MOV TIMES,TIMSV ;SAVE TIME
1940 002016 012767 000001 016442 NOV #1,TIMES ;ONLY TEST ONCE
1941
1942 ;*****
1943 ;TEST 2 RESET TEST FOR REGISTERS
1944 ;*****
1945 TST2: SCOPE
1946 MOV #340,20PS ;LOCK OUT INTERRUPTS
1947 MOV UNNUM,2RSCS2 ;GET UNIT #
1948 MOV #177776,2RSCS1 ;SET ALL
1949 MOV #177777,2RSBA ;POSSIBLE R/W
1950 MOV #177777,2RSDA ;BITS IN THESE REGISTERS
1951 MOV #177777,2RSER
1952 MOV #177777,2RSR
1953 MOV #177777,2RSWC
1954 MOV #177737,2RSCS2 ;CLEAR ALL BITS IN ALL REG.
1955
1956 ;TEST RSCS2 FOR CLEARED BITS
1957
1958 002116 022777 000100 176762 CMP #100,2RSCS2 ;DID THESE BITS GET CLEARED?
1959 002124 001401 BEQ +4 ;YES
1960 002126 104200 HLT !CS2 ;(417) SHOULD BE CLEARED IN CS2
1961 002130 016777 177022 176750 MOV UNNUM,2RSCS2 ;PUT # OF UNIT IN TEST IN CS2
1962 002136 022777 010600 176752 CMP #10600,2RSDS ;IS DPR AND MOL SET?
1963 002144 001401 BEQ +4 ;YES
1964 002146 104040 HLT !DS ;NO WHY NOT?
1965
1966 002150 022777 004200 176726 ;TEST CONTROL AND STATUS REG 1
1967 002156 001401 CMP #4200,2RSCS1 ;DID THE READY BIT SET?
1968 002160 104001 BEQ +4 ;YES
HLT !CS1 ;READY SHOULD BE SET

```

```

1969          ;TEST BUS ADDRESS REGISTER
1970
1971 002162 005777 176724          TST   @RSBA          ; IS BA REG. CLEARED
1972 002166 001401                BEQ   .+4             ; YES
1973 002170 104020                HLT   !BA             ; SHOULD BE 0
1974
1975          ;TEST DISK ADDRESS REGISTER
1976
1977 002172 005777 176716          TST   @RSDA          ; IS DA CLEARED
1978 002176 001401                BEQ   .+4             ; YES
1979 002200 104004                HLT   !DA             ; SHOULD BE 0
1980
1981          ;TEST ERROR REG RSER
1982
1983 002202 005777 176712          TST   @RSER          ; DID RSER CLEAR?
1984 002206 001401                BEQ   .+4             ; YES
1985 002210 104002                HLT   !ER             ; BITS(157015) SHOULD BE CLEARED
1986
1987          ;TEST RS MAINTENANCE REGISTER
1988
1989 002212 032777 000077 176710  BIT   #77,@RSMR      ; DID THESE BITS GET CLEARED
1990 002220 001401                BEQ   .+4             ; YES
1991 002222 104220                HLT   !MR             ; BITS(77) SHOULD BE 0
1992
1993          ;TEST WC REG IT SHOULD NOT CHANGE
1994
1995 002224 022777 177777 176656  CMP   #177777,@RSWC ; DID IT CHANGE?
1996 002232 001401                BEQ   .+4             ; NO
1997 002234 104010                HLT   !WC             ; RESET SHOULD NOT MODIFY RSWC
1998
1999          ;TEST RSAS
2000
2001 002236 005777 176660          TST   @RSAS          ; IS REG CLEAR
2002 002242 001401                BEQ   .+4             ; YES
2003 002244 104100                HLT   !AS             ; NO

```

```

2004
2005
2006
2007 002246 104400
2008
2009 002250 012737 000340 177776
2010 002256 016777 176674 176622
2011 002264 012777 043576 176612
2012 002272 012777 177777 176612
2013 002300 012777 177777 176606
2014 002306 012777 177777 176604
2015 002314 012777 177777 176604
2016 002322 012777 177777 176560
2017 002330 012777 020417 176550
2018 002336 012777 000071 176564
2019 002344 012777 000040 176534
2020 002352 022777 000100 176526
2021 002360 001401
2022 002362 104200
2023 002364 016777 176566 176514
2024 002372 032777 173577 176504
2025 002400 001401
2026 002402 104001
2027
2028
2029 002404 005777 176502
2030 002410 001401
2031 002412 104020
2032
2033
2034
2035 002414 005777 176474
2036 002420 001401
2037 002422 104020
2038
2039
2040
2041 002424 005777 176470
2042 002430 001401
2043 002432 104002
2044
2045
2046 002434 032777 000077 176466
2047 002442 001401
2048 002444 104220
2049
2050
2051 002446 022777 177777 176434
2052 002454 001401
2053 002456 104010

```

```

;*****
;TEST 3 TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
;*****
TST3: SCOPE

TTAGG: MOV #340, @#PS ;LOCK OUT INTERRUPTS
MOV UNNUM, @RSCS2 ;GET UNIT #
MOV #43576, @RSCS1 ;SET ALL
MOV #177777, @RSBA ;POSSIBLE
MOV #177777, @RSDA ;REGISTERS
MOV #177777, @RSER
MOV #177777, @RSDB
MOV #177777, @RSWC
MOV #20417, @RSCS2
MOV #71, @RSMR
MOV #40, @RSCS2 ;CLEAR ALL BITS
CMP #100, @RSCS2 ;DID THE RIGHT BITS CLEAR?
BEQ +4 ;YES
HLT !CS2 ;(417) SHOULD BE CLEARED IN CS2
MOV UNNUM, @RSCS2 ;GET DRIVE NUMBER
BIT #173577, @RSCS1 ;DID ALL BITS GET CLEARED
BEQ +4 ;YES
HLT !CS1 ;NO, ALL BITS SHOULD BE 0
;TEST BUS ADDRESS REGISTER
TST @RSBA ;IS BA REG. CLEARED
BEQ +4 ;YES
HLT !BA ;SHOULD BE 0
;TEST DISK ADDRESS REGISTER
TST @RSDA ;IS DA CLEARED
BEQ +4 ;YES
HLT !BA ;SHOULD BE 0
;TEST ERROR REG RSER
TST @RSER ;DID THESE BITS GET CLEARED
BEQ +4 ;YES
HLT !ER ;BITS(157015) SHOULD BE CLEARED
;TEST RS MAINTENANCE REGISTER
BIT #77, @RSMR ;DID THESE BITS GET CLEARED
BEQ +4 ;YES
HLT !MR ;BITS(77) SHOULD BE 0
;TEST WC REG. IT SHOULD NOT CHANGE
CMP #177777, @RSWC ;DID WC CHANGE
BEQ +4 ;NO
HLT !WC ;WHY DID IT CHANGE?

```



```

2054
2055
2056
2057 002460 104400
2058
2059
2060
2061 002462 104414
2062 002464 016767 176500 015774
2063 002472 012777 003576 176404
2064 002500 022777 005776 176376
2065 002506 001401
2066 002510 104001
2067 002512 012777 002524 176364
2068 002520 022777 004724 176356
2069 002526 001401
2070 002530 104001
2071 002532 012777 001052 176344
2072 002540 022777 005252 176336
2073 002546 001401
2074 002550 104001
2075 002552 104400
2076
2077
2078 002554 012777 043576 176322
2079 002562 005077 176316
2080 002566 022777 004200 176310
2081 002574 001401
2082 002576 104001
2083
2084
2085
2086
2087 002600 104400
2088
2089 002602 000005
2090 002604 022777 000100 176274
2091 002612 001401
2092 002614 104200
2093 002616 012777 021037 176262
2094 002624 022777 000137 176254
2095 002632 001405
2096 002634 017700 176246
2097 002640 012701 000137
2098 002644 104000

```

```

*****
:TEST 4 SET AND CLEAR ALL REGISTERS
*****

```

```

TST4: SCOPE
:CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
:BITS 7,6,5,4,3,2&1

```

```

CLRDK ;CLEAR ALL RS REG
MOV TIMSV,TIMES ;GET TIME
MOV #3576,RSCS1 ;SET DISK FUNCTION BITS
CMP #5776,RSCS1 ;ARE THESE BITS SET?
BEQ +4 ;NO
HLT !CS1 ;SHOULD = 3776
MOV #2524,RSCS1 ;SET THESE BITS
CMP #4724,RSCS1 ;DID THEY SET
BEQ +4 ;YES
HLT !CS1 ;SHOULD BE 2725
MOV #1052,RSCS1 ;SET THESE BITS
CMP #5252,RSCS1 ;ARE THEY =?
BEQ +4 ;YES
HLT !CS1 ;SHOULD = 1252

```

```

TST5: SCOPE
;CLEAR THE FUNCTION BITS

```

```

MOV #43576,RSCS1 ;SET DISK FUNCTION BITS
CLR RSCS1
CMP #4200,RSCS1 ;IS THE READY BIT SET
BEQ +4 ;YES
HLT !CS1 ;RSCS1 SHOULD = 4200

```

```

*****
:TEST 6 TEST RSCS2
*****

```

```

TST6: SCOPE
RESET ;CLEAR WORLD
CMP #100,RSCS2 ;DID THEY CLEAR?
BEQ +4 ;YES
HLT !CS2 ;NO
MOV #21037,RSCS2 ;SET BITS 21017
CMP #137,RSCS2 ;DID THESE BITS GET SET
BEQ 1$ ;YES
MOV RSCS2,BAD ;WHAT CS2 SHOULD =
MOV #137,GOOD ;CS2 = BAD GOOD = CORRECT ANS
HLT

```

2099	002646	012777	020025	176232	IS:	MOV	#20025,RSRCS2	;SET THESE BITS
2100	002654	022777	000125	176224		CMP	#125,RSRCS2	;DID THESE BITS GET SET
2101	002662	001401				BEQ	.+4	;YES
2102	002664	104200				HLT	:CS2	;NO,CS2 SHOULD = 20125
2103	002666	012777	000012	176212		MOV	#12,RSRCS2	;LOAD THESE BITS
2104	002674	022777	000112	176204		CMP	#112,RSRCS2	;DID THESE BITS GET SET IN CS2
2105	002702	001401				BEQ	.+4	;YES
2106	002704	104200				HLT	:CS2	;BAD = CS2 GOOD = CORRECT ANS
2107	002706	012777	177777	176172		MOV	#-1,RSRCS2	;SET BITS
2108	002714	005077	176166			CLR	RSRCS2	;CLEAR THEM
2109	002720	022777	000100	176160		CMP	#100,RSRCS2	;DID CLEAR WORK
2110	002726	001401				BEQ	.+4	;YES
2111	002730	104200				HLT	:CS2	;R/W BITS DID NOT CLEAR
2112	002732	016777	176220	176146		MOV	UNNUM,RSRCS2	;GET UNIT #
2113	002740	104400			TST7:	SCOPE		
2114					;CAN WE	SET ALL	THE RSBA BITS	
2115								
2116	002742	012777	177777	176142		MOV	#177777,RSRBA	;SET THE BITS
2117	002750	022777	177776	176134		CMP	#177776,RSRBA	;DID THEY SET
2118	002756	001401				BEQ	.+4	;YES
2119	002760	104020				HLT	:BA	;BITS 17776 SHOULD BE SET
2120	002762	012777	125252	176122		MOV	#125252,RSRBA	;SET THESE BITS
2121	002770	022777	125252	176114		CMP	#125252,RSRBA	;ARE THEY =
2122	002776	001401				BEQ	.+4	;YES
2123	003000	104020				HLT	:BA	;SHOULD BE 125252
2124	003002	012777	052524	176102		MOV	#52524,RSRBA	;SET THESE BITS
2125	003010	022777	052524	176074		CMP	#52524,RSRBA	;ARE THEY =
2126	003016	001401				BEQ	.+4	;YES
2127	003020	104020				HLT	:BA	;SHOULD BE 52524
2128					TST10:	SCOPE		
2129	003022	104400			;FLOAT A	1 THROUGH	RSBA	
2130								
2131					FLOTBA:	MOV	#2,GOOD	;GET A 2
2132	003024	012701	000002			CLC		;CLEAR CARRY
2133	003030	000241			IS:	MOV	GOOD,RSRBA	;FLOAT NUMBER
2134	003032	010177	176054			MOV	RSRBA,BAD	;GET BA
2135	003036	017700	176050			CMP	GOOD,BAD	;COMPARE BA
2136	003042	020100				BEQ	.+4	;BA CORRECT
2137	003044	001401				HLT		;BAD=BA GOOD=CORRECT AN.
2138	003046	104000				ROL	GOOD	;ROTATE NUMBER
2139	003050	006101				BCC	IS	;LOOP TILL DONE
2140	003052	103367						

```

2141 003054 104400      TST11: SCOPE
2142
2143      ;CLEAR THE RSBA REGISTER
2144
2145 003056 012777 177777 176026      MOV      #177777,@RSBA      ;SET RSBA EQUAL TO ALL ONES
2146 003064 005077 176022      CLR      @RSBA
2147 003070 005777 176016      TST      @RSBA      ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
2148 003074 001401      BEQ      .+4      ;YES
2149 003076 104020      HLT      !BA      ;NO
2150 003100 104400      TST12: SCOPE
2151
2152      ;CAN WE SET ALL BITS IN RSWC REGISTER
2153
2154 003102 012777 177777 176000      MOV      #177777,@RSWC      ;SET WC BITS
2155 003110 022777 177777 175772      CMP      #177777,@RSWC      ;ARE ALL BITS SET
2156 003116 001401      BEQ      .+4      ;YES
2157 003120 104010      HLT      !WC      ;NO
2158 003122 012777 125252 175760      MOV      #125252,@RSWC      ;SET THESE BITS
2159 003130 022777 125252 175752      CMP      #125252,@RSWC      ;ARE THEY =
2160 003136 001401      BEQ      .+4      ;YES
2161 003140 104010      HLT      !WC      ;SHOULD BE 125252
2162 003142 012777 052525 175740      MOV      #52525,@RSWC      ;SET THESE BITS
2163 003150 022777 052525 175732      CMP      #52525,@RSWC      ;ARE THEY =
2164 003156 001401      BEQ      .+4      ;YES
2165 003160 104010      HLT      !WC      ;SHOULD BE 152525
2166 003162 104400      TST13: SCOPE
2167
2168      ;FLOAT A 1 THROUGH RSWC
2169
2170 003164 012701 000001      FLOTWC: MOV      #1,GOOD      ;GET A 1
2171 003170 000241      CLC      ;CLEAR CARRY
2172 003172 010177 175712      IS:      MOV      GOOD,@RSWC      ;FLOAT NUMBER
2173 003176 017700 175706      MOV      @RSWC,BAD      ;GET WC
2174 003202 020100      CMP      GOOD,BAD      ;COMPARE WC
2175 003204 001401      BEQ      .+4      ;WC CORRECT
2176 003206 104000      HLT      ;BAD=WC GOOD=CORRECT ANS
2177 003210 006101      ROL      GOOD      ;ROTATE NUMBER
2178 003212 103367      BCC      IS      ;LOOP TILL DONE

```

```

2179                                     ;CLEAR THE WORD COUNT REGISTER
2180 003214 104400 TST14: SCOPE
2181
2182 003216 012777 177777 175664      MOV    #177777, @RSWC ;SET RSWC REGISTER EQUAL TO ALL ONES
2183 003224 005077 175660              CLR    @RSWC
2184 003230 005777 175654              TST    @RSWC ;DID ALL BITS GET CLEARED
2185 003234 001401                      BEQ    .+4 ;YES
2186 003236 104010                      HLT    !WC ;NO
2187 003240 104400 TST15: SCOPE
2188
2189                                     ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
2190
2191 003242 012777 177777 175644      MOV    #177777, @RSDA ;SET ALL BITS
2192 003250 022777 177777 175636      CMP    #177777, @RSDA ;ARE THE BITS SET
2193 003256 001401                      BEQ    .+4 ;YES
2194 003260 104004                      HLT    !DA ;NO
2195 003262 012777 125252 175624      MOV    #125252, @RSDA ;SET THESE BITS
2196 003270 022777 125252 175616      CMP    #125252, @RSDA ;ARE THEY =
2197 003276 001401                      BEQ    .+4 ;YES
2198 003300 104004                      HLT    !DA ;SHOULD BE 125252
2199 003302 012777 052525 175604      MOV    #52525, @RSDA ;SET THESE BITS
2200 003310 022777 052525 175576      CMP    #52525, @RSDA ;ARE THEY =
2201 003316 001401                      BEQ    .+4 ;YES
2202 003320 104004                      HLT    !DA ;SHOULD BE 52525
2203 003322 104400 TST16: SCOPE
2204
2205                                     ;FLOAT A 1 THROUGH RSDA
2206
2207 003324 012701 000001 FLOTDA: MOV    #1, GOOD ;GET A 1
2208 003330 000241                      CLC ;CLEAR CARRY
2209 003332 010177 175556 IS: MOV    GOOD, @RSDA ;FLOAT NUMBER
2210 003336 017700 175552              MOV    @RSDA, BAD ;GET DA
2211 003342 020100                      CMP    GOOD, BAD ;COMPARE DA
2212 003344 001401                      BEQ    .+4 ;DA CORRECT
2213 003346 104000                      HLT ;BAD=DA GOOD=CORRECT ANS
2214 003350 006101                      ROL    GOOD ;ROTATE NUMBER
2215 003352 103367                      BCC    IS ;LOOP TILL DONE

```

```

2216                                     ;CAN WE CLEAR THE RSDA REG.
2217 003354 104400 TST17: SCOPE
2218                                     ;
2219 003356 012777 177777 175530      MOV    #177777,@RSDA ;SET RSDA TO ALL ONES
2220 003364 005077 175524              CLR    @RSDA
2221 003370 005777 175520              TST    @RSDA ;TEST FOR ZERO RSDA
2222 003374 001401                      BEQ    +4 ;YES
2223 003376 104004                      HLT    !DA ;ANS SHOULD BE 0
2224 003400 104400 TST20: SCOPE
2225                                     ;
2226                                     ;SET AND CLEAR THE RSER REG.
2227                                     ;
2228 003402 012777 177777 175510      MOV    #177777,@RSER ;SET THESE BITS
2229 003410 022777 177017 175502      CMP    #177017,@RSER ;DID THEY SET
2230 003416 001401                      BEQ    +4 ;YES
2231 003420 104002                      HLT    !ER ;RSER SHOULD = 157017
2232 003422 112777 000001 175470      MOVB  #1,@RSER ;A MOVB INST
2233 003430 022777 000001 175462      CMP    #1,@RSER ;SHOULD MODIFY COMPLETE WD
2234 003436 001401                      BEQ    +4 ;OK
2235 003440 104002                      HLT    !ER
2236                                     ;
2237 003442 104400 TST21: SCOPE
2238                                     ;
2239 003444 012777 052005 175446      MOV    #52005,@RSER ;SET THESE BITS
2240 003452 022777 052005 175440      CMP    #52005,@RSER ;DID THEY SET
2241 003460 001401                      BEQ    +4 ;YES
2242 003462 104002                      HLT    !ER ;ER SHOULD = 52005
2243 003464 104400 TST22: SCOPE
2244                                     ;
2245 003466 012777 125012 175424      MOV    #125012,@RSER ;SET THESE BITS
2246 003474 022777 125012 175416      CMP    #125012,@RSER ;DID THEY SET
2247 003502 001401                      BEQ    +4 ;YES
2248 003504 104002                      HLT    !ER ;ER SHOULD = 105012

```

```

2249 003506 104400          TST23: SCOPE
2250
2251 003510 012777 177017 175402      MOV      #177017, @RSER      ;SET THESE BITS
2252 003516 005077 175376          CLR      @RSER              ;CLEAR THEM
2253 003522 005777 175372          TST      @RSER              ;DID THEY CLEAR
2254 003526 001401          BEQ      +4                  ;YES
2255 003530 104002          HLT      !ER                 ;SHOULD = 0
2256 003532 104400          TST24: SCOPE
2257
2258          ;SET AND CLEAR RSMR
2259
2260 003534 012777 000070 175366      MOV      #70, @RSMR         ;SET THESE BITS
2261 003542 017767 175362 175430      MOV      @RSMR, WORK        ;PUT INTO WORKABLE REG
2262 003550 042767 177700 175422      BIC      #177700, WORK      ;CLEAR JUNK
2263 003556 022767 000070 175414      CMP      #70, WORK          ;DID THEY SET
2264 003564 001401          BEQ      +4                  ;YES
2265 003566 104220          HLT      !MR                 ;SHOULD = 70
2266 003570 104400          TST25: SCOPE
2267
2268 003572 012777 000070 175330      MOV      #70, @RSMR         ;SET BITS
2269 003600 005077 175324          CLR      @RSMR              ;CLEAR THEM
2270 003604 032777 000077 175316      BIT      #77, @RSMR         ;DID THEY CLEAR
2271 003612 001401          BEQ      +4                  ;YES
2272 003614 104220          HLT      !MR                 ;BITS (77) SHOULD = 0
2273 003616 104400          TST26: SCOPE
2274
2275 003620 012777 000050 175302      MOV      #50, @RSMR         ;SET BITS
2276 003626 017767 175276 175344      MOV      @RSMR, WORK        ;PUT IN WORKABLE REG
2277 003634 042767 177700 175336      BIC      #177700, WORK      ;CLEAR JUNK
2278 003642 022767 000050 175330      CMP      #50, WORK          ;DID THESE BITS SET
2279 003650 001401          BEQ      +4                  ;YES
2280 003652 104220          HLT      !MR                 ;BITS (50) SHOULD BE SET
2281 003654 104400          TST27: SCOPE
2282
2283 003656 012777 000020 175244      MOV      #20, @RSMR         ;SET BITS
2284 003664 017767 175240 175306      MOV      @RSMR, WORK        ;PUT INTO WORKABLE REG
2285 003672 042767 177700 175300      BIC      #177700, WORK      ;CLEAR JUNK
2286 003700 022767 000020 175272      CMP      #20, WORK          ;DID THEY SET
2287 003706 001401          BEQ      +4                  ;YES
2288 003710 104220          HLT      !MR                 ;MR SHOULD AT LEAST HAVE A (21)

```

```

*****
:TEST 30          LOAD RANDOM NUMBERS INTO RSWC, RSDA AND RSBA
*****
TST30:  SCOPE

2289
2290
2291
2292 003712 104400
2293
2294 003714 012767 001000 175256 RANTS:  MOV #1000,WORK ;MAKE TABLE 1000 WDS LONG
2295 003722 012701 024550          MOV #OUTBUF,R1 ;GET STARTING LOC OF TABLE
2296 003726 004567 020360          JSR  R5,RANDOM ;GENERATE #
2297 003732 012704 024550          MOV #OUTBUF,R4 ;SETUP FOR COMPARE
2298 003736 012767 003744 175044 LOP1:  MOV #LOP1,LAD ;SETUP LOOP ADDR
2299 003744 012703 001000          MOV #1000,R3 ;LOAD TEST COUNTER
2300 003750 005303          4$:  DEC  R3 ;DONE WITH COMPARE?
2301 003752 001413          BEQ  1$ ;YES
2302 003754 016705 175130          MOV  RSWC,R5 ;GET WC ADDRESS
2303 003760 011415          MOV  (R4),(R5) ;LOAD WC
2304 003762 021524          CMP  (R5),(R4)+ ;IS IT CORRECT?
2305 003764 001771          BEQ  4$ ;YES
2306 003766 017700 175116          MOV  @RSWC,BAD ;GET BAD WC
2307 003772 014401          MOV  -(R4),GOOD ;GET GOOD ANS
2308 003774 104000          HLT ;TYPE THEM OUT
2309 003776 005724          TST  (R4)+ ;UPDATE RANDOM NUMBER
2310 004000 000763          BR   4$ ;CONT
2311 004002 012704 024550          1$:  MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
2312 004006 012767 004014 174774 LOP2:  MOV #LOP2,LAD ;SETUP LOOP ADDR
2313 004014 012703 001000          MOV #1000,R3 ;SETUP TEST COUNTER
2314 004020 005303          3$:  DEC  R3 ;DONE YET?
2315 004022 001413          BEQ  1$ ;YES
2316 004024 016705 175064          MOV  RSDA,R5 ;LOAD DA ADDRESS INTO R5
2317 004030 011415          MOV  (R4),(R5) ;LOAD DA
2318 004032 021524          CMP  (R5),(R4)+ ;IS IT CORRECT?
2319 004034 001771          BEQ  3$ ;YES
2320 004036 017700 175052          MOV  @RSDA,BAD ;GET BAD DATA
2321 004042 014401          MOV  -(R4),GOOD ;GET GOOD DATA
2322 004044 104000          HLT ;TYPE IT OUT
2323 004046 005724          TST  (R4)+ ;UPDATE RANDOM NUMBER
2324 004050 000763          BR   3$ ;CONTINUE
2325 004052 012704 024550          1$:  MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
2326 004056 012767 004064 174724 LOP3:  MOV #LOP3,LAD ;SETUP LOOP ADDR
2327 004064 012703 001000          MOV #1000,R3 ;SETUP TEST COUNTER
2328 004070 005303          3$:  DEC  R3 ;DONE YET?
2329 004072 001416          BEQ  2$ ;YES
2330 004074 016705 175012          MOV  RSBA,R5 ;LOAD ADDRESS OF BA INTO R5
2331 004100 011415          MOV  (R4),(R5) ;LOAD BA
2332 004102 042714 000001          BIC  #BIT0,(R4) ;CLEAR BIT 0
2333 004106 021514          CMP  (R5),(R4) ;IS IT CORRECT?
2334 004110 001767          BEQ  3$ ;YES
2335 004112 017700 174774          MOV  @RSBA,BAD ;GET BAD DATA
2336 004116 011401          MOV  (R4),GOOD ;GET GOOD DATA
2337 004120 104000          HLT ;TYPE IT OUT
2338 004122 000400          BR   1$ ;GET OUT
2339 004124 005724          1$:  TST  (R4)+ ;GET NEW NUMBER
2340 004126 000760          BR   3$ ;CONTINUE
2341 004130 000240          2$:  NOP

```

2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393

004132 104400
004134 104414
004136 012777 003566 174740
004144 112777 000005 174772
004152 022777 004766 174724
004160 001401
004162 104001
004164 112777 000032 174712
004172 022777 004632 174704
004200 001401
004202 104001
004204 104400
004206 016777 174744 174672
004214 052777 177400 174664
004222 105077 174720
004226 016701 174724
004232 052701 000100
004236 017700 174644
004242 020001
004244 001401
004246 104000
004250 104400
004252 012777 025252 174630
004260 112777 000377 174662
004266 022777 177652 174614
004274 001401
004276 104010
004300 112777 000123 174602
004306 022777 177523 174574
004314 001401
004316 104010
004320 104400
004322 012777 025252 174562
004330 112777 000377 174614
004336 022777 177652 174546
004344 001401
004346 104020
004350 112777 000125 174534
004356 022777 177524 174526
004364 001401
004366 104020
004370 104414

```
*****  
;TEST 31 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA  
*****  
TST31: SCOPE  
  
BITST: CLRDK ;CLEAR ALL RS REG  
MOV #3566,@RSCS1 ;LOAD CS1  
MOVB #5,@RSCS1B ;LOAD BIT  
CMP #4766,@RSCS1 ;DID IT LOAD?  
BEQ +4 ;YES  
HLT !CS1  
MOVB #32,@RSCS1  
CMP #4632,@RSCS1  
BEQ +4  
HLT !CS1 ;CS1 SHOULD = 4632  
  
TST32: SCOPE  
  
BITCS2: MOV UNNUM,@RSCS2 ;LOAD UNIT NUMBER  
BIS #177400,@RSCS2 ;LOAD ALL BITS  
CLRB @RSCS2B ;CLR UPPER BYTE  
MOV UNNUM,GOOD ;GET UNIT NO.  
BIS #100,GOOD ;SET OR BIT  
MOV @RSCS2,BAD ;GET CS2  
CMP BAD,GOOD ;IS CS2 CORRECT?  
BEQ +4 ;YES  
HLT ;LOAD BYTE DID NOT WORK  
  
TST33: SCOPE  
  
BITWC: MOV #25252,@RSWC ;LOAD WC  
MOVB #377,@RSWCB ;LOAD BIT  
CMP #177652,@RSWC ;DID IT LOAD?  
BEQ +4 ;YES  
HLT !WC ;NO WC SHOULD =177652  
MOVB #123,@RSWC  
CMP #177523,@RSWC  
BEQ +4  
HLT !WC ;WC SHOULD = 177523  
  
TST34: SCOPE  
  
BITBA: MOV #25252,@RSBA ;LOAD DA  
MOVB #377,@RSBAB ;LOAD BIT  
CMP #177652,@RSBA ;DID IT LOAD?  
BEQ +4 ;YES  
HLT !BA ;DA SHOULD =177652  
MOVB #125,@RSBA  
CMP #177524,@RSBA  
BEQ +4  
HLT !BA ;BA SHOULD = 177525  
CLRDK ;CLEAR ALL RS REG
```


B06

44:4000-11-DERSA-4
CERSA.P11 *5731

RS - REC3-4804 BASIC FUNCTION DIAGNOSTIC
*ES+ 000 5: E INSTRUCTIONS ON CS1, CS2, WC AND BA

MAC 11 27.732) 06-OCT-76 08:51 PAGE 67

2394	004372	104400
2395	004374	104414
2396	004376	005077
2397	004402	012777
2398	004410	012767
2399	004416	012701
2400	004422	056701
2401	004428	017700
2402	004434	020100
2403	004436	001404
2404	004438	005367
2405	004440	001371
2406	004442	104200
2407	004444	005201
2408	004446	017700
2409	004448	020100
2410	004450	001401
2411	004452	104600
2412	004454	012701
2413	004456	017700
2414	004458	020100
2415	004474	001401
2416	004476	104000

174524	174516
177777	174562
002000	
000300	
174530	
174454	
174536	
174452	
177777	
174434	

```

7ST35: SCOPE
ZERONE: CLRDK
CLR
MOV
MOV
MOV
BIS
MOV
CMP
BEQ
DEC
BNE
HLT
2$:
CLR
MOV
CMP
BEQ
HLT
3$:
CLR
MOV
CMP
BEQ
HLT

```

```

JRSDB
#177777, JRSDB
#2000, WORK
#300, GOOD
UNNUM, GOOD
JRSDB, BAD
GOOD, BAD
3$
WORK
2$
!CS2
GOOD
JRSDB, BAD
GOOD, BAD
.+4
HLT
#-1, GOOD
JRSDB, BAD
GOOD, BAD
.+4
HLT

```

```

: CLEAR ALL RS REG
: LOAD DB WITH ALL 0
: LOAD DB WITH ALL ONES
: TIME OUT ROUTINE
: GET CORRECT FOR CS2

: GET CS2
: IS IT CORRECT?
: YES
: TO WAIT FOR OR
: TO SET
: OR SHOULD BE SET

: LOAD BAD WITH DB
: IS BAD CORRECT
: YES
: COULD NOT FLOAT 0 THROUGH DB
: LOAD GOOD WITH ANS
: GET DATA FROM DB
: IS DB CORRECT
: YES
: BAD SHOULD = 177777

```

1
2
3
4
5
6
7
8
9
10
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

004500	104400		
004502	104414		
004504	012767	177777	020036
004512	013777	001102	174372
004520	012777	177777	174352
004526	012777	000001	174350
004534	032777	000001	174342
004542	001401		
004544	104001		
004546	005777	174346	
004552	001401		
004554	104002		
004556	022777	177777	174324
004564	001401		
004566	104010		
004570	005777	174320	
004574	001401		
004576	104004		
004600	023777	001102	174304
004606	001401		
004610	104020		
004612	036777	174344	174302
004620	001401		
004622	104100		

```

:*****
:TEST 36 TEST NO-OP FUNCTION
:*****
TST36: SCOPE

NOOP: CLRDK ;CLEAR ALL RS REG
MOV #177777,OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #-1,@RSCW ;LOAD WC WITH -1
MOV #1,@RSCSI ;DO NO-OP FUNCTION
BIT #1,@RSCSI ;DID GO BIT CLEAR
BEQ .+4 ;YES
HLT ;CS1 ;GO BIT SHOULD BE CLEARED
TST @RSER ;DID ANY ERRORS OCCUR?
BEQ .+4 ;NO
HLT ;ER ;ALL ERROR BITS SHOULD BE 0
CMP #-1,@RSCW ;DID WC MOVE?
BEQ .+4 ;NO
HLT ;WC ;WC SHOULD = 1777777
TST @RSDA ;DID DA MOV
BEQ .+4 ;NO
HLT ;DA ;DA SHOULD =0
CMP @#OBUFSV,@RSBA ;DID BA MOVE
BEQ .+4 ;NO
HLT ;BA ;BA MOVED
BIT UNCMP,@RSAS ;AS SHOULD NOT SET ON
BEQ .+4 ;A NO-OP FUNCTION
HLT ;AS ;AS SET WHY?

```

2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479

004624 104400
004626 104414
004630 012777 000007 174262
004636 036777 174320 174256
004644 001001
004646 104100
004650 012767 177777 017672
004656 013777 001102 174226
004664 012777 177777 174216
004672 012777 000001 174204
004700 032777 000001 174176
004706 001401
004710 104001
004712 022777 150600 174176
004720 001401
004722 104040
004724 022777 177777 174156
004732 001401
004734 104010
004736 005777 174152
004742 001401
004744 104004
004746 023777 001102 174136
004754 001401
004756 104020
004760 036777 174176 174134
004766 001001
004770 104100
004772 022777 000007 174120
005000 001401
005002 104002

:TEST 37 TEST NO-OP FUNCTION WITH ERROR BITS SET

TST37: SCOPE

MNOOP: CLRDK ;CLEAR ALL RS REG
MOV #7, @RSER ;LOAD ER
BIT UNCMP, @RSAS ;IS ATA BIT SET?
BNE +4 ;YES
HLT !AS ;AS BIT SHOULD BE SET
MOV #177777, OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #-1, @RSWC ;LOAD WC WITH -1
MOV #1, @RSCSI ;DO NO-OP FUNCTION
BIT #1, @RSCSI ;DID GO BIT CLEAR
BEQ +4 ;YES
HLT !CSI ;GO BIT SHOULD BE CLEARED
CMP #150600, @RSDS ;DID ERR BITS SET?
BEQ +4 ;NO
HLT !DS ;ERR BIT SHOULD BE 0
CMP #-1, @RSWC ;DID WC MOVE?
BEQ +4 ;NO
HLT !WC ;WC SHOULD = 177777
TST @RSDA ;DID DA MOV
BEQ +4 ;NO
HLT !DA ;DA SHOULD =0
CMP @#OBUFSV, @RSBA ;DID BA MOVE
BEQ +4 ;NO
HLT !BA ;BA MOVED
BIT UNCMP, @RSAS ;AS SHOULD BE SET
BNE +4 ;IS IT?
HLT !AS ;NO
CMP #7, @RSER ;DID ER CHANGE?
BEQ +4 ;NO
HLT !ER ;ER SHOULD NOT CHANGE

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140

005004 104400
005006 104414
005010 012767 177777 017532
005016 013777 001102 174066
005024 012777 177777 174056
005032 012777 000021 174044
005040 032777 000001 174036
005046 001401
005050 104001
005052 005777 174042
005056 001401
005060 104002
005062 022777 177777 174020
005070 001401
005072 104010
005074 005777 174014
005100 001401
005102 104004
005104 023777 001102 174000
005112 001401
005114 104020
005116 036777 174040 173776
005124 001401
005126 104100
005130 022777 004220 173746
005136 001401
005140 104040

```
*****  
:TEST 40 TEST NO-OP FUNCTION CODE 21  
*****  
TST40: SCOPE  
  
NOOP21: CLRDK  
MOV #177777,OUTBUF ;CLEAR ALL RS REG  
MOV @#0BUFSV,@RSBA ;DATA TO BE XFERED  
MOV #-1,@RSWC ;SET UP CURRENT ADDRESS  
MOV #21,@RSCS1 ;LOAD WC WITH -1  
BIT #1,@RSCS1 ;DO NO-OP FUNCTION  
BEQ .+4 ;DID GO BIT CLEAR  
HLT :CS1 ;YES  
TST @RSER ;GO BIT SHOULD BE CLEARED  
BEQ .+4 ;DID ANY ERRORS OCCUR?  
HLT :ER ;NO  
CMP #-1,@RSWC ;ALL ERROR BITS SHOULD BE 0  
BEQ .+4 ;DID WC MOVE?  
HLT :WC ;NO  
TST @RSDA ;WC SHOULD = 177777  
BEQ .+4 ;DID DA MOV  
HLT :DA ;NO  
CMP @#0BUFSV,@RSBA ;DA SHOULD =0  
BEQ .+4 ;DID BA MOVE  
HLT :BA ;NO  
BIT UNCMP,@RSAS ;BA MOVED  
BEQ .+4 ;AS SHOULD NOT SET ON  
HLT :AS ;A NO-OP FUNCTION  
CMP #4220,@RSCS1 ;AS SET WHY?  
BEQ .+4 ;IS CS1 CORRECT?  
HLT :DS ;YES  
 ;CS1 SHOULD = 4220
```

```

2511
2512
2513
2514 005142 104400
2515
2516 005144 104414
2517 005146 012777 000007 173744
2518 005148 036777 174002 173740
2519 005162 001001
2520 005164 104100
2521 005166 012767 177777 017354
2522 005174 013777 001102 173710
2523 005202 012777 177777 173700
2524 005210 012777 000021 173666
2525 005216 032777 000001 173660
2526 005224 001401
2527 005226 104001
2528 005230 022777 150600 173660
2529 005236 001401
2530 005240 104040
2531 005242 022777 177777 173640
2532 005250 001401
2533 005254 104010
2534 005254 005777 173634
2535 005260 001401
2536 005262 104004
2537 005264 023777 001102 173620
2538 005272 001401
2539 005274 104020
2540 005276 036777 173660 173616
2541 005304 001001
2542 005306 104100
2543 005310 022777 000007 173602
2544 005316 001401
2545 005320 104002
2546 005322 022777 104220 173554
2547 005330 001401
2548 005332 104040

```

```

:*****
:TEST 41 TEST NO-OP FUNCTION CODE 21 WITH ERROR BITS SET
:*****
TST41: SCOPE

NNOP21: CLRDK
MOV #7,RSER ;CLEAR ALL RS REG
BIT UNCMP,RSAS ;LOAD ER
BNE .+4 ;IS ATA BIT SET?
HLT !AS ;YES
MOV #177777,OUTBUF ;AS BIT SHOULD BE SET
MOV #0BUFSV,RSBA ;DATA TO BE XFERED
MOV #-1,RSWC ;SET UP CURRENT ADDRESS
MOV #21,RS1 ;LOAD WC WITH -1
BIT #1,RS1 ;DO NO-OP FUNCTION
BEQ .+4 ;DID GO BIT CLEAR
HLT !CS1 ;YES
CMP #150600,RS1 ;GO BIT SHOULD BE CLEARED
BEQ .+4 ;DID ERR BITS SET?
HLT !DS ;NO
CMP #-1,RSWC ;ERR BIT SHOULD BE 0
BEQ .+4 ;DID WC MOVE?
HLT !WC ;NO
TST RSDA ;WC SHOULD = 177777
BEQ .+4 ;DID DA MOV
HLT !DA ;NO
CMP #0BUFSV,RSBA ;DA SHOULD =0
BEQ .+4 ;DID BA MOVE
HLT !BA ;NO
BIT UNCMP,RSAS ;BA MOVED
BNE .+4 ;AS SHOULD BE SET
HLT !AS ;IS IT?
CMP #7,RSER ;NO
BEQ .+4 ;DID ER CHANGE?
HLT !ER ;NO
CMP #104220,RS1 ;ER SHOULD NOT CHANGE
BEQ .+4 ;IS CS1 CORRECT?
HLT !DS ;YES
;CS1 SHOULD = 104220

```

```

2549
2550
2551
2552 005334 104400
2553
2554
2555
2556 005336 104414
2557 005340 012777 177777 173542
2558 005346 012777 177777 173540
2559 005354 012777 177017 173536
2560 005362 012777 000070 173540
2561 005370 012777 000011 173506
2562 005376 017700 173504
2563 005402 042700 177640
2564 005406 016701 173544
2565 005412 052701 000100
2566 005416 020100
2567 005420 001401
2568 005422 104000
2569 005424 005777 173464
2570 005430 001401
2571 005432 104004
2572 005434 005777 173460
2573 005440 001401
2574 005442 104002
2575 005444 017767 173460 173526
2576 005452 042767 177707 173520
2577 005460 022767 000070 173512
2578 005466 001401
2579 005470 104220
2580 005472 022777 004210 173404
2581 005500 001401
2582 005502 104001
2583 005504 036777 173452 173410
2584 005512 001401
2585 005514 104100
2586 005516 022777 177777 173364
2587 005524 001401
2588 005526 104010

```

```

:*****
:TEST 42 TEST DRIVE CLEAR FUNCTION WITH ERRORS SET
:*****
↑ST42: SCOPE
:FIRST SET ALL R/W BITS IN DISK REG
:DO DRIVE CLEAR-ALL R/W BITS SHOULD BE CLEARED

DRCLR: CLRDK ;CLEAR ALL RS REG
MOV #177777,@RSWC ;LOAD RSWC
MOV #177777,@RSDA ;SET ALL POSSIBLE
MOV #177017,@RSER ;BITS IN DISK REG
MOV #70,@RSMR ;SET THESE BITS
MOV #11,@RSCS1 ;SET DRIVE CLEAR
MOV @RSCS2,BAD ;GET CS2 DATA
BIC #177640,BAD ;CLEAR JUNK
MOV UNNUM,GOOD ;GET DRIVE UNIT
BIS #100,GOOD ;SET IR BIT
CMP GOOD,BAD ;IS UNIT # THE SAME
BEQ .+4 ;YES
HLT ;UNIT # IN CS2 GOT MODIFIED
TST @RSDA ;DID DA CLEAR
BEQ .+4 ;YES
HLT ;DA SHOULD BE 0
TST @RSER ;DID ER CLEAR
BEQ .+4 ;YES
HLT ;ER SHOULD BE CLEARED
MOV @RSMR,WORK ;GET MR REG
BIC #177707,WORK ;CLEAR JUNK
CMP #70,WORK ;IS 70 STILL SET IN MR?
BEQ .+4 ;YES
HLT ;BITS 70 SHOULD NOT CLEAR
CMP #4210,@RSCS1 ;DID THESE BITS CLEAR?
BEQ .+4 ;YES
HLT ;CS1 SHOULD =4210
BIT UNCMP,@RSAS ;AS SHOULD NOT SET
BEQ .+4 ;ON A DRIVE CLEAR FUN
HLT ;WHY DID AS SET?
CMP #177777,@RSWC ;DID RSWC CHANGE?
BEQ .+4 ;NO
HLT !WC

```

```

2589 ;DO ONE WORD WRITE
2590 ;*****
2591 ;TEST 43 EXECUTE THE ONE WORD WRITE
2592 ;*****
2593 TST43: SCOPE
2594
2595 WRTST: CLRDK ;CLEAR ALL RS REG
2596 005530 104400 ;DATA TO BE X-FERED
2597 005532 104414 ;SET UP CURRENT ADDRESS
2598 005534 012767 177777 017006 ;SET WORD COUNT TO -1
2599 005537 013777 001102 173342 ;SET FUNCTION WITH NO GO BIT
2600 005542 013777 177777 173332 ;DID WC MOVE?
2601 005550 012777 177777 173320 ;NO
2602 005556 012777 000060 173316 ;WC MOVED
2603 005564 022777 177777 173316 ;DID RSBA MOVE?
2604 005572 001401 ;NO
2605 005574 104010 ;BA MOVED
2606 005576 023777 001102 173306 ;SET GO BIT
2607 005604 001401 ;TEST FOR RDY=0
2608 005606 104020 ;RDY=0
2609 005610 052777 000001 173266 ;RDY SHOULD = 0
2610 005616 105777 173262 25: ;WAIT FOR READY
2611 005622 100001 ;SHOULD = 260 RDY NEVER CAME UP
2612 005624 104001 ;IS RSBA CORRECT
2613 005626 004767 016424 ;RSBA OK
2614 005632 104001 ;SHOULD = 1 SHOULD INCREMENT
2615 005634 022777 000001 173252 ;IS ERROR FLAG SET?
2616 005642 001401 ;NO! X-FER OK
2617 005644 104004 ;ERROR DURING X-FER
2618 005646 022777 004260 173230 35: ;FETCH WORD COUNT
2619 005654 001401 ;WORD COUNT DID OVERFLOW
2620 005656 104047 ;SHOULD = 0 FAILED TO INCREMENT
2621 005660 005777 173224 45: ;IS RSBA OK?
2622 005664 001401 ;YES
2623 005666 104010 ;NO
2624 005670 022777 010600 173220 ;GET UNIT #
2625 005676 001401 ;SET IR BIT
2626 005700 104044 ;GET CS2
2627 005702 016701 173250 ;IS CS2 CORRECT?
2628 005706 052701 000100 ;YES
2629 005712 017709 173170 ;BAD = CS2 GOOD IS CORRECT ANS
2630 005716 020100 ;GET BA DATA
2631 005720 001401 ;WHAT RSBA SHOULD EQUAL
2632 005722 104000 ;UPDATE OUTBUFFER
2633 005724 017700 173162 ;IS RSBA CORRECT
2634 005726 013701 001102 ;YES
2635 005730 062701 000002 ;BA FAILED TO INCREMENT
2636 005734 020100 ;DID ANY ERRORS SET?
2637 005742 001401 ;NO
2638 005744 104000 ;NO
2639 005746 005777 173146 ;NO
2640 005752 001401 ;NO
2641 005754 104040 ;NO

```

```

;TEST READ FUNCTION
*****
:TEST 44 EXECUTE THE ONE WORD READ
*****
TST44: SCOPE
RD7ST: CLRDK ;CLEAR ALL RS REG
CLR ;CLR TO READ INTO
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #-1,@RSCW ;SET WORD COUNT TO -1
1$: MOV #71,@RSCS1 ;GO READ
2$: TSTB @RSCS1 ;TEST FOR BUSY=1
BPL .+4 ;BUSY SET
HLT !CS1 ;BUSY NOT SET
JSR PC,WAITRY ;WAIT FOR READY
HLT !CS1 ;TIMEOUT RDY DID NOT SET
CMP #BITO,@RSDA ;WAS RSDA INCREMENTED BY 1
BEQ .+4 ;RSDA OK
HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
3$: CMP #4270,@RSCS1 ;IS ERROR FLAG SET?
BEQ .+4 ;NO! X-FER OK
HLT !CS1!ER!DS ;RSCS1 SHOULD = 270
4$: TST @RSCW ;TEST WC
BEQ .+4 ;WORD COUNT DID OVERFLOW
HLT !WC ;SHOULD = 0
MOV UNNUM,GOOD ;GET CORRECT
BIS #100,GOOD ;ANS OF CS2
MOV @RSCS2,BAD ;GET CS2
CMP GOOD,BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
MOV @RSBA,BAD ;GOOD = CORRECT ANS FOR CS2
MOV @#OBUFSV,GOOD ;FETCH CURRENT ADDRESS
ADD #2,GOOD ;WHAT RSBA SHOULD EQUAL
CMP BAD,GOOD ;UPDATE IT
BEQ .+4 ;IS RSBA CORRECT
HLT ;YES EXECUTE CONTINUE
MOV OUTBUF,BAD ;RSBA FAILED TO INCREMENT
MOV #-1,GOOD ;GET DATA READ FROM DISK
CMP GOOD,BAD ;GET CORRECT ANS
BEQ .+4 ;IS OUTBUF CORRECT
HLT ;YES
;GOOD=CORRECT ANS BAD=DATA READ FROM DISK

```



```

2682
2683
2684
2685 006144 104400
2686
2687
2688
2689
2690 006146 104414
2691 006150 012767 177777 016372
2692 006156 013777 001102 172726
2693 006164 012777 177777 172716
2694 006172 012777 000051 172704
2695 006200 105777 172700
2696 006204 100001
2697 006206 104001
2698 006210 004767 016042
2699 006214 104001
2700 006216 016701 172734
2701 006222 052701 000100
2702 006226 017700 172654
2703 006232 020100
2704 006234 001401
2705 006236 104000
2706 006240 022777 004250 172636 2$:
2707 006246 001401
2708 006250 104046
2709 006252 022777 000001 172634 3$:
2710 006260 001401
2711 006262 104004
2712 006264 005777 172620
2713 006270 001401
2714 006272 104010
2715 006274 017700 172612
2716 006300 013701 001102
2717 006304 062701 000002
2718 006310 020001
2719 006312 001401
2720 006314 104000

```

```

*****
:TEST 45 TEST WRITE CHECK
*****
†ST45: SCOPE
;DO A ONE WORD WRITE CHECK

;* * *EXECUTE THE ONE WORD WRITE CHECK* * *

WRCKT: CLRDK ;CLEAR ALL RS REG
MOV #177777,OUTBUF ;DATA TO BE X-FERED
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #-1,@RSWC ;SET WORD COUNT TO -1
1$: MOV #51,@RSCS1 ;GO WRITE CHECK
2$: TSTB @RSCS1 ;TEST FOR READY
BPL .+4 ;NOT READY
HLT !CS1 ;BUSY FAILED TO SET
RSWCWT: JSR PC,WAITRY ;WAIT FOR READY
HLT !CS1 ;BUSY FAILED TO CLEAR
MOV UNNUM,GOOD ;GET UNIT #
BIS #100,GOOD ;SET BIT IR
MOV @RSCS2,BAD ;GET CS2
CMP GOOD,BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT ;GOOD = CORRECT ANS FOR CS2
2$: CMP #4250,@RSCS1 ;ANY ERRORS?
BEQ .+4 ;X-FER OK
HLT !DA!ER!DS ;ERROR DUR X-FER
3$: CMP #BIT0,@RSDA ;WAS DAR INCREMENTED BY 1
BEQ .+4 ;RSDA OK
HLT !DA ;DAR SHOULD = 1
TST @RSWC ;TEST FOR OVERFLOW
BEQ .+4 ;WORD COUNT DID OVERFLOW
HLT !WC ;SHOULD = 0
MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
ADD #2,GOOD ;UPDATE IT
CMP BAD,GOOD ;IS RSBA CORRECT
BEQ .+4 ;YES EXECUTE CONTINUE
HLT ;RSBA FAILED TO INCREMENT

```

```

2721 006316 016767 172560 172650 NXM:  MOV  OBUFSV,BPORTT ;SAVE -B- PORT BUFFER
2722 006324 012767 024550 172550  MOV  #OUTBUF, OBUFSV ;RESTORE OBUFSV
2723
2724 ;DESELECT THEN SELECT UNIT NUMBER IN RSCS2 CHECK TIMING
2725 ;*****
2726 ;TEST 46 DESELECT THEN SELECT UNIT NUMBER TIMING TEST
2727 ;*****
2728 006332 104400 †ST46: SCOPE
2729
2730 006334 104414 UNITST: CLRDK ;CLEAR ALL RS REG
2731 006336 005004 CLR R4 ;CLEAR R4
2732 006340 020467 172612 CMP R4,UNNUM ;IS THIS CORRECT UNIT #?
2733 006344 001001 3$ 3$ NO THEN USE IT
2734 006346 005204 INC R4 ;GET WRONG DRIVE
2735 006350 012767 177777 016172 3$ MOV #177777,OUTBUF ;DATA TO BE X-FERED
2736 006356 013777 001102 172526 MOV #OBUFSV,RSBA ;SET UP CURRENT ADDRESS
2737 006364 012777 177777 172516 MOV #-1,RSWC ;SET WORD COUNT TO -1
2738 006372 012703 000061 MOV #61,R3 ;GET WRITE FUNCTION
2739 006376 016705 172554 MOV UNNUM,RS ;GET CORRECT UNIT #
2740 006402 012701 172040 MOV #172040,R1 ;GET CS1 REG
2741 006406 010461 000010 MOV R4,10(R1) ;LOAD WRONG UNIT # INTO CS2
2742 006412 000240 NOP ;WAIT FOR DRIVE TO SETTLE
2743 006414 010561 000010 MOV R5,10(R1) ;LOAD CORRECT UNIT #
2744 006420 010311 1$ MOV R3,(R1) ;LOAD FUNCTION IN CS1
2745 006422 004767 015630 JSR PC,WAITRY ;WAIT FOR READY
2746 006426 104001 HLT ;CS1 ;SHOULD = 260 RDY NEVER CAME UP
2747 006430 022777 004260 172446 CMP #4260,RSCS1 ;IS ERROR FLAG SET?
2748 006436 001401 BEQ .+4 ;NO! X-FER OK
2749 006440 104047 HLT ;CS1!ER!DS!DA ;ERROR DURING X-FER
2750 006442 022777 000001 172444 CMP #1,RSDA ;IS RSDA CORRECT
2751 006450 001401 BEQ .+4 ;RSDA OK
2752 006452 104004 HLT ;DA ;SHOULD = 1 SHOULD INCREMENT
2753 006454 005777 172430 TST RSWC ;FETCH WORD COUNT
2754 006460 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
2755 006462 104010 HLT ;WC ;SHOULD = 0 FAILED TO INCREMENT
2756 006464 022777 010600 172424 CMP #10600,RSDS ;IS RSDS OK?
2757 006472 001401 BEQ .+4 ;YES
2758 006474 104044 HLT ;DS!DA ;NO
2759 006476 016701 172454 MOV UNNUM,GOOD ;GET UNIT #
2760 006502 052701 000100 BIS #100,GOOD ;SET IR BIT
2761 006506 017700 172374 MOV RSCS2,BAD ;GET CS2
2762 006512 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2763 006514 001401 BEQ .+4 ;YES
2764 006516 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
2765 006520 017700 172366 MOV RRSBA,BAD ;GET BA DATA
2766 006524 013701 001102 MOV #OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
2767 006530 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
2768 006534 020100 CMP GOOD,BAD ;IS RSBA CORRECT
2769 006536 001401 BEQ .+4 ;YES
2770 006540 104000 HLT ;BA FAILED TO INCREMENT

```

2771
2772
2773
2774
2775
2776
2777
2778 006542 104400
2779
2780 006544 104414
2781 006546 013777 001102 172336
2782 006554 012777 177777 172326
2783 006562 052777 000010 172316
2784 006570 012777 000061 172306
2785 006576 004767 015454
2786 006602 104001
2787 006604 013701 001102
2788 006610 017700 172276
2789 006614 020100
2790 006616 001401
2791 006620 104000
2792 006622 005777 172272
2793 006626 001401
2794 006630 104040
2795 006632 104414
2796 006634 032777 000010 172244
2797 006642 001401
2798 006644 104002

:TEST CURRENT ADDRESS INHIBIT-BAI IN RSCS2
:DO A ONE WORD WRITE AND SEE
:IF RSBA INCREMENTED AFTER THE X-FER

:*****
:TEST 47 TEST BAI IN RSCS2
:*****
TST47: SCOPE

BAITST: CLRDK		:CLEAR ALL RS REG
MOV @#OBUFSV,@RSBA		:SET UP CURRENT ADDR
MOV #-1,@RSMC		:SET WORD COUNT TO -1
BIS #BAI,@RSCS2		:SET BAI BIT
MOV #61,@RSCS1		:WRITE
JSR PC,WAITRY		:WAIT FOR READY
HLT !CS1		:RDY DID NOT SET
IS: MOV @#OBUFSV,GOOD		:WHAT RSBA SHOULD BE
MOV @RSBA,BAD		:WHAT RSBA IS
CMP GOOD,BAD		:COMPARE
BEQ .+4		:YES
HLT		:BAD=OUTBUF GOOD = CORRECT ANS
TST @RSER		:ANY ERRORS?
BEQ .+4		:NO
HLT !DS		:YES
CLRDK		:CLEAR ALL RS REG
BIT #BAI,@RSCS2		:DID BAI CLEAR?
BEQ .+4		:YES
HLT !ER		:BAI DID NOT SET

```

2799
2800
2801
2802 006646 104400
2803
2804 006650 104414
2805 006652 052777 000010 172226
2806 006660 012777 177777 172222
2807 006666 012777 173000 172216
2808 006674 012777 000077 172232
2809 006702 012777 001471 172174
2810 006710 004767 015342
2811 006714 104040
2812 006716 016701 172234
2813 006722 052701 004310
2814 006726 017700 172154
2815 006732 020100
2816 006734 001401
2817 006736 104000
2818 006740 022777 145670 172136
2819 006746 001401
2820 006750 104001
2821 006752 012777 040000 172124
2822 006760 017700 172122
2823 006764 016701 172166
2824 006770 052701 000310
2825 006774 020100
2826 006776 001401
2827 007000 104200
2828
2829
2830
2831
2832 007002 104400
2833
2834 007004 104414
2835 007006 012777 000032 172100
2836 007014 013777 001102 172070
2837 007022 012777 177777 172060
2838 007030 005067 172144
2839 007034 032777 001000 172066
2840 007042 001004
2841 007044 005367 172130
2842 007050 001371
2843 007052 104220

```

```

;*****
;TEST 50 TEST NON-EXISTENT MEMORY ERROR BIT IN CS2
;*****
TST50: SCOPE

```

```

NXMTSM: CLRDK ;CLEAR ALL RS REG
BIS #BAI, ARSCS2 ;SET BAI BIT
MOV #-1, ARSWC ;SET UP WORD COUNT
MOV #173000, ARSBA ;SET UP CURRENT ADDRESS
MOV #77, ARSBAE ;AND EXTENDED BITS
MOV #1471, ARSCS1 ;READ AND LOAD A16 +A17 FOR 18 BIT ADDRESS
JSR PC, WAITRY ;WAIT FOR READY
HLT !DS ;READY NEVER CAME UP
TSTNEM: MOV UNNUM, GOOD ;GET UNIT NO.
BIS #4310, GOOD ;SET BAI+OR BITS
MOV ARSCS2, BAD ;GET CS2
CMP GOOD, BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT ;BAD=CS2 GOOD=CORRECT ANS FOR CS2
CMP #145670, ARSCS1 ;DID TRE SET?
BEQ .+4 ;YES
HLT ;TRE SHOULD SET BECAUSE OF NEM
MOV #TRE, ARSCS1 ;CLEAR TRE
MOV ARSCS2, BAD ;GET CS2
MOV UNNUM, GOOD ;GET DRIVE
BIS #310, GOOD ;SET IR
CMP GOOD, BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT !CS2 ;CS2-BAD GOOD IS CORRECT ANS FOR CS2

```

```

;*****
;TEST 51 TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND ADDR. CONF BIT
;*****
TST51: SCOPE

```

```

BLOCK: CLRDK ;CLEAR ALL RS REG
MOV #32, ARSDA ;DO A SEARCH FOR SECTOR 32
MOV #0BUFSV, ARSBA ;LOAD REGS. TO MAKE
MOV #-1, ARSWC ;SURE THEY DO NOT CHANGE
CLR WORK ;SETUP FOR TIMEOUT ROUTINE
45: BIT #1000, ARSMR ;WAIT FOR DISK TO
BNE 35 ;REACH SECTOR 32
DEC WORK ;TIME OUT
BNE 45 ;ROUTINE
HLT !MR ;COULD NOT FIND SECTOR 32

```

2844	007054	005077	172034		3\$: CLR	DRSDA	;NOW SEARCH FOR 0
2845	007060	012777	000031	172016	MOV	#31,DRSCS1	;DO A BLOCK SEARCH FUNCTION
2846	007066	032777	000200	172022	BIT	#DRY,DRSDS	;IS DRY CLEARED?
2847	007074	001402			BEQ	1\$;YES
2848	007076	104040			HLT	!DS	;DRY SHOULD BE CLEARED DURING A BLOCK SEARCH
2849	007100	000500			BR	0OUT	;GET OUT BECAUSE OF TIMING
2850	007102	032777	020000	172006	1\$: BIT	#20000,DRSDS	;IS PIP SET?
2851	007110	001001			BNE	.+4	;YES
2852	007112	104040			HLT	!DS	;PIP SHOULD BE SET
2853	007114	012701	020000		MOV	#20000,GOOD	;SETUP FOR TIMEOUT ROUTINE
2854	007120	005301			2\$: DEC	GOOD	;DO TIMEOUT
2855	007122	001466			BEQ	TTMOUT	;TIMED OUT
2856	007124	032777	000200	171764	BIT	#DRY,DRSDS	;DID DRY SET?
2857	007132	001772			BEQ	2\$;NO
2858	007134	022777	110600	171754	CMP	#110600,DRSDS	;DID PIP CLEAR?
2859	007142	001401			BEQ	.+4	;YES
2860	007144	104040			HLT	!DS	;PIP BIT DID NOT CLEAR
2861	007146	022777	104230	171730	CMP	#104230,DRSCS1	;DID SC SET?
2862	007154	001401			BEQ	.+4	;YES
2863	007156	104041			HLT	!CS1!DS	;SC DID NOT SET
2864	007160	016767	171772	172012	MOV	UNNUM,WORK	;GET CORRECT AS BIT
2865	007166	005001			CLR	GOOD	;IN RSAS REG
2866	007170	000261			SEC		;THAT SHOULD
2867	007172	006101			5\$: ROL	GOOD	;BE SET
2868	007174	005767	172000		TST	WORK	
2869	007200	001403			BEQ	6\$	
2870	007202	005367	171772		DEC	WORK	
2871	007206	000771			BR	5\$	
2872	007210	020177	171706		6\$: CMP	GOOD,DRSAS	;IS RSAS CORRECT?
2873	007214	001403			BEQ	7\$;YES
2874	007216	017700	171700		MOV	DRSAS,BAD	;NO
2875	007222	104000			HLT		
2876	007224	010177	171672		7\$: MOV	GOOD,DRSAS	;CLEAR AS REG
2877	007230	005777	171666		TST	DRSAS	;DID IT CLEAR?
2878	007234	001401			BEQ	.+4	;YES
2879	007236	104100			HLT	!AS	;NO
2880	007240	022777	010600	171650	CMP	#10600,DRSDS	;DID ATA CLEAR?
2881	007246	001401			BEQ	.+4	;YES
2882	007250	104040			HLT	!DS	;NO
2883	007252	023777	001102	171632	CMP	#0BUFSV,DRSBA	;DID BA MOVE?
2884	007254	001401			BEQ	.+4	;NO
2885	U	104021			HLT	!CS1!BA	;BA MOVED WHY?
2886	007264	022777	177777	171616	CMP	#-1,DRSWC	;DID WC MOVE?
2887	007272	001401			BEQ	.+4	;NO
2888	007274	104010			HLT	!WC	;WC MOVED WHY?
2889	007276	000401			BR	0OUT	;DONE GET OUT
2890	007300	104040			TTMOUT: HLT	!DS	;DYR NEVER CAME UP
2891	007302				0OUT:		;DONE

2892
2893
2894
2895 007302 104420
2896
2897
2898
2899
2900
2901
2902 007304 016767 011156 171656
2903 007312 012767 000010 011146
2904 007320 104414
2905 007322 012703 024464
2906 007326 012300
2907 007330 001513
2908 007332 013777 001102 171552
2909 007340 012777 177777 171542
2910 007346 010077 171532
2911 007352 042700 000001
2912 007356 010001
2913 007360 105777 171520
2914 007364 100375
2915 007366 052701 104200
2916 007372 017700 171506
2917 007376 020100
2918 007400 001401
2919 007402 104000
2920 007404 022777 000001 171506
2921 007412 001401
2922 007414 104043
2923 007416 022777 150600 171472
2924 007424 001401
2925 007426 104043
2926 007430 017700 171452
2927 007434 016701 171516
2928 007440 052701 000100
2929 007444 020100
2930 007446 001401
2931 007450 104000
2932 007452 016701 171504
2933 007456 042701 177400
2934 007462 017700 171434
2935 007466 020001
2936 007470 001401
2937 007472 104100

:TEST 52 ILLEGAL FUNCTION CODE TEST CODE 3 TO 51

TST52: SCOPE

:TEST ILF BIT IN RSER AND ERR BIT IN RSDS
:ALSO CHECKS TO SEE IF WC, BA, OR DA GOT MODIFIED
:IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
:FUNCTION INTO LOCATION ILLTAB: AND 0 IN FOLLOWING LOCATION

```

ILL51:  MOV      TIMES,TIMSV      :SAVE LOOP COUNT
        MOV      #10,TIMES      :LOOP TEN TIMES
        CLDRK                    :CLEAR ALL RS REG
1$:     MOV      #ILLTAB,R3      :GET STARTING ADD OF TABLE
3$:     MOV      (R3)+,BAD      :GET ILL FUN
        BEQ      ILFON          :DONE GET OUT
        MOV      #0BUFV,RSBA    :SET UP REGS.
        MOV      #-1,RSWC      :TO CHECK FOR CHANGE
2$:     MOV      BAD,RSCS1      :DO ILLEGAL FUNCTION
        BIC      #BIT0,BAD      :CLEAR GC BIT
        MOV      BAD,GOOD      :MOV ILLEGAL FUN INTO GOOD
6$:     TSTB     RSCS1          :RDY SET?
        BPL     6$              :NO
        BIS     #104200,GOOD    :SET ERROR BITS
4$:     MOV      RSCS1,BAD      :PUT CS1 INTO BAD
        CMP     GOOD,BAD        :IS CS1 CORRECT?
        BEQ     .+4             :YES
        HLT                    :GOOD IS WHAT CS1 SHOULD =BAD=CS1
        CMP     #1,RSER        :DID ILF SET?
        BEQ     .+4             :YES
        HLT                    :ILF DID NOT SET
        CMP     #150600,RSDS   :IS DS GOOD?
        BEQ     .+4             :YES
        HLT                    :EIPR DID NOT SET
        MOV     RSCS2,BAD      :GET CS2
        MOV     UNUM,GOOD      :GET UNIT #
        BIS     #100,GOOD      :SET IR BIT
        CMP     GOOD,BAD        :IS CS2 CORRECT?
        BEQ     .+4             :YES
        HLT                    :GOOD = CORRECT ANS FOR CS2
        MOV     UNCMP,GOOD     :GET CORRECT DRIVE
        BIC     #177400,GOOD    :CLEAR UNWANTED BITS
        MOV     RSAS,BAD      :GET RSAS REG
        CMP     BAD,GOOD        :DID CORRECT UNIT ANSWER?
        BEQ     .+4             :YES
        HLT                    :NO WRONG DRIVE ANSWERED
  
```

```

2938 007474 023777 001102 171410
2939 007502 001401
2940 007504 104021
2941 007506 022777 177777 171374
2942 007514 001401
2943 007516 104011
2944 007520 005777 171370
2945 007524 001401
2946 007526 104005
2947 007530 104414
2948 007532 005777 171362
2949 007536 001401
2950 007540 104040
2951 007542 022777 004200 171334
2952 007550 001401
2953 007552 104040
2954 007554 000167 177546
2955 007560
2956
2957
2958
2959
2960 007560 104414
2961 007562 012703 024526
2962 007564 012300
2963 007570 001402
2964 007572 000167 000422
2965 007576 013777 001102 171306
2966 007604 012777 177777 171276
2967 007612 010067 171364
2968 007616 042767 177707 171356
2969 007624 010077 171254
2970 007630 042700 000001
2971 007634 010001
2972 007636 105777 171242
2973 007642 100375
2974 007644 052701 144200
2975 007650 017700 171230
2976 007654 020100
2977 007656 001401
2978 007660 104000
2979 007662 022777 000001 171230
2980 007670 001401
2981 007672 104043
2982 007674 022777 150600 171214
2983 007702 001401
2984 007704 104043
2985 007706 005777 171202
2986 007712 001401
2987 007714 104005
2988 007716 022767 000060 171256
2989 007724 001430
2990 007726 022767 000050 171246
2991 007734 001455
2992
2993 007736 017700 171144

```

```

CMP      2#0BUFSV,2RSBA ;DID BA MOVE
BEQ      +4 ;NO
HLT      !CS1!BA ;BA MOVED ON AN ILLEGAL FUNCTION
CMP      #-1,2RSWC ;DID WC MOVE?
BEQ      +4 ;NO
HLT      !CS1!WC ;WC MOVED
TST      2RSDA ;DID DA MOVE
BEQ      +4 ;NO
HLT      !CS1!DA ;DA MOVED
CLROK    ;CLEAR ALL ERRORS
TST      2RSER ;DID ERRORS CLEAR
BEQ      +4 ;YES
HLT      !DS ;ILF DID NOT CLEAR
CMP      #4200,2RSCS1 ;DID ERRORS IN CS1 CLEAR
BEQ      +4 ;YES
HLT      !DS
J        3$ ;CONTINUE UNTIL DONE
;DONE WITH ILLEGAL FUNCTION TEST
;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
;FUNCTION INTO LOCATION ILFTB2: AND 0 IN FOLLOWING LOCATION
ILLFUN: CLROK ;CLEAR ALL RS REG
1$: MOV   #ILFTB2,R3 ;GET TABLE OF ILL FUNS.
3$: MOV   (R3)+,BAD ;GET ILL FUN
BEQ      2$ ;DO THIS FUNCTION
JMP      ILFDNE ;DONE GET OUT
2$: MOV   2#0BUFSV,2RSBA ;SET UP REGS.
MOV      #-1,2RSWC ;TO CHECK FOR CHANGE
MOV      BAD,WORK1 ;SHOULD WE TEST
BIC      #177707,WORK1 ;BA AND WC
MOV      BAD,2RSCS1 ;DO ILLEGAL FUNCTION
10$: BIC  #BIT0,BAD ;CLEAR GO BIT
MOV      BAD,GOOD ;MOV ILLEGAL FUN INTO GOOD
6$: TSTB 2RSCS1 ;RDY SET?
BPL      6$ ;NO
BIS      #144200,GOOD ;SET ERROR BITS
MOV      2RSCS1,BAD ;PUT CS1 INTO BAD
CMP      GOOD,BAD ;IS CS1 CORRECT?
BEQ      +4 ;YES
HLT      ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
CMP      #1,2RSER ;DID ILF SET?
BEQ      +4 ;YES
HLT      !CS1!ER!DS ;ILF DID NOT SET
CMP      #150600,2RSDS ;IS DS GOOD?
BEQ      +4 ;YES
HLT      !CS1!ER!DS ;ERR DID NOT SET
TST      2RSDA ;DID DA MOVE?
BEQ      +4 ;NO
HLT      !CS1!DA ;DA MOVED
CMP      #60,WORK1 ;IS THIS AN ILL WRITE FUN?
BEQ      11$ ;YES
CMP      #50,WORK1 ;IS THIS AN ILL WRITE CHECK FUN?
BEQ      7$ ;YES--BRANCH
;WORK1=70
MOV      2RSCS2,BAD ;GET CS2

```

2994	007742	016701	171210	MOV	UNNUM,GOOD	:GET UNIT #
2995	007746	052701	001100	BIS	#1100,GOOD	:SET IR BIT
2996	007752	020100		CMP	GOOD,BAD	:IS CS2 CORRECT?
2997	007754	001401		BEQ	+.4	:YES
2998	007756	104000		HLT		:GOOD = CORRECT ANS FOR CS2
2999	007760	023777	001102 171124	CMP	2#0BUFSV,2RSBA	:DID BA MOVE
3000	007766	001401		BEQ	+.4	:NO
3001	007770	104021		HLT	:CS1!BA	:BA MOVED ON AN ILLEGAL FUNCTION
3002	007772	022777	177777 171110	CMP	#-1,2RSWC	:DID WC MOVE?
3003	010000	001401		BEQ	+.4	:NO
3004	010002	104011		HLT	:CS1!WC	:WC MOVED
3005	010004	000471		BR	45	:CONTINUE UNTIL DONE
3006						
3007	010006	017700	171074	115: MOV	2RSCS2,BAD	:GET CS2
3008	010012	016701	171140	MOV	UNNUM,GOOD	:GET UNIT #
3009	010016	052701	001300	BIS	#1300,GOOD	:SET IR BIT
3010	010022	020100		CMP	GOOD,BAD	:IS CS2 CORRECT?
3011	010024	001401		BEQ	+.4	:YES
3012	010026	104000		HLT		:GOOD = CORRECT ANS FOR CS2
3013	010030	013767	001102 171142	MOV	2#0BUFSV,WORK	:GET BUFFER ADDR.
3014	010036	062767	000002 171134	ADD	#2,WORK	:UPDATE IT
3015	010044	026777	171130 171040	CMP	WORK,2RSBA	:DID BA MOVE
3016	010052	001401		BEQ	+.4	:YES
3017	010054	104021		HLT	:CS1!BA	:BA MOVED ON AN ILLEGAL FUNCTION
3018	010056	022777	000000 171024	CMP	#0,2RSWC	:DID WC MOVE?
3019	010064	001401		BEQ	+.4	:YES
3020	010066	104011		HLT	:CS1!WC	:BAD WC
3021						
3022	010070	017700	171012	75: MOV	2RSCS2,BAD	:GET CS2
3023	010074	016701	171056	MOV	UNNUM,GOOD	:GET UNIT #
3024	010100	052701	001300	BIS	#1300,GOOD	:SET IR BIT
3025	010104	020100		CMP	GOOD,BAD	:IS CS2 CORRECT?
3026	010106	001401		BEQ	+.4	:YES
3027	010110	104000		HLT		:GOOD = CORRECT ANS FOR CS2
3028	010112	013767	001102 171060	MOV	2#0BUFSV,WORK	:GET BUFFER ADDR.
3029	010120	062767	000002 171052	ADD	#2,WORK	:UPDATE IT
3030	010126	022763	000057 177776	CMP	#57,-2(R3)	:WRITE CHECK REV.?
3031	010134	001003		BNE	85	:NO--BRANCH
3032	010136	162767	000004 171034	SUB	#4,WORK	:YES--BUFFER ADDR. GOES BACKWARDS
3033	010144	026777	171030 170740	CMP	WORK,2RSBA	:DID BA MOVE
3034	010152	001401		BEQ	+.4	:YES
3035	010154	104021		HLT	:CS1!BA	:BA DIDN'T INCREMENT PROPERLY
3036	010156	022777	000000 170724	CMP	#0,2RSWC	:DID WC MOVE?
3037	010164	001401		BEQ	+.4	:YES
3038	010166	104011		HLT	:CS1!WC	:WC DIDN'T INCREMENT PROPERLY
3039	010170	104414		45: CLRCK		:CLEAR ALL ERRORS
3040	010172	022777	004200 170704	CMP	#4200,2RSCS1	:DID ERRORS CLEAR
3041	010200	001401		BEQ	+.4	:YES
3042	010202	104040		HLT	:DS	:NO
3043	010204	005777	170710	TST	2RSER	:DID ERROR CLEAR
3044	010210	001401		BEQ	+.4	:NO
3045	010212	104040		HLT	:DS	:YES
3046	010214	000167	177346	JMP	35	:CONTINUE UNTIL DONE
3047	010220					:DONE WITH ILLEGAL FUNCTION TEST
3048						:*****
3049						:TEST 53 TEST ILLEGAL FUNCTION CODE 67


```

*****
3050                                     TST53: SCOPE
3051 010220 104400
3052
3053 010222 104414 ILF67: CLRDK ;CLEAR ALL RS REG
3054 010224 012777 177777 170656 MOV #-1,RSWC ;SET WC TO -1
3055 010232 012767 024552 170740 MOV #<OUTBUF+2!2>,WORK ;GET OUTBUF ADD.
3056 010240 016777 170734 170644 MOV WORK,RSBA ;LOAD ADDR.
3057 010246 012777 000067 170630 MOV #67,RSCS1 ;DO FUNCTION 67
3058 010254 105777 170624 15: TSTB RSCS1 ;DONE YET?
3059 010260 100375 BPL 15 ;NO
3060 010262 017700 170624 MOV RSBA,BAD ;GET BA REG
3061 010266 012701 024550 MOV #<OUTBUF+28!C3>,GOOD ;GET CORRECT ANS FOR RSBA
3062 010272 020100 CMP GOOD,BAD ;IS RSBA CORRECT?
3063 010274 001401 BEQ .+4 ;YES
3064 010276 104000 HLT ;BAD=RSBA GOOD=CORRECT ANS.
3065 010300 016701 170652 MOV UNNUM,GOOD ;GET UNIT NUMBER
3066 010304 052701 001300 BIS #1300,GOOD ;SET IR AND OR BITS
3067 010310 017700 170572 MOV RSCS2,BAD ;GET CS2
3068 010314 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3069 010316 001401 BEQ .+4 ;YES
3070 010320 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
3071 010322 022777 000001 170570 CMP #1,RSER ;IS RSER CORRECT?
3072 010330 001401 BEQ .+4 ;YES
3073 010332 104002 HLT ;ER IS WRONG
3074 010334 104414 CLRDK ;CLEAR ALL RS REG
3075 010336 005777 170556 TST RSER ;DID ERROR CLEAR
3076 010342 001401 BEQ .+4 ;YES
3077 010344 104040 HLT ;NO
3078 010346 012777 177700 170534 MOV #-100,RSWC ;SET WC TO -100
3079 010354 012767 024566 170616 MOV #OUTBUF+16!2,WORK ;GET OUTBUF ADD.
3080 010362 016777 170612 170522 MOV WORK,RSBA ;LOAD ADDR
3081 010370 012777 000067 170506 MOV #67,RSCS1 ;DO FUNCTION 67
3082 010376 105777 170502 25: TSTB RSCS1 ;DONE YET?
3083 010402 100375 BPL 25 ;NO
3084 010404 012701 024546 MOV #24546,GOOD ;IF PROG. IS MODIFIED THIS
3085 ; NUMBER WILL HAVE TO BE CHANGED.
3086 010410 017700 170476 MOV RSBA,BAD ;GET BA REG
3087 010414 020100 CMP GOOD,BAD ;IS RSBA CORRECT?
3088 010416 001401 BEQ .+4 ;YES
3089 010420 104000 HLT ;BAD=RSBA GOOD=CORRECT ANS.
3090 010422 016701 170530 MOV UNNUM,GOOD ;GET UNIT NUMBER
3091 010426 052701 001200 BIS #1200,GOOD ;SET OR BIT
3092 010432 017700 170450 MOV RSCS2,BAD ;GET CS2
3093 010436 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3094 010440 001401 BEQ .+4 ;YES
3095 010442 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
3096 010444 022777 000001 170446 CMP #1,RSER ;IS RSER CORRECT?
3097 010452 001401 BEQ .+4 ;YES
3098 010454 104002 HLT ;ER IS WRONG
3099 010456 012777 040011 170420 MOV #40011,RSCS1 ;CLEAR ERRORS
3100 010464 022777 004210 170412 CMP #4210,RSCS1 ;DID THEY CLEAR IN CS1
3101 010472 001401 BEQ .+4 ;YES
3102 010474 104040 HLT ;NO
3103 010476 005777 170416 TST RSER ;DID RSER CLEAR
3104 010502 001401 BEQ .+4 ;YES
3105 010504 104040 HLT ;NO

```

F07

MAINDEC-11-DERSA-A
DERSAA.P11 TST53

RS11-R503-R504 BASIC FUNCTION DIAGNOSTIC
TEST ILLEGAL FUNCTION CODE 67

MACY11 27(732) 06-OCT-76 08:51 PAGE 84

3106	010506	005777	170410
3107	010512	001401	
3108	010514	104100	

TST	RSAS
BEQ	+4
HLT	AS

:DID RSAS CLEAR
:YES
:NO

```

3109
3110
3111
3112
3113 010516 104400
3114
3115 010520 016767 170444 007740 PARTST: MOV TIMSV,TIMES ;RESTORE LOOP #
3116 010526 104414 CLRDK ;CLEAR ALL RS REG
3117 010530 012777 000010 170362 1S: MOV #10,RSER ;SET PAR
3118 010536 022777 150600 170352 CMP #150600,RSDS ;DID ERR,ATA AND DRY SET?
3119 010544 001401 BEQ +4 ;YES
3120 010546 104042 HLT !DS!ER ;ER SHOULD SET IF PAR SETS IN RSER
3121 010550 104414 CLRDK ;CLEAR ALL RS REG
3122 010552 005777 170342 TST RSER ;DID PAR CLEAR?
3123 010556 001401 BEQ +4 ;YES
3124 010560 104002 HLT !ER ;PAR DID NOT CLEAR BY CLEAR BIT
3125 010562 022777 010600 170326 CMP #10600,RSDS ;DID ERROR BITS CLEAR
3126 010570 001401 BEQ +4 ;YES
3127 010572 104040 HLT !DS ;NO

```

```

3128 ;CHECK BITS 12 TO 15 FOR 0
3129 ;CHECK SECTOR FRACTION TO WATCH FOR MOVEMENT
3130 ;CHECK CS BITS IN LA AND ADDRESS CONFIRM IN MR REG
3131
3132 ;*****
3133 ;TEST 55 LOOK AHEAD TEST
3134 ;*****
3135 010574 104400 TST55: SCOPE
3136
3137 010576 032777 170000 170320 LATST: BIT #170000,DRSLA ;ARE BITS 12 TO 15 CLEARED?
3138 010604 001401 BEQ +4 ;YES
3139 010606 104204 HLT !LA ;BITS 12 TO 15 SHOULD BE CLEARED
3140 010610 104400 TST56: SCOPE
3141
3142 ;NOW TEST MOVEMENT IN SF BITS
3143
3144 010612 012767 171005 170360 MOV #171005,WORK ;SET UP FOR TIME OUT ROUTINE
3145 010620 017701 170300 MOV DRSLA,GOOD ;GET READING FROM LA
3146 010624 042701 007700 BIC #7700,GOOD ;GET RID OF CS BITS
3147 010630 005367 170344 1$: DEC WORK ;WAIT FOR DISK
3148 010634 001407 BEQ ERRR ;TYPE ERROR
3149 010636 017700 170262 MOV DRSLA,BAD ;READ LA
3150 010642 042700 007700 BIC #7700,BAD ;CLEAR CS BITS
3151 010646 020100 CMP GOOD,BAD ;DID SF BITS CHANGE?
3152 010650 001767 BEQ 1$ ;WAIT FOR TIME OUT
3153 010652 000422 BR LATDON ;LA OK CONT
3154 010654 ERRR:
3155 010654 104402 010660 TYPE +2 ;.ASCIZ <15><12>"SECTOR FRACTIONS NOT MOVING"
3156 010716 104204 HLT !LA ;TYPE LOOK AHEAD REG
3157 010720 LATDON: ;DONE CONT.

```

```

3158 ;*****
3159 ;TEST 5? CHECK CS BITS TO INCREMENT AND ADDRESS CONFIRM BIT IN MR
3160 ;*****
3161 010720 104400 TST57: SCOPE
3162
3163 010722 016767 007540 170240 CSTST: MOV TIMES,TIMSV ;SAVE LOOP CT
3164 010730 012767 000010 007530 MOV #10,TIMES ;LOOP 10 TIMES
3165 010736 104414 CLRDK ;CLEAR ALL RS REG.
3166 010740 012701 001000 MOV #1000,GOOD ;LOAD COUNTER
3167 010744 032777 001000 170156 BIT #BIT9,RSMR ;IS ADD CONFIRM BIT 0?
3168 010752 001407 BEQ ADDCF ;YES CONTINUE
3169 010754 005301 DEC GOOD ;WAIT FOR
3170 010756 001376 BNE -2 ;DISK TO MOVE
3171 010760 032777 001000 170142 BIT #BIT9,RSMR ;IS ADD. CON. BIT BIT 0?
3172 010766 001401 BEQ +4 ;YES
3173 010770 104220 HLT !MR ;ADD. CONF. BIT ALWAYS A 1
3174
3175 ;NOW TEST TA BITS AND ADD. CON. BIT IN MR
3176
3177 010772 012777 177777 170114 ADDCF: MOV #-1,RSDA ;INIT RSDA
3178 011000 012767 177777 170172 1S: MOV #-1,WORK ;SETUP TIMEOUT COUNTER
3179 011006 005277 170102 INC RSDA ;GET NEXT SECTOR
3180 011012 022777 010000 170074 CMP #10000,RSDA ;DONE ALL YET?
3181 011020 001433 BEQ DONCS ;YES
3182 011022 005367 170152 2S: DEC WORK ;DO TIMEOUT ROUTINE
3183 011026 001427 BEQ TMEOUT ;ADD. CON. NEVER CAME UP
3184 011030 032777 001000 170072 BIT #1000,RSMR ;DID ADD CONFIRM BIT SET?
3185 011036 001771 BEQ 2S ;YES
3186 011040 017700 170060 MOV RSLA,BAD ;GET LA
3187 011044 042700 000077 BIC #77,BAD ;CLEAR SF BITS
3188 011050 012767 000006 170124 3S: MOV #6,WORK1 ;SET UP COUNTER
3189 011056 006000 ROR BAD ;MOV SA BITS RIGHT
3190 011060 005367 170116 DEC WORK1 ;DO 6 TIMES
3191 011064 001374 BNE 3S ;DONE YET?
3192 011066 017701 170022 MOV RSDA,GOOD ;GET DA
3193 011072 042701 177700 BIC #177700,GOOD ;CLEAR JUNK
3194 011076 020100 CMP GOOD,BAD ;ARE SA BITS = IN DA AND LA REG.?
3195 011100 001401 BEQ +4 ;OK
3196 011102 104000 HLT ;GOOD =DA BAD = LA
3197 011104 000735 BR 1S ;NO WAIT
3198
3199 011106 104262 TMEOUT: HLT !MR!ER!DS ;ADDRESS CONFIRM BIT NEVER SET COULD BE BAD OR
3200 ;BAD LA OR BAD COMPARE BETWEEN LA AND DA
3201 011110 DONCS: ;TEST DONE CONTINUE

```

```

3202 ;*****
3203 ;TEST 60 PARITY TEST
3204 ;*****
3205 TST60: SCOPE
3206
3207 011110 104400 PART: MOV TIMSV, TIMES ;RESTORE LOOP COUNTER
3208 011112 016767 170052 007346 MOV #PRTP, @#4 ;SETUP TIME OUT VECTOR
3209 011120 012737 011300 000004 MOV #340, @#6
3210 011126 012737 000340 000006 MOV #MPRO, R2 ;GET PAR REG
3211 011134 012702 172100 TSTAGN: TST (R2) ;DOES IT EXIST
3212 011140 005712 MOV #WWP, @R2 ;YES SET WRITE WRONG PARITY
3213 011142 012712 000004 MOV #WORK, R1 ;GET TEST LOCATION
3214 011146 012701 001200 1S: MOV @R1, @R1 ;WRITE WRONG PARITY
3215 011152 011111 TST @R1 ;READ IT
3216 011154 005711 TST @R2 ;DID PARITY ERROR SET?
3217 011156 005712 BMI 2$ ;YES
3218 011160 100402 CLR @R2 ;CLEAR PARITY REG
3219 011162 005012 BR PRTP1 ;GET NEXT PARITY REG
3220 011164 000446 2$: MOV #1, @R2 ;CLEAR WWP AND ENABLE BAD PARITY
3221 011166 012712 000001 CLRDK ;CLEAR ALL RS REG
3222 011172 104414 MOV #177777, OUTBUF ;DATA TO BE X-FERED
3223 011174 012767 177777 013346 MOV #WORK, @RSBA ;SET UP CURRENT ADDRESS
3224 011202 012777 001200 167702 MOV #-1, @R5WC ;SET WORD COUNT TO -1
3225 011210 012777 177777 167672 MOV #61, @RSCS1 ;GO WRITE
3226 011215 012777 000061 167660 3$: TSTB @RSCS1 ;DONE YET?
3227 011224 105777 167654 BPL 3$ ;NO WAIT
3228 011230 100375 CLR @R2 ;CLEAR PARITY REG
3229 011232 005012 CLR WORK ;WRITE GOOD PARITY
3230 011234 005067 167740 MOV @RSCS2, BAD ;GET CS2
3231 011240 017700 167642 MOV #20100, GOOD
3232 011244 012701 020100 BIS UNNUM, GOOD ;GET CORRECT AND FOR CS2
3233 011250 056701 167702 CMP GOOD, BAD ;IS CS2 CORRECT?
3234 011254 020100 BEQ .+4 ;YES
3235 011256 001401 HLT ;CS2 SHOULD = GOOD
3236 011260 104000 144260 167614 CMP #144260, @RSCS1 ;IS CS1 CORRECT?
3237 011262 022777 BEQ .+4 ;YES
3238 011264 001401 HLT ;DS
3239 011270 104040 JMP NOPAR ;GET OUT
3240
3241 ;TRAPOUT ROUTINE
3242
3243 011300 022626 PRTP: CMP (6)+, (6)+ ;CLEAR STACK
3244 011302 022702 172136 PRTP1: CMP #172136, R2 ;DONE YET?
3245 011306 001404 BEQ NOPAR ;YES NO PAR REG
3246 011310 052702 000002 ADD #2, R2 ;NO TRY AGAIN
3247 011314 000167 177620 JMP TSTAGN ;RETRY
3248
3249 011320 012737 000006 000004 NOPAR: MOV #6, @#4
3250 011326 005037 000006 CLR @#6

```

```

3251 :*****
3252 :TEST 61 TEST WRITE CHECK ERROR
3253 :*****
3254 011332 10440C †TST61: SCOPE
3255
3256 :WRITE A WORD OF 0 AND FLOAT A 1 THROUGH IT TO CAUSE WCE
3257 :SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
3258 :TO CAUSE WCE
3259
3260 011334 104414 WCETST: CLRDK ;CLEAR ALL RS REG
3261 011336 005067 013206 CLR OUTBUF ;WRITE A WD OF 0
3262 011342 013777 001102 167542 WCETT: MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3263 011350 012777 177777 167532 MOV #-1,@RSWC ;SET WORD COUNT TO -1
3264 011356 012777 000061 167520 MOV #61,@RSCS1 ;GO WRITE
3265 011364 105777 167514 3$: TSTB @RSCS1 ;DONE YET?
3266 011370 100375 BPL 3$ ;NO WAIT
3267 011372 032767 040000 167564 BIT #BIT14,ONCEE ;WRITE A 1 OR 0?
3268 011400 001410 BEQ 2$ ;WRITE A 0
3269 011402 012767 177777 013140 MOV #-1,OUTBUF ;WRITE A 1
3270 011410 000241 CLC ;CLEAR CARRY
3271 011412 006167 013132 6$: ROL OUTBUF ;FLOAT A 0 THROUGH BAD WD
3272 011416 103123 BCC WCEDON ;DONE GET OUT
3273 011420 000406 BR 5$ ;CHECK WCE
3274 011422 005067 013122 2$: CLR OUTBUF ;WRITE A 0
3275 011426 000261 SEC ;SET CARRY
3276 011430 006167 013114 1$: ROL OUTBUF ;FLOAT A 1
3277 011434 103503 BCS WCEDNE ;GET OUT WHEN DONE
3278 011436 013777 001102 167446 5$: MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3279 011441 012777 177777 167436 MOV #-1,@RSWC ;SET WORD COUNT TO -1
3280 011452 005077 167436 CLR @RSDA
3281 011456 012777 000051 167420 4$: MOV #51,@RSCS1 ;GO WRITE CHECK
3282 011464 105777 167414 TSTB @RSCS1 ;READY YET?
3283 011470 100375 BPL 4$ ;NO WAIT
3284 011472 017700 167410 MOV @RSCS2,BAD ;GET CS2
3285 011476 016701 167454 MOV UNNUM,GOOD ;SET UNIT #
3286 011502 052701 040300 BIS #40300,GOOD ;SET BITS
3287 011506 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3288 011510 001413 BEQ 7$ ;YES
3289 011512 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
3290 011514 016700 013030 MOV OUTBUF,BAD ;GET BAD WD THAT SHOULD CAUSE WCE
3291 011520 005001 CLR GOOD ;GET GOOD WD IF WRITING 0
3292 011522 032767 040000 167434 BIT #BIT14,ONCEE ;ARE WE WRITING 1 OR 0
3293 011530 001402 BEQ 8$ ;0
3294 011532 012701 177777 MOV #-1,GOOD ;GET GOOD WD FOR 1
3295 011536 104000 8$: HLT ;GOOD = CORRECT WD WRITTEN
3296 ;BAD = INCORRECT WD THAT WCE DID NOT CATCH

```

3297	011540	022777	144250	167336	7\$:	CMP	#144250,@RSCS1	;DID TRE SET?
3298	011546	001401				BEQ	.+4	;YES
3299	011550	104043				HLT	!CS1!ER!DS	;TRE SHOULD SET IF WCE SETS
3300	011552	017700	167334			MOV	@RSBA,BAD	;FETCH CURRENT ADDRESS
3301	011556	013701	0C1102			MOV	@#OBUFSV,GOOD	;WHAT RSBA SHOULD EQUAL
3302	011562	062701	000002			ADD	#2,GOOD	;UPDATE IT
3303	011566	020001				CMP	BAD,GOOD	;IS RSBA CORRECT
3304	011570	001401				BEQ	.+4	;YES EXECUTE CONTINUE
3305	011572	104000				HLT		;RSBA FAILED TO INCREMENT
3306	011574	104414				CLRDK		;CLEAR ALL RS REG
3307	011576	016701	167354			MOV	UNNUM,GOOD	;PUT DRIVE IN GOOD
3308	011602	052701	000100			BIS	#100,GOOD	;SET IR BIT
3309	011606	017700	167274			MOV	@RSCS2,BAD	;GET CS2
3310	011612	020100				CMP	GOOD,BAD	;IS CS2 CORRECT
3311	011614	001401				BEQ	.+4	;YES
3312	011616	104000				HLT		;BAD =CS2 GOOD IS CORRECT ANS
3313	011620	022777	004200	167256		CMP	#4200,@RSCS1	;DID TRE CLEAR?
3314	011626	001401				BEQ	.+4	;YES
3315	011630	104001				HLT	!CS1	;TRE DID NOT CLEAR WITH CLEAR
3316	011632	032767	040000	167324		BIT	#BIT14,ONCEE	;FLOATION A 1 OR 0?
3317	011640	001673				BEQ	1\$;FLOAT 1
3318	011642	000663				BR	6\$;FLOAT 0
3319	011644	052767	040000	167312	WCEDNE:	BIS	#BIT14,ONCEE	;SET BIT14
3320	011652	104414				CLRDK		
3321	011654	012767	177777	012666		MOV	#-1,OUTBUFF	
3322	011662	000167	177454			JMP	WCETT	;NOW WRITE -1 IN OUTBUF
3323	011666	042767	040000	167270	WCEDON:	BIC	#BIT14,ONCEE	;CLEAR TEST FLAG


```

3324                                     :*****
3325                                     :TEST 62 TEST PROGRAM ERROR BIT IN RSCS2
3326                                     :*****
3327 011674 104400 TST62: SCOPE
3328
3329 011676 104414 PGETST: CLRDK ;CLEAR ALL RS REG
3330 011700 012767 177777 012642 MOV #177777,OUTBUF ;DATA TO BE X-FERED
3331 011706 013777 001102 167176 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
3332 011714 012777 177000 167166 MOV #177000,@RSWC ;SET WORD COUNT
3333 011722 012777 000061 167154 MOV #61,@RSCS1 ;GO WRITE
3334 011730 105777 167150 2S: TSTB @RSCS1 ;IS RDY CLEARED YET?
3335 011734 100775 BMI 2S ;NO WAIT
3336 011736 012777 000071 167140 MOV #71,@RSCS1 ;GO READ
3337 011744 004767 012306 JSR PC,WAITRY ;WAIT FOR READY
3338 011750 104001 HLT !CS1 ;RDY NEVER CAME UP
3339 011752 022777 144260 167124 CMP #144260,@RSCS1 ;IS CS1 CORRECT?
3340 011760 001401 BEQ .+4 ;YES
3341 011762 104001 HLT !CS1 ;TRE SHOULD SET BY SETTING PGE
3342 011764 005777 167132 TST @RSAS ;AS SHOULD = 0
3343 011770 001401 BEQ .+4 ;YES
3344 011772 104100 HLT !AS ;RSAS SHOULD = 0
3345 011774 016701 167156 MOV UNNUM,GOOD ;GET UNIT #
3346 012000 052701 002200 BIS #2200,GOOD ;SET PGE, IR, AND OR
3347 012004 017700 167076 MOV @RSCS2,BAD ;GET CS2
3348 012010 020100 CMP GOOD,BAD ;IS IT CORRECT?
3349 012012 001401 BEQ .+4 ;YES
3350 012014 104000 HLT ;BAD = CS2
3351 012016 005777 167066 TST @RSWC ;SHOULD NOT BE 0
3352 012022 001001 BNE .+4 ;BECAUSE PGE SHOULD ABORT
3353 012024 104011 HLT !WC!CS1 ;CURRENT OPERATION
3354 012026 005777 167066 TST @RSER ;DID ANY ERRORS SET?
3355 012032 001401 BEQ .+4 ;NO
3356 012034 104040 HLT !DS ;RMR SHOULD BE SET
3357 012036 052777 040000 167040 BIS #TRE,@RSCS1 ;CLEAR ERRORS
3358 012044 042701 002000 BIC #PGE,GOOD ;CLEAR PGE ERROR
3359 012050 017700 167032 MOV @RSCS2,BAD ;GET CS2
3360 012054 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3361 012056 001401 BEQ .+4 ;YES
3362 012060 104200 HLT !CS2 ;PGE DID NOT CLEAR BY CLEARING TRE BAD = CS2
3363 012062 022777 004260 167014 CMP #4260,@RSCS1 ;DID SC CLEAR
3364 012070 001401 BEQ .+4 ;YES
3365 012072 104040 HLT !DS ;DID NOT CLEAR BY CLEARING TRE

```

```

3366
3367
3368
3369 012074 104400
3370
3371 012076 104414
3372 012100 013777 001102 167004
3373 012106 012777 177700 166774
3374 012114 012703 172060
3375 012120 011304
3376 012122 042704 000017
3377 012126 022704 000020
3378 012132 001372
3379 012134 012777 000031 166742
3380 012142 012777 007777 166744
3381 012150 004767 012102
3382 012154 104001
3383 012156 022777 000004 166734
3384 012164 001401
3385 012166 104002
3386 012170 022777 007777 166716
3387 012176 001001
3388 012200 104004
3389 012202 022777 150600 166706
3390 012210 001401
3391 012212 104042
3392 012214 022777 104230 166662
3393 012222 001401
3394 012224 104040
3395 012226 104414
3396 012230 005777 166664
3397 012234 001401
3398 012236 104002
3399 012240 022777 004200 166636
3400 012246 001401
3401 012250 104040

```

```

;*****
;TEST 63 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSDA
;*****
TST63: SCOPE

```

```

RMRT1: CLRDK ;CLEAR ALL RS REG
MOV @#0BUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #177700, @RSWC ;SET WORD COUNT
MOV #172060, R3 ;GET RSLA REG
1$: MOV @R3, R4 ;WAIT FOR
BIC #17, R4 ;THE MIDDLE
CMP #20, R4 ;OF SECTOR 0
BNE 1$ ;BEFORE DOING A SEARCH
MOV #31, @RSCS1 ;SEARCH
MOV #7777, @RSDA ;CAUSE ERROR
JSR PC, WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY NEVER CAME UP
CMP #4, @RSER ;DID RMR SET?
BEQ +4 ;YES
HLT !ER ;ER SHOULD = 4
CMP #7777, @RSDA ;DID DA GET MODIFIED?
BNE +4 ;NO
HLT !DA
CMP #150600, @RSDS ;DID ERR SET?
BEQ +4 ;YES
HLT !DS!ER ;ER DID NOT SET BECAUSE OF RMR
CMP #104230, @RSCS1 ;IS CS1 CORRECT?
BEQ +4 ;YES
HLT !DS ;CS1 SHOULD = 144260
CLRDK ;CLEAR ALL RS REG
TST @RSER ;DID RMR CLEAR?
BEQ +4 ;YES
HLT !ER ;RMR DID NOT CLEAR WITH A CLEAR
CMP #4200, @RSCS1 ;IS CS1 CORRECT?
BEQ +4 ;YES
HLT !DS ;NO

```

```

3402
3403
3404
3405
3406 012252 104400
3407
3408 012254 104414
3409 012256 013777 001102 166626
3410 012264 012777 177700 166616
3411 012272 012777 000061 166604
3412 012300 105777 166600
3413 012304 100775
3414 012306 012777 177773 166604

```

```

;*****
;TEST 64 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSER
;*****
TST64: SCOPE

```

```

RMRT2: CLRDK ;CLEAR ALL RS REG
MOV @#0BUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #177700, @RSWC ;SET WORD COUNT
MOV #61, @RSCS1 ;GO WRITE
2$: TSTB @RSCS1 ;IS RDY SET?
BMI 2$ ;YES WAIT FOR IT TO CLEAR
MOV #177773, @RSER ;CAUSE ERROR

```

NOEC-11-DERSA-2
DERSA.P11

TST64

RS11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
TEST RMR IN RSEP REGISTER TRYING TO WRITE INTO RSR

3415	012314	004767	011736		
3416	012320	104001			
3417	012320	022777	000004	166570	
3418	012330	001401			
3419	012332	104002			
3420	012334	022777	150600	166554	
3421	012342	001401			
3422	012344	104042			
3423	012346	022777	144260	166530	
3424	012354	001401			
3425	012356	104040			
3426	012360	104414			
3427					
3428					
3429					
3430					
3431	012362	104400			
3432					
3433	012364	104414			
3434	012366	013777	001102	166516	
3435	012374	012777	177700	166506	
3436	012402	012777	000061	166474	
3437	012410	105777	166470		
3438	012414	100775			
3439	012416	012777	000030	166460	
3440	012424	004767	011626		
3441	012430	104001			
3442	012432	022777	000004	166460	
3443	012440	001401			
3444	012442	104002			
3445	012444	022777	150600	166444	
3446	012452	001401			
3447	012454	104042			
3448	012456	022777	144260	166420	
3449	012464	001401			
3450	012466	104040			
3451	012470	104414			

```

JSR PC, WAITRY
HLT !CSI
CMP #4, RSR
BEQ +4
HLT !ER
CMP #150600, RSDS
BEQ +4
HLT !DS!ER
CMP #144260, RSCS1
BEQ +4
HLT !DS
CLRDK

```

```

:RDY NEVER CAME UP
:DID RMR SET?
: YES
: ER SHOULD = 4
:DID ERR SET?
: YES
: ERR DID NOT SET BECAUSE OF RMR
: IS CSI CORRECT?
: YES
: CSI SHOULD = 144260
: CLEAR ALL RS REG

```

:TEST 65 TEST RMR IN RSEP REGISTER TRYING TO WRITE INTO RSCS1

TST65: SCOPE

```

RMRT3: CLRDK
MOV #0BUFSV, RSB
MOV #177700, RSWC
MOV #61, RSCS1
25: TSTB RSCS1
BMI 25
MOV #30, RSCS1
JSR PC, WAITRY
HLT !CSI
CMP #4, RSR
BEQ +4
HLT !ER
CMP #150600, RSDS
BEQ +4
HLT !DS!ER
CMP #144260, RSCS1
BEQ +4
HLT !DS
CLRDK

```

```

: CLEAR ALL RS REG
: SET UP CURRENT ADDRESS
: SET WORD COUNT
: GO WRITE
: IS RDY SET?
: YES WAIT FOR IT TO CLEAR
: CAUSE ERROR
: WAIT FOR READY
: RDY NEVER CAME UP
: DID RMR SET?
: YES
: ER SHOULD = 4
: DID ERR SET?
: YES
: ERR DID NOT SET BECAUSE OF RMR
: IS CSI CORRECT?
: YES
: CSI SHOULD = 144260
: CLEAR ALL RS REG

```

```

3453
3454
3455 012472 104400
3456
3457 012474 104414
3458 012476 013777 001102 166406
3459 012504 012777 177700 166376
3460 012512 012703 172060
3461 012516 011304
3462 012520 042704 000017
3463 012524 022704 000020
3464 012530 001372
3465 012532 012777 000031 166344
3466 012540 012777 000000 166354
3467 012546 005067 166426
3468 012552 032777 000200 166336
3469 012560 001004
3470 012562 005267 166412
3471 012566 001371
3472 012570 104001
3473 012572 005777 166322
3474 012576 001401
3475 012600 104002
3476 012602 022777 110600 166306
3477 012610 001401
3478 012612 104042
3479 012614 022777 104230 166262
3480 012622 001401
3481 012624 104040
3482 012626 104414
3483 012630 022777 004200 166246
3484 012636 001401
3485 012640 104040

```

```

*****
:TEST 66 TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS
*****
TST66: SCOPE

RMRT4: CLRDK
MOV 2#0BUFSV,RSBA :CLEAR ALL RS REG
MOV #177700,RSWC :SET UP CURRENT ADDRESS
MOV #172060,R3 :SET WORD COUNT
15: MOV R3,R4 :GET RSLA REG
BIC #17,R4 :WAIT FOR
CMP #20,R4 :THE MIDDLE
BNE 15 :OF SECTOR 0
MOV #31,RS0CS1 :BEFORE DOING A SEARCH
MOV #0,RSAS :SEARCH
CLR WORK :TRY TO CAUSE ERROR
25: BIT #BIT7,RS0SDS :CLEAR COUNTER
BNE 35 :WAIT FOR DRY
INC WORK :READY CONT
BNE 25 :COUNT
HLT !CS1 :RETRY
35: TST RSER :RDY NEVER CAME UP
BEQ +4 :DID RMR SET?
HLT !ER :NO
CMP #110600,RS0SDS :ER SHOULD = 0
BEQ +4 :DID ERR SET?
HLT !DS!ER :NO
CMP #104230,RS0CS1 :DS SHOULD = 110600
BEQ +4 :IS CS1 CORRECT?
HLT !DS :YES
CLRDK :CS1 SHOULD = 144260
CMP #4200,RS0CS1 :CLEAR ALL RS REG
BEQ +4 :IS CS1 CORRECT?
HLT !DS :YES
: :NO

```

```

3486
3487
3488
3489 012642 104400
3490
3491
3492
3493 012644 104414
3494 012646 012767 177500 166330
3495 012654 005767 166306
3496 012660 001003
3497 012662 012767 177600 166314
3498 012670 016777 166310 166212 1$:
3499 012676 012767 177777 011644
3500 012704 013777 001102 166200
3501 012712 052777 000010 166166 4$:
3502 012720 012777 000061 166156
3503 012726 105777 166152 5$:
3504 012732 100375
3505 012734 005077 166154 2$:
3506 012740 005067 011604
3507 012744 013777 001102 166140
3508 012752 012777 177777 166130
3509 012760 012702 172060
3510 012764 011203 3$:
3511 012766 042703 000077
3512 012772 022703 004000
3513 012776 001372
3514 013000 012777 000061 166076
3515 013006 011203 6$:
3516 013010 042703 000017
3517 013014 022703 000020
3518 013020 001372
3519 013022 012777 000040 166056
3520 013030 016777 166122 166050

```

```

*****
:TEST 67 TEST DCK IN RSER
*****
TST67: SCOPE

:DO A WRITE AND THEN A CLEAR FUNCTION THAT SHOULD CAUSE DCK TO SET

DCKTST: CLRDK
MOV #177500,WORK2 ;CLEAR ALL RS REG
TST RS04DT ;GET WC FOR RS04
BNE IS ;IS THIS A RS04?
MOV #177600,WORK2 ;YES
MOV WORK2,RSWC ;NO
MOV #-1,OUTBUF ;LOAD WC
MOV #0,OUTBUF ;WRITE -1
MOV #0,OUTBUF,RSBA ;SET UP CURRENT ADDRESS
BIS #10,RS0CS2 ;SET BAI BIT
MOV #61,RS0CS1 ;GO WRITE
TSTB RS0CS1 ;IS RDY SET?
BPL SS ;WAIT FOR WRITE TO FINISH
CLR RS0CA ;SET DSK ADDRESS TO 0
CLR OUTBUF ;WRITE 0
MOV #0,OUTBUF,RSBA ;SET IP CURRENT ADDRESS
MOV #-1,RSWC ;LOAD WC
MOV #172060,R2 ;PUT RSLA ADDR INTO R2
MOV (R2),R3 ;GET LA AND WAIT FOR
BIC #77,R3 ;SECTOR 40
CMP #4000,R3 ;BEFORE
BNE 3$ ;WRITING
MOV #61,RS0CS1 ;GO WRITE
MOV (R2),R3 ;GET RSLA AND WAIT FOR
BIC #17,R3 ;MIDDLE OF SECTOR
CMP #20,R3 ;0 BEFORE EXECUTING
BNE 6$ ;A CLEAR FUNCTION
MOV #40,RS0CS2 ;CLEAR ALL REG. DO IT THIS WAY
MOV UNNUM,RS0CS2 ;DO NOT USE TRAP

```

3521	013036	105777	166042	INCW:	TSTB	RS051	: IS BUSY CLEARED
3522	013042	100401			BMI	5\$: FLAG CLEARED
3523	013044	104001			HLT	:CS1	: RDY NEVER CAME UP
3524	013046	013777	001102	166036	6\$: MOV	#0BUFSV,RSBA	: SET UP CURRENT ADDRESS
3525	013054	016777	166124	166026	MOV	WORK2,RSWC	: LOAD WC
3526	013062	012777	000071	166014	MOV	#71,RS051	: GO READ
3527	013070	105777	166010		5\$: TSTB	RS051	: IS RDY SET?
3528	013074	100375			BPL	5\$: WAIT FOR READ TO FINISH
3529	013076	022777	100000	166014	CMP	#100000,RSER	: DID DCK SET?
3530	013104	001401			BEQ	.+4	: YES
3531	013106	104002			HLT	:ER	: DCK DID NOT SET
3532	013110	022777	150600	166000	CMP	#150600,RS05	: DID ERR SET?
3533	013116	001401			BEQ	.+4	: YES
3534	013120	104040			HLT	:DS	: ER DID NOT SET BY DCK
3535	013122	022777	144270	155754	CMP	#144270,RS051	: IS CS1 CORRECT?
3536	013130	001401			BEQ	.+4	: YES
3537	013132	104044			HLT	:DS!DA	
3538	013134	017700	165746		MOV	RS052,BAD	: GET CS2
3539	013140	016701	166012		MOV	UNNUM,GOOD	: GET UNIT #
3540	013144	052701	000100		BIS	#100,GOOD	: SET IR
3541	013150	020100			CMP	GOOD,BAD	: IS CS2 CORRECT?
3542	013152	001401			BEQ	.+4	: YES
3543	013154	104000			HLT		
3544	013156	012701	177700		MOV	#177700,GOOD	: NO
3545	013162	017700	165722		15\$: MOV	RSWC,BAD	: DID TRANSFER STOP AT END OF SECTOR?
3546	013166	020100			CMP	GOOD,BAD	
3547	013170	001401			BEQ	.+4	: YES
3548	013172	104000			HLT		: NO
3549	013174	012701	024550		MOV	#OUTBUF,GOOD	: GET BA
3550	013200	005767	165762		TST	RS04DT	: RS04?
3551	013204	001003			BNE	2\$: YES
3552	013206	062701	000200		ADD	#200,GOOD	: RS03
3553	013212	000402			BR	3\$	
3554	013214	062701	000400		2\$: ADD	#400,GOOD	: GET CORRECT ANS FOR BA
3555	013220	017700	165666		3\$: MOV	RSBA,BAD	: GET BA
3556	013224	020001			CMP	BAD,GOOD	: IS BA CORRECT?
3557	013226	001401			BEQ	.+4	: YES
3558	013230	104000			HLT		: NO
3559	013232	104414			CLROK		: CLEAR ALL RS REG
3560	013234	005777	165660		TST	RSER	: DID DCK CLEAR?
3561	013240	001401			BEQ	.+4	: YES
3562	013242	104002			HLT	:ER	: DCK DID NOT CLEAR WITH CLEAR
3563	013244	012777	177500	165636	MOV	#177500,RSWC	: CLEAR DCK ON
3564	013252	013777	001102	165632	MOV	#0BUFSV,RSBA	: DRIVE BY WRITING
3565	013260	012777	000061	165616	MOV	#61,RS051	: GOOD DATA
3566	013266	105777	165612		4\$: TSTB	RS051	: ON DRIVE
3567	013272	100375			BPL	4\$	

3568
3569
3570
3571
3572
3573
3574
3575
3576
3577 013274 104400
3578
3579 013276 104414
3580 013300 012777 177777 165602
3581 013306 013777 001102 165576
3582 013314 012777 007777 165572
3583 013322 012777 000061 165554
3584 013330 004767 010722
3585 013334 104001
3586 013336 027727 165552 010000 55:
3587 013344 001401
3588 013346 104004
3589
3590
3591
3592
3593 013350 104400
3594
3595
3596
3597 013352 104414
3598 013354 012777 177777 165526
3599 013362 013777 001102 165522
3600 013370 012777 017777 165516
3601 013376 012777 000061 165500
3602 013404 105777 165474 75:
3603 013410 100401
3604 013412 000774
3605 013414 022777 002000 165476
3606 013422 001401
3607 013424 104002
3608 013426 022777 150600 165462
3609 013434 001401
3610 013436 104140
3611 013440 022777 144260 165436
3612 013446 001401
3613 013450 104001
3614 013452 104414
3615 013454 005777 165440
3616 013460 001401
3617 013462 104002

;TEST THE ABILITY OF THE DISK CONTROL TO
;INCREMENT THE TRACK REGISTER.

;A ONE WORD WRITE WILL BE EXECUTED

;RSDA=7777 RSWC = -1
;AT THE COMPLETION OF THE WRITE RSDA = 10000

;TEST 70 TEST DISK ADDRESS REGISTER

TST70: SCOPE

DKADR: CLRDK ;CLEAR ALL RS REG
MOV #177777, @RSWC ;SET WORD COUNT TO -1
MOV @#0BUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #7777, @RSDA ;SET RSDA TO ALL ONES
MOV #61, @RSCSI ;GO WRITE ONE WORD
JSR PC, WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY DID NOT COME UP
CMP @RSDA, #10000 ;DOES RSDA=0
BEQ +4 ;RSDA OK
HLT !DA ;DA DID NOT INCREMENT

;TEST 71 TEST IAE ERROR

TST71: SCOPE

;IAE ERROR SHOULD SET ERR,ATA AND SC BITS

IAERR: CLRDK ;CLEAR ALL RS REG
MOV #177777, @RSWC ;SET WC TO -1
MOV @#0BUFSV, @RSBA ;SET UP BUS ADDRESS
MOV #17777, @RSDA ;SET DA TO RECEIVE ERROR
MOV #61, @RSCSI ;GO WRITE ONE WD
75: TSTB @RSCSI ;TEST FOR ERR OR RDY
BMI +4 ;OK CONT.
BR 75 ;WAIT
CMP #2000, @RSER ;DID IAE SET?
BEQ +4 ;YES
HLT !ER ;IAE SHOULD BE SET
CMP #150600, @RSDS ;DID ERR SET?
BEQ +4 ;YES
HLT !DS!AS ;ERR SHOULD BE SET
CMP #144260, @RSCSI ;DID SC SET?
BEQ +4 ;YES
HLT !CS1 ;SC SHOULD BE SET
CLRDK ;CLEAR ALL RS REG
TST @RSER ;CLR ERRORS?
BEQ +4 ;YES
HLT !ER ;ERR DID NOT CLR WITH 40 IN CS2

```

3618 ; IN THIS ROUTINE THE PROGRAM WILL GENERATE A
3619 ; NON-EXISTENT DISK ERROR
3620
3621 ;*****
3622 ; TEST 72 TEST FOR NON-EXISTENT DISK ERROR
3623 ;*****
3624 TST72: SCOPE
3625
3626 NEDTST: CLRDK ; CLEAR ALL RS REG
3627 013464 104400 MOV #401,WORK ; SET UP FOR N.E.D. NUMBER
3628 013470 012767 000401 165502 CLR GOOD ; LOOK FOR NON EXISTENT DRIVES
3629 013476 005001 BIT WORK,UNITSV ; ON THE SYSTEM
3630 013500 036767 165474 165452 1$: BEQ 3$ ; FOUND NON EXISTENT DRIVE
3631 013510 005201 INC GOOD ; CONTAINS UNIT #
3632 013512 006167 165462 ROL WORK ; KEEP LOOKING
3633 013516 103452 BCS NEDDON ; COULD NOT FIND ANY NON EXISTENT DRIVES
3634 013520 000767 BR 1$ ; LOOK FOR NED
3635 013522 010177 165360 3$: MOV GOOD,RS2 ; LOAD NED IN CS2
3636 013526 005077 165362 CLR RSDA ; WRITE DRIVE REG
3637 013532 005777 165362 TST RSER ; DID ANY BITS SET IN RSER?
3638 013536 001401 BEQ .+4 ; NO
3639 013540 104040 HLT DS ; WHY DID RSER CHANGE?
3640 013542 017700 165340 MOV RS2,BAD ; GET CS2
3641 013546 052701 010100 BIS #10100,GOOD ; SET NED AND IR
3642 013552 020100 CMP GOOD,BAD ; IS CS2 CORRECT?
3643 013554 001401 BEQ .+4 ; YES
3644 013556 104000 HLT ; GOOD=CORRECT CS2 BAD=CS2
3645 013560 022777 160200 165316 CMP #160200,RS1 ; IS CS1 CORRECT?
3646 013566 001401 BEQ .+4 ; YES
3647 013570 104200 HLT CS2 ; TRE SHOULD SET BY NED ERROR
3648 013572 005777 165324 TST RSAS ; DID ANY BITS SET?
3649 013576 001401 BEQ .+4 ; NO
3650 013600 104100 HLT AS ; WHY DID AT BITS SET?
3651 013602 112777 000100 165334 MOVB #100,RS1B ; CLEAR TRE
3652 013610 032777 010000 165270 BIT #NED,RS2 ; DID NED CLEAR
3653 013616 001401 BEQ .+4 ; YES
3654 013620 104200 HLT CS2 ; NED DID NOT CLEAR
3655 013622 017767 165266 165350 MOV RSDA,WORK ; READ DRIVE REG
3656 013630 032777 010000 165250 BIT #NED,RS2 ; DID NED SET?
3657 013636 001001 BNE .+4 ;
3658 013640 104040 HLT DS ; NED DID NOT SET
3659 013642 000431 BR NNDD ; GET OUT
3660 013644 032767 010000 165312 NEDDON: BIT #BIT12,ONCEE ; DWAS THIS TYPED BEFORE?
3661 013652 001025 BNE NNDD ; YES
3662 013654 104402 013660 TYPE .+2 ; .ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
3663 013726 052767 010000 165230 NNDD: BIS #BIT12,ONCEE ; SET TYPED FLAG

```



```

3664
3665
3666
3667 013734 104400
3668
3669 013736 104414
3670 013740 012777 177577 165142
3671 013746 005767 165214
3672 013752 001003
3673 013754 012777 177677 165126
3674 013762 012777 007777 165124
3675 013770 013777 001102 165114
3676 013776 012777 000061 165100
3677 014004 004767 010246
3678 014010 104001
3679 014012 022777 001000 165100
3680 014020 001401
3681 014022 104002
3682 014024 022777 152600 165004
3683 014032 001401
3684 014034 104040
3685 014036 005777 165042
3686 014042 100401
3687 014044 104001
3688 014046 104414
3689 014050 022777 010600 165040
3690 014056 001401
3691 014060 104040
3692 014062 005777 165032
3693 014066 001401
3694 014070 104002

```

```

:*****
:TEST 73 TEST THAT DAO IN RSER AND LBT IN RSDS DO SET
:*****
†ST73: SCOPE
DAOTST: CLDK
MOV #-201, @RSWC :CLEAR ALL RS REG
TST RSO4D† :LOAD WC FOR RSO4
BNE IS :IS THIS A RSO4?
MOV #-101, @RSWC :YES
MOV #7777, @RSDA :NO
MOV @#OBUFSV, @RSBA :SET RSDA=TO ALL ONES
MOV #61, @RSCS1 :CURRENT ADDRESS=OUTBUF
JSR PC, WAITRY :WRITE
HLT !CS1 :WAIT FOR READY
CMP #1000, @RSER :RDY DID NOT SET
BEQ +4 :DID DAO SET?
HLT !ER :YES
CMP #152600, @RSDS :DAO DID NOT SET
BEQ +4 :DID LBT SET?
HLT !DS :YES
TST @RSCS1 :LBT DID NOT SET
BMI +4 :IS ERROR FLAG SET
HLT !CS1 :ERROR IS SET
CLDK :SC DID NOT SET
CMP #10600, @RSDS :CLEAR ALL RS REG
BEQ +4 :DID ATA +LBT CLEAR
HLT !DS :YES
TST @RSER :ATA DID NOT CLEAR BY CLR BIT
BEQ +4 :DID DAO CLEAR?
HLT !ER :YES
:DAO DID NOT CLEAR WITH CLEAR

```

```

3695
3696
3697
3698 014072 104400
3699
3700 014074 104414
3701 014076 012777 177601 165004
3702 014104 005767 165056
3703 014110 001003
3704 014112 012777 177701 164770
3705 014120 012777 007777 164766
3706 014126 013777 001102 164756
3707 014134 012777 000061 164742
3708 014142 004767 010110
3709 014146 104001
3710 014150 005777 164744
3711 014154 001401
3712 014156 104002
3713 014160 022777 012600 164730
3714 014166 001401
3715 014170 104040
3716 014172 005777 164706
3717 014176 100001
3718 014200 104001
3719 014202 104414
3720 014204 022777 010600 164704
3721 014212 001401
3722 014214 104040

```

```

:*****
:TEST 74 TEST THAT LBT DOES SET AND DAO DOES NOT
:*****
TST74: SCOPE

DAOTT: CLRDK ;CLEAR ALL RS REG
MOV #-177, @RSWC ;LOAD WC FOR RS04
TST RS04DT ;IS THIS A RS04?
BNE !S ;YES
MOV #-77, @RSWC ;NO
MOV #7777, @RSDA ;SET RSDA=TO ALL ONES
1S: MOV @#0BUFSV, @RSBA ;CURRENT ADDRESS=OUTBUF
2S: MOV #61, @RSCS1 ;WRITE
JSR PC, WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY DID NOT SET
TST @RSER ;ANY ERRORS?
BEQ +4 ;NO
HLT !ER ;YES
CMP #12600, @RSDS ;DID LBT SET?
BEQ +4 ;YES
HLT !DS ;LBT DID NOT SET
TST @RSCS1 ;IS ERROR FLAG SET
BPL +4 ;NO
HLT !CS1 ;ERROR
CLRDK ;CLEAR ALL RS REG
CMP #10600, @RSDS ;DID LBT CLEAR
BEQ +4 ;YES
HLT !DS ;ATA DID NOT CLEAR BY CLR BIT

```

```

3723 ;*****
3724 ;TEST 75 EXECUTE FUNCTION WITH ERROR BITS SET
3725 ;*****
3726 014216 104400 TST75: SCOPE
3727
3728 014220 104414 ERTST: CLROK ;CLEAR ALL RS REG
3729 014222 012777 177017 164670 MOV #177017,RSER ;LOAD ER
3730 014230 017700 164666 MOV #RSAS,BAD ;GET AS REG
3731 014234 016701 164722 MOV UNCMP,GOOD ;GET UNIT ATA BIT
3732 014240 042701 177400 BIC #177400,GOOD ;CLEAR JUNK
3733 014244 020100 CMP GOOD,BAD ;IS AS REG CORRECT?
3734 014246 001401 BEQ +4 ;YES
3735 014250 104100 HLT !AS ;AS BIT SHOULD BE SET
3736 014252 022777 104200 164624 CMP #104200,RS0CS1 ;DID ERRS SET IN CS1?
3737 014260 001401 BEQ +4 ;YES
3738 014262 104040 HLT !DS ;CS1 SHOULD =104200
3739 014264 016777 164672 164630 MOV UNCMP,RSAS ;CLEAR ATA BIT
3740 014272 005777 164624 TST RSAS ;DID IT CLEAR?
3741 014276 001401 BEQ +4 ;YES
3742 014300 104100 HLT !AS ;COULD NOT CLEAR AS BIT
3743 ;BY LOADING A 1 INTO IT
3744 014302 022777 004200 164574 CMP #4200,RS0CS1 ;DID SC CLEAR BY
3745 014310 001401 BEQ +4 ;CLEARING ATA
3746 014312 104002 HLT !ER ;NO
3747 014314 012767 177777 010226 MOV #177777,OUTBUF ;DATA TO BE XFERRED
3748 014322 013777 001102 164562 MOV #0BUFSV,RSBA ;SET UP CURRENT ADDRESS
3749 014330 012777 177777 164552 MOV #-1,RSWC ;LOAD WC WITH -1
3750 014336 012777 000071 164540 MOV #71,RS0CS1 ;DO READ FUNCTION
3751 014344 032777 000001 164532 BIT #1,RS0CS1 ;DID GO BIT CLEAR
3752 014352 001401 BEQ +4 ;YES
3753 014354 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
3754 014356 105777 164522 1$: TSTB RS0CS1 ;WAIT FOR READY
3755 014362 100375 BPL 1$ ;WAIT
3756 014364 022777 144270 164512 CMP #144270,RS0CS1 ;DID ERRS CLEAR BY SETTING GO BIT?
3757 014372 001401 BEQ +4 ;YES
3758 014374 104002 HLT !ER ;NO

```

3759	014376	017700	164504		MOV	QARSCS2,BAD	;GET CS2
3760	014402	012701	001100		MOV	#1100,GOOD	;GET CORRECT ANS
3761	014406	056701	164544		BIS	UNNUM,GOOD	;GET UNIT #
3762	014412	020100			CMP	GOOD,BAD	;IS CS2 CORRECT?
3763	014414	001401			BEQ	.+4	;YES
3764	014416	104000			HLT		;GOOD = WHAT CS2 SHOULD =
3765	014420	022777	150600	164470	CMP	#150600,QARSDS	;DID ERR BITS SET?
3766	014426	001401			BEQ	.+4	;NO
3767	014430	104040			HLT	;DS	;ERR BIT SHOULD BE 1
3768	014432	022777	177777	164450	CMP	#-1,QARSWC	;DID WC MOVE?
3769	014440	001401			BEQ	.+4	;NO
3770	014442	104010			HLT	;WC	;WC SHOULD = 1777777
3771	014444	005777	164444		TST	QARSDA	;DID DA MOV
3772	014450	001401			BEQ	.+4	;NO
3773	014452	104004			HLT	;DA	;DA SHOULD =0
3774	014454	023777	001102	164430	CMP	Q#0BUFSV,QARSB A	;DID BA MOVE
3775	014462	001401			BEQ	.+4	;NO
3776	014464	104020			HLT	;BA	;BA MOVED
3777	014466	036777	164470	164426	BIT	UNCMP,QARSAS	;AS SHOULD BE SET
3778	014474	001001			BNE	.+4	;IS IT?
3779	014476	104100			HLT	;AS	;NO
3780	014500	022777	177017	164412	CMP	#177017,QARSER	;DID ER CHANGE?
3781	014506	001401			BEQ	.+4	;NO
3782	014510	104002			HLT	;ER	;ER SHOULD NOT CHANGE

```

3783
3784
3785
3786 014512 104400
3787
3788 014514 104414
3789 014516 052777 000020 164362
3790 014524 005777 164400
3791 014530 017700 164352
3792 014534 012701 000120
3793 014540 056701 164412
3794 014544 020100
3795 014546 001401
3796 014550 104000
3797 014552 012777 000010 164350
3798 014560 022777 000010 164332
3799 014566 001401
3800 014570 104140
3801 014572 017700 164324
3802 014576 016701 164360
3803 014602 042701 177400
3804 014606 020100
3805 014610 001401
3806 014612 104100
3807 014614 022777 104200 164262
3808 014622 001401
3809 014624 104040
3810 014626 104414
3811 014630 022777 004200 164246
3812 014636 001401
3813 014640 104002
3814 014642 017700 164240
3815 014646 016701 164304
3816 014652 052701 000100
3817 014656 020100
3818 014660 001401
3819 014662 104000

```

```

*****
;TEST 76 PAT AND MCPE TEST
*****
TST76: SCOPE

PATST: CLRDK ;CLEAR ALL RS REG
BIS #BIT4,RS2 ;SET PAT
TST RSMR ;READ DRIVE REG
MOV RSCS2,BAD ;GET CS2
MOV #120,GOOD ;MDPE SHOULD
BIS UNNUM,GOOD ;NOT SET
CMP GOOD,BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT ;BAD = CS2 GOOD = CORRECT ANS
MOV #10,RSMR ;CAUSE PAR TO SET IN RSER
CMP #10,RSER ;DID PAR SET?
BEQ .+4 ;YES
HLT ;AS!DS
MOV RAS,BAD ;GET AS REG
MOV UNCMP,GOOD ;GET UNIT ATA BIT
BIC #177400,GOOD ;CLEAR JUNK
CMP GOOD,BAD ;IS AS REG CORRECT?
BEQ .+4 ;YES
HLT ;AS ;AS BIT SHOULD BE SET
CMP #104200,RS1 ;DID ERRS SET IN CS1?
BEQ .+4 ;YES
HLT ;DS ;CS1 SHOULD =104200
CLRDK ;CLEAR RS REG
CMP #4200,RS1 ;IS CS1 CORRECT?
BEQ .+4 ;CLEARING ATA
HLT ;ER ;NO
MOV RSCS2,BAD ;CHECK TO SEE
MOV UNNUM,GOOD ;IF PAT CLEARS
BIS #100,GOOD
CMP GOOD,BAD
BEQ .+4
HLT ;PAT DID NOT CLEAR

```

```

3820
3821
3822
3823 014664 104400
3824
3825 014666 104414
3826 014670 052777 000010 164210
3827 014676 012767 177777 007644
3828 014704 013777 001102 164200
3829 014712 012777 177000 164170
3830 014720 052777 000020 164160
3831 014726 012777 000071 164150
3832 014734 105777 164144
3833 014740 100375
3834 014742 022777 144270 164134
3835 014750 001401
3836 014752 104040
3837 014754 023777 001102 164130
3838 014762 001401
3839 014764 104020
3840 014766 022777 177000 164114
3841 014774 001401
3842 014776 104010
3843
3844
3845
3846
3847 015000 104400
3848 015002 104414
3849 015004 052777 000010 164074
3850 015012 012767 177777 007530
3851 015020 013777 001102 164064
3852 015026 012777 177000 164054
3853 015034 012777 000071 164042
3854 015042 100774
3855 015044 052777 000020 164034
3856 015052 105777 164026
3857 015056 100375
3858 015060 022777 144270 164016
3859 015066 001401
3860 015070 104002
3861 015072 017700 164010
3862 015076 012701 000730
3863 015102 056701 164050
3864 015106 020100
3865 015110 001401
3866 015112 104000

```

```

;*****
;TEST 77 SET PAT BIT AND LOAD FUNCTION
;*****
TST77: SCOPE
SETPAT: CLRDK ;CLEAR ALL REG
BIS #BAI, @RSCS2 ;SET BAI
MOV #177777, @OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #-1000, @RSWC ;LOAD WC WITH -1
BIS #BIT4, @RSCS2 ;SET PAT BIT
MOV #71, @RSCS1 ;DO READ FUNCTION
1$: TSTB @RSCS1 ;WAIT FOR READY
BFL 1$ ;WAIT
CMP #144270, @RSCS1 ;DID CS1 GET LOADED?
BEQ .+4 ;NO
HLT !DS ;IT SHOULD NOT
CMP @#OBUFSV, @RSBA ;DID BA MOVE?
BEQ .+4 ;NO
HLT !BA ;YES
CMP #-1000, @RSWC ;DID WC MOVE?
BEQ .+4 ;NO
HLT !WC ;YES WHY?

```

```

;*****
;TEST 100 DO FUNCTION THEN SET PAT BIT
;*****
TST100: SCOPE
FUNDO: CLRDK ;CLEAR ALL REG
BIS #BAI, @RSCS2 ;SET BAI
MOV #177777, @OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #-1000, @RSWC ;LOAD WC WITH -1
3$: MOV #71, @RSCS1 ;DO A READ
BMI 3$ ;WAIT FOR BUSY
BIS #BIT4, @RSCS2 ;SET PAT
2$: TSTB @RSCS1 ;WAIT FOR READY
BFL 2$
CMP #144270, @RSCS1 ;DID MCPE SET?
BEQ .+4 ;NO
HLT !ER ;YES
MOV @RSCS2, @BAD ;GET CS2
MOV #730, @GOOD ;GET CORRECT ANS
BIS @UNNUM, @GOOD ;GET UNIT #
CMP @GOOD, @BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT ;GOOD = WHAT CS2 SHOULD =

```

```

3867 015114 022777 010600 163774      CMP      #10600, @RSD5      ;DID ERR BITS SET?
3868 015122 001401      BEQ      .+4           ;NO
3869 015124 104040      HLT      !DS          ;ERR BIT SHOULD BE 1
3870 015126 005777 163766      TST      @RSER       ;IS ER CLEAR?
3871 015132 001401      BEQ      .+4           ;YES
3872 015134 104044      HLT      !DS!DA      ;NO ERRORS SHOULD BE SET
3873 015136 104414      CLRDK    ;CLEAR ALL RS REG
3874 015140 022777 004200 163736      CMP      #4200, @RSCS1 ;IS CS1 CORRECT?
3875 015146 001401      BEQ      .+4           ;YES
3876 015150 104040      HLT      !DS
3877
3878 ;*****
3879 ;TEST 101 TEST PAR BY SETTING PAT
3880 ;*****
3881 TST101: SCOPE
3882 015152 104400      PATTST: CLROK
3883 015154 104414      BIS      #BAI, @RSCS2 ;CLEAR ALL RS REG
3884 015156 052777 000010 163722      MOV      #177777, @OUTBUF ;SET BAI
3885 015164 012767 177777 007356      MOV      @#OBUFSV, @RSBA ;DATA TO BE XFERED
3886 015172 013777 001102 163712      MOV      #-1000, @RSWC ;SET UP CURRENT ADDRESS
3887 015200 012777 177000 163702      MOV      #61, @RSCS1 ;LOAD WC WITH -1
3888 015206 012777 000061 163670      MOV      @RSCS1 ;DO WRITE FUNCTION
3889 015214 105777 163664      TSTB    @RSCS1 ;WAIT FOR READY
3890 015220 100775      BMI     1$           ;WAIT
3891 015222 052777 000020 163656      BIS      #BIT4, @RSCS2 ;SET PAT
3892 015230 105777 163650      TSTB    @RSCS1 ;WAIT FOR READY
3893 015234 100375      BPL     2$           ;
3894 015236 022777 144260 163640      CMP      #144260, @RSCS1 ;DID MCPE SET?
3895 015244 001401      BEQ      .+4           ;NO
3896 015246 104002      HLT      !ER         ;YES
3897 015250 017700 163632      MOV      @RSCS2, BAD ;GET CS2
3898 015254 012701 000230      MOV      #230, GOOD ;GET CORRECT ANS-- DO NOT CHECK IR - REASON 50 CYCLE
3899 015260 056701 163672      BIS      UNNUM, GOOD ;GET UNIT #
3900 015264 042700 000100      BIC      #BIT6, BAD ;CLEAR IR BIT FOR CS2 COMPARE
3901 015270 020100      CMP      GOOD, BAD ;IS CS2 CORRECT?
3902 015272 001401      BEQ      .+4           ;YES
3903 015274 104000      HLT      ;GOOD = WHAT CS2 SHOULD =
3904 015276 022777 150600 163612      CMP      #150600, @RSD5 ;DID ERR BITS SET?
3905 015304 001401      BEQ      .+4           ;NO
3906 015306 104040      HLT      !DS          ;ERR BIT SHOULD BE 1
3907 015310 022777 000010 163602      CMP      #10, @RSER   ;DID PAR SET?
3908 015316 001401      BEQ      .+4           ;YES
3909 015320 104044      HLT      !DS!DA      ;NO
3910 015322 104414      CLRDK    ;CLEAR ALL RS REG
3911 015324 022777 004200 163552      CMP      #4200, @RSCS1 ;IS CS1 CORRECT?
3912 015332 001401      BEQ      .+4           ;YES
3913 015334 104040      HLT      !DS
3914 015336 005777 163556      TST      @RSER       ;DID PAR CLEAR?
3915 015342 001401      BEQ      .+4           ;YES
3916 015344 104040      HLT      !DS

```

```

3916
3917
3918
3919 015346 104400
3920
3921 015350 104414
3922 015352 012777 007777 163534
3923 015360 012777 177700 163522
3924 015366 005767 163574
3925 015372 001403
3926 015374 012777 177600 163506
3927 015402 013777 001102 163502
3928 015410 012777 000061 163466
3929 015416 004767 006634
3930 015422 104001
3931 015424 005777 163470
3932 015430 001401
3933 015432 104002
3934 015434 022777 012600 163454
3935 015442 001401
3936 015444 104040
3937 015446 005777 163432
3938 015452 100001
3939 015454 104001
3940 015456 104414
3941 015460 022777 010600 163430
3942 015466 001401
3943 015470 104040

```

```

:*****
:TEST 102 TEST THE ABILITY TO FILL THE LAST SECTOR
:*****
↑TST102: SCOPE
LASTSC: CLRDK
MOV #7777, @RSDA ;CLEAR ALL RS REG
MOV #-100, @RSCW ;SET RSDA=TO ALL ONES
TST RSC4D↑ ;WORD COUNT=-100
BEQ 1$ ;IS THIS A RS04?
;NO
MOV #-200, @RSCW ;YES
MOV @RSC4SV, @RSC4A ;CURRENT ADDRESS=OUTBUF
MOV #61, @RSCS1 ;WRITE
JSR PC, WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY DID NOT SET
TST @RSC4R ;DID ANY ERROR BITS SET?
BEQ +4 ;NO
HLT !ER ;GOT AN ERROR
CMP #12600, @RSC4S ;DID LBT SET?
BEQ +4 ;YES
HLT !DS ;LBT DID NOT SET
TST @RSC4S1 ;IS ERROR FLAG SET
BPL +4 ;ERROR IS SET
HLT !CS1 ;SC DID NOT SET
CLRDK ;CLEAR ALL RS REG
CMP #10600, @RSC4S ;DID ATA +LBT CLEAR
BEQ +4 ;YES
HLT !DS ;ATA DID NOT CLEAR BY CLR BIT

```



```

3944
3945
3946
3947
3948
3949
3950
3951 015472 104400
3952
3953 015474 104414
3954 015476 012767 177777 007044
3955 015504 013777 001102 163400
3956 015512 012777 177600 163370
3957 015520 005767 163442
3958 015524 001003
3959 015526 012777 177700 163354
3960 015534 052777 000010 163344 5$:
3961 015542 012777 000061 163334
3962 015550 105777 163330 3$:
3963 015554 100375
3964 015556 005077 163332
3965 015562 012767 177777 006760
3966 015570 013777 001102 163314
3967 015576 012777 177777 163304
3968 015604 052777 000010 163274
3969 015612 012777 000061 163264 1$:
3970 015620 105777 163260
3971 015624 100375
3972 015626 042777 000010 163252
3973 015634 005767 163326
3974 015640 001404
3975 015642 012767 000200 163330
3976 015650 000403
3977 015652 012767 000100 163320 7$:
3978 015660 013701 001102 8$:
3979 015664 012721 177777
3980 015670 005021 6$:
3981 015672 005367 163302
3982 015676 001374
3983 015700 005077 163210
3984 015704 013777 001102 163200
3985 015712 005767 163250
3986 015716 001404
3987 015720 012777 177600 163162
3988 015726 000403
3989 015730 012777 177700 163152 9$:
3990 015736 012777 000051 163140 10$:

```

```

:FILL SECTOR WITH ALL ONES.
:NOW WRITE 1ST WORD IN SECTOR
:TEST REMAINING 63 WORDS FOR 0

```

```

:*****
:TEST 103 TEST FOR ZERO'S IN PARTIAL FILLED SECTOR
:*****

```

TST103: SCOPE

```

SECT: CLDK ;CLEAR ALL RS REG
MOV #-1,OUTBUF ;PUT -1 INTO OUTBUF
MOV #0,OUTBUF,RSBA ;SET UP CURRENT ADDR
MOV #-200,RSWC ;LOAD WC FOR RS04
TST RS04DT ;RS04?
BNE 5$ ;YES
MOV #-100,RSWC ;SET WORD COUNT TO -100
5$: BIS #BAI,RS0CS2 ;SET BAI BIT
MOV #61,RS0CS1 ;WRITE
3$: TSTB RS0CS1 ;IS R0Y SET?
BPL 3$ ;NO
CLR RS0DA ;SET DSK ADDRESS TO 0
MOV #-1,OUTBUF ;PUT 177777 INTO OUTBUF
MOV #0,OUTBUF,RSBA ;SET UP CURRENT ADDR
MOV #-1,RSWC ;SET WORD COUNT TO -1
BIS #10,RS0CS2 ;SET BAI BIT
MOV #61,RS0CS1 ;WRITE
1$: TSTB RS0CS1 ;IS R0Y SET?
BPL 1$ ;NO
BIC #10,RS0CS2 ;CLEAR BAI BIT
TST RS04DT ;RS04?
BEQ 7$ ;NO
MOV #200,WORK ;YES
BR 8$ ;CONT
7$: MOV #100,WORK ;SET UP BUFFER
8$: MOV #0,OUTBUF,RSBA ;GET STARTING ADDR OF BUF
MOV #-1,(R1)+ ;LOAD FIRST WD WITH -1
6$: CLR (R1)+ ;LOAD REST WITH 0
DEC WORK ;DONE YET?
BNE 6$ ;NO
CLR RS0DA ;SET DSK ADDRESS TO 0
MOV #0,OUTBUF,RSBA ;SET UP CURRENT ADDR
TST RS04DT ;RS04?
BEQ 9$ ;NO
MOV #-200,RSWC ;YES
BR 10$ ;CONT
9$: MOV #-100,RSWC ;SET WORD COUNT TO -100
10$: MOV #51,RS0CS1 ;WRITE CHECK

```

3991 015744 032777
 3992 015752 001774
 3993 015754 016701
 3994 015760 052701
 3995 015764 017700
 3996 015770 020100
 3997 015772 001401
 3998 015774 104002

000200 163132 25:
 163176
 000100
 163116

BIT #200, @RSCS1
 BEG 25
 MOV UNNUM, GOOD
 BIS #100, GOOD
 MOV @RSCS2, BAD
 CMP GOOD, BAD
 BEQ +4
 HLT !ER

; IS RDY SET?
 ; NO
 ; GET UNIT #
 ; SET IR BIT
 ; GET CS2
 ; IS CS2 CORRECT?
 ; YES
 ; THERE WAS A WRITE CHECK ERROR

4000
 4001
 4002
 4003 015776 104400
 4004

 ; TEST 104 IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDR

 †ST104: SCOPE

4005 016000 104414
 4006 016002 016767
 4007 016010 012767
 4008 016016 012767
 4009 016024 012767
 4010 016032 005737
 4011
 4012
 4013 016036 012767
 4014 016044 012737
 4015 016052 005037
 4016 016056 012737
 4017 016064 012737
 4018 016072 012737
 4019 016100 012737
 4020 016106 012737
 4021 016114 012737
 4022 016122 012737
 4023 016130 012702

002460 163160
 000010 002450
 016522 161760
 000340 161754
 177572
 016514 161740
 007600 172356
 172340
 000200 172342
 002000 172344
 177406 172300
 177406 172302
 177406 172304
 177406 172316
 000001 177572
 040000

EXTTST: CLRDK

MOV TIMES, TIMSV
 MOV #10, TIMES
 MOV #EXTRP, 4
 MOV #340, 6
 TST @#SR0
 MOV #EXTRP, 4
 MOV #7600, @#KIPAR7
 CLR @#KIPAR0
 MOV #200, @#KIPAR1
 MOV #2000, @#KIPAR2
 MOV #400*256.-400+UP+RW, @#KIPDR0
 MOV #400*256.-400+UP+RW, @#KIPDR1
 MOV #400*256.-400+UP+RW, @#KIPDR2
 MOV #400*256.-400+UP+RW, @#KIPDR7
 MOV #1, @#SR0
 MOV #40000, R2

; CLEAR ALL RS REG.
 ; SAVE LOOP #
 ; LOOP 10 TIMES
 ; SETUP TIMEOUT TRAP
 ; IF MEMORY MANAGEMENT IS NOT
 ; AVAILABLE THE PROGRAM WILL TRAP
 ; AND TRANSFER TO END OF THE TEST
 ; OPEN I/O REGISTERS
 ; FREE FIRST 4K
 ; ENABLE SECOND 4K
 ; SET KIPDR0=RW UP 400 BLOCKS
 ; SET KIPDR1=RW UP 400 BLOCKS
 ; SET KIPDR2=RW UP 400 BLOCKS
 ; SET KIPDR7=RW UP 400 BLOCKS
 ; TURN ON MEMORY MANAGEMENT
 ; R2 EQUALS BASE ADDR

E09

41NOEC-11-DERSA-A
DERSAA.P11 TST104

RS11-REC3-REC4 BASIC FUNCTION DIAGNOSTIC
IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS BITS

MACY11 27(732) 06-OCT-76 08:51 PAGE 109

4024	016134	012712	177777			75:	MOV	#177777, (R2)	; INSERT PATTERN INTO 200000
4025	016140	012777	177776	162742			MOV	#-2, @RSWC	; SETUP WORDCOUNT
4026	016146	012777	177777	162736			MOV	#177777, @RSBA	; SETUP BUS ADDR
4027	016154	012777	000061	162722			MOV	#61, @RSCS1	; WRITE TWO WORDS ON DISK. RSBA
4028									; STARTS AT 177777 TO FORCE CARRY
4029									; TO SET A16
4030	016162	105777	162716				TSTB	@RSCS1	; WAIT FOR READY
4031	016166	100375					BPL	.-4	
4032	016170	005777	162710				TST	@RSCS1	
4033	016174	100002					BPL	15	
4034	016176	104046					HLT	!ER!DA!DS	; STATUS ERROR AFTER 2 WORD WRITE
4035	016200	000447					BR	25	; USING MEXD
4036	016202	022777	004660	162674		15:	CMP	#4660, @RSCS1	; IS CS1 CORRECT
4037	016210	001402					BEQ	35	; YES
4038	016212	104002					HLT	!ER	; CS2 DID NOT COMPARE
4039	016214	000441					BR	25	
4040	016216	005012				35:	CLR	(R2)	; CLEAR LOCATION 200000
4041	016220	005077	162670				CLR	@RSDA	; SETUP DA
4042	016224	012777	177777	162660			MOV	#177777, @RSBA	; SETUP BA
4043	016232	012777	177776	162650			MOV	#-2, @RSWC	; SETUP WC
4044	016240	012777	000071	162636			MOV	#71, @RSCS1	; READ TWO WORDS INTO LOCATIONS
4045									; 177777 AND 200000.
4046	016246	105777	162632				TSTB	@RSCS1	; WAIT FOR READY
4047	016252	100375					BPL	.-4	
4048	016254	005777	162624				TST	@RSCS1	; ANY ERRORS?
4049	016260	100002					BPL	45	; BRANCH IF NO
4050	016262	104002					HLT	!ER	; ERROR AFTER READING 2 WORDS
4051	016264	000415					BR	25	
4052	016266	022777	004670	162610		45:	CMP	#4670, @RSCS1	; IS CS1 CORRECT?
4053	016274	001402					BEQ	55	; YES
4054	016276	104002					HLT	!ER	; CS1 DID NOT COMPARE
4055	016300	000407					BR	25	; READ STARTING AT 177777
4056	016302	022712	177777			55:	CMP	#177777, (R2)	; WAS DATA READ INTO LOCATION
4057	016306	001404					BEQ	25	; 200000 CORRECTLY? - BRANCH IF YES
4058	016310	012701	177777				MOV	#177777, GOOD	
4059	016314	011200					MOV	(R2), BAD	
4060	016316	104000					HLT		; DATA COMPARE ERROR AT 200000
4061	016320	000240				25:	NOP		

4062	016322	104414			EXTT1:	CLRDK			;CLEAR ALL REG
4063	016324	012737	004000	172344		MOV	#4000, @#KIPAR2		
4064	016332	012702	040000			MOV	#40000, R2		;R2 EQUALS THE BASE ADDR
4065	016336	012712	177777		7S:	MOV	#177777, (R2)		;INSERT PATTERN INTO 400000
4066	016342	012777	177777	162542		MOV	#177777, @RSBA		;SETUP BUS ADDR
4067	016350	012777	177776	162532		MOV	#-2, @RSWC		;LOAD WC
4068	016356	012777	000461	162520		MOV	#461, @RSCS1		;SET BIT A16 AND WRITE
4069	016364	105777	162514			TSTB	@RSCS1		;WAIT FOR READY
4070	016370	100375				BPL	.-4		
4071	016372	005777	162506			TST	@RSCS1		;ANY ERRORS?
4072	016376	100001				BPL	4S		;BRANCH IF NO
4073	016400	104002				HLT	!ER		;ERROR AFTER READING 2 WORDS
4074	016402	022777	005260	162474	4S:	CMP	#5260, @RSCS1		;IS CS1 CORRECT?
4075	016410	001401				BEQ	10S		;BRANCH IF YES
4076	016412	104002				HLT	!ER		;NO CS1 DID NOT COMPARE
4077									;READ STARTING AT 377777
4078	016414	005012			10S:	CLR	(R2)		;CLEAR LOCATION 400000
4079									;READ TWO WORDS STARTING AT 377777
4080	016416	012777	177776	162464		MOV	#-2, @RSWC		;SETUP WC
4081	016424	005077	162464			CLR	@RSDA		;SETUP DA
4082	016430	012777	177777	162454		MOV	#177777, @RSBA		
4083	016436	012777	000471	162440		MOV	#471, @RSCS1		;CLEAR A 17 SET A16, READ
4084	016444	105777	162434			TSTB	@RSCS1		;WAIT FOR READY
4085	016450	100375				BPL	.-4		
4086	016452	005777	162426			TST	@RSCS1		;ANY ERRORS?
4087	016456	100002				BPL	11S		;BRANCH IF NO
4088	016460	104002				HLT	!ER		;ERROR WHILE READING TWO WORDS
4089	016462	000414				BR	EXTRP		
4090	016464	022777	005270	162412	11S:	CMP	#5270, @RSCS1		;IS CS1 CORRECT?
4091	016472	001401				BEQ	12S		;BRANCH IF YES
4092	016474	104002				HLT	!ER		;CS1 DID NOT COMPARE
4093									;READ STARTING AT 377777
4094	016476	022712	177777		12S:	CMP	#177777, (R2)		;WAS DATA READ INTO LOCATION 400000
4095	016502	001404				BEQ	EXTRP		;CORRECTLY? - BRANCH IF YES
4096	016504	012701	177777			MOV	#177777, GOOD		
4097	016510	011200				MOV	(R2), BAD		
4098	016512	104000				HLT			;DATA COMPARE ERROR AT 400000 IF
4099									;RECEIVED=0 - LOCATION WASN'T ACCESSED
4100	016514	005037	177572		EXTRP:	CLR	@SRD		;TURN OFF MEMORY MANAGEMENT
4101	016520	000401				BR	EXT1		
4102	016522	000240			EXTTRP:	NOP			;UPDATE TEST NUMBERS
4103	016524	012706	000500		EXT1:	MOV	#500, SP		;RESTORE STACK
4104	016530	012767	000006	161246	MEMOUT:	MOV	#6, 4		
4105	016536	005067	161244			CLR	6		

```

*****
;TEST 105 TEST PROGRAM INTERRUPT BY MOVING 300 INTO RSCS1
*****
TST105: SCOPE
QES: CLRDK ; CLEAR ALL DRIVES
      MOV #500,SP ; SETUP STACK
      MOV #PGTRAP,RSEVEC ; SET UP VECTOR
      MOV #340,RSEVCPS ; SET TRAP PS
      MOV #200,R#PS ; SET PS AT PRIORITY 4
      MOV #300,RSCS1 ; THIS SHOULD CAUSE A TRAP
      MOV #500,WORK ; SETUP LOOP
IS: DEC WORK ; DEC LOOP SHOULD
     BNE IS ; INTERRUPE BEFORE LOOP IS DONE
     HLT !CS1 ; SHOULD NEVER GET HERE
     JMP QESDON ; GET OUT

PGTRAP: CMP (6)+(6)+ ; TRAP OK
        CMP #4200,RSCS1 ; DID IE CLEAR?
        BEQ +4 ; YES
        HLT !CS1 ; IE SHOULD BE CLEARED

QESDON:
*****
;TEST 106 TEST THAT DISK DOES NOT INTERRUPT WHEN PS IS AT 5
*****
TST106: SCOPE
INTRS: MOV #500,SP ; SETUP STACK
       MOV TIMSV,TIMES ; RESTORE LOOP COUNTER
       CLRDK ; CLEAR ALL RS REG
       MOV #INT112,RSEVEC ; SET UP INTERRUPT VECTOR
       MOV #340,RSEVCPS ; SET PRIO.
       MOV #240,R#PS ; LOCK OUT ALL INTERRUPTS ABOVE
       MOV R#PS,BAD ; GET PS
       MOV #177777,RSEVC ; SET WORD COUNT TO -1
       MOV R#OBUSV,R#SBA ; LOAD CURRENT ADDRESS
       MOV #161,RSCS1 ; GO WRITE (INTERRUPT ENABLED)
       JSR PC,WAIRY ; WAIT FOR READY
       HLT !CS1 ; NO RDY NEVER CAME UP
       BR INTDON ; RESTART ROUTINE

;PROCESSOR SHOULD NOT TRAP TO INT112
INT112: MOV #240,GOOD ; WHAT PS SHOULD HAVE
        HLT ; GOOD = CORRECT ANS FOR PS
INTDON: ; DONE GET OUT

```

```

4152
4153
4154
4155 016744 104400
4156
4157 016746 012706 000500
4158 016752 104414
4159 016754 012777 017050 162156
4160 016762 012777 000340 162152
4161 016770 012737 000200 177776
4162 016776 013700 177776
4163 017002 012701 000200
4164 017006 012777 177777 162074
4165 017014 013777 001102 162070
4166 017022 012777 000161 162054
4167 017030 005067 162144
4168 017034 005267 162140
4169 017040 001375
4170 017042 104000
4171 017044 104042
4172 017046 000405
4173 017050 022777 004260 162026 INT114:
4174 017056 001401
4175 017060 104001
4176 017062
4177
4178
4179
4180
4181 017062 104400
4182
4183 017064 012706 000500
4184 017070 012737 000200 177776
4185 017076 012777 017154 162034
4186 017104 104414
4187 017106 012777 000340 162026
4188 017114 012777 177777 161772
4189 017122 012777 177600 161760
4190 017130 013777 001102 161754
4191 017136 012777 000161 161740
4192 017144 004767 005106
4193 017150 104042
4194 017152 000406
4195 017154 022777 144260 161722 ERRINT:
4196 017162 001401
4197 017164 104042
4198 017166 022626
4199 017170

```

```

;*****
;TEST 107 TEST THAT DISK DOES INTERRUPT WHEN PS IS AT 4
;*****
TST107: SCOPE
INTR4: MOV #500,SP ;SETUP STACK
        CLRDK ;CLEAR ALL RS REG
        MOV #INT114,RSVEC ;SET UP DISK TRAP VECTOR
        MOV #340,RSVCPS ;SET PRIO.
        MOV #200,PS ;SET PROCESSOR TO PRIORITY 4
        MOV PS,BAD ;GET PS
        MOV #200,GOOD ;GET CORRECT PS
        MOV #177777,RSWC ;SET WORD COUNT TO -1
        MOV #0BUFSV,RSBA ;LOAD CURRENT ADDRESS
        MOV #161,RSCSI ;WRITE (INTERRUPT ENABLE)
        CLR WORK ;WAIT FOR INTERRUPT TO OCCUR
        INC WORK
        BNE -4 ;GOOD=CORRECT PS BAD=WRONG PS
        HLT ;CONT
        BR DONINT ;DID IE CLEAR?
INT114: CMP #4260,RSCSI ;YES
        BEQ +4 ;WHY DID NOT IE CLEAR
        HLT !CSI
DONINT:

```

```

;*****
;TEST 110 TEST INTERRUPT ON ERROR
;*****
TST110: SCOPE
ERINT: MOV #500,SP ;SETUP STACK
        MOV #200,PS ;SET PS AT PRI 4
        MOV #ERRINT,RSVEC ;SET UP INTERRUPT ADD.
        CLRDK ;CLEAR ALL RS REG
        MOV #340,RSVCPS ;SET PRIO.
        MOV #177777,RSDA ;SET RSDA=TO ALL ONES
        MOV #177600,RSWC ;WORD COUNT=-200
        MOV #0BUFSV,RSBA ;CURRENT ADDRESS=OUTBUF
        MOV #161,RSCSI ;WRITE
        JSR PC,WAITRY ;WAIT FOR READY
        HLT !ER!DS ;Y DIDN'T PGM INTERRUPT IS RDY SET?
        BR FINTST ;GET OUT
ERRINT: CMP #144260,RSCSI ;IS CSI RIGHT?
        BEQ +4 ;YES
        HLT !ER!DS
        CMP (6)+,(6)+ ;CLEAR STACK
FINTST:

```

```

4200 :*****
4201 :TEST 111 DYNAMIC FUNCTION TEST
4202 :*****
4203 017170 104400 †TST111: SCOPE
4204 :EXECUTE FUNCTION MODIFY UNIT # AND DO A DRIVE SEARCH
4205 :DRIVE SEARCH WILL ONLY BE DONE IF THERE ARE AT LEAST 2 DRIVES
4206 :2ND DRIVE MAY NOT BE TESTED YET SO IF THIS TEST FAILS CHECK 2ND DRIVE
4207 :BEFORE TRYING TO DEBUG THIS TEST
4208
4209 017172 104414 MODNUM: CLRDK ;CLEAR ALL RS REG
4210 017174 016767 001266 161766 MOV TIMES,TIMSV ;SAVE LOOP COUNT
4211 017202 012767 000010 001256 MOV #10,TIMES ;LOOP ONLY 10 TIMES
4212 017210 005067 161766 CLR WORK1 ;CLEAR WORK LOC.
4213 017214 005003 CLR R3
4214 017216 005004 CLR R4
4215 017220 012702 020120 MOV #DVTAB,R2 ;SETUP TABLE
4216 017224 012767 000401 161746 MOV #401,WORK ;SETUP TO TEST FOR MORE DRIVES
4217 017232 036767 161742 161720 7$: BIT WORK,UNITSV ;IS DRIVE ON SYSTEM?
4218 017240 001403 BEQ 6$ ;NO
4219 017242 020467 161710 CMP R4,UNNUM ;IS THIS THE SAME DRIVE?
4220 017246 001017 BNE 8$ ;NO
4221 017250 005204 6$: INC R4 ;UPDATE 2ND UNIT #
4222 017252 000241 CLC
4223 017254 006167 161720 ROL WORK ;CHECK FOR NEXT DRIVE
4224 017260 103364 BCC 7$ ;NOT DONE YET
4225 017262 032767 000010 161712 BIT #BIT3,WORK1 ;MULTI DRIVE?
4226 017270 001016 BNE 12$ ;YES
4227 017272 016705 161660 MOV UNNUM,RS ;LOAD UNIT NO
4228 017276 005205 INC R5 ;CHANGE IT
4229 017300 042705 177770 BIC #177770,R5 ;CLEAR JUNK
4230 017304 000410 BR 12$
4231 017306 052767 000010 161666 8$: BIS #BIT3,WORK1 ;SET FOUND MULTI DRIVE
4232 017314 010422 MOV R4,(R2)+ ;LOAD UNT # INTO TABLE
4233 017316 005203 INC R3 ;COUNT # OF DRIVES
4234 017320 010367 161652 MOV R3,SAVEE ;SAVE IT
4235 017324 000751 BR 6$
4236 017326 012777 017442 161604 12$: MOV #TSTVEC,RSVEC ;SETUP INT. TRAP
4237 017334 012777 000340 161600 MOV #340,RSVCPS ;SETUP PRIO.
4238 017342 005067 005202 CLR OUTBUF ;CLR TO READ INTO
4239 017346 013777 001102 161536 MOV #0BUF SV,RSBA ;SET UP CURRENT ADDRESS
4240 017354 012777 177000 161526 MOV #-1000,RSWC ;SET WORD COUNT
4241 017362 012777 000060 161524 1$: MOV #60,RSDA ;LOAD DA
4242 017370 012702 020120 MOV #DVTAB,R2 ;GET TABLE
4243 017374 012777 000161 161502 MOV #161,RSCSI ;GO WRITE
4244 017402 005703 TST R3 ;MORE THEN 1 DRIVE?
4245 017404 001003 BNE 13$ ;YES
4246 017406 010577 161474 MOV R5,RSWSC2 ;NO MODIFY UNIT #
4247 017412 000407 BR 14$

```

4248	017414	012277	161466		13\$:	MOV	(R2)+, @RSCS2	: LOAD UNIT#
4249	017420	012777	000131	161456		MOV	#131, @RSCS1	: DO SEARCH
4250	017426	005303				DEC	R3	: DONE ALL DRIVES YET?
4251	017430	001371				BNE	13\$: NO
4252	017432	012737	000200	177776	14\$:	MOV	#200, @#PS	: ENABLE INTERRUPTS
4253	017440	000001			WTDV:	WAIT		
4254	017442	016777	161510	161436	TSTVEC:	MOV	UNNUM, @RSCS2	: GET 1ST DRIVE
4255	017450	017700	161430			MOV	@RSCS1, BAD	: GET CS1
4256	017454	042700	100000			BIC	#BIT15, BAD	: CLEAR SC
4257	017460	012701	004260			MOV	#4260, GOOD	: GET CORRECT ANS
4258	017464	020100				CMP	GOOD, BAD	: IS CS1 CORRECT?
4259	017466	001402				BEQ	4\$: NO! X-FER OK
4260	017470	104000				HLT		: CS1 SHOULD = 14270 OR 4270
4261	017472	104140				HLT	!DS!AS	
4262	017474	005777	161410		4\$:	TST	@RSCS1	: TEST WC
4263	017500	001401				BEQ	.+4	: WORD COUNT DID OVERFLOW
4264	017502	104010				HLT	!WC	: SHOULD = 0
4265	017504	016701	161446			MOV	UNNUM, GOOD	: GET CORRECT
4266	017510	052701	000100			BIS	#100, GOOD	: ANS OF CS2
4267	017514	017700	161366			MOV	@RSCS2, BAD	: GET CS2
4268	017520	020100				CMP	GOOD, BAD	: IS CS2 CORRECT?
4269	017522	001401				BEQ	.+4	: YES
4270	017524	104000				HLT		: GOOD = CORRECT ANS FOR CS2
4271	017526	017700	161360			MOV	@RSBA, BAD	: FETCH CURRENT ADDRESS
4272	017532	013701	001102			MOV	@#OBUFSV, GOOD	: WHAT RSBA SHOULD EQUAL
4273	017536	062701	002000			ADD	#2000, GOOD	: UPDATE IT
4274	017542	020001				CMP	BAD, GOOD	: IS RSBA CORRECT
4275	017544	001401				BEQ	.+4	: YES EXECUTE CONTINUE
4276	017546	104000				HLT		: RSBA FAILED TO INCREMENT
4277	017550	005767	161412		2\$:	TST	RS04DT	: IS THIS A RS04?
4278	017554	001005				BNE	5\$: YES
4279	017556	022777	000070	161330		CMP	#70, @RSDA	: IS DA CORRECT?
4280	017564	001406				BEQ	3\$: YES
4281	017566	104104				HLT	!DA!AS	: DA NOT CORRECT
4282	017570	022777	000064	161316	5\$:	CMP	#64, @RSDA	: WAS RSDA INCREMENTED
4283	017576	001401				BEQ	.+4	: RSDA OK
4284	017600	104046				HLT	!DA!ER!DS	: RSDA SHOULD CONTAIN A 64
4285	017602	012777	040000	161274	3\$:	MOV	#TRE, @RSCS1	: CLEAR ALL ERRORS IF ANY
4286	017610	032767	000010	161364		BIT	#BIT3, WORK1	: MULTI DRIVE?
4287	017616	001461				BEQ	WTDV1	: NO
4288	017620	012777	017644	161312	1\$:	MOV	#TTVEC, @RSVEC	: SETUP INT FOR NEXT DRIVE
4289	017626	012716	017642			MOV	#WTDV2, (SP)	: GET WAIT
4290	017632	012777	000100	161244		MOV	#100, @RSCS1	: SET IE
4291	017640	000002				RTI		: RETURN
4292	017642	000001			WTDV2:	WAIT		


```

:SERVICE ROUTINE FOR SEARCH FUNCTIONS
↑TVEC: CLR R2 ;CLEAR UNIT #
4293 017644 005002
4294 017646 000241
4295 017650 012767 000401 161322
4296 017656 036777 161316 161236 1S: BIT #401,WORK ;DID THIS DRIVE INT?
4297 017664 001006 161316 161236 1S: BNE WORK,ARSAS ;YES
4298 017666 005202 INC R2 ;UPDATE UNIT #
4300 017670 000241
4301 017672 006167 161302 ROL WORK
4302 017676 103367 BCC 1S
4303 017700 104100 HLT AS ;WHY DID WE INT WITH NO ATA??
4304 017702 010277 161200 2S: MOV R2,ARSCS2 ;GET DRIVE
4305 017706 022777 110600 161202 2S: CMP #110600,ARSDS ;DID PIP CLEAR?
4306 017714 001401 BEQ +4 ;YES
4307 017716 104140 HLT DS:AS ;PIP BIT DID NOT CLEAR
4308 017720 022777 104230 161156 2S: CMP #104230,ARSCS1 ;DID SC SET?
4309 017726 001401 BEQ +4 ;YES
4310 017730 104140 HLT AS:DS ;SC DID NOT SET
4311 017732 005367 161240 DEC SAVEE ;COUNT # OF INT
4312 017736 001411 BEQ WTDV1 ;DONE YET?
4313 017740 016777 161234 161154 MOV WORK,ARSAS ;CLEAR AS
4314 017746 012777 000100 161130 MOV #100,ARSCS1 ;SET IE
4315 017754 012716 017642 MOV #WTDV2,(SP) ;RETURN TO WAIT
4316 017760 000002
4317 017762 012737 000340 177776 WTDV1: MOV #340,#PS
4318 017770 012706 000500 MOV #500,SP ;CLEAR STACK
4319 017774 016777 161142 161136 MOV RSVCP5,ARSEV ;RESTORE INT VECTOR
4320 020002 005077 161134 CLR ARSVCP5
4321 020006 104400 MODDON: SCOPE ;DONE
4322 020010 016767 161154 000450 MOV TIMSV,TIMES ;RESTORE LOOP COUNT
4323 020016 012737 000340 177776 MOV #340,#PS ;RESTORE PS
4324 020024 012767 000001 160746 MOV #1,ICNT ;FUGE TEST NUMBERS
4325 020032 000137 001606 OUT: JMP #TRYNX ;TEST NEXT DRIVE
4326 .SBTTL ;DONE - BELL AND SCOPE ROUTINE
4327
4328 020036 104400 DONE: SCOPE ;TERMINATING SCOPE FOR LOOPING
4329 020040 062767 000001 160740 ADD #1,PCNT+2 ;ADD 1 TO THE PASS COUNT
4330 020046 005567 160732 ADC PCNT ;MAKE IT DOUBLE PREC.
4331 020052 032737 002000 177570 BIT #SW10,#SWR ;RING THE BELL?
4332 020060 001004 BNE 1S ;NO!
4333 020062 104402 020066 TYPE .+2 ;ASCIZ <BELL><177>
4334 020072 013700 000042 4S: MOV #42,R0 ;GET MONITOR ADDRESS
4335 020076 001405 BEQ SEND1 ;IF NONE
4336 020100 000005 RESET
4337 020102 004710 SENDAD: JSR 7,(0) ;GO TO MONITOR
4338 020104 000240 000240 000240 JSR 240,240,240 ;SAVE ROOM FOR ACT11
4339 020112 000137 001206 SEND1: JMP #BEGIN ;RETURN
4340
4341 020116 000000 .TBIT: 0 ;T BIT FLAG
4342
4343 020120 000010 DVTAB: .BLKW 10

```

.SBTTL STYPE - TTY TYPEOUT ROUTINE

;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ADR" - TYPES THE
;MESSAGE STARTING IN LOCATION "ADR:" 2) "TYPE CHAR" - TYPES
;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE") - TYPES
;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
;IS IN FILCHR+1.

4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383

020140 010446
020142 010546
020144 017605 000004
020150 032705 177400
020154 001002
020156 016605 000004
020162 105715
020164 001423
020166 122715 000012
020172 001012
020174 116704 160615
020200 116777 160610 160612 5\$:
020206 105777 160604
020212 100375
020214 005304
020216 001370
020220 112577 160574 4\$:
020224 105777 160566
020230 100375
020232 000753
020234 017646 000004 2\$:
020240 062766 000002 000006
020246 022666 000004
020252 001006
020254 062705 000002
020260 042705 000001
020264 010566 000004
020270 012605 3\$:
020272 012604
020274 000002

.TYPE: MOV R4,-(6) ;SAVE R4
MOV R5,-(6) ;SAVE R5
MOV @4(6),R5 ;GET ADDRESS TO BE TYPED
BIT #177400,R5 ;IS IT A TYPED?
BNE 1\$;NO
MOV 4(6),R5 ;GET ADDRESS OF CHARACTER
1\$: TSTB (R5) ;TERMINATOR?
BEQ 2\$;GET OUT IF SO
CMPB #12,(R5) ;IS THE CHAR A LINE FEED
BNE 4\$;NO - GET OUT
MOVB FILCHR+1,R4 ;GET THE FILL COUNT
5\$: MOVB FILCHR,@TPB ;TYPE A FILLER
TSTB @TPS ;DONE YET?
BPL .-4 ;NO - WAIT
DEC R4 ;DEC COUNT
BNE 5\$;LOOP UNTIL 0
4\$: MOVB (R5)+,@TPB ;LOAD AND TYPE THE CHARACTER
TSTB @TPS ;IS THE PRINTER READY
BPL .-4 ;WAIT UNTIL IT IS
BR 1\$;GET THE NEXT CHARACTER
2\$: MOV @4(6),-(6) ;GET ADDRESS TO BE TYPED
ADD #2,6(6) ;ADD 2 TO THE ADDRESS
CMP (6)+,4(6) ;IS IT .+2?
BNE 3\$;NO
ADD #2,R5 ;ADD 2 TO THE ADDRESS
BIC #1,R5 ;BACK UP TO AN EVEN BYTE
MOV R5,4(6) ;RESTORE ADDRESS
3\$: MOV (6)+,R5 ;RESTORE R5
MOV (6)+,R4 ;RESTORE R4
RTI ;RETURN

```

4384          .SBTTL          $SCOPE - SCOPE LOOP HANDLER
4385
4386          ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
4387          ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
4388          ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
4389          ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
4390
4391 020276 032737 000400 177570 .SCOPE: BIT      #SW8, @#SWR      ;LOOP ON SPEC. TEST?
4392 020304 001404          BEQ      1$              ;NO LOOP ON SPEC. TEST
4393 020306 123767 177570 160464          CMPB   @#SWR, ICNT    ;ON RIGHT TEST? *SW7-0*
4394 020314 001453          BEQ      .OVER          ;NOT RIGHT TEST
4395 020316 032737 040000 177570 1$: BIT      #SW14, @#SWR   ;LOOP ON TEST?
4396 020324 001045          BNE      .KIT              ;LOOP ON TEST IS SET
4397 020326 000416          BR       3$              ;SKIP - NOP FOR XOR TESTER
4398 020330 013746 000004          MOV     @#4, -(6)     ;PUSH @#4 ON STACK
4399 020334 012737 020354 000004          MOV     #4$, @#4     ;SET FOR TIMEOUT
4400 020342 005737 177060          TST    @#177060     ;ERROR ON XOR?
4401 020346 012637 000004          MOV     (6)+, @#4    ;POP STACK INTO @#4
4402 020352 000422          BR     .SVLAD        ;NO ERROR - GO TO NEXT TEST
4403 020354 022626          4$: CMP     (6)+, (6)+ ;CLEAR STACK
4404 020356 012637 000004          MOV     (6)+, @#4    ;POP STACK INTO @#4
4405 020362 000426          BR     .KIT          ;ERROR - LOOP ON TEST
4406 020364 032737 004000 177570 3$: BIT      #SW11, @#SWR   ;KILL ITERATIONS
4407 020372 001012          BNE     .SVLAD        ;YES - KILL ITERATIONS
4408 020374 105767 160401          TSTB   ICNT+1        ;FIRST ONE?
4409 020400 001404          BEQ     2$          ;BRANCH IF FIRST
4410 020402 126767 000060 160371          CMPB   TIMES, ICNT+1 ;DONE?
4411 020410 003013          BGT     .KIT          ;BRANCH IF NOT
4412 020412 112767 000001 160361 2$: MOVB   #1, ICNT+1    ;FIRST ITERATION
4413 020420 105267 160354          .SVLAD: INCB   ICNT   ;COUNT TEST NUMBERS
4414 020424 011667 160360          MOV     (6), LAD     ;SAVE LOOP ADDRESS
4415 020430 016737 160344 177570          MOV     ICNT, @#DISPLAY ;DISPLAY TEST NO. AND ITERATION COUNT
4416 020436 000002          RTI                    ;RETURN
4417
4418 020440 105267 160335          .KIT: INCB   ICNT+1   ;INC THE ITERATION COUNT
4419 020444 016737 160330 177570 .OVER: MOV     ICNT, @#DISPLAY ;SET UP DISPLAY
4420 020452 005767 160332          TST    LAD           ;FIRST ONE?
4421 020456 001760          BEQ     .SVLAD        ;YES
4422 020460 016716 160324          MOV     LAD, (6)     ;FUDGE RETURN ADDRESS
4423 020464 000002          RTI                    ;FIXES PS
4424
4425 020466 000100          TIMES: 100          ;RUN 100 TIMES

```

```

4426          .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)
4427
4428          ; THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
4429          ; ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
4430          ; AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
4431          ; "HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
4432          ; (HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADITIONAL TYPEOUTS.
4433
4434 020470 032737 002000 177570 .HLT: BIT      #SW10,@#SWR      ;BELL ON ERROR?
4435 020476 001402          BEQ      1$              ;NO - SKIP
4436 020500 104402          TYPE     ,BELL          ;RING BELL
4437 020504 005267 160272          INC      ERRORS        ;COUNT THE NUMBER OF ERRORS
4438 020510 032737 C30000 177570 1$: BIT      #SW13,@#SWR      ;SKIP TYPEOUT IF SET
4439 020516 001025          BNE      2$              ;SKIP TYPEOUTS
4440 020520 104402 020524          TYPE     ,.+2          ;.ASCIZ <15><12>
4441 020530 011667 160256          MOV      (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
4442 020534 162767 000002 160250  SUB      #2,HLTADR      ;FUDGE ADDRESS
4443 020542 117767 160244 000054  MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUMENT
4444 020550 016746 160236          MOV      HLTADR,-(6)    ;PUT HLTADR ON STACK
4445 020554 104404          TYPEO    TYPE          ;TYPE STACK IN OCTAL
4446 020556 104402 020562          TYPE     ,.+2          ;.ASCIZ " "
4447 020566 004767 002752          JSR      PC,RSREG      ;GO TO USER ERROR ROUTINE
4448 020572 005737 177570 2$: TST      @#SWR          ;HALT ON ERROR
4449 020576 100001          BPL      .+4            ;SKIP IF CONTINUE
4450 020600 000000          HALT                    ;HALT ON ERROR!
4451 020602 032737 001000 177570  BIT      #SW9,@#SWR      ;CHECK FOR INHIBIT LOOP ON ERROR
4452 020610 001003          BNE      3$              ;SKIP IF LOOP ON ERROR
4453 020612 105067 160163          CLRB    ICNT+1         ;CLEAR ITERATION COUNT
4454 020616 000002          RTI                      ;RETURN
4455 020620 000167 177614 3$: JMP      .KIT          ;LOOP ON TEST UNTIL NO ERRORS
4456
4457 020624 000000          .HLTCT: 0              ;HLT ARGUMENT

```

```

4458
4459
4460
4461
4462
4463
4464 020626 012767 170101 000160 .SBTTL SOCIAL - OCTAL TYPEOUT ROUTINE
4465 020634 000411
4466 020636 112767 000001 000150 :THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
4467 020644 000402 :ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
4468 020646 005067 000142 :16 BITS. IT IS CALLED VIA THE TYOCT, TYPBIT, OR TYPCS MACPO'S.
4469 020652 112767 177772 000135 .TYPE3: MOV #170101,.PR ;SET BIT FLAG AND 16. CHARACTER COUNT
4470 020660 .PTIT: BR .PTIT ;NOW TYPE IT IN BIT FORM
4471 020660 010446 MOV R4,-(6) ;SET ZERO FILL SWITCH
4472 020662 010546 MOV R5,-(6) ;SKIP
4473 020664 016605 000010 000150 .TYPE0: MOV #1,.PR ;SUPPRESS LEADING ZERO'S
4474 020670 012704 021016 MOV 10(6),R5 ;SET COUNT
4475 020674 105014 CLRB (4) ;PUSH R4 ON STACK
4476 020676 000411 BR .PRF ;PUSH R5 ON STACK
4477 020700 105014 .PRL: CLRB (4) ;GET THE DATA
4478 020702 032767 000100 000104 BIT #100,.PR ;SET POINTER TO FIRST ASCII CHAR.
4479 020710 001004 BNE .PRF ;CLEAR FIRST BYTE
4480 020712 006105 ROL R5 ;ROTATE FIRST BIT
4481 020714 106114 ROLB (4) ;CLEAR BYTE OF CHARACTER
4482 020716 006105 ROL R5 ;BIT TYPING MODE?
4483 020720 106114 ROLB (4) ;YES - SKIP 2 ROTATES
4484 020722 006105 .PRF: ROL R5 ;ROTATE BIT INTO C
4485 020724 106114 ROLB (4) ;PACK IT
4486 020726 105714 TSTB (4) ;ROTATE BIT INTO C
4487 020730 001402 BEQ .+6 ;PACK IT
4488 020732 105267 000056 INCB .PR ;IS IT ZERO?
4489 020736 105767 000052 TSTB .PR ;SKIP INC
4490 020742 001402 BEQ .+6 ;SET FILL SWITCH
4491 020744 152724 000060 BISB #'0,(4)+ ;CHECK FILL SWITCH
4492 020750 105267 000041 INCB .PR+1 ;SKIP BITSET
4493 020754 001351 BNE .PRL ;MAKE INTO ASCII CHAR
4494 020756 022704 021016 CMP #.PR+2,R4 ;INC COUNT
4495 020762 001002 BNE .+6 ;REPEAT
4496 020764 112724 000060 MOVB #'0,(4)+ ;EMPTY BUFFER?
4497 020770 105014 CLRB (4) ;SKIP IF NOT
4498 020772 104402 021016 TYPE .PR+2 ;LOAD 1 ZERO
4499 020776 012605 MOV (6)+,R5 ;NULL TERMINATOR
4500 021000 012604 MOV (6)+,R4 ;TYPE IT
4501 021002 016666 000002 000004 MOV 2(6),4(6) ;POP STACK INTO R5
4502 021010 012616 MOV (6)+,(6) ;POP STACK INTO R4
4503 021012 000002 RTI ;GET RID OF
4504 ;DATA WORD
4505 021014 000012 .PR: .BLKW 12 ;RETURN
;COUNT, SWITCH, AND OUTPUT BUFFER

```

.SBTTL \$POWER - POWER DOWN AND UP ROUTINES

: THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
: THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
: WAIT FOR POWER TO GO DOWN AND BE RESTORED.
: IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
: THE PROGRAM WILL HALT AT '.ILLUP'.

4506					
4507					
4508					
4509					
4510					
4511					
4512					
4513					
4514	021040	012777	021164	000124	
4515	021046	012777	000340	000120	
4516	021054	010046			
4517	021056	010146			
4518	021060	010246			
4519	021062	010346			
4520	021064	010446			
4521	021066	010546			
4522	021070	010667	000074		
4523	021074	012777	021104	000070	
4524	021102	000000			
4525					
4526	021104	016706	000060		
4527	021110	005001			
4528	021112	005201			
4529	021114	001376			
4530	021116	012605			
4531	021120	012604			
4532	021122	012603			
4533	021124	012602			
4534	021126	012601			
4535	021130	012600			
4536	021132	012737	021040	000024	
4537	021140	012737	000340	000026	
4538	021146	104402	021152		
4539	021162	000002			
4540					
4541	021164	000000			
4542	021166	000776			
4543					
4544	021170	000000			
4545	021172	000024	000026		

```

.POWER: MOV      #.ILLUP, @.PUVEC ; SET FOR FAST UP
        MOV      #340, @.PUVECS+2 ; PRIO:7
        MOV      R0, -(6) ; PUSH R0 ON STACK
        MOV      R1, -(6) ; PUSH R1 ON STACK
        MOV      R2, -(6) ; PUSH R2 ON STACK
        MOV      R3, -(6) ; PUSH R3 ON STACK
        MOV      R4, -(6) ; PUSH R4 ON STACK
        MOV      R5, -(6) ; PUSH R5 ON STACK
        MOV      SP, .SAVR6 ; SAVE SP
        MOV      #.POWUP, @.PUVEC ; SET UP VECTOR
        HALT ; WAIT FOR PF

```

```

.PCWLP: MOV      .SAVR6, SP ; GET SP
        CLR      R1 ; WAIT LOOP FOR THE TTY
1$: INC      R1 ; WAIT FOR THE INC
        BNE     1$ ; OF WORD
        MOV      (6)+, R5 ; POP STACK INTO R5
        MOV      (6)+, R4 ; POP STACK INTO R4
        MOV      (6)+, R3 ; POP STACK INTO R3
        MOV      (6)+, R2 ; POP STACK INTO R2
        MOV      (6)+, R1 ; POP STACK INTO R1
        MOV      (6)+, R0 ; POP STACK INTO R0
        MOV      #.POWER, @#24 ; SET UP THE POWER DOWN VECTOR
        MOV      #340, @#26 ; PRIO:7
        TYPE    ..+2 ; .ASCIZ (15)(12)"POWER"
        RTI ; RETURN

```

```

.ILLUP: HALT ; THE POWER UP SEQUENCE WAS STARTED
        BR      .-2 ; BEFORE THE POWER DOWN WAS COMPLETE

```

```

.SAVR6: 0 ; PUT THE SP HERE
.PUVEC: 24,26 ; POWER UP VECTOR

```

.SBTTL SRDOCT - OCTAL INPUT ROUTINE

: THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
: IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.

4546									
4547									
4548									
4549									
4550									
4551	021175	011646			.RDOCT:	MOV	(6),-(6)		: MOVE THE PC
4552	021200	016666	000004	000202		MOV	4(6),2(6)		: MOVE THE PS
4553	021206	010146				MOV	R1,-(6)		: PUSH R1 ON STACK
4554	021210	010246				MOV	R2,-(6)		: PUSH R2 ON STACK
4555	021212	010346				MOV	R3,-(6)		: PUSH R3 ON STACK
4556	021214	104412			4S:	RDLIN			: READ A LINE INTO INPUT
4557	021216	005001				CLR	R1		: INIT DATA WORD
4558	021220	012703	021420			MOV	#INPUT,R3		: INIT POINTER
4559	021224	112302			1S:	MOVB	(3)+,R2		: GET A BYTE
4560	021226	001417				BEG	2S		: GET OUT IF ZERO
4561	021230	122702	000060			CMPB	#'0,R2		: CHECK FOR 0 OR GREATER
4562	021234	003022				BGT	3S		: ERROR - LESS THAN 0
4563	021236	122702	000067			CMPB	#'7,R2		: CHECK FOR 7 OR LESS
4564	021242	002417				BLT	3S		: ERROR - GREATER THAN 7
4565	021244	006002				ROR	R2		: GET
4566	021246	006002				ROR	R2		: INTO
4567	021250	006002				ROR	R2		: POSITION
4568	021252	006101				ROL	R1		: FIRST BIT
4569	021254	006102				ROL	R2		: GET
4570	021256	006101				ROL	R1		: SECOND BIT
4571	021260	006102				ROL	R2		: GET
4572	021262	006101				ROL	R1		: THIRD BIT
4573	021264	000757				BR	1S		: LOOP
4574	021266	010166	000012		2S:	MOV	R1,12(6)		: SAVE THE RESULT
4575	021272	012603				MOV	(6)+,R3		: POP STACK INTO R3
4576	021274	012602				MOV	(6)+,R2		: POP STACK INTO R2
4577	021276	012601				MOV	(6)+,R1		: POP STACK INTO R1
4578	021300	000002				RTI			: RETURN
4579									
4580	021302				3S:				
4581	021302	104402	021306			TYPE	4S+2		: .ASCIZ "'?'(15)<12>
4582	021312	000740				BR	4S		: TRY AGAIN

4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632

021314 010546
021316 012705 021420
021322 022705 021440
021328 001412
021330 105737 177560
021334 100375
021336 113715 177562
021342 142715 000200
021346 122715 000177
021352 001005
021354
021354 104402 021360
021364 000754
021366 111527 000000
021372 104402 021370
021376 122725 000015
021402 001347
021404 105065 177777
021410 104402 000012
021414 012605
021416 000002
021420 000020

.SBTTL SRDLIN - TTY INPUT ROUTINE
: THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
: INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
: INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
: THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.

```
.RDLIN: MOV R5, -(6) ;SAVE R5
15: MOV #INPUT, R5 ;GET ADDRESS
25: CMP #INPUT+16., R5 ;BUFFER FULL?
; YES - TYPE "?"
BEQ 45 ;WAIT FOR
TSTB @#177560 ;A CHARACTER
BPL -4 ;GET CHARACTER
MOVB @#177562, (5) ;GET RID OF JUNK
BICB #200, (5) ;IS IT A RUBOUT
CMPB #177, (5) ;SKIP IF NOT
BNE 35
45: TYPE 15+2 ;ASCIZ "?"(15)(12)
BR 15 ;ZAP THE BUFFER AND LOOP
35: MOVB (5), #0 ;SET UP FOR TYPING
TYPE 35+2 ;ECHO IT
CMPB #15, (5)+ ;CHECK FOR RETURN
BNE 25 ;LOOP IF NOT RETURN
CLRB -1(5) ;ZAP RETURN (THE 15)
TYPE 12 ;TYPE A LINE FEED
MOV (6)+, R5 ;RESTORE R5
RTI ;RETURN
```

INPUT: .BLKB 16. ;TTY INPUT AREA
.SBTTL STRAP - TRAP HANDLER

: THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
: SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
: THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
: FOLLOW THIS MACRO.

```
.TRAP: MOV (6), -(6) ;GET ADDRESS OF TRAP +2
SUB #2, (6) ;MAKE IT ADDRESS OF TRAP
MOV @ (6), (6) ;GET TRAP INSTRUCTION
ADD #.TRAP+2-TRAP, (6) ;GET DATA AND MAKE IT AN OFFSET
.TRP: MOV @ (6)+, PC ;GO TO PROPER SUBROUTINE
```

.SCOPE = TRAP+0 (104400)
.TYPE = TRAP+2 (104402)
.TYPE0 = TRAP+4 (104404)
.TYPES = TRAP+6 (104406)
.RDOCT = TRAP+10 (104410)
.RDLIN = TRAP+12 (104412)
.CLROK = TRAP+14 (104414)


```

4633          ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
4634          ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
4635
4636          ;WORD COUNT REGISTER
4637 021476 016777 15605F 157404 SRSWC: MOV SWR, @R5WC ;MOV SWR INTO WORD COUNT REG
4638 021504 017737 15740C 177570      MOV @R5WC, @#DISPLAY ;DISPLAY IN LIGHTS
4639 021512 000771      BR SRSWC
4640
4641          ;CURRENT ADDRESS REGISTER
4642 021514 016777 156050 157370 SRSBA: MOV SWR, @R5BA ;MOV SWR INTO CURRENT ADDR REG
4643 021522 017737 157364 177570      MOV @R5BA, @#DISPLAY ;SHOW IN LIGHTS
4644 021530 000771      BR SRSBA
4645
4646          ;DISK ADDRESS REGISTER
4647 021532 016777 156032 157354 SRSDA: MOV SWR, @R5DA ;MOV SWR INTO DISK ADDR REG
4648 021540 017737 157350 177570      MOV @R5DA, @#DISPLAY ;SHOW IN LIGHTS
4649 021546 000771      BR SRSDA
4650
4651          ;DRIVE STATUS REGISTER
4652 021550 016777 156014 157340 SRSDS: MOV SWR, @R5DS ;MOV SWR INTO DRIVE STATUS
4653 021556 017737 157334 177570      MOV @R5DS, @#DISPLAY ;SHOW IN LIGHTS
4654 021564 000771      BR SRSDS
4655
4656          ;DRIVE ERROR REGISTER
4657 021566 016777 155776 157324 SRSER: MOV SWR, @R5SER ;LOAD ER REG
4658 021574 017737 157320 177570      MOV @R5SER, @#DISPLAY ;DISPLAY IT IN LIGHTS
4659 021602 000771      BR SRSER ;LOOP
4660
4661          ;WATCH LOOK AHEAD REGISTER
4662 021604 017737 157314 177570 SRSLA: MOV @R5SLA, @#DISPLAY ;SHOW IN LIGHTS
4663 021612 000774      BR SRSLA

```

```

4664                                     :RSCS2 REGISTER
4665 021614 016777 155750 157264 SRCS2: MOV SWR,@RSCS2 ;LOAD CS2
4666 021622 017737 157260 177570      MOV @RSCS2,@#DISPLAY ;DISPLAY IT
4667 021630 000771      BR SRCS2
4668
4669                                     :RSAS REGISTER
4670 021632 016777 155732 157262 SRAS: MOV SWR,@RSAS ;LOAD RSAS
4671 021640 017737 157256 177570      MOV @RSAS,@#DISPLAY ;DISPLAY IT
4672 021646 000771      BR SRAS
4673
4674                                     :RSMR REGISTER
4675 021650 016777 155714 157252 RSMRR: MOV SWR,@RSMR ;LOAD RSMR
4676 021656 017737 157246 177570      MOV @RSMR,@#DISPLAY ;DISPLAY IT
4677 021664 000771      BR RSMRR
4678
4679                                     :DISK CONTROL STATUS REGISTER
4680 021666 012737 000340 177776 SRSCS1: MOV #340,@#PS ;LOCK UP INTERRUPTS
4681 021674 012777 177777 157206      MOV #177777,@RSWC ;SET WORD COUNT -1 WORD
4682 021702 013777 001102 157202      MOV @#BUFSV,@RSBA ;SET UP CURRENT ADDRESS
4683 021710 016777 155654 157166      MOV SWR,@RSCS1 ;MOV SWR INTO CONTROL REG
4684 021716 032777 000001 157160      BIT #BIT0,@RSCS1 ;IS FUNCTION BITS SET
4685 021724 001760      BEQ SRSCS1 ;FUNCTION BITS NOT SET
4686 021726 105777 157152      DKBUSY: TSTB @RSCS1 ;TEST FOR DISK READY
4687 021732 100375      BPL DKBUSY ;DISK STILL NOT READY
4688 021734 000754      BR SRSCS1 ;DISK NOT BUSY SECT NEW CR

```

4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734

021736 012706 000500
021742 000240
021744 104402 021750
021770 104410
021772 012667 157160
021776 104402 022002
022036 104410
022040 012667 157134
022044 005167 157130
022050 104402 022054
022104 104410
022106 012667 002436
022112 016767 155452 157064
022120 104414
022122 052777 000010 156756
022130 016767 155434 157044
022136 042767 170000 157036
022144 016777 157032 156742
022152 016777 157022 156730
022160 032767 010000 155402
022166 001412
022170 012777 024552 156714
022176 012777 000071 156700
022204 105777 156674
022210 100375
022212 000407
022214 012777 024550 156670
022222 012777 000061 156654
022230 000765
022232 017737 156646 177570
022240 005777 156640
022244 100001
022246 104014
022250 026767 155314 156726
022256 001315
022260 000717

```
: THIS ROUTINE GIVES THE OPERATOR THE ABILITY TO  
: SELECT DA, WC, UNIT # AND DESIRED PATTERN. PATTERN = NUMBER TYPED  
: WITH SW12 SET THE PROGRAM WILL LOOP ON A READ WITH LOC OUTBUF+2 AS  
: THE BA ADDR. WITH BIT12 0 IN THE SWR THE PROGRAM  
: WILL WRITE WITH OUTBUF AS THE BA ADDR. BAI IS ALWAYS SET  
: SWITCHES 0 TO 11 WILL DETERMINE THE DA  
: EXAMPLE: TYPE UNIT # 5  
: TYPE POSITIVE (OCTAL) WC 64  
: TYPE PATTERN DESIRED 1252525  
: :  
TKSEL: MOV #500, SP ;SET STACK  
NOP  
TYPE ,.+2 ;.ASCIZ <15><12>"TYPE UNIT # "  
RDOCT  
MOV (6)+, UNNUM  
TYPE ,.+2 ;.ASCIZ <15><12>"TYPE POSITIVE (OCTAL) WC "  
RDOCT  
MOV (6)+, WORK  
COM WORK  
TYPE ,.+2 ;.ASCIZ <15><12>"TYPE PATTERN DESIRED "  
RDOCT  
MOV (6)+, OUTBUF  
TK1: MOV SWR, WORK2 ;SAVE SWR  
TK2: CLDK ;CLEAR ALL RS REG  
BIS #BIT3, @RSCS2 ;SET BAI  
MOV SWR, WORK1 ;GET SWR FOR DSK ADDR  
BIC #170000, WORK1 ;CLEAR UNIT #  
TKS: MOV WORK1, @RSDA ;LOAD THE DA  
MOV WORK, @RSMC ;LOAD WORD COUNT  
BIT #BIT12, SWR ;READ?  
BEQ WTE ;NO  
MOV #OUTBUF+2, @RSBA ;LOAD CURRENT ADDRESS  
MOV #71, @RSCS1 ;GO AND READ  
WT: TSTB @RSCS1 ;TEST FOR READY  
BPL .-4  
BR SWRCHG  
WTE: MOV #OUTBUF, @RSBA  
MOV #61, @RSCS1  
BR WT  
SWRCHG: MOV @RSCS1, @DISPLAY ;DISPLAY CS1  
TST @RSCS1 ;ANY ERRORS?  
BPL IS ;NO  
HLT !DA!WC  
IS: CMP SWR, WORK2 ;DID SWR CHANGE?  
BNE TK1 ;YES  
BR TK2 ;NO
```

4735
4736
4737
4738 022262 104400
4739
4740 022264
4741 022264 104402 022270
4742 022330 000000
4743 022332 013767 177570 156616
4744 022340 104414
4745 022342 005067 002202
4746 022346 012777 024550 156536
4747 022354 012777 007700 156532
4748 022362 012777 177777 156520
4749 022370 012777 000061 156506
4750 022376 105777 156502 6S:
4751 022402 100375
4752 022404 104402 022410
4753 022456 000000
4754 022460 012777 000000 156426 1S:
4755 022466 022777 014600 156422
4756 022474 001401
4757 022476 104044
4758 022500 012777 000100 156406
4759 022506 022777 010600 156402
4760 022514 001401
4761 022516 104044
4762 022520 104402 022524
4763 022560 000000
4764 022562 022777 014600 156326 2S:
4765 022570 001401
4766 022572 104044
4767 022574 012777 000300 156312
4768 022602 022777 010600 156306
4769 022610 001401
4770 022612 104044
4771 022614 104402 022620
4772 022654 000000
4773 022656 022777 014600 156232 3S:
4774 022664 001401
4775 022666 104044
4776 022670 012777 000700 156216
4777 022676 022777 010600 156212
4778 022704 001401
4779 022706 104044
4780 022710 104402 022714
4781 022750 000000

```

*****
;TEST 112 WRITE LOCK TEST
*****
†ST112: SCOPE

WRTLCK:
TYPE      ,.+2      ;.ASCIZ <15><12>"LOAD SW WITH UNIT # AND CONT"
HALT
MOV       @#SWR,UNNUM ;GET UNIT #
CLDRK
CLR       OUTBUF     ;CLEAR ALL REG
MOV       #OUTBUF,@RSBA ;PUT A 0 INTO DATA BUFFER
MOV       #7700,@RSDA ;SETUP REG TO
MOV       #-1,@RSWC   ;TO A WRITE
MOV       #61,@RSCS1
TSTB     @RSCS1      ;WAIT FOR DCNE
BPL      6S
TYPE      ,.+2      ;.ASCIZ <15><12>"SET WRITE LOCK ENABLE AND CONTINUE"
HALT
MOV       #0,@RSDA
CMP       #14600,@RSDS 1S:
BEQ      .+4
HLT      !DS!DA        ;DS SHOULD=14600
MOV       #100,@RSDA
CMP       #10600,@RSDS
BEQ      .+4
HLT      !DS!DA        ;DS SHOULD=10600
TYPE      ,.+2      ;.ASCIZ <15><12>"SET WRT LOC SW 0 AND CONT"
HALT
CMP       #14600,@RSDS 2S:
BEQ      .+4
HLT      !DS!DA        ;DS SHOULD=14600
MOV       #300,@RSDA
CMP       #10600,@RSDS
BEQ      .+4
HLT      !DS!DA        ;DS SHOULD=10600
TYPE      ,.+2      ;.ASCIZ <15><12>"SET WRT LOC SW 1 AND CONT"
HALT
CMP       #14600,@RSDS 3S:
BEQ      .+4
HLT      !DS!DA        ;DS SHOULD=14600
MOV       #700,@RSDA
CMP       #10600,@RSDS
BEQ      .+4
HLT      !DS!DA        ;DS SHOULD=10600
TYPE      ,.+2      ;.ASCIZ <15><12>"SET WRT LCK SW 2 AND CONT"
HALT

```

4792	022752	022777	014600	156136	4\$:	CMP	#14600, ARSDS	
4783	022760	001401				BEQ	.+4	
4784	022762	104044				HLT	!DS!DA	;DS SHOULD=14600
4785	022764	012777	001700	156122		MOV	#1700, ARSDA	
4786	022772	022777	010600	156116		CMP	#10600, ARSDS	
4787	023000	001401				BEQ	.+4	
4788	023002	104044				HLT	!DS!DA	;DS SHOULD=10600
4789	023004	104402	023010			TYPE	,.+2	;.ASCIZ <15><12>"SET WRT LCK SW 3 AND CONT"
4790	023044	000000			HALT:	HALT		
4791	023046	022777	014600	156042		CMP	#14600, ARSDS	
4792	023054	001401				BEQ	.+4	
4793	023056	104044				HLT	!DS!DA	;DS SHOULD=14600
4794	023060	012777	003700	156026		MOV	#3700, ARSDA	
4795	023066	022777	010600	156022		CMP	#10600, ARSDS	
4796	023074	001401				BEQ	.+4	
4797	023076	104044				HLT	!DS!DA	;DS SHOULD=10600
4798	023100	104402	023104			TYPE	,.+2	;.ASCIZ <15><12>"SET WRT LCK SW 4 AND CONT"
4799	023140	000000				HALT		
4800	023142	022777	014600	155746		CMP	#14600, ARSDS	
4801	023150	001401				BEQ	.+4	
4802	023152	104044				HLT	!DS!DA	;DS SHOULD=14600
4803	023154	012777	006000	155732		MOV	#6000, ARSDA	
4804	023162	022777	010600	155726		CMP	#10600, ARSDS	
4805	023170	001401				BEQ	.+4	
4806	023172	104044				HLT	!DS!DA	;DS SHOULD=10600
4807	023174	104402	023200			TYPE	,.+2	;.ASCIZ <15><12>"SET WRT LCK SW 5 AND CONT"
4808	023234	000000				HALT		
4809	023236	022777	014600	155652		CMP	#14600, ARSDS	
4810	023244	001401				BEQ	.+4	
4811	023246	104044				HLT	!DS!DA	;DS SHOULD=14600
4812	023250	012777	007700	155636		MOV	#7700, ARSDA	
4813	023256	022777	014600	155632		CMP	#14600, ARSDS	
4814	023264	001401				BEQ	.+4	
4815	023266	104044				HLT	!DS!DA	;DS SHOULD=14600
4816	023270	012767	177777	001252		MOV	#-1, OUTBUF	;PUT A 1 INTO DATA BUFFER
4817	023276	012777	024550	155606		MOV	#OUTBUF, ARSBA	;SETUP REG TO
4818	023304	012777	007700	155602		MOV	#7700, ARSDA	;TO A WRITE
4819	023312	012777	177777	155570		MOV	#-1, ARSWC	
4820	023320	012777	000061	155556		MOV	#61, ARSCS1	;TRY TO WRITE
4821	023326	105777	155552		7\$:	TSTB	ARSCS1	;WAIT FOR DONE
4822	023332	100375				BPL	7\$	
4823	023334	105777	155544		5\$:	TSTB	ARSCS1	;WAIT FOR READY
4824	023340	100375				BPL	5\$	
4825	023342	022777	154600	155546		CMP	#154600, ARSDS	
4826	023350	001401				BEQ	.+4	
4827	023352	104044				HLT	!DS!DA	;DS SHOULD=154600

4828	023354	022777	004000	155536		CMP	#4000, @RSER	
4829	023362	001401				BEG	.+4	
4830	023364	104044				HLT	!DS!DA	;ER SHOULD=4000
4831	023366	104414				CLRDK		;CLEAR ALL REG
4832	023370	022777	014600	.55520		CMP	#14600, @RSOS	;DID WLE CLEAR?
4833	023376	001401				BEG	.+4	;NO
4834	023400	104040				HLT	!DS	;A CLEAR SHOULD NOT CLEAR WLE
4835	023402	005067	001142			CLR	OUTBUF	;PUT A 0 INTO DATA BUFFER
4836	023406	012777	024550	155476		MOV	#OUTBUF, @RSBA	;SETUP REG TO
4837	023414	012777	007700	155472		MOV	#7700, @RSDA	;TO A WRITE
4838	023422	012777	177777	155460		MOV	#-1, @RSWC	
4839	023430	012777	000071	155446		MOV	#71, @RSCS1	;DO A READ TO SEE IF DISK DID
4840	023436	105777	155442		8\$:	TSTB	@RSCS1	;ACTUALLY GET WRITTEN ON TO
4841	023442	100375				BPL	8\$;WAIT FOR DONE
4842	023444	005767	001100			TST	OUTBUF	;IS DATA STILL 0 ON THE DSK?
4843	023450	001401				BEG	.+4	;YES
4844	023452	104040				HLT	!DS	;NO DSK DID GET WRITTEN ONTO WITH WLE SET
4845	023454	104402	023460			TYPE	..+2	;ASCIZ <15><12>"RESET ALL W/L SWITCHES"
4846	023512	000000				HALT		
4847	023514	022777	010600	155374		CMP	#10600, @RSOS	;DID WLE CLEAR
4848	023522	001401				BEG	.+4	;YES
4849	023524	104 '0				HLT	!DS	;NO
4850								;CLEAR ALL DISK
4851	023526	012777	000040	155352		CLRDK: MOV	#40, @RSCS2	;CLEAR ALL DSK REG
4852	023534	016777	155416	155344		MOV	UNNUM, @RSCS2	;GET UNIT NUMBER
4853	023542	000002				RTI		

;ERROR TYPTXTOUT ROUTINE

```

4854
4855
4856 023544 005767 175054 RSREG: TST .HLTCT ; SHOULD WE TYPTXT GOOD AND BAD
4857 023550 001021 BNE 8$ ; NO
4858 023552 104402 023556 TYPE .+2 ; .ASCIZ " BAD="
4859 023564 010046 MOV BAD,-(6) ; PUT BAD ON STACK
4860 023566 104404 TYPE0 ; TYPE STACK IN OCTAL
4861 023570 104402 023574 TYPE .+2 ; .ASCIZ " GOOD="
4862 023604 010146 MOV GOOD,-(6) ; PUT GOOD ON STACK
4863 023606 104404 TYPE0 ; TYPE STACK IN OCTAL
4864 023610 000167 000432 JMP PTDONE ; GET OUT
4865 023614
4866 023614 104402 023620 8$: TYPE .+2 ; .ASCIZ " CS1="
4867 023626 017746 155252 MOV @RSCS1,-(6) ; PUT @RSCS1 ON STACK
4868 023632 104404 TYPE0 ; TYPE STACK IN OCTAL
4869 023634
4870 023634 104402 023640 1$: TYPE .+2 ; .ASCIZ " ER="
4871 023646 017746 155246 MOV @RSEER,-(6) ; PUT @RSEER ON STACK
4872 023652 104404 TYPE0 ; TYPE STACK IN OCTAL
4873 023654
4874 023654 104402 023660 2$: TYPE .+2 ; .ASCIZ " CS2="
4875 023666 017746 155214 MOV @RSCS2,-(6) ; PUT @RSCS2 ON STACK
4876 023672 104404 TYPE0 ; TYPE STACK IN OCTAL
4877 023674 032767 000200 174722 BIT #200,.HLTCT ; TYPTXT SECOND SET ?
4878 023702 001076 BNE SEEC ; YES
4879 023704 032767 000100 174712 BIT #AS,.HLTCT ; TYPTXT ER ?
4880 023712 001410 BEQ 3$ ; NO
4881 023714 104402 023720 TYPE .+2 ; .ASCIZ " AS="
4882 023726 017746 155170 MOV @RSAS,-(6) ; PUT @RSAS ON STACK
4883 023732 104404 TYPE0 ; TYPE STACK IN OCTAL
4884 023734 032767 000020 174662 3$: BIT #BA,.HLTCT ; TYPTXT BUS ASSRESS
4885 023742 001410 BEQ 4$ ; NO
4886 023744 104402 023750 TYPE .+2 ; .ASCIZ " BA="
4887 023756 017746 155130 MOV @RSBA,-(6) ; PUT @RSBA ON STACK
4888 023762 104404 TYPE0 ; TYPE STACK IN OCTAL
4889 023764 032767 000004 174632 4$: BIT #DA,.HLTCT ; TYPTXT DA ?
4890 023772 001410 BEQ 5$ ; NO
4891 023774 104402 024000 TYPE .+2 ; .ASCIZ " DA="
4892 024006 017746 155102 MOV @RSDA,-(6) ; PUT @RSDA ON STACK
4893 024012 104404 TYPE0 ; TYPE STACK IN OCTAL
4894 024014 032767 000010 174602 5$: BIT #WC,.HLTCT ; TYPTXT WC?
4895 024022 001410 BEQ 6$ ; NO
4896 024024 104402 024030 TYPE .+2 ; .ASCIZ " WC="
4897 024036 017746 155046 MOV @RSWC,-(6) ; PUT @RSWC ON STACK
4898 024042 104404 TYPE0 ; TYPE STACK IN OCTAL

```

4899	024044	032767	000040	174552	6\$:	BIT	#DS,.HLTCT	;DRIVE STATUS
4900	024052	001475				BEQ	PTDONE	;NO
4901	024054	104402	024060			TYPE	+2	;ASCIZ "DS="
4902	024066	017746	155024			MOV	ARSDS,-(6)	;PUT ARSDS ON STACK
4903	024072	104404				TYPE0		;TYPE STACK IN OCTAL
4904	024074	000167	000146			JMP	PTDONE	;GET OUT
4905	024100	042767	000200	174516	SEEC:	BIC	#200,.HLTCT	;CLEAR COMMON BIT
4906	024106	032767	000240	174510		BIT	#DT,.HLTCT	;TYPTXT DRIVE TYPE?
4907	024114	001410				BEQ	9\$;NO
4908	024116	104402	024122			TYPE	+2	;ASCIZ "DT="
4909	024130	017746	154776			MOV	ARSDT,-(6)	;PUT ARSDT ON STACK
4910	024134	104404				TYPE0		;TYPE STACK IN OCTAL
4911	024136	032767	000210	174460	9\$:	BIT	#DB,.HLTCT	;TYPTXT DATA BUFFER
4912	024144	001410				BEQ	10\$;NO
4913	024146	104402	024152			TYPE	+2	;ASCIZ "DB="
4914	024160	017746	154742			MOV	ARSDB,-(6)	;PUT ARSDB ON STACK
4915	024164	104404				TYPE0		;TYPE STACK IN OCTAL
4916	024166	032767	000220	174430	10\$:	BIT	#MR,.HLTCT	;TYPTXT MN?
4917	024174	001410				BEQ	11\$;NO
4918	024176	104402	024202			TYPE	+2	;ASCIZ "MR="
4919	024210	017746	154714			MOV	ARSMR,-(6)	;PUT ARSMR ON STACK
4920	024214	104404				TYPE0		;TYPE STACK IN OCTAL
4921	024216	032767	000204	174400	11\$:	BIT	#LA,.HLTCT	;TYPTXT LA?
4922	024224	001410				BEQ	PTDONE	;NO
4923	024226	104402	024232			TYPE	+2	;ASCIZ "LA="
4924	024240	017746	154660			MOV	ARSLA,-(6)	;PUT ARSLA ON STACK
4925	024244	104404				TYPE0		;TYPE STACK IN OCTAL
4926	024246	052767	100000	154710	PTDONE:	BIS	#BIT15,ONCEE	;SET FORND ERROR FLAG
4927	024254	000207				RTS	PC	
4928								
4929	024256	005067	154716		WAITRY:	CLR	WORK	;CLEAR COUNTER
4930	024262	105777	154616		1\$:	TSTB	ARSCS1	;TEST READY
4931	024266	100406				BMI	2\$;OK CONT
4932	024270	005267	154704			INC	WORK	;UPDATE COUNTER
4933	024274	005767	154700			TST	WORK	;DONE YET?
4934	024300	001403				BEQ	3\$;READY DID NOT COME UP
4935	024302	000767				BR	1\$;CONTINUE WAITING
4936	024304	062716	000002		2\$:	ADD	#2,(SP)	;UPDATE RETURN PC
4937	024310	000207			3\$:	RTS	PC	;RETURN

;RANDOM DATA GENERATOR SUBROUTINE

```

4938
4939
4940 024312 016767 000136 000140 RANDOM: MOV LONUM, LOSAV
4941 024320 016767 000132 000134 MOV HINUM, HISAV
4942 024326 016700 000122 RAND1: MOV LONUM, R0 ;SET UP R0 WITH 5 DIGITS LOW
4943 024332 016704 000120 MOV HINUM, R4 ;SET UP R1 WITH 5 DIGITS HIGH
4944 024336 012703 000007 MOV #7, R3 ;SET UP SHIFT COUNT
4945 024342 005002 CLR R2 ;CLEAR R2
4946 024344 006300 SHIFT: ASL R0 ;SHIFT R0 LEFT AND
4947 024346 006104 ROL R4 ;ROTATE CARRY INTO LSB OF R1 INTO
4948 024350 006102 ROL R2 ;ROTATE CARRY OUT OF R1 INTO R2
4949 024352 005303 DEC R3 ;DECREMENT R3
4950 024354 001373 BNE SHIFT ;CONTINUE SHIFT LOOP
4951 024356 066700 000072 ADD LONUM, R0 ;ADDN IN NUMBER TO MAKE X 129
4952 024362 005504 ADC R4 ;PROPOGATE CARRY
4953 024364 066704 000066 ADD HINUM, R4 ;ADDN IN NUMBER TO MAKE X 129
4954 024370 005502 ADC R2 ;PROPOGATE CARRY
4955 024372 062700 001057 ADD #1057, R0 ;ADDN LOW CONSTANT
4956 024376 005504 ADC R4 ;PROPOGATE CARRIES
4957 024400 005502 ADC R2 ;PROPOGATE AGAIN
4958 024402 062704 047401 ADD #47401, R4 ;ADDN HIGH CONSTANT
4959 024406 005502 ADC R2 ;PROPOGATE CARRY
4960 024410 062702 000006 ADD #6, R2 ;ADDN HIGHEST CONSTANT
4961 024414 062700 000002 ADD #2, R0 ;REPRIME R0 WITH HIGH DIGIT
4962 024420 005504 ADC R4 ;PROPOGATE CARRY
4963 024422 010067 000026 MOV R0, LONUM ;PUT R0 BACK IN LONUM
4964 024426 010021 MOV R0, (R1)+ ;LOAD WC
4965 024430 005367 154544 DEC WORK
4966 024434 001406 BEQ EXGEN
4967 024436 010467 000014 MOV R4, HINUM ;PUT R1 BACK IN HINUM
4968 024442 010421 MOV R4, (1)+ ;HOLD HINUM FOR PROGRAM
4969 024444 005367 154530 DEC WORK
4970 024450 001326 BNE RAND1
4971 024452 000205 EXGEN: RTS ;RETURN TO PROGRAM
4972 024454 000000 LONUM: 0
4973 024456 000000 HINUM: 0
4974 024460 000000 LOSAV: 0
4975 024462 000000 HISAV: 0
4976 024464 RANEND:

```

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

: TABLES FOR ILLEGAL FUNCTION TESTS

ILLTAB:

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

ILFTB2:

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

OUTBUF: .BLKW 300
INBUF: .BLKW 300
.END

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

RDOCF = 010772	3168	3177*												
AS = 000100	1799#	2003	2444	2454	2476	2507	2520	2542	2585	2879	2937	3108	3244	
	3610	3650	3735	3742	3779	3800	3806	4261	4281	4303	4307	4310	4879	
ATA = 100000	1820#													
RDB1 = 001172	1834#													
SA = 000020	1797#	1973	2031	2037	2119	2123	2127	2149	2388	2392	2441	2473	2504	
	2539	2605	2885	2940	3001	3017	3035	3776	3839	4884				
BAD = 000000	1729#	2096*	2135*	2136	2173*	2174	2210*	2211	2306*	2320*	2335*	2365*	2366	
	2401*	2402	2408*	2409	2413*	2414	2562*	2563*	2566	2626*	2627	2630*	2633	
	2667*	2668	2671*	2674	2677*	2679	2702*	2703	2715*	2718	2761*	2762	2765*	
	2768	2798*	2789	2814*	2815	2822*	2825	2874*	2906*	2910	2911*	2912	2915*	
	2917	2926*	2929	2934*	2935	2962*	2967	2969	2970*	2971	2975*	2976	2993*	
	2996	3007*	3010	3022*	3025	3060*	3062	3067*	3068	3086*	3087	3092*	3093	
	3149*	3150*	3151	3186*	3187*	3189*	3194	3230*	3233	3284*	3287	3290*	3300*	
	3303	3309*	3310	3347*	3348	3359*	3360	3538*	3541	3545*	3546	3555*	3556	
	3640*	3642	3730*	3733	3759*	3762	3791*	3794	3801*	3804	3814*	3817	3861*	
	3864	3896*	3899*	3900	3995*	3996	4059*	4097*	4140*	4162*	4255*	4256*	4258	
	4267*	4268	4271*	4274	4859									
BAI = 000010	1823#	2783	2796	2805	3826	3849	3883	3960						
BAITST = 006544	2780#													
BEGIN = 001206	1674	1692	1854#	4339										
BELL = 000007	1703#	4334	4436											
BITBA = 004322	2384#													
BITCS2 = 004206	2360#													
BITST = 004134	2347#													
BITMC = 004252	2372#													
BIT0 = 000001	1712#	1920	1938	2332	2606	2656	2709	2911	2970	4684				
BIT1 = 000002	1713#													
BIT10 = 002000	1722#													
BIT11 = 004000	1673	1691	1723#	1865										
BIT12 = 010000	1724#	3560	3663	4718										
BIT13 = 020000	1725#	1884	1904	1924										
BIT14 = 040000	1665	1726#	3267	3292	3316	3319	3323							
BIT15 = 100000	1727#	1887	1906	1913	4256	4926								
BIT2 = 000004	1714#													
BIT3 = 000010	1715#	4225	4231	4286	4713									
BIT4 = 000020	1716#	3789	3830	3855	3890									
BIT5 = 000040	1717#													
BIT6 = 000100	1718#	3899												
BIT7 = 000200	1719#	3468												
BIT8 = 000400	1720#													
BIT9 = 001000	1721#	3167	3171											
BLOCK = 007004	2834#													
RPORTT = 001174	1837#	2721*												
CHKDV = 001622	1916	1920#												
CLROK = 104414	2061	2347	2393	2395	2422	2450	2485	2516	2556	2595	2646	2690	2730	
	2780	2795	2804	2834	2904	2947	2960	3039	3053	3074	3116	3121	3165	
	3221	3260	3306	3320	3329	3371	3395	3408	3426	3433	3451	3457	3482	
	3493	3559	3579	3597	3614	3626	3669	3688	3700	3719	3728	3788	3810	
	3825	3848	3873	3882	3909	3921	3940	3953	4005	4062	4111	4136	4158	
	4186	4209	4632#	4712	4744	4831								
CSTST = 010722	3163#													
CS1 = 000001	1793#	1968	2026	2066	2070	2074	2082	2352	2356	2429	2461	2492	2527	
	2582	2609	2611	2617	2653	2655	2661	2697	2699	2746	2749	2786	2820	
	2863	2885	2922	2925	2940	2943	2946	2981	2984	2987	3001	3004	3017	
	3020	3035	3038	3299	3315	3338	3341	3353	3382	3416	3441	3472	3523	

0000 =.000001

1728#	2097*	2132*	2134	2136	2139*	2170*	2172	2174	2177*	2207*	2209	2211
2214*	2307*	2321*	2336*	2363*	2364*	2366	2399*	2400*	2402	2407*	2409	2412*
2414	2564*	2565*	2566	2624*	2625*	2627	2631*	2632*	2633	2655*	2666*	2668
2672*	2673*	2674	2678*	2679	2701*	2701*	2703	2716*	2717*	2718	2759*	2760*
2762	2766*	2767*	2768	2787*	2789	2812*	2813*	2815	2823*	2824*	2825	2853*
2854*	2865*	2867*	2872	2876	2912*	2915*	2917	2927*	2928*	2929	2932*	2933*
2935	2971*	2974*	2976	2994*	2995*	2996	3008*	3009*	3010	3023*	3024*	3025
3061*	3062	3065*	3066*	3068	3084*	3087	3090*	3091*	3093	3145*	3146*	3151
3166*	3169*	3192*	3193*	3194	3231*	3232*	3233	3285*	3286*	3287	3291*	3294*
3301*	3302*	3303	3307*	3308*	3310	3345*	3346*	3348	3358*	3360	3539*	3540*
3541	3544*	3546	3549*	3552*	3554*	3556	3628*	3631*	3635	3641*	3642	3731*
3732*	3733	3760*	3761*	3762	3792*	3793*	3794	3802*	3803*	3804	3815*	3816*
3817	3862*	3863*	3864	3897*	3898*	3900	3993*	3994*	3996	4058*	4096*	4149*
4163*	4257*	4258	4265*	4266*	4268	4272*	4273*	4274	4862			
4790#												
4941	4943	4953	4967*	4973#								
4941*	4975#											
1698#	1960	1964	1968	1973	1979	1985	1991	1997	2003	2022	2026	2031
2037	2043	2048	2053	2066	2070	2074	2082	2092	2098	2102	2106	2111
2119	2123	2127	2138	2149	2157	2161	2165	2176	2186	2194	2198	2202
2213	2223	2231	2235	2242	2248	2255	2265	2272	2280	2288	2308	2322
2327	2352	2356	2368	2376	2380	2388	2392	2406	2411	2416	2429	2432
2435	2438	2441	2444	2454	2461	2464	2467	2470	2473	2476	2479	2492
2495	2498	2501	2504	2507	2510	2520	2527	2530	2533	2536	2539	2542
2545	2548	2568	2571	2574	2579	2582	2585	2588	2602	2605	2609	2611
2614	2617	2620	2623	2629	2635	2638	2653	2655	2658	2661	2664	2670
2676	2681	2697	2699	2705	2708	2711	2714	2720	2746	2749	2752	2755
2758	2764	2770	2786	2791	2794	2798	2811	2817	2820	2827	2843	2848
2852	2860	2863	2875	2879	2882	2885	2888	2890	2919	2922	2925	2931
2937	2940	2943	2946	2950	2953	2978	2981	2984	2987	2998	3001	3004
3012	3017	3020	3027	3035	3038	3042	3045	3064	3070	3073	3077	3089
3095	3098	3102	3105	3108	3120	3124	3127	3139	3156	3173	3196	3199
3235	3238	3289	3295	3299	3305	3312	3315	3338	3341	3344	3350	3353
3356	3362	3365	3382	3385	3388	3391	3394	3398	3401	3416	3419	3422
3425	3441	3444	3447	3450	3472	3475	3478	3481	3485	3523	3531	3534
3537	3543	3548	3558	3562	3585	3588	3607	3610	3613	3617	3639	3644
3647	3650	3654	3658	3678	3681	3684	3687	3691	3694	3709	3712	3715
3718	3722	3735	3738	3742	3746	3753	3758	3764	3767	3770	3773	3776
3779	3782	3796	3800	3806	3809	3813	3819	3836	3839	3842	3860	3866
3869	3872	3876	3895	3902	3905	3908	3912	3915	3930	3933	3936	3939
3943	3998	4034	4038	4050	4054	4060	4073	4076	4088	4092	4098	4120
4126	4145	4150	4170	4171	4175	4193	4197	4260	4261	4264	4270	4276
4281	4284	4303	4307	4310	4731	4757	4761	4766	4770	4775	4779	4784
4788	4793	4797	4802	4806	4811	4815	4827	4830	4834	4844	4849	
1737#	4441*	4442*	4443	4444	4458							
3597#												
1733#	1861*	4324*	4393	4408	4410	4412*	4413*	4415	4418*	4419	4425	4453*
1824#												
2907	2955#											
2964	3047#											
2961	4998#											
3053#												
2960#												
2905	4980#											
2902#												
5008#												

HLTAOR 001012
IAERR 013352
ICNT 001000
IE = 000100
ILFDN 007560
ILFDNE 010220
ILFTB2 024526
ILF67 010222
ILLFUN 007560
ILLTAB 024464
ILLS1 007304
INBLF 025350

NYMTSM	006650	2804#												
OBUSV	001102	1744#	1864*	1937*	2424	2439	2456	2471	2487	2502	2522	2537	2597	2603
		2631	2648	2672	2692	2716	2721	2722*	2736	2766	2781	2787	2836	2883
		2908	2938	2965	2999	3013	3028	3262	3278	3301	3231	3372	3409	3434
		3458	3500	3507	3524	3564	3581	3599	3675	3706	3748	3774	3828	3837
		3851	3885	3927	3955	3966	3978	3984	4142	4165	4190	4239	4272	4682
ONCEE	001164	1673*	1691*	1831#	1863*	1865	1887*	1906	1913*	1920	1938*	3267	3292	3316
		3319*	3323*	3660	3663*	4926*								
OUU	007302	2849	2889	2891#										
OR	= 000200	1811#												
OUT	020032	4325#												
OLTBUF	024550	1864	1937	2295	2297	2311	2325	2423*	2455*	2486*	2521*	2596*	2647*	2677
		2691*	2722	2735*	3055	3061	3079	3222*	3261*	3269*	3271*	3274*	3276*	3290
		3321*	3330*	3499*	3506*	3549	3747*	3827*	3850*	3884*	3954*	3965*	4238*	4710*
		4720	4725	4745*	4746	4816*	4817	4835*	4836	4842	5007#			
PART	011112	3207#												
PARTST	010520	3115#												
PATST	014514	3788#												
PATTST	015154	3882#												
PC	= %000007	1711#	2610*	2654*	2698*	2745*	2785*	2810*	3337*	3381*	3415*	3440*	3584*	3677*
		3708*	3929*	4144*	4192*	4447*	4624*	4927*	4937*					
PCNT	001004	1735#	4329*	4330*	4341									
PGE	= 002000	1812#	3358											
PGETST	011676	3329#												
PGTRAP	016624	4113	4123#											
PIP	= 020000	1817#												
POTP	011300	3208	3243#											
POTP1	011302	3219	3244#											
PS	= 177776	1699#	1700	1945*	2009*	4115*	4139*	4140	4161*	4162	4184*	4252*	4317*	4323*
		4680#												
PSW	= 177776	1700#												
PTDONE	024246	4864	4900	4904	4922	4926#								
QES	016544	4111#												
QES00N	016640	4121	4127#											
QQ	= 000001	1869#	1941#	2004#	2054#	2075#	2084#	2113#	2129#	2141#	2150#	2166#	2180#	2187#
		2203#	2217#	2224#	2237#	2243#	2249#	2256#	2266#	2273#	2281#	2289#	2342#	2358#
		2370#	2382#	2394#	2417#	2445#	2480#	2511#	2549#	2590#	2641#	2682#	2725#	2775#
		2799#	2829#	2892#	3048#	3110#	3132#	3140#	3158#	3202#	3251#	3324#	3366#	3403#
		3428#	3452#	3486#	3574#	3590#	3621#	3664#	3695#	3723#	3783#	3820#	3844#	3877#
		3916#	3948#	4000#	4106#	4129#	4152#	4178#	4200#	4735#				
		2296	4940#											
RANDOM	024312	4942#	4970											
RAND1	024326	4976#												
RANEND	024464	1827#												
RANNU	001154	2294#												
RANTS	003714	4556	4631#											
ROLIN	= 104412	4630#	4702	4705	4709									
RDOCT	= 104410	2646#												
ROTST	005760	3371#												
RHRT1	012076	3408#												
RHRT2	012254	3433#												
RHRT3	012364	3457#												
RHRT4	012474	1755#	1881	2001	2442	2452	2474	2505	2518	2540	2583	2872	2874	2876*
RSAS	001122	2877	2934	3106	3342	3466*	3648	3730	3739*	3740	3777	3801	4297	4313*
		4670*	4671	4882										
RSBA	001112	1751#	1948*	1971	2012*	2029	2116*	2117	2120*	2121	2124*	2125	2134*	2135

		2145*	2146*	2147	2330	2335	2384*	2386	2389*	2390	2424*	2439	2456*	2471
		2487*	2502	2522*	2537	2597*	2603	2630	2648*	2671	2692*	2715	2736*	2765
		2781*	2788	2807*	2836*	2883	2908*	2938	2965*	2999	3015	3033	3056*	3060
		3080*	3086	3223*	3262*	3278*	3300	3331*	3372*	3409*	3434*	3458*	3500*	3507*
		3524*	3555	3564*	3581*	3599*	3675*	3706*	3748*	3774	3828*	3837	3851*	3885*
		3927*	3955*	3966*	3984*	4026*	4042*	4066*	4082*	4142*	4155*	4190*	4239*	4271
		4642*	4643	4682*	4720*	4725*	4746*	4817*	4836*	4887				
RSB08	001152	1767*	2385*											
RSB0E	001134	1760*	2808*											
RSCS1	001104	1748*	1947*	1966	2011*	2024	2063*	2064	2067*	2068	2071*	2072	2078*	2079*
		2080	2348*	2350	2353*	2354	2426*	2427	2458*	2459	2489*	2490	2508	2524*
		2525	2546	2561*	2580	2599*	2606*	2607	2615	2650*	2651	2659	2694*	2695
		2706	2747	2784*	2809*	2818	2821*	2845*	2861	2910*	2913	2916	2951	2969*
		2972	2975	3040	3057*	3058	3081*	3082	3099*	3100	3225*	3226	3236	3264*
		3265	3281*	3282	3297	3313	3333*	3334	3336*	3339	3357*	3363	3379*	3392
		3399	3411*	3412	3423	3436*	3437	3439*	3448	3465*	3479	3483	3502*	3503
		3514*	3521	3526*	3527	3535	3565*	3566	3583*	3601*	3602	3611	3645	3676*
		3685	3707*	3716	3736	3744	3750*	3751	3754	3756	3807	3811	3831*	3832
		3834	3853*	3856	3858	3874	3887*	3888	3891	3893	3910	3928*	3937	3961*
		3962	3969*	3970	3990*	3991	4027*	4030	4032	4036	4044*	4046	4048	4052
		4068*	4069	4071	4074	4083*	4084	4086	4090	4116*	4124	4143*	4166*	4173
		4191*	4195	4243*	4249*	4255	4285*	4290*	4308	4314*	4683*	4684	4686	4721*
		4722	4726*	4728	4729	4749*	4750	4820*	4821	4823	4839*	4840	4867	4930
RSCS18	001144	1764*	2349*	3651*										
RSCS2	001106	1749*	1875*	1879*	1890*	1946*	1953*	1958	1961*	2010*	2017*	2019*	2020	2023*
		2090	2093*	2094	2096	2099*	2100	2103*	2104	2107*	2108*	2109	2112*	2360*
		2361*	2365	2401	2562	2626	2667	2702	2761	2783*	2796	2805*	2814	2822
		2926	2993	3007	3022	3067	3092	3230	3284	3309	3347	3359	3501*	3519*
		3520*	3538	3635*	3640	3652	3656	3759	3789*	3791	3814	3826*	3830*	3849*
		3855*	3861	3883*	3890*	3896	3960*	3968*	3972*	3995	4246*	4248*	4254*	4267
		4304*	4665*	4666	4713*	4851*	4852*	4875						
RSCS28	001146	1765*	2362*											
RSCS3	001136	1761*												
RSDA	001114	1752*	1949*	1977	2013*	2035	2191*	2192	2195*	2196	2199*	2200	2209*	2210
		2219*	2220*	2221	2316	2320	2436	2468	2499	2534	2558*	2569	2612	2656
		2709	2750	2835*	2844*	2944	2985	3177*	3179*	3180	3192	3280*	3380*	3386
		3505*	3582*	3586	3600*	3636*	3655	3674*	3705*	3771	3922*	3964*	3983*	4041*
		4081*	4188*	4241*	4279	4282	4647*	4648	4716*	4747*	4754*	4758*	4767*	4776*
		4785*	4794*	4803*	4812*	4818*	4837*	4892						
RSD8	001126	1757*	2015*	2396*	2397*	2408	2413	4914						
RSDS	001116	1753*	1962	2462	2529	2621	2756	2846	2850	2856	2858	2880	2923	2982
		3118	3125	3389	3420	3445	3468	3476	3532	3608	3682	3689	3713	3720
		3765	3867	3903	3934	3941	4305	4652*	4653	4755	4759	4764	4768	4773
		4777	4782	4786	4791	4795	4800	4804	4809	4813	4825	4832	4847	4902
		1759*	1891	1895	1899	1901	4909							
RSDT	001132	1754*	1876*	1950*	1983	2014*	2041	2228*	2229	2232*	2233	2239*	2240	2245*
RSER	001120	2246	2251*	2252*	2253	2430	2451*	2477	2493	2517*	2543	2559*	2572	2636
		2792	2920	2948	2979	3043	3071	3075	3096	3103	3117*	3122	3354	3383
		3396	3414*	3417	3442	3473	3529	3560	3605	3615	3637	3679	3692	3710
		3729*	3780	3798	3870	3906	3913	3931	4657*	4658	4828	4871		
RSLA	001124	1756*	3137	3145	3149	3186	4662	4924						
RSNR	001130	1758*	1951*	1989	2018*	2046	2260*	2261	2268*	2269*	2270	2275*	2276	2283*
		2284	2560*	2575	2839	3167	3171	3184	3790	3797*	4675*	4676	4919	
RSMRR	021650	1687	4675*	4677										
RSREG	023544	4447	4856*											
RSVCPS	001142	1763*	4114*	4138*	4160*	4187*	4237*	4319	4320*					

TST25	003570	2266#																		
TST26	003616	2273#																		
TST27	003654	2281#																		
TST3	002246	2007#																		
TST30	003712	2292#																		
TST31	004132	2345#																		
TST32	004204	2358#																		
TST33	004250	2370#																		
TST34	004320	2382#																		
TST35	004372	2394#																		
TST36	004500	2420#																		
TST37	004624	2448#																		
TST4	002460	2057#																		
TST40	005004	2483#																		
TST41	005142	2514#																		
TST42	005334	2552#																		
TST43	005530	2593#																		
TST44	005756	2644#																		
TST45	006144	2685#																		
TST46	006332	2728#																		
TST47	006542	2778#																		
TST5	002552	2075#																		
TST50	006646	2802#																		
TST51	007002	2832#																		
TST52	007302	2895#																		
TST53	010220	3051#																		
TST54	010516	3113#																		
TST55	010574	3135#																		
TST56	010610	3140#																		
TST57	010720	3161#																		
TST6	002600	2087#																		
TST60	011110	3205#																		
TST61	011332	3254#																		
TST62	011674	3327#																		
TST63	012074	3369#																		
TST64	012252	3406#																		
TST65	012362	3431#																		
TST66	012472	3455#																		
TST67	012642	3489#																		
TST7	002740	2113#																		
TST70	013274	3577#																		
TST71	013350	3593#																		
TST72	013464	3624#																		
TST73	013734	3667#																		
TST74	014072	3698#																		
TST75	014216	3726#																		
TST76	014512	3786#																		
TST77	014664	3823#																		
TTAGG	002250	2009#																		
TTMOUT	007300	2855	2890#																	
TTVEC	017644	4288	4294#																	
TYPE =	104402	1886	1908	1912	1928	1929	3155	3662	4333	4436	4440	4446	4498	4538						
		4581	4601	4604	4608	4627#	4701	4704	4708	4741	4752	4762	4771	4780						
		4789	4798	4807	4845	4858	4861	4866	4870	4874	4881	4886	4891	4896						
		4901	4908	4913	4918	4923														
TYPE0 =	104404	4445	4628#	4860	4863	4868	4872	4876	4883	4888	4893	4898	4903	4910						

TYPES = 104406	4915	4920	4925												
UNCMP 001162	1911	1927	4629#												
	1830#	1882*	1888	1915*	1922*	2442	2452	2474	2505	2518	2540	2583	2932		
	3731	3739	3777	3802											
UNITST 006234	2730#														
UNITSV 001160	1829#	1881*	1888	3629	4217										
LNNUM 001156	1828#	1883*	1890	1910	1917*	1923*	1926	1946	1961	2010	2023	2112	2360		
	2363	2400	2564	2624	2665	2700	2732	2739	2759	2812	2823	2864	2927		
	2994	3008	3023	3065	3090	3232	3285	3307	3345	3520	3539	3761	3793		
	3815	3863	3898	3993	4219	4227	4254	4265	4703*	4743*	4852				
UP = 000000	1781#	4018	4019	4020	4021										
WAITRY 024256	2610	2654	2698	2745	2785	2810	3337	3381	3415	3440	3584	3677	3708		
	3929	4144	4192	4929#											
WC = 000010	1796#	1997	2053	2157	2161	2165	2186	2376	2380	2435	2467	2498	2533		
	2588	2602	2620	2664	2714	2755	2888	2943	3004	3020	3038	3353	3770		
	3842	4264	4731	4894											
WCE = 040000	1814#														
WCEDNE 011644	3277	3319#													
WCEDON 011666	3272	3323#													
WCETST 011334	3260#														
WCETT 011342	3262#	3322													
WORK 001200	1850#	2261*	2262*	2263	2276*	2277*	2278	2284*	2285*	2286	2294*	2398*	2404*		
	2575*	2576*	2577	2838*	2841*	2864*	2868	2870*	3013*	3014*	3015	3028*	3029*		
	3032*	3033	3055*	3056	3079*	3080	3144*	3147*	3178*	3182*	3213	3223	3229*		
	3467*	3470*	3627*	3629	3632*	3655*	3975*	3977*	3981*	4117*	4118*	4167*	4168*		
	4216*	4217	4223*	4296*	4297	4301*	4313	4706*	4707*	4717	4929*	4932*	4933		
	4965*	4969*													
WORK1 001202	1851#	2967*	2968*	2988	2990	3188*	3190*	4212*	4225	4231*	4286	4714*	4715*		
	4716														
WORK2 001204	1852#	3494*	3497*	3498	3525	4711*	4732								
WRCKT 006146	2690#														
WRTLCK 022264	1670	4740#													
WRTST 005532	2595#														
WT 022204	4722#	4727													
WTDV 017440	4253#														
WTDV1 017762	4287	4312	4317#												
WTDV2 017642	4289	4292#	4315												
WTE 022214	4719	4725#													
WMP = 000004	1835#	3212													
ZERONE 004374	2395#														
SENDAD 020102	1653	4337#													
SEND1 020112	4335	4339#													
= 026150	1661#	1662#	1664#	1666#	1669#	1672#	1678#	1690#	1731#	1742#	1866	1886	1908		
	1929	1930#	1959	1963	1967	1972	1978	1984	1990	1996	2002	2021	2025		
	2030	2036	2042	2047	2052	2065	2069	2073	2081	2091	2101	2105	2110		
	2118	2122	2126	2137	2148	2156	2160	2164	2175	2185	2193	2197	2201		
	2212	2222	2230	2234	2241	2247	2254	2264	2271	2279	2287	2351	2355		
	2367	2375	2379	2387	2391	2410	2415	2428	2431	2434	2437	2440	2443		
	2453	2460	2463	2466	2469	2472	2475	2478	2491	2494	2497	2500	2503		
	2506	2509	2519	2526	2529	2532	2535	2538	2541	2544	2547	2567	2570		
	2573	2578	2581	2584	2587	2601	2604	2608	2613	2616	2619	2622	2628		
	2634	2637	2652	2657	2660	2663	2669	2675	2680	2696	2704	2707	2710		
	2713	2719	2748	2751	2754	2757	2763	2769	2790	2793	2797	2816	2819		
	2826	2851	2859	2862	2878	2881	2884	2887	2918	2921	2924	2930	2936		
	2939	2942	2945	2949	2952	2977	2980	2983	2986	2997	3000	3003	3011		
	3016	3019	3026	3034	3037	3041	3044	3063	3069	3072	3076	3088	3094		

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS BASIC FUNCTION DIAGNOSTIC

	4230	4953	4954	4956	4957	4959	4962								
	4933	2633	2767	2767	3014	3029	3246	3302	3552	3554	4273	4329	4375	4378	4623
	4936	4951	4953	4955	4958	4960	4961								
	4946	2215	3272	4224	4302										
	4948	3533													
	4950	1889	1900	1907	1959	1963	1967	1972	1978	1984	1990	1996	2002	2021	
	4952	2036	2042	2047	2052	2065	2069	2073	2081	2091	2095	2101	2105	2110	
	4954	2126	2137	2146	2156	2160	2164	2175	2185	2193	2197	2201	2212	2222	
	4955	2241	2247	2254	2264	2271	2279	2287	2301	2305	2315	2319	2329	2334	
	4956	2367	2375	2379	2387	2391	2403	2410	2415	2428	2431	2434	2437	2440	
	4957	2463	2466	2469	2472	2478	2491	2494	2497	2500	2503	2506	2509	2526	
	4958	2535	2538	2544	2547	2567	2570	2573	2578	2581	2584	2587	2601	2604	
	4959	2613	2622	2628	2634	2637	2657	2660	2663	2669	2675	2680	2704	2707	
	4960	2710	2713	2751	2754	2757	2763	2769	2790	2793	2797	2816	2819	2826	
	4961	2847	2855	2862	2869	2873	2878	2881	2884	2887	2907	2918	2921	2924	
	4962	2930	2936	2942	2949	2952	2963	2977	2980	2983	2986	2989	2991	2997	
	4963	3000	3003	3011	3016	3019	3026	3034	3041	3063	3069	3072	3076	3088	
	4964	3094	3097	3101	3104	3107	3119	3123	3138	3148	3152	3172	3181	3183	
	4965	3185	3195	3234	3237	3245	3268	3288	3293	3304	3311	3317	3340	3343	
	4966	3249	3255	3261	3264	3284	3290	3293	3307	3400	3418	3421	3446	3449	
	4967	3474	3477	3480	3484	3530	3533	3542	3547	3557	3561	3587	3606	3612	
	4968	3616	3630	3638	3643	3646	3649	3680	3683	3690	3693	3711	3714	3734	
	4969	3737	3741	3745	3752	3757	3763	3769	3772	3775	3781	3795	3799	3808	
	4970	3812	3818	3835	3838	3841	3859	3865	3868	3871	3894	3901	3904	3911	
	4971	3914	3923	3932	3935	3942	3974	3986	3992	3997	4037	4053	4075	4091	
	4972	4125	4174	4196	4218	4259	4263	4269	4275	4280	4283	4287	4306	4312	
	4973	4361	4392	4394	4409	4421	4435	4487	4490	4560	4593	4685	4719	4756	
	4974	4769	4774	4778	4783	4787	4792	4796	4801	4805	4810	4814	4826	4833	
	4975	4848	4880	4885	4890	4895	4900	4907	4912	4917	4922	4934	4966	4983	
	4976	4985	4990	4995	5000	5005	5010	5015	5020	5025	5030	5035	5040	5045	
BCT	4411	4562													
BIC	1673	1863	1887	1913	2262	2277	2285	2332	2563	2576	2911	2933	2968	2970	3146
	3150	3187	3193	3323	3358	3376	3462	3511	3516	3732	3803	3899	3972	4229	4256
	4379	4715	4905												
B:CB	4597														
B:CS	1691	1903	1938	2361	2364	2400	2565	2606	2625	2666	2701	2760	2783	2805	2813
	2824	2915	2928	2974	2995	3009	3024	3066	3091	3232	3286	3308	3319	3346	3357
	3501	3540	3641	3663	3761	3789	3793	3816	3826	3830	3849	3855	3863	3883	3890
	3898	3960	3968	3994	4231	4266	4713	4926							
B:GA	4491														
B:IT	1865	1884	1888	1904	1906	1920	1924	1989	2024	2046	2270	2427	2442	2452	2459
	2474	2490	2505	2518	2525	2540	2583	2796	2839	2846	2850	2856	3137	3167	3171
	3184	3267	3292	3316	3468	3629	3652	3656	3660	3751	3777	3991	4217	4225	4286
	4297	4331	4357	4391	4395	4406	4434	4438	4451	4478	4684	4719	4877	4879	4884
	4889	4894	4899	4906	4911	4916	4921								
B:LT	4564														
B:NI	3217	3335	3413	3438	3522	3603	3686	3854	3889	4931					
B:NE	1885	1892	1896	1902	1905	1921	1925	2405	2453	2475	2519	2541	2733	2840	2842
	2851	3031	3170	3191	3352	3378	3387	3464	3469	3471	3496	3513	3518	3551	3657
	3661	3672	3703	3778	3958	3982	4119	4169	4220	4226	4245	4251	4278	4298	4332
	4358	4363	4369	4377	4396	4407	4439	4452	4479	4493	4495	4529	4599	4606	4733
	4857	4878	4950	4970											
BFL	2608	2652	2696	2714	2973	3059	3083	3227	3266	3283	3504	3528	3567	3717	3755
	3833	3857	3892	3938	3963	3971	4031	4033	4047	4049	4070	4072	4085	4087	4367
	4372	4449	4595	4687	4723	4730	4751	4822	4824	4841					
BR	1880	1894	1898	1914	1918	1931	2310	2324	2338	2340	2849	2871	2889	3005	3153

CLC
CLC

CLC
#28

OMP8
COM
DEC

EMT
HALT
INC
INCB
JMP

JSP

MOV

3197	3219	3273	3318	3553	3604	3634	3659	3976	3988	4035	4039	4051	4055	4089
4101	4146	4172	4194	4230	4235	4247	4373	4397	4402	4405	4465	4467	4476	4542
4573	4582	4602	4639	4644	4649	4654	4659	4663	4667	4672	4677	4588	4724	4727
4734	4935													
2133	2171	2208	3270	4222	4295	4300								
1861	1862	1875	1893	1897	1923	2079	2108	2146	2183	2220	2252	2269	2396	2407
2647	2731	2838	2844	2865	3218	3228	3229	3250	3261	3274	3280	3291	3467	3505
3506	3628	3636	3964	3980	3983	4015	4040	4041	4078	4081	4100	4105	4167	4212
4213	4214	4238	4294	4323	4468	4527	4557	4745	4835	4929	4945			
2362	4453	4475	4477	4497	4607									
1891	1895	1899	1901	1958	1962	1966	1995	2020	2051	2064	2068	2072	2080	2090
2094	2100	2104	2109	2117	2121	2125	2136	2155	2159	2163	2174	2192	2196	2200
2211	2229	2233	2240	2246	2263	2278	2286	2304	2318	2333	2350	2354	2366	2374
2378	2396	2390	2402	2409	2414	2433	2439	2462	2465	2471	2477	2496	2502	2508
2528	2531	2537	2543	2546	2566	2577	2580	2586	2600	2603	2612	2615	2621	2627
2633	2656	2659	2668	2674	2679	2703	2706	2709	2718	2732	2747	2750	2756	2762
2768	2789	2815	2818	2825	2858	2861	2872	2880	2883	2886	2917	2920	2923	2929
2935	2938	2941	2951	2976	2979	2982	2988	2990	2996	2999	3002	3010	3015	3018
3025	3030	3033	3036	3040	3062	3068	3071	3087	3093	3096	3100	3118	3125	3151
3183	3194	3233	3236	3243	3244	3287	3297	3303	3310	3313	3339	3348	3360	3363
3377	3383	3386	3389	3392	3399	3417	3420	3423	3442	3445	3448	3463	3476	3479
3483	3512	3517	3529	3532	3535	3541	3546	3556	3586	3605	3608	3611	3642	3645
3679	3682	3669	3713	3720	3733	3736	3744	3756	3762	3765	3768	3774	3780	3794
3798	3804	3807	3811	3817	3834	3837	3840	3858	3864	3867	3874	3893	3900	3903
3906	3910	3934	3941	3996	4036	4052	4056	4074	4090	4094	4123	4124	4173	4195
4198	4219	4258	4268	4274	4279	4282	4305	4308	4376	4403	4494	4592	4732	4755
4759	4764	4768	4773	4777	4782	4786	4791	4795	4800	4804	4809	4813	4825	4828
4832	4847													
4362	4393	4410	4561	4563	4598	4605								
4707														
1877	2300	2314	2328	2404	2841	2854	2870	3147	3169	3182	3190	3981	4118	4250
4311	4368	4949	4965	4969										
1698														
1662	1930	4450	4524	4541	4742	4753	4763	4772	4781	4790	4799	4808	4846	
1879	1917	2734	3179	3470	3631	4168	4221	4228	4233	4299	4437	4528	4932	
4413	4418	4488	4492											
1667	1670	1674	1679	1680	1681	1682	1683	1684	1685	1686	1687	1688	1692	1867
1932	2954	2964	3046	3239	3247	3322	4121	4325	4339	4455	4864	4904		
2296	2610	2654	2698	2745	2785	2810	3337	3381	3415	3440	3584	3677	3708	3929
4144	4192	4337	4447											
1954	1855	1856	1857	1858	1859	1860	1864	1874	1876	1881	1882	1883	1890	1910
1922	1926	1937	1939	1940	1945	1946	1947	1948	1949	1950	1951	1952	1953	1961
2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2023	2062	2063	2067
2071	2078	2093	2096	2097	2099	2103	2107	2112	2116	2120	2124	2132	2134	2135
2145	2154	2158	2162	2170	2172	2173	2182	2191	2195	2199	2207	2209	2210	2219
2229	2239	2245	2251	2260	2261	2268	2275	2276	2283	2284	2294	2295	2297	2298
2299	2302	2303	2306	2307	2311	2312	2313	2316	2317	2320	2321	2325	2326	2327
2330	2331	2335	2336	2348	2360	2363	2365	2372	2384	2397	2398	2399	2401	2408
2412	2413	2423	2424	2425	2426	2451	2455	2456	2457	2458	2486	2487	2488	2489
2517	2521	2522	2523	2524	2557	2558	2559	2560	2561	2562	2564	2575	2596	2597
2598	2599	2624	2626	2630	2631	2648	2649	2650	2665	2667	2671	2672	2677	2678
2691	2692	2693	2694	2700	2702	2715	2716	2721	2722	2735	2736	2737	2738	2739
2740	2741	2743	2744	2759	2761	2765	2766	2781	2782	2784	2787	2788	2806	2807
2808	2809	2812	2814	2821	2822	2823	2835	2836	2837	2845	2853	2864	2874	2876
2902	2903	2905	2906	2908	2909	2910	2912	2916	2926	2927	2932	2934	2961	2962
2965	2966	2967	2969	2971	2975	2993	2994	3007	3008	3013	3022	3023	3028	3054

	3055	3056	3057	3060	3061	3065	3067	3078	3079	3080	3081	3084	3086	3090	3092
	3099	3115	3117	3144	3145	3149	3163	3164	3166	3177	3178	3186	3188	3192	3207
	3208	3209	3210	3212	3213	3214	3220	3222	3223	3224	3225	3230	3231	3249	3262
	3263	3264	3269	3278	3279	3281	3284	3285	3290	3294	3300	3201	3307	3309	3321
	3330	3331	3332	3333	3336	3345	3347	3359	3372	3373	3374	3375	3379	3380	3409
	3410	3411	3414	3434	3435	3436	3439	3458	3459	3460	3461	3465	3466	3494	3497
	3498	3499	3500	3502	3507	3508	3509	3510	3514	3515	3519	3520	3524	3525	3526
	3538	3539	3544	3545	3549	3555	3563	3564	3565	3580	3581	3582	3583	3598	3599
	3600	3601	3627	3635	3640	3655	3670	3673	3674	3675	3676	3701	3704	3705	3706
	3707	3729	3730	3731	3739	3747	3748	3749	3750	3759	3760	3791	3792	3797	3801
	3802	3814	3815	3827	3828	3829	3831	3850	3851	3852	3853	3861	3862	3884	3885
	3896	3887	3896	3897	3922	3923	3926	3927	3928	3954	3955	3956	3959	3961	3965
	3966	3967	3969	3975	3977	3978	3979	3984	3987	3989	3990	3993	3995	4006	4007
	4008	4009	4013	4014	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026
	4027	4042	4043	4044	4058	4059	4063	4064	4065	4066	4067	4068	4080	4082	4083
	4096	4097	4103	4104	4112	4113	4114	4115	4116	4117	4134	4135	4137	4138	4139
	4140	4141	4142	4143	4149	4157	4159	4160	4161	4162	4163	4164	4165	4166	4183
	4184	4185	4187	4188	4189	4190	4191	4210	4211	4215	4216	4227	4232	4234	4236
	4237	4239	4240	4241	4242	4243	4246	4248	4249	4252	4254	4255	4257	4265	4267
	4271	4272	4285	4288	4289	4290	4296	4304	4313	4314	4315	4317	4318	4319	4322
	4323	4324	4334	4354	4355	4356	4359	4374	4380	4381	4382	4398	4399	4401	4404
	4414	4415	4419	4422	4441	4444	4464	4471	4472	4473	4474	4499	4500	4501	4502
	4514	4515	4516	4517	4518	4519	4520	4521	4522	4523	4526	4530	4531	4532	4533
	4534	4535	4536	4537	4551	4552	4553	4554	4555	4558	4574	4575	4576	4577	4590
	4591	4609	4620	4622	4624	4637	4638	4642	4643	4647	4648	4652	4653	4657	4658
	4662	4665	4666	4670	4671	4675	4676	4680	4681	4682	4683	4699	4703	4706	4710
	4711	4714	4716	4717	4720	4721	4725	4726	4728	4743	4746	4747	4748	4749	4754
	4758	4767	4776	4785	4794	4803	4812	4816	4817	4818	4819	4820	4836	4837	4838
	4839	4851	4852	4859	4862	4867	4871	4875	4882	4887	4892	4897	4902	4909	4914
	4919	4924	4940	4941	4942	4943	4944	4963	4964	4967	4968				
NOVB	2232	2349	2353	2373	2377	2385	2389	3651	4364	4365	4370	4412	4443	4466	4469
	4496	4559	4596	4603											
NOP	2341	2742	4061	4102	4700										
RESET	1954	2089	4336												
ROL	2139	2177	2214	2867	3271	3276	3632	4223	4301	4480	4482	4484	4568	4569	4570
	4571	4572	4947	4948											
ROLB	4481	4483	4485												
ROR	3189	4565	4566	4567											
RTI	4291	4316	4383	4416	4423	4454	4503	4539	4578	4610	4853				
RTS	4927	4937	4971												
SEC	2866	3275													
SUB	3032	4442	4621												
TRAP	4623	4626	4627	4628	4629	4630	4631	4632							
TST	1971	1977	1983	2001	2029	2035	2041	2147	2184	2221	2253	2309	2323	2339	2430
	2436	2468	2493	2499	2534	2569	2572	2618	2636	2662	2712	2753	2792	2868	2877
	2944	2948	2985	3043	3075	3103	3106	3122	3211	3215	3216	3342	3351	3354	3396
	3473	3495	3550	3560	3615	3637	3648	3671	3685	3692	3702	3710	3716	3740	3771
	3790	3870	3913	3924	3931	3937	3957	3973	3985	4010	4032	4048	4071	4086	4244
	4262	4277	4400	4420	4448	4729	4842	4856	4933						
TSTB	2607	2651	2695	2913	2972	3058	3082	3226	3265	3282	3334	3412	3437	3503	3521
	3527	3566	3602	3754	3832	3856	3888	3891	3962	3970	4030	4046	4069	4084	4360
	4366	4371	4408	4486	4489	4594	4686	4722	4750	4821	4823	4840	4930		
WAIT	4253	4292													
ASCIZ	1887	1909	1930	3156	3663	4334	4441	4447	4539	4582	4602	4702	4705	4709	4742
	4753	4763	4772	4781	4790	4799	4808	4846	4859	4862	4867	4871	4875	4882	4887
	4892	4897	4902	4909	4914	4919	4924								

.BLKB	4612																			
.BLKB	4343	4505	5007	5008																
.ENABL	1	1645																		
.ENOC	5009																			
	1728	1855	1861	1872	1944	2007	2057	2075	2087	2113	2129	2141	2150	2166	2180					
	2187	2203	2217	2224	2237	2243	2249	2256	2266	2273	2281	2292	2345	2358	2370					
	2382	2394	2420	2448	2483	2514	2552	2593	2644	2685	2728	2778	2802	2832	2895					
	3051	3113	3135	3140	3161	3205	3254	3327	3369	3406	3431	3455	3489	3577	3593					
	3624	3667	3698	3726	3786	3823	3847	3880	3919	3951	4003	4109	4132	4155	4181					
	4203	4331	4333	4334	4340	4396	4424	4438	4440	4448	4456	4522	4530	4539	4540					
	4593	4613	4627	4628	4629	4630	4631	4632	4633	4738										
.E.EN	1887	1909	1930	3156	3663	4334	4441	4447	4539	4582	4602	4702	4705	4709	4742					
	4753	4763	4772	4781	4790	4799	4808	4846	4859	4862	4867	4871	4875	4882	4887					
	4892	4897	4902	4909	4914	4919	4924													
.IF	1712	1854	1861	1869	1941	2004	2054	2075	2084	2113	2129	2141	2150	2166	2180					
	2187	2203	2217	2224	2237	2243	2249	2256	2266	2273	2281	2289	2342	2358	2370					
	2382	2394	2417	2445	2480	2511	2549	2590	2641	2682	2725	2775	2799	2829	2892					
	3048	3110	3132	3140	3158	3202	3251	3324	3366	3403	3428	3452	3486	3574	3590					
	3621	3664	3695	3723	3783	3820	3844	3877	3916	3948	4000	4106	4129	4152	4178					
	4200	4331	4332	4334	4339	4391	4424	4434	4438	4447	4451	4522	4530	4538	4539					
	4592	4612	4626	4627	4628	4629	4630	4631	4632	4735										
.IFF	1855	4332	4340	4396	4434	4451	4540	4593	4613	4626	4627	4628	4629	4630	4631					
	4632																			
.IIF	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	4341	4416	4420	4425	4457					
	4458	4612																		
.IRP	4398	4401	4404	4471	4499	4516	4530	4553	4575	4626										
.LIST	1	1645	1662	1728	1869	1873	1887	1909	1930	1941	1945	2004	2008	2054	2058					
	2075	2076	2084	2088	2113	2114	2129	2130	2141	2142	2150	2151	2166	2167	2180					
	2181	2187	2188	2203	2204	2217	2218	2224	2225	2237	2238	2243	2244	2249	2250					
	2256	2257	2266	2267	2273	2274	2281	2282	2289	2293	2342	2346	2358	2359	2370					
	2371	2382	2383	2394	2395	2417	2421	2445	2449	2480	2484	2511	2515	2549	2553					
	2590	2594	2641	2645	2682	2686	2725	2729	2775	2779	2799	2803	2829	2833	2892					
	2896	3048	3052	3110	3114	3132	3136	3140	3141	3156	3158	3162	3202	3206	3251					
	3255	3324	3328	3366	3370	3403	3407	3428	3432	3452	3456	3486	3490	3574	3578					
	3590	3594	3621	3625	3663	3664	3668	3695	3699	3723	3727	3783	3787	3820	3824					
	3844	3848	3877	3881	3916	3920	3948	3952	4000	4004	4106	4110	4129	4133	4152					
	4156	4178	4182	4200	4204	4334	4441	4447	4539	4582	4602	4613	4626	4627	4628					
	4629	4630	4631	4632	4633	4702	4705	4709	4735	4739	4742	4753	4763	4772	4781					
	4790	4799	4808	4846	4859	4862	4867	4871	4875	4882	4887	4892	4897	4902	4909					
	4914	4919	4924																	
.MACR	1783																			
.MACRO	1	4613																		
.MCALL	1645	1728	1869	1941	2004	2054	2075	2084	2113	2129	2141	2150	2166	2180	2187					
	2203	2217	2224	2237	2243	2249	2256	2266	2273	2281	2289	2342	2358	2370	2382					
	2394	2417	2445	2480	2511	2549	2590	2641	2682	2725	2775	2799	2829	2892	3048					
	3110	3132	3140	3158	3202	3251	3324	3366	3403	3428	3452	3486	3574	3590	3621					
	3664	3695	3723	3783	3820	3844	3877	3916	3948	4000	4106	4129	4152	4178	4200					
	4735																			
.MLIST	1	1645	1662	1728	1869	1873	1887	1909	1930	1941	1945	2004	2008	2054	2058					
	2075	2076	2084	2088	2113	2114	2129	2130	2141	2142	2150	2151	2166	2167	2180					
	2181	2187	2188	2203	2204	2217	2218	2224	2225	2237	2238	2243	2244	2249	2250					
	2256	2257	2266	2267	2273	2274	2281	2282	2289	2293	2342	2346	2358	2359	2370					
	2371	2382	2383	2394	2395	2417	2421	2445	2449	2480	2484	2511	2515	2549	2553					
	2590	2594	2641	2645	2682	2686	2725	2729	2775	2779	2799	2803	2829	2833	2892					
	2896	3048	3052	3110	3114	3132	3136	3140	3141	3156	3158	3162	3202	3206	3251					
	3255	3324	3328	3366	3370	3403	3407	3428	3432	3452	3456	3486	3490	3574	3578					

	3590	3594	3621	3625	3663	3664	3668	3695	3699	3723	3727	3783	3787	3820	3824
	3844	3848	3877	3881	3916	3920	3948	3952	4000	4004	4106	4110	4129	4133	4152
	4156	4178	4182	4200	4204	4334	4441	4447	4539	4582	4602	4613	4626	4627	4628
	4629	4630	4631	4632	4633	4702	4705	4709	4735	4739	4742	4753	4763	4772	4781
	4780	4799	4808	4846	4859	4862	4867	4871	4875	4882	4887	4892	4897	4902	4909
	4914	4919	4924												
PAGE	1696	1731	4384	4426	4458	4506	4546	4583							
REF	1	549	1097												
REF	1662														
SMTL	1869	1941	2004	2054	2084	2289	2342	2417	2445	2480	2511	2549	2590	2641	2682
	2725	2775	2799	2829	2892	3048	3110	3132	3158	3202	3251	3324	3366	3403	3429
	3452	3486	3574	3590	3621	3664	3695	3723	3783	3820	3844	3877	3916	3948	4000
	4106	4129	4152	4178	4200	4326	4344	4384	4426	4458	4506	4546	4583	4613	4735
TITLE	1645														

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DERSAA.SEQ/SOL/CRF/PAGNUM/NL:TOC=DERSAA.SML,DERSAA.P11
RUN-TIME: 29 46 7 SECONDS
RUN-TIME RATIO: 196/83=2.3
CORE USED: 22K (43 PAGES)