

RH11-RS03

BASIC FUNCT AL DIAL
MD-11-DZRSB-E

EP-DZRSB-E-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames. The left side of the card features a vertical column of frames, each containing a small diagram or schematic. The remaining frames in the grid contain larger, more detailed diagrams and text, likely representing various functional dial settings or configurations for the MD-11-DZRSB-E device. The text and diagrams are printed in a light color against a dark background, typical of microfiche technology.

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRSB-E-D
PRODUCT NAME:	RH11-RS03-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC
DATE CREATED:	AUGUST 1976
MAINTAINER:	DIAGNOSTIC GROUP

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 DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1973,1974,1975,1976 BY DIGITAL EQUIPMENT CORPORATION

MAINDEC-11-DZRSB-E
DZRSB.P11

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

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

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 3
DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03,RS03/LA AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03,RS03/LA 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 OK BK OF MEMORY, AND AN RH11 CONTROLLER WITH A RS03, RS03/LA 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

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148

E01

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 5

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 4
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. IF SWITCHLESS CPU, SIMPLY
- 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, RS03/LA AND RS04 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/LA/04 REGISTERS USING THE SWITCH REGISTER. ON SWITCH-LESS MACHINES THESE ROUTINES ARE USEFULL FOR SCOPING. SIMPLY STRIKE ↑ ANYTIME AFTER PRESSING START TO ENTER OR CHANGE VALUE DESIRED.

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

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC.176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(I.E.) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

- 1. <CR> IF NO CHANGES ARE TO BE MADE.
- 2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>.
- 3. ↑ TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED

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
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

F01

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 6

205

KEYING IN SWREG VALUE.

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 5
DESCRIPTION

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUT: YES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

SWITCH SETTINGS ARE:

- SW<15> = 1 HALT ON ERROR
- SW<14> = 1 LOOP ON TEST
- SW<13> = 1 INHIBIT TYPEOUTS
- SW<12> = 1 INHIBIT OBUFSV FROM CHANGING WHEN LOOKING FOR MEMORY ON -B- PORT
- 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 SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE 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

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
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261

262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 6
DESCRIPTION

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

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.

101

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 9

318
319

10. TEST DESCRIPTION

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 7
DESCRIPTION

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
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

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

THIS TEST GENERATES RANDOM NUMBERS AND LOADS THEM INTO RSWC, RSDA AND RSBA.

6. TEST "CLEAR BIT" IN RSCS2

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

7. TEST DLT AND TRE BITS

DO A READ FROM THE SILO. THIS SHOULD CAUSE A DLT AND A TRE ERROR BECAUSE THE SILO IS EMPTY.

8. CLEAR DLT AND TRE

CLEAR BY SETTING TRE IN RSCS1 AND TEST.

9. LOAD RSOB WITH ALL ONES AND ALL ZEROS

LOAD RSOB 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.

10. TEST FOR 66 LOCATIONS IN SILO

K01

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 11

376
377

THIS IS DONE BY PUTTING A BINARY COUNT IN EVERY LOCATION AND

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 8
DESCRIPTION378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
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

CHECKING THE OUTPUT FOR 66 WORDS.

11. TEST DLT ERROR

THIS IS DONE BY LOADING THE SILO WITH 67 WORDS WITHOUT READING ANY OUT. THIS SHOULD CAUSE DLT TO SET.

12. FLOAT A "1" AND A "0" THROUGH THE SILO

LOAD THE SILO WITH A WORD OF ZEROS AND FLOAT A "1" THROUGH THE WORD. THEN LOAD THE SILO WITH A WORD OF ALL ONES AND FLOAT A "0" THROUGH THE WORD. CHECK THE OUTPUT OF THE SILO FOR THE CORRECT ANSWER.

13. TEST NO-OP FUNCTION

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

14. 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.

15. 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. THIS IS TESTED ON -A- AND -B- PORT.

16. 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. THIS IS TESTED ON -A- AND -B- PORT.

17. 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. THIS IS TESTED ON -A- AND -B- PORT.

18. 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

MO1

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 13

434
435

RSMR. IF IT DOESN'T SET, TIMEOUT. WHEN YOU GET THERE DO A
BLOCK SEARCH FOR SECTOR 0. NOW WE KNOW THAT WE HAVE TIME TO

MAINDEC-11-DZRSB-E RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 9
DESCRIPTION

436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491

TEST FOR DRY AND PIP BITS BEFORE FINDING SECTOR 0. FOR PIP 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.

19. 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.

20. TEST PAR IN RSER

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

21. TEST DPR AND MOL IN RSDS

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

22. 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.

23. 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.

24. 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

492
493

ARE THEN CLEARED. A WORD OF 0 IS NOW WRITTEN ON THE DISK AND
A 1 IS FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE

494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
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

WRITE CHECK FUNCTION TEST IS REPEATED.

25. 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.

26. 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.

27. 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.

28. TEST DISK ADDRESS REGISTER

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

29. 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.

30. 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.

31. 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.

32. TEST BAI IN RSCS2

D02

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-RSC3LA-RSC3-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 17

550
551

SET BAI IN RSC52. DO A ONE WORD WRITE AND CHECK RSBA TO SEE

552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582

IF IT INCREMENTED.

33. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS A16 AND A17 IN RSCS1 FOR AN 18 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.

34. 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.

35. 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.

36. 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.

G02

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-RS03LA-RSQ3-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 20

639

H02

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 21

640
641 000001
642 104000
643 177776
644 177776
645 000007
646 000000
647 000001
648 000002
649 000003
650 000004
651 000005
652 000006
653 000007
654 000001
655 000002
656 000004
657 000010
658 000020
659 000040
660 000100
661 000200
662 000400
663 001000
664 002000
665 004000
666 010000
667 020000
668 040000
669 100000
670 000001
671 000000
672

N= 1
HLT= EMT
PS= 177776
PSW= PS
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
GOOD= %1
BAD= %0

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

;FOR GOOD DATA
;FOR BAD DATA

673	001000		.=	1000	
674					
675	001000	000000			
676	001002	000000			
677	001004	000000	000000		
678	001010	000000			
679	001012	000000			
680	001014	001000			
681	001016	177564			
682	001020	177560			
683	001022	177562			
684	001024	177566			
685	001026	177570			
686	001030	177570			
687					
688		001100	.=	1100	
689	001100	000000			
690	001102	000000			
691					
692					
693					
694	001104	172040			
695	001106	172050			
696	001110	172042			
697	001112	172044			
698	001114	172046			
699	001116	172052			
700	001120	172054			
701	001122	172056			
702	001124	172060			
703	001126	172062			
704	001130	172064			
705	001132	172066			
706	001134	000204			
707	001136	000206			
708	001140	172041			
709	001142	172051			
710	001144	172043			
711	001146	172045			
712					
713					
714					
715		177572			
716		172340			
717		172342			
718		172344			
719		172356			
720		172300			
721		172302			
722		172304			
723		172316			
724		000006			
725		000000			
726					


```

ICNT: 0
ERRORS: 0
PCNT: 0,0
LAD: 0
HLTADR: 0
FILCHR: 1000
TPS: 177564
TKS: 177560
TKB: 177562
TPB: 177566
SWR: 177570
DISPLAY: 177570

;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
;SWITCH REGISTER
;DISPLAY REGISTER

;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
RSDB: 172062
RSMR: 172064
RSDT: 172066
RSVEC: 204
RSVCPS: 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
;INTERUPT VECTOR
;INTERUPT 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

SR0=177572
KIPAR0=172340
KIPAR1=172342
KIPAR2=172344
KIPAR7=172356
KIPDR0=172300
KIPDR1=172302
KIPDR2=172304
KIPDR7=172316
RW=6
UP=00
    
```

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

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

```

769          ;WORKING LOCATIONS
770
771 001150 146723      RANNU: 146723      ;RANDOM NUMBER PRIME
772 001152 000000      UNNUM: 0          ;UNIT CURRENTLY BEING TESTED
773 001154 000000      UNITSV: 0         ;SET BIT=UNIT ON BUS
774 001156 000000      UNCMP: 0         ;FOR COMPARING FOR # OF DEVICE
775 001160 000000      ONCEE: 0         ;DID WE TEST ANY DRIVES
776 001162 000000      R504DT: 0        ;CLR IF R503 SET IF R504
777 001164 000000      TIMSV: 0         ;SAVE LOC FOR TIME
778 001166 000000      AOB1: 0          ;PORT SWITCH
779          000004      WWP=4           ;WRITE WRONG PARITY
780          172100      MPRO=172100     ;PARITY REG
781 001170 000000      BPORTT: 0        ;BUFFER ADDR FOR -B- PORT
782 001172 000000      SAVEE: 0         ;WORK LOC
783
784          ;DISCRIPTION OF ONCEE BITS
785          ;BIT0 MEANS FOUND DRIVE
786          ;BIT11 DO TKSEL TEST
787          ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
788          ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
789          ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
790          ;BIT15 MEANS ERROR FOUND
791
792          ;RH11 WORK REGISTERS
793          ;(CAN BE CHANGED IN ANY ROUTINE)
794 001174 000000      WORK: 0
795 001176 000000      WORK1: 0
796 001200 000000      WORK2: 0
797
798 001202 012706 000500      BEGIN: MOV #500,SP      ;SET STACK TO *** 500 ***
799 001206 012737 022770 000024      MOV #.POWER,2#24    ;SET UP PF VECTOR
800 001214 012737 000340 000026      MOV #340,2#26      ;LOCK OUT THE WORLD
801 001222 012737 022416 000030      MOV #.HLT,2#30     ;SET EMT VECTOR
802 001230 012737 000340 000032      MOV #340,2#32     ;LOCK UP
803 001236 012737 023422 000034      MOV #.TRAP,2#34   ;SET TRAP VECTOR
804 001244 012737 000340 000036      MOV #340,2#36     ;LOCK UP
805 001252 005037 001000      CLR ICNT          ;INIT ICNT
806 001256 005037 001010      CLR LAD           ;INIT LAD
807 001262 042737 143777 001160      BIC #143777,ONCEE ;CLEAR ONCEE
808 001270 012737 027364 001102      MOV #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
809 001276 032737 004000 001160      BIT #BIT11,ONCEE  ;DO TKSEL TEST?
810 001304 001402          BEQ .+6           ;NO
811 001306 000137 024014          JMP 2#TKSEL       ;YES
812 001312 104420          SUSWR           ;SIZE FOR HDWR SWR

```


M02

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST1

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 26
TEST FOR DRIVES

```

866 001634 006337 001156      TRYNX:  ASL      UNCMP      ;CHECK NEXT BIT FOR DRIVE
867 001640 103403                BCS      CHCKDV    ;DID WE TEST ANY REG?
868 001642 005237 001152      INC      UNNUM     ;INC UNIT #
869 001646 000671                BR       STTEST   ;CHECK FOR NEXT DRIVE
870
871 001650 032737 000001 001160  CHCKDV: BIT      #BIT0,ONCEE ;DID WE TEST ANY DRIVES?
872 001656 001057                BNE     DONEE    ;YES WE DID TEST A DRIVE
873 001660 012737 100000 001156  MOV     #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
874 001666 005037 001152      CLR     UNNUM     ;ANY AS BITS, THUS DEFAULTS TO
875 001672 032777 020000 177126  BIT     #BIT13,ASWR ;INHIBIT TYPE OUT?
876 001700 001045                BNE     4$       ;YES
877 001702 013746 001152      MOV     UNNUM,-(6) ;PUT UNNUM ON STACK
878 001706 104406                TYPES   ;TYPE STACK IN OCTAL - SUPRESS
879 001710 104402 000040      TYPE   ,40       ;TYPE SPACE
880 001714 104402 001720      TYPE   ,.+2      ;ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
881 002012 000000                HALT    ;WAIT
882 002014 000402                4$:    BR       NOWGO ;TEST DRIVE 0
883 002016 000137 021762  DONEE: JMP      DONE  ;GET OUT
884
885 ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
886 ;TO CLEAR ALL THE RH AND RS REGISTERS
887
888 002022 012737 027364 001102  NOWGO: MOV     #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
889 002030 052737 000001 001160  BIS     #BIT0,ONCEE ;SET FOUND DRIVE FLAG
890 002036 013737 022414 001164  MOV     TIMES,TIMSV ;SAVE TIME
891 002044 012737 000001 022414  MOV     #1,TIMES  ;ONLY TEST ONCE
892
893 ;*****
894 ;TEST 2 RESET TEST FOR REGISTERS
895 ;*****
896 002052 104400      TST2:  SCOPE
897 002054 012737 000340 177776  MOV     #340,ASPS  ;LOCK OUT INTERUPTS
898 002062 013777 001152 177016  MOV     UNNUM,ARSCS2 ;GET UNIT #
899 002070 012777 177776 177006  MOV     #177776,ARSCS1 ;SET ALL
900 002076 012777 177777 177006  MOV     #177777,ARSCS1 ;POSSIBLE R/W
901 002104 012777 177777 177002  MOV     #177777,ARSDA ;BITS IN THESE REGISTERS
902 002112 012777 177777 177000  MOV     #177777,ARSER
903 002120 012777 177777 177002  MOV     #177777,ARSMR
904 002126 012777 177777 176754  MOV     #177777,ARSWC
905 002134 012777 177737 176744  MOV     #177737,ARSCS2
906 002142 000005      RESET ;CLEAR ALL BITS IN ALL REG.
907
908 ;TEST RSCS2 FOR CLEARED BITS
909 002144 022777 000100 176734  CMP     #100,ARSCS2 ;DID THESE BITS GET CLEARED?
910 002152 001401                BEQ     .+4       ;YES
911 002154 104200                HLT     !CS2     ;(417) SHOULD BE CLEARED IN CS2
912 002156 013777 001152 176722  MOV     UNNUM,ARSCS2 ;PUT # OF UNIT IN TEST IN CS2
913 002164 022777 010600 176724  CMP     #10600,ARSDS ;IS DPR AND MOL SET?
914 002172 001401                BEQ     .+4       ;YES
915 002174 104040                HLT     !DS     ;NO WHY NOT?
916
917 002176 022777 004200 176700  ;TEST CONTROL AND STATUS REG 1
918 002204 001401                CMP     #4200,ARSCS1 ;DID THE READY BIT SET?
919 002206 104001                BEQ     .+4       ;YES
919 002206 104001                HLT     !CS1     ;READY SHOULD BE SET

```

```

920                                     ;TEST BUS ADDRESS REGISTER
921
922 002210 005777 176676             TST   @RSBA           ;IS BA REG. CLEARED
923 002214 001401                   BEQ   :+4             ;YES
924 002216 104020                   HLT   !BA             ;SHOULD BE 0
925
926                                     ;TEST DISK ADDRESS REGISTER
927
928 002220 005777 176670             TST   @RSDA           ;IS DA CLEARED
929 002224 001401                   BEQ   :+4             ;YES
930 002226 104004                   HLT   !DA             ;SHOULD BE 0
931
932                                     ;TEST ERROR REG RSER
933
934 002230 005777 176654             TST   @RSER           ;DID RSER CLEAR?
935 002234 001401                   BEQ   :+4             ;YES
936 002236 104002                   HLT   !ER             ;BITS(157015) SHOULD BE CLEARED
937
938                                     ;TEST RS MAINTENANCE REGISTER
939
940 002240 032777 000077 176662     BIT   #77,@RSMR       ;DID THESE BITS GET CLEARED
941 002246 001401                   BEQ   :+4             ;YES
942 002250 104220                   HLT   !MR             ;BITS(77) SHOULD BE 0
943
944                                     ;TEST WC REG IT SHOULD NOT CHANGE
945
946 002252 022777 177777 176630     CMP   #177777,@RSWC  ;DID IT CHANGE?
947 002260 001401                   BEQ   :+4             ;NO
948 002252 104010                   HLT   !WC             ;RESET SHOULD NOT MODIFY RSWC
949
950                                     ;TEST RSAS
951
952 002264 005777 176632             TST   @RSAS           ;IS REG CLEAR
953 002270 001401                   BEQ   :+4             ;YES
954 002272 104100                   HLT   !AS             ;NO

```

B03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST3

RH11-RSC3LA-RSC3-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 28
TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS

```

955 ;*****
956 ;TEST 3 TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
957 ;*****
958 002274 104400 TST3: SCOPE
959
960 002276 012737 000340 177776 TTAGG: MOV #340, @PS ;LOCK OUT INTERRUPTS
961 002304 013777 001152 176574 MOV UNNUM, @RSCS2 ;GET UNIT #
962 002312 012777 043576 176564 MOV #43576, @RSCS1 ;SET ALL
963 002320 012777 177777 176564 MOV #177777, @RSBA ;POSSIBLE
964 002326 012777 177777 176560 MOV #177777, @RSDA ;REGISTERS
965 002334 012777 177777 176556 MOV #177777, @RSER
966 002342 012777 177777 176556 MOV #177777, @RSDB
967 002350 012777 177777 176532 MOV #177777, @RSWC
968 002356 012777 020417 176522 MOV #20417, @RSCS2
969 002364 012777 000071 176536 MOV #71, @RSMR
970 002372 012777 000040 176506 MOV #40, @RSCS2 ;CLEAR ALL BITS
971 002400 022777 000100 176500 CMP #100, @RSCS2 ;DID THE RIGHT BITS CLEAR?
972 002406 001401 BEQ +4 ;YES
973 002410 104200 HLT ;CS2 ;(417) SHOULD BE CLEARED IN CS2
974 002412 013777 001152 176466 MOV UNNUM, @RSCS2 ;GET DRIVE NUMBER
975 002420 032777 173577 176456 BIT #173577, @RSCS1 ;DID ALL BITS GET CLEARED
976 002426 001401 BEQ +4 ;YES
977 002430 104001 HLT ;CS1 ;NO, ALL BITS SHOULD BE 0
978 ;TEST BUS ADDRESS REGISTER
979
980 002432 005777 176454 TST @RSBA ;IS BA REG. CLEARED
981 002436 001401 BEQ +4 ;YES
982 002440 104020 HLT ;BA ;SHOULD BE 0
983
984 ;TEST DISK ADDRESS REGISTER
985
986 002442 005777 176446 TST @RSDA ;IS DA CLEARED
987 002446 001401 BEQ +4 ;YES
988 002450 104020 HLT ;BA ;SHOULD BE 0
989
990 ;TEST ERROR REG RSER
991
992 002452 005777 176442 TST @RSER ;DID THESE BITS GET CLEARED
993 002456 001401 BEQ +4 ;YES
994 002460 104002 HLT ;ER ;BITS(157015) SHOULD BE CLEARED
995
996 ;TEST RS MAINTENANCE REGISTER
997 002462 012777 000077 176440 BIT #77, @RSMR ;DID THESE BITS GET CLEARED
998 002470 012777 000077 176440 BEQ +4 ;YES
999 002472 104220 HLT ;MR ;BITS(77) SHOULD BE 0
1000
1001 ;TEST WC REG. IT SHOULD NOT CHANGE
1002 002474 022777 177777 176406 CMP #177777, @RSWC ;DID WC CHANGE
1003 002502 001401 BEQ +4 ;NO
1004 002504 104010 HLT ;WC ;WHY DID IT CHANGE?

```

```

1005 ;*****
1006 ;TEST 4 SET AND CLEAR ALL REGISTERS
1007 ;*****
1008 002506 104400 †TST4: SCOPE
1009 ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
1010 ;BITS 7,6,5,4,3,2&1
1011
1012 002510 104414 CLRDK ;CLEAR ALL RS REG
1013 002512 013737 001164 022414 MOV TIMSV,TIMES ;GET TIME
1014 002520 012777 003576 176356 MOV #3576,RSCS1 ;SET DISK FUNCTION BITS
1015 002526 022777 007776 176350 CMP #7776,RSCS1 ;ARE THESE BITS SET?
1016 002534 001401 BEQ +4 ;NO
1017 002536 104001 HLT !CS1 ;SHOULD = 7776
1018 002540 012777 002524 176336 MOV #2524,RSCS1 ;SET THESE BITS
1019 002546 022777 006724 176330 CMP #6724,RSCS1 ;DID THEY SET
1020 002554 001401 BEQ +4 ;YES
1021 002556 104001 HLT !CS1 ;SHOULD BE 6724
1022 002560 012777 001052 176316 MOV #1052,RSCS1 ;SET THESE BITS
1023 002566 022777 005252 176310 CMP #5252,RSCS1 ;ARE THEY =?
1024 002574 001401 BEQ +4 ;YES
1025 002576 104001 HLT !CS1 ;SHOULD = 5252
1026 002600 104400
1027
1028
1029 002602 012777 043576 176274 MOV #43576,RSCS1 ;SET DISK FUNCTION BITS
1030 002610 005077 176270 CLR RSCS1
1031 002614 022777 004200 176262 CMP #4200,RSCS1 ;IS THE READY BIT SET
1032 002622 001401 BEQ +4 ;YES
1033 002624 104001 HLT !CS1 ;RSCS1 SHOULD = 4200
1034
1035 ;*****
1036 ;TEST 6 TEST RSCS2
1037 ;*****
1038 002626 104400 †TST6: SCOPE
1039
1040 002630 000005 RESET ;CLEAR WORLD
1041 002632 022777 000100 176246 CMP #100,RSCS2 ;DID THEY CLEAR?
1042 002640 001401 BEQ +4 ;YES
1043 002642 104200 HLT !CS2 ;NO
1044 002644 012777 021037 176234 MOV #21037,RSCS2 ;SET BITS 21017
1045 002652 022777 021137 176226 CMP #21137,RSCS2 ;DID THESE BITS GET SET
1046 002660 001405 BEQ 18 ;YES
1047 002662 017700 176220 MOV RSCS2,BAD
1048 002666 012701 021137 MOV #21137,GOOD ;WHAT CS2 SHOULD =
1049 002672 104000 HLT ;CS2 = BAD GOOD = CORRECT ANS

```

1050	002674	012777	020025	176204	15:	MOV	#20025,@RSCS2	:SET THESE BITS
1051	002702	022777	020125	176176		CMP	#20125,@RSCS2	:DID THESE BITS GET SET
1052	002710	001401				BEQ	.+4	:YES
1053	002712	104200				HLT	:CS2	:NO CS2 SHOULD = 20125
1054	002714	012777	000012	176164		MOV	#12,@RSCS2	:LOAD THESE BITS
1055	002722	022777	000112	176156		CMP	#112,@RSCS2	:DID THESE BITS GET SET IN CS2
1056	002730	001401				BEQ	.+4	:YES
1057	002732	104200				HLT	:CS2	:BAD = CS2 GOOD = CORRECT ANS
1058	002734	012777	177777	176144		MOV	#-1,@RSCS2	:SET BITS
1059	002742	005077	176140			CLR	@RSCS2	:CLEAR THEM
1060	002746	022777	000100	176132		CMP	#100,@RSCS2	:DID CLEAR WORK
1061	002754	001401				BEQ	.+4	:YES
1062	002756	104200				HLT	:CS2	:R/W BITS DID NOT CLEAR
1063	002760	013777	001152	176120		MOV	UNNUM,@RSCS2	:GET UNIT #
1064	002766	104400						
1065					TST7:	SCOPE		
1066					;CAN WE	SET ALL THE RSBA BITS		
1067	002770	012777	177777	176114		MOV	#177777,@RSBA	:SET THE BITS
1068	002776	022777	177776	176106		CMP	#177776,@RSBA	:DID THEY SET
1069	003004	001401				BEQ	.+4	:YES
1070	003006	104020				HLT	:BA	:BITS 17776 SHOULD BE SET
1071	003010	012777	125252	176074		MOV	#125252,@RSBA	:SET THESE BITS
1072	003016	022777	125252	176066		CMP	#125252,@RSBA	:ARE THEY =
1073	003024	001401				BEQ	.+4	:YES
1074	003026	104020				HLT	:BA	:SHOULD BE 125252
1075	003030	012777	052524	176054		MOV	#52524,@R' 9A	:SET THESE BITS
1076	003036	022777	052524	176046		CMP	#52524,@RSBA	:ARE THEY =
1077	003044	001401				BEQ	.+4	:YES
1078	003046	104020				HLT	:BA	:SHOULD BE 52524
1079								
1080	003050	104400			TST10:	SCOPE		
1081					;FLOAT A 1	THROUGH RSBA		
1082								
1083	003052	012701	000002		FLOTBA:	MOV	#2,GOOD	:GET A 2
1084	003056	000241				CLC		:CLEAR CARRY
1085	003060	010177	176026		15:	MOV	GOOD,@RSBA	:FLOAT NUMBER
1086	003064	017700	176022			MOV	@RSBA,BAD	:GET BA
1087	003070	020100				CMP	GOOD,BAD	:COMPARE BA
1088	003072	001401				BEQ	.+4	:BA CORRECT
1089	003074	104000				HLT		:BAD=BA GOOD=CORRECT ANS
1090	003076	006101				ROL	GOOD	:ROTATE NUMBER
1091	003100	103367				BCC	15	:LOOP TILL DONE

E03

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST6

R411-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 31
TEST R5C52

```

1092 003102 104400          TST11: SCOPE
1093
1094          ;CLEAR THE RSBA REGISTER
1095
1096 003104 012777 177777 176000      MOV      #177777, @RSBA      ;SET RSBA EQUAL TO ALL ONES
1097 003112 005077 175774              CLR      @RSBA
1098 003116 005777 175770              TST      @RSBA              ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1099 003122 001401                      BEQ      .+4                ;YES
1100 003124 104020                      HLT      !BA                ;NO
1101 003126 104400          TST12: SCOPE
1102
1103          ;CAN WE SET ALL BITS IN RSWC REGISTER
1104
1105 003130 012777 177777 175752      MOV      #177777, @RSWC      ;SET WC BITS
1106 003136 022777 177777 175744      CMP      #177777, @RSWC      ;ARE ALL BITS SET
1107 003144 001401                      BEQ      .+4                ;YES
1108 003146 104010                      HLT      !WC                ;NO
1109 003150 012777 125252 175732      MOV      #125252, @RSWC      ;SET THESE BITS
1110 003156 022777 125252 175724      CMP      #125252, @RSWC      ;ARE THEY =
1111 003164 001401                      BEQ      .+4                ;YES
1112 003166 104010                      HLT      !WC                ;SHOULD BE 125252
1113 003170 012777 052525 175712      MOV      #52525, @RSWC       ;SET THESE BITS
1114 003176 022777 052525 175704      CMP      #52525, @RSWC       ;ARE THEY =
1115 003204 001401                      BEQ      .+4                ;YES
1116 003206 104010                      HLT      !WC                ;SHOULD BE 152525
1117 003210 104400          TST13: SCOPE
1118
1119          ;FLOAT A 1 THROUGH RSWC
1120
1121 003212 012701 000001      FLOTWC: MOV      #1, GOOD      ;GET A 1
1122 003216 000241              CLC                          ;CLEAR CARRY
1123 003220 010177 175664      IS:     MOV      GOOD, @RSWC   ;FLOAT NUMBER
1124 003224 017700 175660      MOV      @RSWC, BAD          ;GET WC
1125 003230 020100              CMP      GOOD, BAD           ;COMPARE WC
1126 003232 001401                      BEQ      .+4                ;WC CORRECT
1127 003234 104000                      HLT                          ;BAD=WC GOOD=CORRECT ANS
1128 003236 006101              ROL      GOOD                ;ROTATE NUMBER
1129 003240 103367              BCC      IS                  ;LOOP TILL DONE

```

F03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST6

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 32
TEST RSCS2

```

1130          ;CLEAR THE WORD COUNT REGISTER
1131 003242 104400 TST14: SCOPE
1132
1133 003244 012777 177777 175636      MOV      #177777, @RSWC      ;SET RSWC REGISTER EQUAL TO ALL ONES
1134 003252 005077 175632              CLR      @RSWC
1135 003254 005777 175626              TST      @RSWC              ;DID ALL BITS GET CLEARED
1136 003262 001401              BEQ      .+4                ;YES
1137 003264 104010              HLT      !WC                ;NO
1138 003266 104400 TST15: SCOPE
1139
1140          ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1141
1142 003270 012777 177777 175616      MOV      #177777, @RSDA      ;SET ALL BITS
1143 003276 022777 177777 175610      CMP      #177777, @RSDA      ;ARE THE BITS SET
1144 003304 001401              BEQ      .+4                ;YES
1145 003306 104004              HLT      !DA                ;NO
1146 003310 012777 125252 175576      MOV      #125252, @RSDA      ;SET THESE BITS
1147 003316 022777 125252 175570      CMP      #125252, @RSDA      ;ARE THEY =
1148 003324 001401              BEQ      .+4                ;YES
1149 003326 104004              HLT      !DA                ;SHOULD BE 125252
1150 003330 012777 052525 175556      MOV      #52525, @RSDA       ;SET THESE BITS
1151 003336 022777 052525 175550      CMP      #52525, @RSDA       ;ARE THEY =
1152 003344 001401              BEQ      .+4                ;YES
1153 003346 104004              HLT      !DA                ;SHOULD BE 52525
1154 003350 104400 TST16: SCOPE
1155
1156          ;FLOAT A 1 THROUGH RSDA
1157
1158 003352 012701 000001 FLOTDA: MOV      #1, GOOD      ;GET A 1
1159 003356 000241          CLC                      ;CLEAR CARRY
1160 003360 010177 175530 15:      MOV      GOOD, @RSDA        ;FLOAT NUMBER
1161 003364 017700 175524          MOV      @RSDA, BAD        ;GET DA
1162 003370 020100          CMP      GOOD, BAD         ;COMPARE DA
1163 003372 001401              BEQ      .+4                ;DA CORRECT
1164 003374 104000          HLT                      ;BAD=DA GOOD=CORRECT ANS
1165 003376 006101          ROL      GOOD              ;ROTATE NUMBER
1166 003400 103367          BCC     15                 ;LOOP TILL DONE

```


G03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST6

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 33
TEST RSC52

```

1167          ;CAN WE CLEAR THE RSDA REG.
1168 003402 104400 TST17: SCOPE
1169
1170 003404 012777 177777 175502      MOV      #177777, @RSDA      ;SET RSDA TO ALL ONES
1171 003412 005077 175476              CLR      @RSDA
1172 003416 005777 175472              TST      @RSDA      ;TEST FOR ZERO RSDA
1173 003422 001401              BEQ      .+4          ;YES
1174 003424 104004              HLT      !DA          ;ANS SHOULD BE 0
1175 003426 104400 TST20: SCOPE
1176
1177          ;SET AND CLEAR THE RSER REG.
1178
1179 003430 012777 177777 175462      MOV      #177777, @RSER      ;SET THESE BITS
1180 003436 022777 177017 175454      CMP      #177017, @RSER      ;DID THEY SET
1181 003444 001401              BEQ      .+4          ;YES
1182 003446 104002              HLT      !ER          ;RSER SHOULD = 157017
1183 003450 112777 000001 175442      MOVB     #1, @RSER          ;A MOVB INST
1184 003456 022777 000001 175434      CMP      #1, @RSER          ;SHOULD MODIFY COMPLETE WD
1185 003464 001401              BEQ      .+4          ;OK
1186 003466 104002              HLT      !ER
1187
1188 003470 104400 TST21: SCOPE
1189
1190 003472 012777 052005 175420      MOV      #52005, @RSER      ;SET THESE BITS
1191 003500 022777 052005 175412      CMP      #52005, @RSER      ;DID THEY SET
1192 003506 001401              BEQ      .+4          ;YES
1193 003510 104002              HLT      !ER          ;ER SHOULD = 52005
1194 003512 104400 TST22: SCOPE
1195
1196 003514 012777 125012 175376      MOV      #125012, @RSER     ;SET THESE BITS
1197 003522 022777 125012 175370      CMP      #125012, @RSER     ;DID THEY SET
1198 003530 001401              BEQ      .+4          ;YES
1199 003532 104002              HLT      !ER          ;ER SHOULD = 105012

```

H03

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST6

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 34
TEST RSC52

1200	003534	104400			TST23: SCOPE			
1201								
1202	003536	012777	177017	175354	MOV	#177017, @RSER	; SET THESE BITS	
1203	003544	005077	175350		CLR	@RSER	; CLEAR THEM	
1204	003550	005777	175344		TST	@RSER	; DID THEY CLEAR	
1205	003554	001401			BEQ	+4	; YES	
1206	003556	104002			HLT	!ER	; SHOULD = 0	
1207	003560	104400			TST24: SCOPE			
1208								
1209					; SET AND CLEAR RSMR			
1210								
1211	003562	012777	000070	175340	MOV	#70, @RSMR	; SET THESE BITS	
1212	003570	017737	175334	001174	MOV	@RSMR, WORK	; PUT INTO WORKABLE REG	
1213	003576	042737	177700	001174	BIC	#177700, WORK	; CLEAR JUNK	
1214	003604	022737	000070	001174	CMP	#70, WORK	; DID THEY SET	
1215	003612	001401			BEQ	+4	; YES	
1216	003614	104220			HLT	!MR	; SHOULD = 70	
1217	003616	104400			TST25: SCOPE			
1218								
1219	003620	012777	000070	175302	MOV	#70, @RSMR	; SET BITS	
1220	003626	005077	175276		CLR	@RSMR	; CLEAR THEM	
1221	003632	032777	000077	175270	BIT	#77, @RSMR	; DID THEY CLEAR	
1222	003640	001401			BEQ	+4	; YES	
1223	003642	104220			HLT	!MR	; BITS (77) SHOULD = 0	
1224	003644	104400			TST26: SCOPE			
1225								
1226	003646	012777	000050	175254	MOV	#50, @RSMR	; SET BITS	
1227	003654	017737	175250	001174	MOV	@RSMR, WORK	; PUT IN WORKABLE REG	
1228	003662	042737	177700	001174	BIC	#177700, WORK	; CLEAR JUNK	
1229	003670	022737	000050	001174	CMP	#50, WORK	; DID THESE BITS SET	
1230	003676	001401			BEQ	+4	; YES	
1231	003700	104220			HLT	!MR	; BITS (50) SHOULD BE SET	
1232	003702	104400			TST27: SCOPE			
1233								
1234	003704	012777	000020	175216	MOV	#20, @RSMR	; SET BITS	
1235	003712	017737	175212	001174	MOV	@RSMR, WORK	; PUT INTO WORKABLE REG	
1236	003720	042737	177700	001174	BIC	#177700, WORK	; CLEAR JUNK	
1237	003726	022737	000020	001174	CMP	#20, WORK	; DID THEY SET	
1238	003734	001401			BEQ	+4	; YES	
1239	003736	104220			HLT	!MR	; MR SHOULD AT LEAST HAVE A (21)	

27

```

1240 ;*****
1241 ;TEST 30 LOAD RANDOM NUMBERS INTO RSWC, RSDA AND RSBA
1242 ;*****
1243 TST30: SCOPE
1244
1245 003740 104400
1245 003742 012737 001000 001174 RANTS: MOV #1000,WORK ;MAKE TABLE 1000 WDS LONG
1246 003750 012701 027364 MOV #OUTBUF,R1 ;GET STARTING LOC OF TABLE
1247 003754 004537 026634 JSR R5,RANDOM ;GENERATE #
1248 003760 012704 027364 MOV #OUTBUF,R4 ;SETUP FOR COMPARE
1249 003764 012737 003772 001010 LOP1: MOV #LOP1,LAD ;SETUP LOOP ADDR
1250 003772 012703 001000 4$: MOV #1000,R3 ;LOAD TEST COUNTER
1251 003776 005303 BEQ R3 ;DONE WITH COMPARE?
1252 004000 001413 1$: BEQ R5 ;YES
1253 004002 013705 001110 MOV RSWC,R5 ;GET WC ADDRESS
1254 004006 011415 MOV (R4),(R5) ;LOAD WC
1255 004010 021524 CMP (R5),(R4)+ ;IS IT CORRECT?
1256 004012 001771 BEQ 4$ ;YES
1257 004014 017700 175070 MOV @RSWC,BAD ;GET BAD WC
1258 004020 014401 MOV -(R4),GOOD ;GET GOOD ANS
1259 004022 104000 HLT ;TYPE THEM OUT
1260 004024 005724 TST (R4)+ ;UPDATE RANDOM NUMBER
1261 004026 000763 BR 4$ ;CONT
1262 004030 012704 027364 1$: MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
1263 004034 012737 004042 001010 LOP2: MOV #LOP2,LAD ;SETUP LOOP ADDR
1264 004042 012703 001000 3$: MOV #1000,R3 ;SETUP TEST COUNTER
1265 004046 005303 BEQ R3 ;DONE YET?
1266 004050 001413 1$: BEQ R5 ;YES
1267 004052 013705 001114 MOV RSDA,R5 ;LOAD DA ADDRESS INTO R5
1268 004056 011415 MOV (R4),(R5) ;LOAD DA
1269 004060 021524 CMP (R5),(R4)+ ;IS IT CORRECT?
1270 004062 001771 BEQ 3$ ;YES
1271 004064 017700 175024 MOV @RSDA,BAD ;GET BAD DATA
1272 004070 014401 MOV -(R4),GOOD ;GET GOOD DATA
1273 004072 104000 HLT ;TYPE IT OUT
1274 004074 005724 TST (R4)+ ;UPDATE RANDOM NUMBER
1275 004076 000763 BR 3$ ;CONTINUE
1276 004100 012704 027364 1$: MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
1277 004104 012737 004112 001010 LOP3: MOV #LOP3,LAD ;SETUP LOOP ADDR
1278 004112 012703 001000 3$: MOV #1000,R3 ;SETUP TEST COUNTER
1279 004116 005303 BEQ R3 ;DONE YET?
1280 004120 001416 2$: BEQ R5 ;YES
1281 004122 013705 001112 MOV RSBA,R5 ;LOAD ADDRESS OF BA INTO R5
1282 004126 011415 MOV (R4),(R5) ;LOAD BA
1283 004130 042714 000001 BIC #BIT0,(R4) ;CLEAR BIT 0
1284 004134 021514 CMP (R5),(R4) ;IS IT CORRECT?
1285 004136 001767 BEQ 3$ ;YES
1286 004140 017700 174746 MOV @RSBA,BAD ;GET BAD DATA
1287 004144 011401 MOV (R4),GOOD ;GET GOOD DATA
1288 004146 104000 HLT ;TYPE IT OUT
1289 004150 000400 BR 1$ ;GET OUT
1290 004152 005724 1$: TST (R4)+ ;GET NEW NUMBER
1291 004154 000760 BR 3$ ;CONTINUE
1292 004156 000240 2$: NOP

```

J03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST31

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 36
TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA

```
1293 ;*****
1294 ;TEST 31 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
1295 ;*****
1296 004160 104400 TST31: SCOPE
1297
1298 004162 104414 BITST: CLRDK ;CLEAR ALL RS REG
1299 004164 012777 003566 174712 MOV #3566,@RSCS1 ;LOAD CS1
1300 004172 112777 000005 174740 MOVB #5,@RSCS1B ;LOAD BIT
1301 004200 022777 006766 174676 CMP #6766,@RSCS1 ;DID IT LOAD?
1302 004206 001401 BEQ +4 ;YES
1303 004210 104001 HLT !CS1
1304 004212 112777 000032 174664 MOVB #32,@RSCS1
1305 004220 022777 006632 174656 CMP #6632,@RSCS1
1306 004226 001401 BEQ +4
1307 004230 104001 HLT !CS1 ;CS1 SHOULD = 6632
1308
1309 004232 104400 TST32: SCOPE
1310
1311 004234 013777 001152 174644 BITCS2: MOV UNNUM,@RSCS2 ;LOAD UNIT NUMBER
1312 004242 052777 177400 174636 BIS #177400,@RSCS2 ;LOAD ALL BITS
1313 004250 105077 174666 CLR8 @RSCS2B ;CLR UPPER BYTE
1314 004254 013701 001152 MOV UNNUM,GOOD ;GET UNIT NO.
1315 004260 052701 000100 BIS #100,GOOD ;SET OR BIT
1316 004264 017700 174616 MOV @RSCS2,BAD ;GET CS2
1317 004270 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1318 004272 001401 BEQ +4 ;YES
1319 004274 104000 HLT ;LOAD BYTE DID NOT WORK
1320
1321 004276 104400 TST33: SCOPE
1322
1323 004300 012777 025252 174602 BITWC: MOV #25252,@RSWC ;LOAD WC
1324 004306 112777 000377 174630 MOVB #377,@RSWCB ;LOAD BIT
1325 004314 022777 177652 174566 CMP #177652,@RSWC ;DID IT LOAD?
1326 004322 001401 BEQ +4 ;YES
1327 004324 104010 HLT !WC ;NO WC SHOULD =177652
1328 004326 112777 000123 174554 MOVB #123,@RSWC
1329 004334 022777 177523 174546 CMP #177523,@RSWC
1330 004342 001401 BEQ +4
1331 004344 104010 HLT !WC ;WC SHOULD = 177523
1332
1333 004346 104400 TST34: SCOPE
1334
1335 004350 012777 025252 174534 BITBA: MOV #25252,@RSBA ;LOAD BA
1336 004356 112777 000377 174562 MOVB #377,@RSBAB ;LOAD BIT
1337 004364 022777 177652 174520 CMP #177652,@RSBA ;DID IT LOAD?
1338 004372 001401 BEQ +4 ;YES
1339 004374 104020 HLT !BA ;BA SHOULD =177652
1340 004376 112777 000125 174506 MOVB #125,@RSBA
1341 004404 022777 177524 174500 CMP #177524,@RSBA
1342 004412 001401 BEQ +4
1343 004414 104020 HLT !BA ;BA SHOULD = 177525
1344 004416 104414 CLRDK ;CLEAR ALL RS REG
```

K03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST35

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 37
TEST DATA LATE IN CS2

```

1345 ;*****
1346 ;TEST 35 TEST DATA LATE IN CS2
1347 ;*****
1348 004420 104400 †TST35: SCOPE
1349
1350 ;DO A READ FROM SILO: SHOULD GET DLT + TRE ERROR BECAUSE SILO IS EMPTY
1351
1352 004422 104414 SILOB: CLRDK ;CLEAR ALL RS REG
1353 004424 017700 174476 MOV @RSDB,BAD ;READ FROM EMPTY SILO
1354 004430 017700 174452 MOV @RSCS2,BAD ;GET CS2
1355 004434 012701 100100 MOV #100100,GOOD ;GET CORRECT ANS
1356 004440 053701 001152 BIS UNNUM,GOOD ;FOR CS2
1357 004444 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1358 004446 001401 BEQ .+4 ;YES
1359 004450 104200 HLT !CS2 ;SHOULD HAVE DLT ERROR
1360 004452 022777 144200 174424 CMP #144200,@RSCS1 ;DID SC AND TRE SET?
1361 004460 001401 BEQ .+4 ;YES
1362 004462 104001 HLT !CS1 ;SC AND TRE SHOULD BE SET
1363 004464 012777 040000 174412 MOV #TRE,@RSCS1 ;CLEARS ERROR BIT
1364 004472 032777 140000 174404 BIT #140000,@RSCS1 ;DID SC + TRE CLEAR
1365 004500 001401 BEQ .+4 ;YES
1366 004502 104001 HLT !CS1 ;TRE AND SC SHOULD BE 0
1367 004504 017700 174376 MOV @RSCS2,BAD ;GET CS2
1368 004510 042701 100000 BIC #BIT15,GOOD ;GET CORRECT ANS
1369 004514 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1370 004516 001401 BEQ .+4 ;YES
1371 004520 104200 HLT !CS2 ;DLT SHOULD BE 0
1372 ;*****
1373 ;TEST 36 LOAD RSDB WITH ALL ONES AND ALL ZEROS
1374 ;*****
1375 004522 104400 †TST36: SCOPE
1376
1377 004524 104414 ZERONE: CLRDK ;CLEAR ALL RS REG
1378 004526 005077 174374 CLR @RSDB ;LOAD DB WITH ALL 0
1379 004532 012777 177777 174366 MOV #177777,@RSDB ;LOAD DB WITH ALL ONES
1380 004540 012737 002000 001174 MOV #2000,WORK ;TIME OUT ROUTINE
1381 004546 012701 000300 MOV #300,GOOD ;GET CORRECT FOR CS2
1382 004552 053701 001152 BIS UNNUM,GOOD
1383 004556 017700 174324 2$: MOV @RSCS2,BAD ;GET CS2
1384 004562 020100 CMP GOOD,BAD ;IS IT CORRECT?
1385 004564 001404 BEQ 3$ ;YES
1386 004566 005337 001174 DEC WORK ;TO WAIT FOR OR
1387 004572 001371 BNE 2$ ;TO SET
1388 004574 104200 HLT !CS2 ;OR SHOULD BE SET
1389 004576 005001 3$: CLR GOOD
1390 004600 017700 174322 MOV @RSDB,BAD ;LOAD BAD WITH DB
1391 004604 020100 CMP GOOD,BAD ;IS BAD CORRECT
1392 004606 001401 BEQ .+4 ;YES
1393 004610 104000 HLT ;COULD NOT FLOAT 0 THROUGH DB
1394 004612 012701 177777 MOV #-1,GOOD ;LOAD GOOD WITH ANS
1395 004616 017700 174304 MOV @RSDB,BAD ;GET DATA FROM DB
1396 004622 020100 CMP GOOD,BAD ;IS DB CORRECT
1397 004624 001401 BEQ .+4 ;YES
1398 004626 104000 HLT ;BAD SHOULD = 177777

```

L03

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST36

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 38
LOAD RSDB WITH ALL ONES AND ALL ZEROS

```

1399 004630 104400          TST37: SCOPE
1400                                ;TEST FOR 66 LOCATIONS IN SILO PUT COUNT IN EVERY LOCATION
1401
1402 004632 104414          SILO:  CLRDK                                ;CLEAR ALL RS REG
1403 004634 005001          CLR      R1                                ;CLEAR COUNTER
1404 004636 005201          1$:  INC      R1                                ;INCREMENT COUNTER
1405 004640 010177 174262    MOV      R1,RSDB                            ;LOAD SILO
1406 004644 022701 000102    CMP      #66.,R1                            ;LAST LOC. YET?
1407 004650 001372          BNE      1$                                ;NO LOOP AGAIN
1408 004652 012701 000200    MOV      #200,GOOD                          ;GET CORRECT ANS FOR CS2
1409 004656 053701 001152    BIS      UNNUM,GOOD
1410 004662 017700 174220    MOV      RS2CS2,BAD
1411 004666 020100          CMP      GOOD,BAD
1412 004670 001401          BEQ      .+4
1413 004672 104200          HLT      !CS2
1414 004674 005001          CLR      GOOD
1415 004676 005201          2$:  INC      GOOD
1416 004700 022701 000103    CMP      #67.,GOOD
1417 004704 001405          BEQ      3$
1418 004706 017700 174214    MOV      RSDB,BAD
1419 004712 020100          CMP      GOOD,BAD
1420 004714 001770          BEQ      2$
1421 004716 104000          HLT
1422 004720 032777 000200 174160 3$: BIT      #OR,RS2CS2
1423 004726 001401          BEQ      .+4
1424 004730 104200          HLT      !CS2
1425
1426                                ;NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT ERROR
1427
1428 004732 005001          4$:  CLR      R1                                ;CLEAR COUNTER
1429 004734 005201          INC      R1                                ;ADD 1 TO COUNT
1430 004736 010177 174164    MOV      R1,RSDB                            ;PUT INTO COUNTER
1431 004742 022701 000103    CMP      #67.,R1                            ;DONE YET?
1432 004746 001401          BEQ      .+4
1433 004750 000771          BR       4$
1434 004752 032777 100000 174126 BIT      #DLT,RS2CS2
1435 004760 001001          BNE      .+4
1436 004762 104200          HLT      !CS2
1437
1438                                ;DOES SILO CHANGE WITH 67TH WORD: IT SHOULD NOT
1439
1440 004764 017700 174136    MOV      RSDB,BAD
1441 004770 012701 000001    MOV      #1,GOOD
1442 004774 020100          CMP      GOOD,BAD
1443 004776 001401          BEQ      .+4
1444 005000 104000          HLT
1445 005002 104400          TST40: SCOPE

```

;SILO SHOULD NOT HAVE MOVED

M03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST36

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 39
LOAD RSDB WITH ALL ONES AND ALL ZEROS

```

1446                ;FLOAT A 1 AND A 0 THROUGH THE SILO
1447
1448 005004 104414  SILOFL: CLRDK                ;CLEAR ALL RS REG
1449 005006 000241                ;CLEAR CARRY TO FLOAT A 0
1450 005010 012701 000001        ;GET UP DATA FOR INPUT TO SILO
1451 005014 010177 174106        ;LOAD DB
1452 005020 006101                ;SHIFT BIT
1453 005022 103401                ;DONE YET SHIFTING?
1454 005024 000773                ;NO
1455 005026 012701 177776        ;SET ALL ONES
1456 005032 000261                ;SET CARRY TO ROL
1457 005034 010177 174066        ;LOAD SILO
1458 005040 006101                ;SHIFT 0
1459 005042 103774                ;LOOP TILL DONE
1460
1461                ;NOW TEST OUTPUT
1462
1463 005044 000241                ;CLEAR CARRY
1464 005046 012701 000001        ;CORRECT ANS
1465 005052 017700 174050        ;GET DATA FROM DB
1466 005056 020100                ;IS DB DATA GOOD?
1467 005060 001401                ;YES
1468 005062 104000                ;DB COULD NOT BUBBLE CORRECTLY
1469 005064 006101                ;SETUP FOR NEXT ANS
1470 005066 103401                ;DONE YET?
1471 005070 000770                ;NO
1472 005072 012701 177776        ;SETUP FOR ANS
1473 005076 017700 174024        ;GET DATA FROM DB
1474 005102 020100                ;IS IT CORRECT?
1475 005104 001401                ;YES
1476 005106 104000                ;DB WRONG
1477 005110 000261                ;SET CARRY TO ROL
1478 005112 006101                ;SETUP FOR NEXT ANS
1479 005114 103770                ;LOOP TILL DONE

```

N03

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST41

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 40
TEST NO-OP FUNCTION

```
1480 ;*****
1481 ;TEST 41 TEST NO-OP FUNCTION
1482 ;*****
1483 005116 104400 †TST41: SCOPE
1484
1485 005120 104414 NOOP: CLRDK ;CLEAR ALL RS REG
1486 005122 012737 177777 027364 MOV #177777,OUTBUF ;DATA TO BE XFERED
1487 005130 013777 001102 173754 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1488 005136 012777 177777 173744 MOV #-1,@RSWC ;LOAD WC WITH -1
1489 005144 012777 000001 173732 MOV #1,@RSCS1 ;DO NO-OP FUNCTION
1490 005152 032777 000001 173724 BIT #1,@RSCS1 ;DID GO BIT CLEAR
1491 005160 001401 BEQ +4 ;YES
1492 005162 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
1493 005164 005777 173730 TST @RSER ;DID ANY ERRORS OCCUR?
1494 005170 001401 BEQ +4 ;NO
1495 005172 104002 HLT !ER ;ALL ERROR BITS SHOULD BE 0
1496 005174 022777 177777 173706 CMP #-1,@RSWC ;DID WC MOVE?
1497 005202 001401 BEQ +4 ;NO
1498 005204 104010 HLT !WC ;WC SHOULD = 177777
1499 005206 005777 173702 TST @RSDA ;DID DA MOV
1500 005212 001401 BEQ +4 ;NO
1501 005214 104004 HLT !DA ;DA SHOULD =0
1502 005216 023777 001102 173666 CMP @#OBUFSV,@RSBA ;DID BA MOVE
1503 005224 001401 BEQ +4 ;NO
1504 005226 104020 HLT !BA ;BA MOVED
1505 005230 033777 001156 173664 BIT UNCMP,@RSAS ;AS SHOULD NOT SET ON
1506 005236 001401 BEQ +4 ;A NO-OP FUNCTION
1507 005240 104100 HLT !AS ;AS SET WHY?
```



```

1508 :*****
1509 :TEST 42 TEST NO-OP FUNCTION WITH ERROR BITS SET
1510 :*****
1511 TST42: SCOPE
1512
1513 MNOOP: CLRDK ;CLEAR ALL RS REG
1514 MOV #7, @RSER ;LOAD ER
1515 BIT UNCMP, @RSAS ;IS ATA BIT SET?
1516 BNE +4 ;YES
1517 HLT !AS ;AS BIT SHOULD BE SET
1518 MOV #177777, @OUTBUF ;DATA TO BE XFERED
1519 MOV @@OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
1520 MOV #-1, @RSWC ;LOAD WC WITH -1
1521 MOV #1, @RSCSI ;DO NO-OP FUNCTION
1522 BIT #1, @RSCSI ;DID GO BIT CLEAR
1523 BEQ +4 ;YES
1524 HLT !CSI ;GO BIT SHOULD BE CLEARED
1525 CMP #150600, @RSOS ;DID ERR BITS SET?
1526 BEQ +4 ;NO
1527 HLT !DS ;ERR BIT SHOULD BE 0
1528 CMP #-1, @RSWC ;DID WC MOVE?
1529 BEQ +4 ;NO
1530 HLT !WC ;WC SHOULD = 1777777
1531 TST @RSDA ;DID DA MOV
1532 BEQ +4 ;NO
1533 HLT !DA ;DA SHOULD =0
1534 CMP @@OBUFSV, @RSBA ;DID BA MOVE
1535 BEQ +4 ;NO
1536 HLT !BA ;BA MOVED
1537 BIT UNCMP, @RSAS ;AS SHOULD BE SET
1538 BNE +4 ;IS I?
1539 HLT !AS ;NO
1540 CMP #7, @RSER ;DID ER CHANGE?
1541 BEQ +4 ;NO
1542 HLT !ER ;ER SHOULD NOT CHANGE

```

```

1543 ;*****
1544 ;TEST 43 TEST NO-OP FUNCTION CODE 21
1545 ;*****
1546 005422 104400 TST43: SCOPE
1547
1548 005424 104414 NOOP21: CLRDK ;CLEAR ALL RS REG
1549 005426 012737 177777 027364 MOV #177777,OUTBUF ;DATA TO BE XFERED
1550 005434 013777 001102 173450 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
1551 005442 012777 177777 173440 MOV #-1,@RSWC ;LOAD WC WITH -1
1552 005450 012777 000021 173426 MOV #21,@RSCS1 ;DO NO-OP FUNCTION
1553 005456 032777 000001 173420 BIT #1,@RSCS1 ;DID GO BIT CLEAR
1554 005464 001401 BEQ .+4 ;YES
1555 005466 104001 HLT :CS1 ;GO BIT SHOULD BE CLEARED
1556 005470 005777 173424 TST @RSER ;DID ANY ERRORS OCCUR?
1557 005474 001401 BEQ .+4 ;NO
1558 005476 104002 HLT :ER ;ALL ERROR BITS SHOULD BE 0
1559 005500 022777 177777 173402 CMP #-1,@RSWC ;DID WC MOVE?
1560 005506 001401 BEQ .+4 ;NO
1561 005510 104010 HLT :WC ;WC SHOULD = 1777777
1562 005512 005777 173376 TST @RSDA ;DID DA MOV
1563 005516 001401 BEQ .+4 ;NO
1564 005520 104004 HLT :DA ;DA SHOULD =0
1565 005522 023777 001102 173362 CMP @#0BUFSV,@RSBA ;DID BA MOVE
1566 005530 001401 BEQ .+4 ;NO
1567 005532 104020 HLT :BA ;BA MOVED
1568 005534 033777 001156 173360 BIT UNCMP,@RSAS ;AS SHOULD NOT SET ON
1569 005542 001401 BEQ .+4 ;A NO-OP FUNCTION
1570 005544 104100 HLT :AS ;AS SET WHY?
1571 005546 022777 004220 173330 CMP #4220,@RSCS1 ;IS CSI CORRECT?
1572 005554 001401 BEQ .+4 ;YES
1573 005556 104040 HLT :DS ;CSI SHOULD = 4220

```

```

1574
1575
1576
1577 005560 104400
1578
1579 005562 104414
1580 005564 012777 000007 173326
1581 005572 033777 001156 173322
1582 005600 001001
1583 005602 104100
1584 005604 012737 177777 027364
1585 005612 013777 001102 173272
1586 005620 012777 177777 173262
1587 005626 012777 000021 173250
1588 005634 032777 000001 173242
1589 005642 001401
1590 005644 104001
1591 005646 022777 150600 173242
1592 005654 001401
1593 005656 104040
1594 005660 022777 177777 173222
1595 005666 001401
1596 005670 104010
1597 005672 005777 173216
1598 005676 001401
1599 005700 104004
1600 005702 023777 001102 173202
1601 005710 001401
1602 005712 104020
1603 005714 033777 001156 173200
1604 005722 001001
1605 005724 104100
1606 005726 022777 000007 173164
1607 005734 001401
1608 005736 104002
1609 005740 022777 104220 173136
1610 005746 001401
1611 005750 104040

```

```

:*****
:TEST 44 TEST NO-OP FUNCTION CODE 21 WITH ERROR BITS SET
:*****
TST44: 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,RSCSI ;LOAD WC WITH -1
BIT #1,RSCSI ;DO NO-OP FUNCTION
BEQ +4 ;DID GO BIT CLEAR
HLT !CSI ;YES
CMP #150600,RSDS ;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,RSCSI ;ER SHOULD NOT CHANGE
BEQ +4 ;IS CSI CORRECT?
HLT !DS ;YES
;CSI SHOULD = 104220

```

E04

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST45

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 44
TEST DRIVE CLEAR FUNCTION WITH ERRORS SET

1612				
1613				
1614				
1615	005752	104400		
1616				
1617				
1618				
1619	005754	104414		
1620	005756	012777	177777	173124
1621	005764	012777	177777	173122
1622	005772	012777	177017	173120
1623	006000	012777	000070	173122
1624	006006	012777	000011	173070
1625	006014	017700	173066	
1626	006020	042700	177640	
1627	006024	013701	001152	
1628	006030	052701	000100	
1629	006034	020100		
1630	006036	001401		
1631	006040	104000		
1632	006042	005777	173046	
1633	006046	001401		
1634	006050	104004		
1635	006052	005777	173042	
1636	006056	001401		
1637	006060	104002		
1638	006062	017737	173042	001174
1639	006070	042737	177707	001174
1640	006076	022737	000070	001174
1641	006104	001401		
1642	006106	104220		
1643	006110	022777	004210	172766
1644	006116	001401		
1645	006120	104001		
1646	006122	033777	001156	172772
1647	006130	001401		
1648	006132	104100		
1649	006134	022777	177777	172746
1650	006142	001401		
1651	006144	104010		

```

:*****
:TEST 45 TEST DRIVE CLEAR FUNCTION WITH ERRORS SET
:*****
TST45: SCOPE
:FIRST SET ALL R/W BITS IN DISK REG
:DO DRIVE CLEAR-ALL R/W BITS SHOULD BE CLEARED

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

```

F04

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST45

RH11-RSC3LA-RSC3-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 45
TEST DRIVE CLEAR FUNCTION WITH ERRORS SET

```

1652 ;DO ONE WORD WRITE
1653 ;*****
1654 ;TEST 46 EXECUTE THE ONE WORD WRITE
1655 ;*****
1656 006146 104400 †TST46: SCOPE
1657
1658 006150 104414 WRTST: CLRDK ;CLEAR ALL RS REG
1659 006152 012737 177777 027364 MOV #177777,OUTBUF ;DATA TO BE X-FERED
1660 006160 013777 001102 172724 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1661 006166 012777 177777 172714 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1662 006174 012777 000060 172702 1S: MOV #60,@RSCS1 ;SET FUNCTION WITH NO GO BIT
1663 006202 022777 177777 172700 CMP #-1,@RSWC ;DID WC MOVE?
1664 006210 001401 BEQ .+4 ;NO
1665 006212 104010 HLT !WC ;WC MOVED
1666 006214 023777 001102 172670 CMP @#OBUFSV,@RSBA ;DID RSBA MOVE?
1667 006222 001401 BEQ .+4 ;NO
1668 006224 104020 HLT !BA ;BA MOVED
1669 006226 052777 000001 172650 2S: BIS #BIT0,@RSCS1 ;SET GO BIT
1670 006234 105777 172644 TSTB @RSCS1 ;TEST FOR RDY=0
1671 006240 100001 BPL .+4 ;RDY=0
1672 006242 104001 HLT !CS1 ;RDY SHOULD = 0
1673 006244 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
1674 006250 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1675 006252 022777 000001 172634 CMP #1,@RSDA ;IS RSDA CORRECT
1676 006260 001401 BEQ .+4 ;RSDA OK
1677 006262 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1678 006264 022777 004260 172612 3S: CMP #4260,@RSCS1 ;IS ERROR FLAG SET?
1679 006272 001401 BEQ .+4 ;NO! X-FER OK
1680 006274 104047 HLT !CS1!ER!DS!DA ;ERROR DURING X-FER
1681 006276 005777 172606 4S: TST @RSWC ;FETCH WORD COUNT
1682 006302 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1683 006304 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1684 006306 022777 010600 172602 CMP #10600,@RSDS ;IS RSDS OK?
1685 006314 001401 BEQ .+4 ;YES
1686 006316 104044 HLT !DS!DA ;NO
1687 006320 013701 001152 MOV UNUM,GOOD ;GET UNIT #
1688 006324 052701 000100 BIS #100,GOOD ;SET IR BIT
1689 006330 017700 172552 MOV @RSCS2,BAD ;GET CS2
1690 006334 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1691 006336 001401 BEQ .+4 ;YES
1692 006340 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1693 006342 017700 172544 MOV @RSBA,BAD ;GET BA DATA
1694 006346 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1695 006352 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
1696 006356 020100 CMP GOOD,BAD ;IS RSBA CORRECT
1697 006360 001401 BEQ .+4 ;YES
1698 006362 104000 HLT ;BA FAILED TO INCREMENT
1699 006364 005777 172530 TST @RSER ;DID ANY ERRORS SET?
1700 006370 001401 BEQ .+4 ;NO
1701 006372 104040 HLT !DS

```

```

1702 ;TEST READ FUNCTION
1703
1704 ;*****
1705 ;TEST 47 EXECUTE THE ONE WORD READ
1706 ;*****
1707 006374 104400 TST47: SCOPE
1708
1709 006376 104414 RDTST: CLRDK ;CLEAR ALL RS REG
1710 006400 005037 027364 CLR OUTBUF ;CLR TO READ INTO
1711 006404 013777 001102 172500 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1712 006412 012777 177777 172470 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1713 006420 012777 000071 172456 1S: MOV #71,@RSCS1 ;GO READ
1714 006426 105777 172452 2S: TSTB @RSCS1 ;TEST FOR BUSY=1
1715 006432 100001 BPL .+4 ;BUSY SET
1716 006434 104001 HLT !CS1 ;BUSY NOT SET
1717 006436 004737 026600 JSR PC, WAITRY ;WAIT FOR READY
1718 006442 104001 HLT !CS1 ;TIMEOUT RDY DID NOT SET
1719 006444 022777 000001 172442 CMP #BIT0,@RSDA ;WAS RSDA INCREMENTED BY 1
1720 006452 001401 BEQ .+4 ;RSDA OK
1721 006454 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
1722 006456 022777 004270 172420 3S: CMP #4270,@RSCS1 ;IS ERROR FLAG SET?
1723 006464 001401 BEQ .+4 ;NO! X-FER OK
1724 006466 104043 HLT !CS1!ER!DS ;RSCS1 SHOULD = 270
1725 006470 005777 172414 4S: TST @RSWC ;TEST WC
1726 006474 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1727 006476 104010 HLT !WC ;SHOULD = 0
1728 006500 013701 001152 MOV UNNUM,GOOD ;GET CORRECT
1729 006504 052701 000100 BIS #100,GOOD ;ANS OF CS2
1730 006510 017700 172372 MOV @RSCS2,BAD ;GET CS2
1731 006514 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1732 006516 001401 BEQ .+4 ;YES
1733 006520 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1734 006522 017700 172364 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1735 006526 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1736 006532 062701 000002 ADD #2,GOOD ;UPDATE IT
1737 006536 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1738 006540 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1739 006542 104000 HLT ;RSBA FAILED TO INCREMENT
1740 006544 013700 027364 MOV OUTBUF,BAD ;GET DATA READ FROM DISK
1741 006550 012701 177777 MOV #-1,GOOD ;GET CORRECT ANS
1742 006554 020100 CMP GOOD,BAD ;IS OUTBUF CORRECT
1743 006556 001401 BEQ .+4 ;YES
1744 006560 104000 HLT ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK

```

```

1745 ;*****
1746 ;TEST 50 TEST WRITE CHECK
1747 ;*****
1748 006562 104400 TST50: SCOPE
1749 ;DO A ONE WORD WRITE CHECK
1750
1751 ;* * *EXECUTE THE ONE WORD WRITE CHECK* * *
1752
1753 006564 104414 WRCKT: CLRDK ;CLEAR ALL RS REG
1754 006566 012737 177777 027364 MOV #177777,OUTBUF ;DATA TO BE X-FERED
1755 006574 013777 001102 172310 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1756 006602 012777 177777 172300 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1757 006610 012777 000051 172266 1$: MOV #51,@RSCS1 ;GO WRITE CHECK
1758 006616 105777 172262 2$: TSTB @RSCS1 ;TEST FOR READY
1759 006622 100001 BPL .+4 ;NOT READY
1760 006624 104001 HLT !CS1 ;BUSY FAILED TO SET
1761 006626 004737 026600 RSWCNT: JSR PC,WAITRY ;WAIT FOR READY
1762 006632 104001 HLT !CS1 ;BUSY FAILED TO CLEAR
1763 006634 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1764 006640 052701 000100 BIS #100,GOOD ;SET BIT IR
1765 006644 017700 172236 MOV @RSCS2,BAD ;GET CS2
1766 006650 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1767 006652 001401 BEQ .+4 ;YES
1768 006654 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1769 006656 022777 004250 172220 2$: CMP #4250,@RSCS1 ;ANY ERRORS?
1770 006664 001401 BEQ .+4 ;X-FER OK
1771 006666 104046 HLT !DA!ER!DS ;ERROR DUR X-FER
1772 006670 022777 000001 172216 3$: CMP #BIT0,@RSDA ;WAS DAR INCREMENTED BY 1
1773 006676 001401 BEQ .+4 ;RSDA OK
1774 006700 104004 HLT !DA ;DAR SHOULD = 1
1775 006702 005777 172202 TST @RSWC ;TEST FOR OVERFLOW
1776 006706 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1777 006710 104010 HLT !WC ;SHOULD = 0
1778 006712 017700 172174 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1779 006716 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1780 006722 062701 000002 ADD #2,GOOD ;UPDATE IT
1781 006726 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1782 006730 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1783 006732 104000 HLT ;RSBA FAILED TO INCREMENT

```

```

1784 ;DO ONE WORD WRITE ON -B- PORT
1785 ;IF A 1 WD TRANSFER KEEPS SETTING NEM PROGRAM WILL GO AND UPDATE
1786 ;ADDRESS (OBUFSV) ON -B- PORT BY 4K AND TRY TRANSFER AGAIN UNTILL IT
1787 ;REACHES 28K. IF NO TRANSFER IT THEN SKIPS WRITE,
1788 ;READ AND WRITE CHECK ON -B- PORT
1789 ;TO INHIBIT OBUFSV FROM CHANGING SET BIT 12
1790
1791 ;*****
1792 ;TEST 51 EXECUTE THE ONE WORD WRITE ON -B- PORT
1793 ;*****
1794 G06734 104400 TST51: SCOPE
1795
1796 006736 104414 WRTSTR: CLRDK ;CLEAR ALL RS REG
1797 006740 013737 001102 001174 MOV @#OBUFSV,WORK ;GET LOC OF RSBA TO LOAD
1798 006746 012777 177777 172220 MOV #177777,@WORK ;DATA TO BE X-FERED
1799 006754 013777 001102 172130 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1800 006762 012777 177777 172120 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1801 006770 012777 002061 172106 MOV #2061,@RSCS1 ;TEST B PORT
1802 006776 105777 172102 2$: TSTB @RSCS1 ;TEST FOR RDY=0
1803 007002 100001 BPL .+4 ;RDY=0
1804 007004 104001 HLT !CS1 ;RDY SHOULD = 0
1805 007006 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
1806 007012 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1807 007014 032777 010000 172004 BIT #BIT12,@SWR ;INHIBIT ADDRESS?
1808 007022 001006 BNE 3$ ;YES
1809 007024 032777 004000 172054 BIT #BIT11,@RSCS2 ;DID NEM SET?
1810 007032 001402 BEQ 3$ ;NO
1811 007034 000137 025632 JMP FINDM ;GO FIND MEMORY ON PORT B
1812 007040 022777 000001 172046 3$: CMP #1,@RSDA ;IS RSDA CORRECT
1813 007046 001401 BEQ .+4 ;RSDA OK
1814 007050 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1815 007052 022777 006260 172024 CMP #6260,@RSCS1 ;IS CS1 CORRECT?
1816 007060 001401 BEQ 4$ ;YES
1817 007062 104001 HLT !CS1
1818 007064 005777 172020 4$: TST @RSWC ;FETCH WORD COUNT
1819 007070 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1820 007072 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1821 007074 022777 010600 172014 CMP #10600,@RSDS ;IS RSDS OK?
1822 007102 001401 BEQ .+4 ;YES
1823 007104 104044 HLT !DS!DA ;NO
1824 007106 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1825 007112 052701 000100 BIS #100,GOOD ;SET IR BIT
1826 007116 017700 171764 MOV @RSCS2,BAD ;GET CS2
1827 007122 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1828 007124 001401 BEQ .+4 ;YES
1829 007126 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1830 007130 017700 171756 MOV @RSBA,BAD ;GET BA DATA
1831 007134 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1832 007140 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
1833 007144 020100 CMP GOOD,BAD ;IS RSBA CORRECT
1834 007146 001401 BEQ .+4 ;YES
1835 007150 104000 HLT ;BA FAILED TO INCREMENT

```



```

1836                                     ;*****
1837                                     ;TEST 52 EXECUTE THE ONE WORD READ ON -B- PORT
1838                                     ;*****
1839 007152 104400 TST52: SCOPE
1840
1841 007154 104414 RDTSTB: CLRDK ;CLEAR ALL RS REG
1842 007156 005037 001174 CLR WORK ;CLR TO READ INTO
1843 007162 013777 001102 171722 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
1844 007170 012777 177777 171712 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1845 007176 012777 002071 171700 MOV @2071,@RSCS1 ;B PORT
1846 007204 105777 171674 25: TSTB @RSCS1 ;TEST FOR BUSY=1
1847 007210 100001 BPL .+4 ;BUSY SET
1848 007212 104001 HLT !CS1 ;BUSY NOT SET
1849 007214 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
1850 007220 104001 HLT !CS1 ;TIMEOUT RDY DID NOT SET
1851 007222 022777 000001 171664 CMP @BIT0,@RSDA ;WAS RSDA INCREMENTED BY 1
1852 007230 001401 BEQ .+4 ;RSDA OK
1853 007232 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
1854 007234 022777 006270 171642 CMP @6270,@RSCS1 ;TST B PORT
1855 007242 001401 BEQ 45 ;OK
1856 007244 104001 HLT !CS1 ;CS1 SHOULD = 6270
1857 007246 005777 171636 46: TST @RSWC ;TEST WC
1858 007252 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1859 007254 104010 HLT !WC ;SHOULD = 0
1860 007256 013701 001152 MOV UNNUM,GOOD ;GET CORRECT
1861 007262 052701 000100 BIS #100,GOOD ;ANS OF CS2
1862 007266 017700 171614 MOV @RSCS2,BAD ;GET CS2
1863 007272 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1864 007274 001401 BEQ .+4 ;YES
1865 007276 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1866 007300 017700 171606 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1867 007304 013701 001102 MOV @#0BUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1868 007310 062701 000002 ADD #2,GOOD ;UPDATE IT
1869 007314 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1870 007316 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1871 007320 104000 HLT ;RSBA FAILED TO INCREMENT
1872 007322 013737 001102 001174 MOV @#0BUFSV,WORK ;GET DATA READ FROM DISK
1873 007330 017700 171640 MOV @WORK,BAD
1874 007334 012701 177777 MOV #-1,GOOD ;GET CORRECT ANS
1875 007340 020100 CMP GOOD,BAD ;IS OUTBUF CORRECT
1876 007342 001401 BEQ .+4 ;YES
1877 007344 104000 HLT ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK

```

K04

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST53

RH11-RS03, A-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 50
TEST WRITE CHECK ON -B- PORT

```

1878 ;*****
1879 ;TEST 53 TEST WRITE CHECK ON -B- PORT
1880 ;*****
1881 007346 104400 †TST53: SCOPE
1882
1883
1884 007350 104414 WRCKTB: CLRDK ;CLEAR ALL RS REG
1885 007352 013737 001102 001174 MOV @#0BUFSV,WORK ;GET LOC FOR
1886 007360 012777 177777 171606 MOV #177777,@WORK ;DATA TO BE X-FERED
1887 007366 013777 001102 171516 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
1888 007374 012777 177777 171506 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1889 007402 012777 702051 171474 MOV #2051,@RSCS1 ;B PORT
1890 007410 105777 71470 25: TSTB @RSCS1 ;TEST FOR READY
1891 007414 100001 BPL .+4 ;NOT READY
1892 007416 104001 HLT !CS1 ;BUSY FAILED TO SET
1893 007420 004737 06600 JSR PC, WAITRY ;WAIT FOR READY
1894 007424 104001 HLT !CS1 ;BUSY FAILED TO CLEAR
1895 007426 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1896 007432 052701 000100 BIS #100,GOOD ;SET BIT IR
1897 007436 017700 171444 MOV @RSCS2,BAD ;GET CS2
1898 007442 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1899 007444 001401 BEQ .+4 ;YES
1900 007446 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1901 007450 022777 006250 171426 CMP #6250,@RSCS1 ;IS CS1 CORRECT?
1902 007456 001401 BEQ 3$ ;YES
1903 007460 104001 HLT !CS1 ;CS1 SHOULD = 6250
1904 007462 022777 000001 171424 3$: CMP #BIT0,@RSDA ;WAS DAR INCREMENTED BY 1
1905 007470 001401 BEQ .+4 ;RSDA OK
1906 007472 104004 HLT !DA ;DAR SHOULD = 1
1907 007474 005777 171410 TST @RSWC ;TEST FOR OVERFLOW
1908 007500 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1909 007502 104010 HLT !WC ;SHOULD = 0
1910 007504 017700 171402 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1911 007510 013701 001102 MOV @#0BUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1912 007514 062701 000002 ADD #2,GOOD ;UPDATE IT
1913 007520 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1914 007522 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1915 007524 104000 HLT ;RSBA FAILED TO INCREMENT

```


MO4

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST54

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 52
DESELECT THEN SELECT UNIT NUMBER TIMING TEST

```

1966 ;TEST CURRENT ADDRESS INHIBT-BAI IN RSCS2
1967 ;DO A ONE WORD WRITE AND SEE
1968 ;IF RSBA INCREMENTED AFTER THE X-FER
1969
1970 ;*****
1971 ;TEST 55 TEST BAI IN RSCS2
1972 ;*****
1973 007752 104400 TST55: SCOPE
1974
1975 007754 104414 BAITST: CLDK ;CLEAR ALL RS REG
1976 007756 013777 001102 171126 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDR
1977 007764 012777 177777 171116 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1978 007772 052777 000010 171106 BIS #BAI,@RSCS2 ;SET BAI BIT
1979 010000 012777 000061 171076 MOV #61,@RSCS1 ;WRITE
1980 010006 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
1981 010012 104001 HLT !CS1 ;RDY DID NOT SET
1982 010014 013701 001102 171066 1$: MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD BE
1983 010020 017700 MOV @RSBA,BAD ;WHAT RSBA IS
1984 010024 020100 CMP GOOD,BAD ;COMPARE
1985 010026 001401 BEQ .+4 ;YES
1986 010030 104000 HLT ;BAD=OUTBUF GOOD = CORRECT ANS
1987 010032 005777 171062 TST @RSER ;ANY ERRORS?
1988 010036 001401 BEQ .+4 ;NO
1989 010040 104040 HLT !DS ;YES
1990 010042 104414 CLDK ;CLEAR ALL RS REG
1991 010044 032777 000010 171034 BIT #BAI,@RSCS2 ;DID BAI CLEAR?
1992 010052 001401 BEQ .+4 ;YES
1993 010054 104002 HLT !ER ;BAI DID NOT SET

```

N04

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST56

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 53
TEST NON-EXISTENT MEMORY ERROR BIT IN CS2

```

1994 ;*****
1995 ;TEST 56 TEST NON-EXISTENT MEMORY ERROR BIT IN CS2
1996 ;*****
1997 010056 104400 TST56: SCOPE
1998
1999 010060 104414 NXMTSM: CLRDK ;CLEAR ALL RS REG
2000 010062 052777 000010 171016 BIS #BAI,ARSCS2 ;SET BAI BIT
2001 010070 012777 177600 171012 MOV #-200,ARSWC ;SET UP WORD COUNT
2002 010076 012777 173000 171006 MOV #173000,ARSB A ;SET UP CURRENT ADDRESS
2003 010104 012777 001471 170772 MOV #1471,ARSCS1 ;READ AND LOAD A16 +A17 FOR 18 BIT ADDRESS
2004 010112 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
2005 010116 104040 HLT !DS ;READY NEVER CAME UP
2006 010120 013701 001152 TSTNEM: MOV UNNUM,GOOD ;GET UNIT NO.
2007 010124 052701 004310 BIS #4310,GOOD ;SET BAI+OR BITS
2008 010130 017700 170752 MOV ARSCS2,BAD ;GET CS2
2009 010134 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2010 010136 001401 BEQ .+4 ;YES
2011 010140 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS FOR CS2
2012 010142 022777 145670 170734 CMP #145670,ARSCS1 ;DID TRE SET?
2013 010150 001401 BEQ .+4 ;YES
2014 010152 104001 HLT !CS1 ;TRE SHOULD SET BECAUSE OF NEM
2015 010154 012777 040000 170722 MOV #TRE,ARSCS1 ;CLEAR TRE
2016 010162 017700 170720 MOV ARSCS2,BAD ;GET CS2
2017 010166 013701 001152 MOV UNNUM,GOOD ;GET DRIVE
2018 010172 052701 000310 BIS #310,GOOD ;SET IR
2019 010176 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2020 010200 001401 BEQ .+4 ;YES
2021 010202 104200 HLT !CS2 ;CS2=BAD GOOD IS CORRECT ANS FOR CS2
2022

```

```

2023 ;*****
2024 ;TEST 57 TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND ADDR. CONF BIT
2025 ;*****
2026 010204 104400 TST57: SCOPE
2027
2028 010206 104414 BLOCK: CLRDK ;CLEAR ALL RS REG
2029 010210 012777 000032 170676 MOV #32,ARSDA ;DO A SEARCH FOR SECTOR 32
2030 010216 013777 001102 170666 MOV @#0BUFSV,ARSB A ;LOAD REGS. TO MAKE
2031 010224 012777 177777 170656 MOV #-1,ARSWC ;SURE THEY DO NOT CHANGE
2032 010232 005037 001174 CLR WORK ;SETUP FOR TIMEOUT ROUTINE
2033 010236 032777 001000 170664 4$: BIT #1000,ARSMR ;WAIT FOR DISK TO
2034 010244 001004 BNE 3$ ;REACH SECTOR 32
2035 010246 005337 001174 DEC WORK ;TIME OUT
2036 010252 001371 BNE 4$ ;ROUTINE
2037 010254 104220 HLT !MR ;COULD NOT FIND SECTOR 32

```

B05

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST57

R411-RSC3LA-RSC3-RSD4 BASIC FUNCTION DIAGNOSTIC
TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND

MACY11 27(732) 25-SEP-76 09:06 PAGE 54
ADDR. CONF BIT

2038	010256	005077	170632		35:	CLR	DRSDA	:NOW SEARCH FOR 0
2039	010262	012777	000031	170614		MOV	#31,DRSCS1	:DO A BLOCK SEARCH FUNCTION
2040	010270	032777	000200	170620		BIT	#DRY,DRSDS	:IS DRY CLEARED?
2041	010276	001402				BEQ	15	:YES
2042	010300	104040				HLT	!DS	:DRY SHOULD BE CLEARED DURING A BLOCK SEARCH
2043	010302	000500				BR	00UT	:GET OUT BECAUSE OF TIMING
2044	010304	032777	020000	170604	15:	BIT	#20000,DRSDS	:IS PIP SET?
2045	010312	001001				BNE	+4	:YES
2046	010314	104040				HLT	!DS	:PIP SHOULD BE SET
2047	010316	012701	020000			MOV	#20000,GOOD	:SETUP FOR TIMEOUT ROUTINE
2048	010322	005301			25:	DEC	GOOD	:DO TIMEOUT
2049	010324	001466				BEQ	TTMOUT	:TIMED OUT
2050	010326	032777	000200	170562		BIT	#DRY,DRSDS	:DID DRY SET?
2051	010334	001772				BEQ	25	:NO
2052	010336	022777	110600	170552		CMP	#110600,DRSDS	:DID PIP CLEAR?
2053	010344	001401				BEQ	+4	:YES
2054	010346	104040				HLT	!DS	:PIP BIT DID NOT CLEAR
2055	010350	022777	104230	170526		CMP	#104230,DRSCS1	:DID SC SET?
2056	010356	001401				BEQ	+4	:YES
2057	010360	104040				HLT	!CS1!DS	:SC DID NOT SET
2058	010362	013737	001152	001174		MOV	UNNUM,WORK	:GET CORRECT AS BIT
2059	010370	005001				CLR	GOOD	:IN RSAS REG
2060	010372	000261				SEC		:THAT SHOULD
2061	010374	006101			55:	ROL	GOOD	:BE SET
2062	010376	005737	001174			TST	WORK	
2063	010402	001403				BEQ	65	
2064	010404	005337	001174			DEC	WORK	
2065	010410	000771				BR	55	
2066	010412	020177	170504		65:	CMP	GOOD,DRSAS	:IS RSAS CORRECT?
2067	010416	001403				BEQ	75	:YES
2068	010420	017700	170476			MOV	DRSAS,BAD	:NO
2069	010424	104000				HLT		
2070	010426	010177	170470		75:	MOV	GOOD,DRSAS	:CLEAR AS REG
2071	010432	005777	170464			TST	DRSAS	:DID IT CLEAR?
2072	010436	001401				BEQ	+4	:YES
2073	010440	104100				HLT	!AS	:NO
2074	010442	022777	010600	170446		CMP	#10600,DRSDS	:DID ATA CLEAR?
2075	010450	001401				BEQ	+4	:YES
2076	010452	104040				HLT	!DS	:NO
2077	010454	023777	001102	170430		CMP	#30BUFSV,DRSBA	:DID BA MOVE?
2078	010462	001401				BEQ	+4	:NO
2079	010464	104021				HLT	!CS1!BA	:BA MOVED WHY?
2080	010466	022777	177777	170414		CMP	#-1,DRSWC	:DID WC MOVE?
2081	010474	001401				BEQ	+4	:NO
2082	010476	104010				HLT	!WC	:WC MOVED WHY?
2083	010500	000401				BR	00UT	:DONE GET OUT
2084	010502	104040			TTMOUT:	HLT	!DS	:DZR NEVER CAME UP
2085	010504				00UT:			:DONE

C05

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST60

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 55
ILLEGAL FUNCTION CODE TEST CODE 3 TO 51

```

2086                                     ;*****
2087                                     ;TEST 60                               ILLEGAL FUNCTION CODE TEST CODE 3 TO 51
2088                                     ;*****
2089 010504 104400                       TST60: SCOPE
2090
2091                                     ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
2092                                     ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2093                                     ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2094                                     ;FUNCTION INTO LOCATION ILLTAB: AND 0 IN FOLLOWING LOCATION
2095
2096 010506 013737 022414 001164 ILL51: MOV      TIMES,TIMSV      ;SAVE LOOP COUNT
2097 010514 012737 000010 022414      MOV      #10,TIMES      ;LOOP TEN TIMES
2098 010522 104414                      CLRDK                      ;CLEAR ALL RS REG
2099 010524 012703 027300              15:    MOV      #ILLTAB,R3      ;GET STARTING ADD OF TABLE
2100 010530 012300                      35:    MOV      (R3)+,BAD      ;GET ILL FUN
2101 010532 001513                      BEQ                      ;DONE GET OUT
2102 010534 013777 001102 170350      MOV      @#0BUFSV,@RSBA      ;SET UP REGS.
2103 010542 012777 177777 170340      MOV      #-1,@RSWC          ;TO CHECK FOR CHANGE
2104 010550 010077 170330              25:    MOV      BAD,@RSCS1      ;DO ILLEGAL FUNCTION
2105 010554 042700 000001              BIC      #BIT0,BAD          ;CLEAR GO BIT
2106 010560 010001                      MOV      BAD,GOOD          ;MOV ILLEGAL FUN INTO GOOD
2107 010562 105777 170316              65:    TSTB      @RSCS1        ;RDY SET?
2108 010566 100375                      BPL      65                ;NO
2109 010570 052701 104200              BIS      #104200,GOOD      ;SET ERROR BITS
2110 010574 017700 170304              45:    MOV      @RSCS1,BAD      ;PUT CS1 INTO BAD
2111 010600 020100                      CMP      GOOD,BAD          ;IS CS1 CORRECT?
2112 010602 001401                      BEQ      .+4                ;YES
2113 010604 104000                      HLT                      ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2114 010606 022777 000001 170304      CMP      #1,@RSER          ;DID ILF SET?
2115 010614 001401                      BEQ      .+4                ;YES
2116 010616 104043                      HLT      !CS1!ER!DS        ;ILF DID NOT SET
2117 010620 022777 150600 170270      CMP      #150600,@RSDS     ;IS DS GOOD?
2118 010626 001401                      BEQ      .+4                ;YES
2119 010630 104043                      HLT      !CS1!ER!DS        ;ERR DID NOT SET
2120 010632 017700 170250              MOV      @RSCS2,BAD        ;GET CS2
2121 010636 013701 001152              MOV      UNNUM,GOOD        ;GET UNIT #
2122 010642 052701 000100              BIS      #100,GOOD         ;SET IR BIT
2123 010646 020100                      CMP      GOOD,BAD          ;IS CS2 CORRECT?
2124 010650 001401                      BEQ      .+4                ;YES
2125 010652 104000                      HLT                      ;GOOD = CORRECT ANS FOR CS2
2126 010654 013701 001156              MOV      UNCMP,GOOD        ;GET CORRECT DRIVE
2127 010660 042701 177400              BIC      #177400,GOOD      ;CLEAR UNWANTED BITS
2128 010664 017700 170232              MOV      @RSAS,BAD        ;GET RSAS REG
2129 010670 020001                      CMP      BAD,GOOD          ;DID CORRECT UNIT ANSWER?
2130 010672 001401                      BEQ      .+4                ;YES
2131 010674 104100                      HLT      !AS                ;NO WRONG DRIVE ANSWERED

```

```

2132 010676 023777 001102 170206      CMP      @#0BUFSV,@RSBA      ;DID BA MOVE
2133 010704 001401                BEQ      +4                  ;NO
2134 010706 104021                HLT      !CS1!BA            ;BA MOVED ON AN ILLEGAL FUNCTION
2135 010710 022777 177777 170172      CMP      #-1,@RSWC          ;DID WC MOVE?
2136 010716 001401                BEQ      +4                  ;NO
2137 010720 104011                HLT      !CS1!WC            ;WC MOVED
2138 010722 005777 170166      TST      @RSDA              ;DID DA MOVE
2139 010726 001401                BEQ      +4                  ;NO
2140 010730 104005                HLT      !CS1!DA            ;DA MOVED
2141 010732 104414                CLRDK                     ;CLEAR ALL ERRORS
2142 010734 005777 170160      TST      @RSER              ;DID ERRORS CLEAR
2143 010740 001401                BEQ      +4                  ;YES
2144 010742 104040                HLT      !DS                 ;ILF DID NOT CLEAR
2145 010744 022777 004200 170132      CMP      #4200,@RSCSI       ;DID ERRORS IN CS1 CLEAR
2146 010752 001401                BEQ      +4                  ;YES
2147 010754 104040                HLT      !DS
2148 010756 000137 010530      JMP      3$                 ;CONTINUE UNTIL DONE
2149 010762                ILFDN:                     ;DONE WITH ILLEGAL FUNCTION TEST
2150                ;*****
2151                ;TEST 61                ILLEGAL FUNCTION CODE TEST CODE 53 TO 77
2152                ;*****
2153 010762 104400      TST61: SCOPE
2154
2155                ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
2156                ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2157                ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2158                ;FUNCTION INTO LOCATION ILFTB2: AND 0 IN FOLLOWING LOCATION
2159
2160 010764 104414      ILLFUN: CLRDK              ;CLEAR ALL RS REG
2161 010766 012703 027342      1$: MOV      #ILFTB2,R3      ;GET TABLE OF ILL FUNS.
2162 010772 005037 001174      3$: CLR      WORK           ;CLEAR WORK
2163 010776 012300                MOV      (R3)+,BAD          ;GET ILL FUN
2164 011000 001554                BEQ      ILFDNE             ;DONE GET OUT
2165 011002 013777 001102 170102      MOVB    @#0BUFSV,@RSBA     ;SET UP REGS.
2166 011010 012777 177777 170072      MOV      #-1,@RSWC         ;TO CHECK FOR CHANGE
2167 011016 010037 001176                MOV      BAD,WORK1         ;SHOULD WE TEST
2168 011022 042737 177707 001176      BIC     #177707,WORK1      ;BA AND WC
2169 011030 022737 000060 001176      CMP     #60,WORK1         ;TO INC
2170 011036 001003                BNE     2$                 ;NO
2171 011040 012737 000007 001174      MOV     #7,WORK           ;YES
2172 011046 010077 170032      2$: MOV     BAD,@RSCSI      ;DO ILLEGAL FUNCTION
2173 011052 042700 000001      10$: BIC     #BIT0,BAD      ;CLEAR GO BIT
2174 011056 010001                MOV     BAD,GOOD          ;MOV ILLEGAL FUN INTO GOOD

```


E05

2175	011060	105777	170020		65:	TSTB	@RSCS1	;RDY SET?
2176	011064	103375				BPL	65	;NO
2177	011066	052701	144200			BIS	#14200,GOOD	;SET ERROR BITS
2178	011072	017700	170006			MOV	@RSCS1,BAD	;PUT CS1 INTO BAD
2179	011076	020100				CMP	GOOD,BAD	;IS CS1 CORRECT?
2180	011100	001401				BEQ	+.4	;YES
2181	011102	104000				HLT		;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2182	011104	022777	000001	170006		CMP	#1,@RSER	;DID ILF SET?
2183	011112	001401				BEQ	+.4	;YES
2184	011114	104043				HLT	!CS1!ER!DS	;ILF DID NOT SET
2185	011116	022777	150600	167772		CMP	#150600,@RSDS	;IS DS GOOD?
2186	011124	001401				BEQ	+.4	;YES
2187	011126	104043				HLT	!CS1!ER!DS	;ERR DID NOT SET
2188	011130	005777	167760			TST	@RSDA	;DID DA MOVE?
2189	011134	001401				BEQ	+.4	;NO
2190	011136	104005				HLT	!CS1!DA	;DA MOVED
2191	011140	005737	001174			TST	WORK	;IS THIS AN ILL WRITE FUN?
2192	011144	001025				BNE	11\$;YES
2193	011146	017700	167734			MOV	@RSCS2,BAD	;GET CS2
2194	011152	013701	001152			MOV	UNNUM,GOOD	;GET UNIT #
2195	011156	052701	001100			BIS	#1100,GOOD	;SET IR BIT
2196	011162	020100				CMP	GOOD,BAD	;IS CS2 CORRECT?
2197	011164	001401				BEQ	+.4	;YES
2198	011166	104000				HLT		;GOOD = CORRECT ANS FOR CS2
2199	011170	023777	001102	167714		CMP	@#0BUFSV,@RSBA	;DID BA MOVE
2200	011176	001401				BEQ	+.4	;NO
2201	011200	104021				HLT	!CS1!BA	;BA MOVED ON AN ILLEGAL FUNCTION
2202	011202	022777	177777	167700		CMP	#-1,@RSWC	;DID WC MOVE?
2203	011210	001401				BEQ	+.4	;NO
2204	011212	104011				HLT	!CS1!WC	;WC MOVED
2205	011214	000137	011302			JMP	4\$;CONTINUE UNTIL DONE
2206	011220	017700	167662		11\$:	MOV	@RSCS2,BAD	;GET CS2
2207	011224	013701	001152			MOV	UNNUM,GOOD	;GET UNIT #
2208	011230	052701	001300			BIS	#1300,GOOD	;SET IR BIT
2209	011234	020100				CMP	GOOD,GOOD	;IS CS2 CORRECT?
2210	011236	001401				BEQ	+.4	;YES
2211	011240	104000				HLT		;GOOD = CORRECT ANS FOR CS2
2212	011242	013737	001102	001174		MOV	@#0BUFSV,WORK	;GET BUFFER ADDR.
2213	011250	062737	000002	001174		ADD	#2,WORK	;UPDATE IT
2214	011256	023777	001174	167626		CMP	WORK,@RSBA	;DID BA MOVE
2215	011264	001401				BEQ	+.4	;YES
2216	011266	104021				HLT	!CS1!BA	;BA MOVED ON AN ILLEGAL FUNCTION
2217	011270	022777	000000	167612		CMP	#0,@RSWC	;DID WC MOVE?
2218	011276	001401				BEQ	+.4	;NO
2219	011300	104011				HLT	!CS1!WC	;WC MOVED
2220	011302	104414			4\$:	CLRDK		;CLEAR ALL ERRORS
2221	011304	022777	004200	167572		CMP	#4200,@RSCS1	;DID ERRORS CLEAR
2222	011312	001401				BEQ	+.4	;YES
2223	011314	104040				HLT	!DS	;NO
2224	011316	005777	167576			TST	@RSER	;DID ERROR CLEAR
2225	011322	001401				BEQ	+.4	;NO
2226	011324	104040				HLT	!DS	;YES
2227	011326	000137	010772			JMP	3\$;CONTINUE UNTIL DONE
2228	011332				ILFDNE:			;DONE WITH ILLEGAL FUNCTION TEST

F05

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST62

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 58
TEST ILLEGAL FUNCTION CODE 67

```

2229 ;*****
2230 ;TEST 62 TEST ILLEGAL FUNCTION CODE 67
2231 ;*****
2232 011332 104400 †TST62: SCOPE
2233
2234 011334 104414 ILF67: CLRDK ;CLEAR ALL RS REG
2235 011336 012777 177777 167544 MOV #-1,RSWC ;SET WC TO -1
2236 011344 013737 001102 001174 MOV @#OBUFSV,WORK ;GET OUTBUF ADD.
2237 011352 062737 000002 001174 ADD #2,WORK ;FOR TEST
2238 011360 013777 001174 167524 MOV WORK,RSBA ;LOAD ADDR.
2239 011366 012777 000067 167510 MOV #67,RSCS1 ;DO FUNCTION 67
2240 011374 105777 167504 15: TSTB @RSCS1 ;DONE YET?
2241 011400 100375 BPL 15 ;NO
2242 011402 017700 167504 MOV @RSBA,BAD ;GET BA REG
2243 011406 013701 001102 MOV @#OBUFSV,GOOD ;GET CORRECT ANS FOR RSBA
2244 011412 020100 CMP GOOD,BAD ;IS RSBA CORRECT?
2245 011414 001401 BEQ .+4 ;YES
2246 011416 104000 HLT ;BAD=RSBA GOOD=CORRECT ANS.
2247 011420 013701 001152 MOV UNNUM,GOOD ;GET UNIT NUMBER
2248 011424 052701 001300 BIS #1300,GOOD ;SET IR AND OR BITS
2249 011430 017700 167452 MOV @RSCS2,BAD ;GET CS2
2250 011434 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2251 011436 001401 BEQ .+4 ;YES
2252 011440 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
2253 011442 022777 000001 167450 CMP #1,@RSER ;IS RSER CORRECT?
2254 011450 001401 BEQ .+4 ;YES
2255 011452 104002 HLT !ER ;ER IS WRONG
2256 011454 104414 CLRDK ;CLEAR ALL RS REG
2257 011456 005777 167436 TST @RSER ;DID ERROR CLEAR
2258 011462 001401 BEQ .+4 ;YES
2259 011464 104040 HLT !DS ;NO
2260 011466 012777 177700 167414 MOV #-100,RSWC ;SET WC TO -100
2261 011474 013737 001102 001174 MOV @#OBUFSV,WORK ;GET OUTBUF ADD.
2262 011502 062737 000200 001174 ADD #200,WORK ;FOR TEST
2263 011510 013777 001174 167374 MOV WORK,RSBA ;LOAD ADDR
2264 011516 012777 000067 167360 MOV #67,RSCS1 ;DO FUNCTION 67
2265 011524 105777 167354 25: TSTB @RSCS1 ;DONE YET?
2266 011530 100375 BPL 25 ;NO

```

```

2267 011532 013701 001102      3S:  MOV      @#0BUFSV,GOOD ;GET CORRECT ANS.
2268 011536 017700 167350      MOV      @RSBA,BAD ;GET BA REG
2269 011542 020100      CMP      GOOD,BAD ;IS RSBA CORRECT?
2270 011544 001401      BEQ      .+4 ;YES
2271 011546 104000      HLT ;BAD=RSBA GOOD=CORRECT ANS.
2272 011550 013701 001152      MOV      UNNUM,GOOD ;GET UNIT NUMBER
2273 011554 052701 001300      BIS      #1300,GOOD ;SET IR AND OR BITS
2274 011560 017700 167322      MOV      @RSCS2,BAD ;GET CS2
2275 011564 020100      CMP      GOOD,BAD ;IS CS2 CORRECT?
2276 011566 001401      BEQ      .+4 ;YES
2277 011570 104000      HLT ;BAD=CS2 GOOD=CORRECT ANS
2278 011572 022777 000001 167320      CMP      #1,@RSER ;IS RSER CORRECT?
2279 011600 001401      BEQ      .+4 ;YES
2280 011602 104002      HLT ;ER IS WRONG
2281 011604 012777 040011 167272      MOV      #40011,@RSCS1 ;CLEAR ERRORS
2282 011612 022777 004210 167264      CMP      #4210,@RSCS1 ;DID THEY CLEAR IN CS1
2283 011620 001401      BEQ      .+4 ;YES
2284 011622 104040      HLT ;NO
2285 011624 005777 167270      TST      @RSER ;DID RSER CLEAR
2286 011630 001401      BEQ      .+4 ;YES
2287 011632 104040      HLT ;NO
2288 011634 005777 167262      TST      @RSAS ;DID RSAS CLEAR
2289 011640 001401      BEQ      .+4 ;YES
2290 011642 104100      HLT ;NO
2291
2292
2293 ;*****
2294 ;TEST 63 TEST PAR IN RSER
2295 ;*****
2296 TST63: SCOPE
2297 011646 013737 001164 022414 PARTST: MOV      TIMSV,TIMES ;RESTORE LOOP #
2298 011654 104414      CLDK ;CLEAR ALL RS REG
2299 011656 012777 000010 167234 1S:  MOV      #10,@RSER ;SET PAR
2300 011664 022777 150600 167224      CMP      #150600,@RSOS ;DID ERR,ATA AND DRY SET?
2301 011672 001401      BEQ      .+4 ;YES
2302 011674 104042      HLT ;ER SHOULD SET IF PAR SETS IN RSER
2303 011676 104414      CLDK ;CLEAR ALL RS REG
2304 011700 005777 167214      TST      @RSER ;DID PAR CLEAR?
2305 011704 001401      BEQ      .+4 ;YES
2306 011706 104002      HLT ;PAR DID NOT CLEAR BY CLEAR BIT
2307 011710 022777 010600 167200      CMP      #10600,@RSOS ;DID ERROR BITS CLEAR
2308 011716 001401      BEQ      .+4 ;YES
2309 011720 104040      HLT ;NO

```

H05

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST63

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 60
TEST PAR IN RSER

```
2310 ;CHECK BITS 12 TO 15 FOR 0
2311 ;CHECK SECTOR FRACTION TO WATCH FOR MOVEMENT
2312 ;CHECK CS BITS IN LA AND ADDRESS CONFIRM IN MR REG
2313
2314 ;*****
2315 ;TEST 64 LOOK AHEAD TEST
2316 ;*****
2317 011722 104400 TST64: SCOPE
2318
2319 011724 032777 170000 167172 LATST: BIT #170000,RSLA ;ARE BITS 12 TO 15 CLEARED?
2320 011732 001401 BEQ +4 ;YES
2321 011734 104204 HLT !LA ;BITS 12 TO 15 SHOULD BE CLEARED
2322 011736 104400 TST65: SCOPE
2323
2324 ;NOW TEST MOVEMENT IN SF BITS
2325
2326 011740 012737 171005 001174 MOV #171005,WORK ;SET UP FOR TIME OUT ROUTINE
2327 011746 017701 167152 MOV RSRLA,GOOD ;GET READING FROM LA
2328 011752 042701 007700 BIC #7700,GOOD ;GET RID OF CS BITS
2329 011756 005337 001174 IS: DEC WORK ;WAIT FOR DISK
2330 011762 001407 BEQ ERRR ;TYPE ERROR
2331 011764 017700 167134 MOV RSRLA,BAD ;READ LA
2332 011770 042700 007700 BIC #7700,BAD ;CLEAR CS BITS
2333 011774 020100 CMP GOOD,BAD ;DID SF BITS CHANGE?
2334 011776 001767 BEQ IS ;WAIT FOR TIME OUT
2335 012000 000422 BR LATDON ;LA OK CONT
2336 012002 ERRR:
2337 012002 104402 012006 TYPE +2 ;.ASCIZ <15><12>"SECTOR FRACTIONS NOT MOVING"
2338 012044 104204 HLT !LA ;TYPE LOOK AHEAD REG
2339 012046 LATDON: ;DONE CONT.
```

```

2340 ;*****
2341 ;TEST 66 CHECK CS BITS TO INCREMENT AND ADDRESS CONFIRM BIT IN MR
2342 ;*****
2343 012046 104400 TST66: SCOPE
2344
2345 012050 013737 022414 001164 CSTST: MOV TIMES,TIMSV ;SAVE LOOP CT
2346 012056 012737 000010 022414 MOV #10,TIMES ;LOOP 10 TIMES
2347 012064 104414 CLRDK ;CLEAR ALL RS REG.
2348 012066 012701 001000 MOV #1000,GOOD ;LOAD COUNTER
2349 012072 032777 001000 167030 BIT #BIT9,QRSMR ;IS ADD CONFIRM BIT 0?
2350 012100 001407 BEQ ADDCF ;YES CONTINUE
2351 012102 005301 DEC GOOD ;WAIT FOR
2352 012104 001376 BNE -2 ;DISK TO MOVE
2353 012106 032777 001000 167014 BIT #BIT9,QRSMR ;IS ADD. CON. BIT BIT 0?
2354 012114 001401 BEQ +4 ;YES
2355 012116 104220 HLT !MR ;ADD. CONF. BIT ALWAYS A 1
2356
2357 ;NOW TEST TA BITS AND ADD. CON. BIT IN MR
2358
2359 012120 012777 177777 166766 ADDCF: MOV #-1,QRSDA ;INIT RSDA
2360 012126 012737 177777 001174 1$: MOV #-1,WORK ;SETUP TIMEOUT COUNTER
2361 012134 005277 166754 INC QRSDA ;GET NEXT SECTOR
2362 012140 022777 010000 166746 CMP #10000,QRSDA ;DONE ALL YET?
2363 012146 001433 BEQ DONCS ;YES
2364 012150 005337 001174 2$: DEC WORK ;DO TIMEOUT ROUTINE
2365 012154 001427 BEQ TMEOUT ;ADD. CON. NEVER CAME UP
2366 012156 032777 001000 166744 BIT #1000,QRSMR ;DID ADD CONFIRM BIT SET?
2367 012164 001771 BEQ 2$ ;YES
2368 012166 017700 166732 MOV QRSLA,BAD ;GET LA
2369 012172 042700 000077 BIC #77,BAD ;CLEAR SF BITS
2370 012176 012737 000006 001176 MOV #6,WORK1 ;SET UP COUNTER
2371 012204 006000 3$: ROR BAD ;MOV SA BITS RIGHT
2372 012206 005337 001176 DEC WORK1 ;DO 6 TIMES
2373 012212 001374 BNE 3$ ;DONE YET?
2374 012214 017701 166674 MOV QRSDA,GOOD ;GET DA
2375 012220 042701 177700 BIC #17700,GOOD ;CLEAR JUNK
2376 012224 020100 CMP GOOD,BAD ;ARE SA BITS = IN DA AND LA REG.?
2377 012226 001401 BEQ +4 ;OK
2378 012230 104000 HLT ;GOOD =DA BAD = LA
2379 012232 000735 BR 1$ ;NO WAIT
2380
2381 012234 104262 TMEOUT: HLT !MR!ER!DS ;ADDRESS CONFIRM BIT NEVER SET COULD BE BAD OR
2382 ;BAD LA OR BAD COMPARE BETWEEN LA AND DA
2383 012236 DONCS: ;TEST DONE CONTINUE

```

```

*****
;TEST 67 PARITY TEST
*****
TST67: SCOPE

2384
2385
2386
2387 012236 104400
2388
2389 012240 013737 001164 022414 PART: MOV TIMSV,TIMES ;RESTORE LOOP COUNTER
2390 012246 012737 012426 000004 ;MOV #PRT,2#4 ;SETUP TIME OUT VECTOR
2391 012254 012737 000340 000006 ;MOV #340,2#6
2392 012262 012702 172100 ;MOV #MPRO,R2 ;GET PAR REG
2393 012266 005712 TSTAGN: TST (R2) ;DOES IT EXIST
2394 012270 012712 000004 ;MOV #WWP,2R2 ;YES SET WRITE WRONG PARITY
2395 012274 012701 001174 ;MOV #WORK,R1 ;GET TEST LOCATION
2396 012300 011111 1S: MOV 2R1,2R1 ;WRITE WRONG PARITY
2397 012302 005711 TST 2R1 ;READ IT
2398 012304 005712 TST 2R2 ;DID PARITY ERROR SET?
2399 012306 100402 BMI 2S ;YES
2400 012310 005012 CLR 2R2 ;CLEAR PARITY REG
2401 012312 000446 BR PRTP1 ;GET NEXT PARITY REG
2402 012314 012712 000001 2S: MOV #1,2R2 ;CLEAR WWP AND ENABLE BAD PARITY
2403 012320 104414 CLRDK ;CLEAR ALL RS REG
2404 012322 012737 177777 027364 MOV #177777,OUTBUF ;DATA TO BE X-FERED
2405 012330 012777 001174 166554 MOV #WORK,2RSBA ;SET UP CURRENT ADDRESS
2406 012336 012777 177777 166544 MOV #-1,2R5WC ;SET WORD COUNT TO -1
2407 012344 012777 000061 166532 MOV #61,2RSCS1 ;GO WRITE
2408 012352 105777 166526 3S: TSTB 2RSCS1 ;DONE YET?
2409 012356 100375 BPL 3S ;NO WAIT
2410 012360 005012 CLR 2R2 ;CLEAR PARITY REG
2411 012362 005037 001174 CLR WORK ;WRITE GOOD PARITY
2412 012366 017700 166514 MOV 2RSCS2,BAD ;GET CS2
2413 012372 012701 020100 MOV #20100,GOOD
2414 012376 053701 001152 BIS UNNUM,GOOD ;GET CORRECT AND FOR CS2
2415 012402 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2416 012404 001401 BEQ .+4 ;YES
2417 012406 104000 HLT ;CS2 SHOULD = GOOD
2418 012410 022777 144260 166466 CMP #144260,2RSCS1 ;IS CS1 CORRECT?
2419 012416 001401 BEQ .+4 ;YES
2420 012420 104040 HLT ;DS
2421 012422 000137 012446 JMP NOPAR ;GET OUT

;TRAPOUT ROUTINE
2422
2423
2424
2425 012426 022626 PRTP: CMP (6)+,(6)+ ;CLEAR STACK
2426 012430 022702 172136 PRTP1: CMP #172136,R2 ;DONE YET?
2427 012434 001404 BEQ NOPAR ;YES NO PAR REG
2428 012436 062702 000002 ADD #2,R2 ;NO TRY AGAIN
2429 012442 000137 012266 JMP TSTAGN ;RETRY
2430
2431 012446 012737 000006 000004 NOPAR: MOV #6,2#4
2432 012454 005037 000006 CLR 2#6

```

K05

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST70

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 63
TEST WRITE CHECK ERROR

```
2433 :*****
2434 :TEST 70 TEST WRITE CHECK ERROR
2435 :*****
2436 012460 104400 †TST70: SCOPE
2437
2438 ;WRITE A WORD OF 0 AND FLOAT A 1 THROUGH IT TO CAUSE WCE
2439 ;SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
2440 ;TO CAUSE WCE
2441
2442 012462 104414 WCETST: CLRDK ;CLEAR ALL RS REG
2443 012464 005037 027364 CLR OUTBUF ;WRITE A WD OF 0
2444 012470 013777 001102 166414 WCETT: MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
2445 012476 012777 177777 166404 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2446 012504 012777 000061 166372 MOV #61,@RSCS1 ;GO WRITE
2447 012512 105777 166366 3$: TSTB @RSCS1 ;DONE YET?
2448 012516 100375 BPL 3$ ;NO WAIT
2449 012520 032737 040000 001160 BIT #BIT14,ONCEE ;WRITE A 1 OR 0?
2450 012526 001410 BEQ 2$ ;WRITE A 0
2451 012530 012737 177777 027364 MOV #-1,OUTBUF ;WRITE A 1
2452 012536 000241 CLC ;CLEAR CARRY
2453 012540 006137 027364 6$: ROL OUTBUF ;FLOAT A 0 THROUGH BAD WD
2454 012544 103123 BCC WCEDON ;DONE GET OUT
2455 012546 000406 BR 5$ ;CHECK WCE
2456 012550 005037 027364 2$: CLR OUTBUF ;WRITE A 0
2457 012554 000261 SEC ;SET CARRY
2458 012556 006137 027364 1$: ROL OUTBUF ;FLOAT A 1
2459 012562 103503 BCS WCEDNE ;GET OUT WHEN DONE
2460 012564 013777 001102 166320 5$: MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
2461 012572 012777 177777 166310 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2462 012600 005077 166310 CLR @RSDA
2463 012604 012777 000051 166272 MOV #51,@RSCS1 ;GO WRITE CHECK
2464 012612 105777 166266 4$: TSTB @RSCS1 ;READY YET?
2465 012616 100375 BPL 4$ ;NO WAIT
2466 012620 017700 166262 MOV @RSCS2,BAD ;GET CS2
2467 012624 013701 001152 MOV UNNUM,GOOD ;SET UNIT #
2468 012630 052701 040300 BIS #40300,GOOD ;SET BITS
2469 012634 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2470 012636 001413 BEQ 7$ ;YES
2471 012640 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
2472 012642 013700 027364 MOV OUTBUF,BAD ;GET BAD WD THAT SHOULD CAUSE WCE
2473 012646 005001 CLR GOOD ;GET GOOD WD IF WRITING 0
2474 012650 032737 040000 001160 BIT #BIT14,ONCEE ;ARE WE WRITING 1 OR 0
2475 012656 001402 BEQ 8$ ;0
2476 012660 012701 177777 MOV #-1,GOOD ;GET GOOD WD FOR 1
2477 012664 104000 8$: HLT ;GOOD = CORRECT WD WRITTEN
2478 ;BAD = INCORRECT WD THAT WCE DID NOT CATCH
```

L05

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST70

R411-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 64
TEST WRITE CHECK ERROR

2479	012666	022777	144250	166210	75:	CMP	#144250, @RSCS1	; DID TRE SET?
2480	012674	001401				BEQ	.+4	; YES
2481	012676	104043				HLT	!CS1!ER!DS	; TRE SHOULD SET IF WCE SETS
2482	012700	017700	166206			MOV	@RSBA, BAD	; FETCH CURRENT ADDRESS
2483	012704	013701	001102			MOV	@#OBUFSV, GOOD	; WHAT RSBA SHOULD EQUAL
2484	012710	062701	000002			ADD	#2, GOOD	; UPDATE IT
2485	012714	020001				CMP	BAD, GOOD	; IS RSBA CORRECT
2486	012716	001401				BEQ	.+4	; YES EXECUTE CONTINUE
2487	012720	104000				HLT		; RSBA FAILED TO INCREMENT
2488	012722	104414				CLRDK		; CLEAR ALL RS REG
2489	012724	013701	001152			MOV	UNNUM, GOOD	; PUT DRIVE IN GOOD
2490	012730	052701	000100			BIS	#100, GOOD	; SET IR BIT
2491	012734	017700	166146			MOV	@RSCS2, BAD	; GET CS2
2492	012740	020100				CMP	GOOD, BAD	; IS CS2 CORRECT
2493	012742	001401				BEQ	.+4	; YES
2494	012744	104000				HLT		; BAD =CS2 GOOD IS CORRECT ANS
2495	012746	022777	004200	166130		CMP	#4200, @RSCS1	; DID TRE CLEAR?
2496	012754	001401				BEQ	.+4	; YES
2497	012756	104001				HLT	!CS1	; TRE DID NOT CLEAR WITH CLEAR
2498	012760	032737	040000	001160		BIT	#BIT14, ONCEE	; FLOATION A 1 OR 0?
2499	012766	001673				BEQ	1\$; FLOAT 1
2500	012770	000663				BR	6\$; FLOAT 0
2501	012772	052737	040000	001160	WCEDNE:	BIS	#BIT14, ONCEE	; SET BIT14
2502	013000	104414				CLRDK		
2503	013002	012737	177777	027364		MOV	#-1, OUTBUFF	
2504	013010	000137	012470			JMP	WCETT	; NOW WRITE -1 IN OUTBUF
2505	013014	042737	040000	001160	WCEDON:	BIC	#BIT14, ONCEE	; CLEAR TEST FLAG

M05

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST71

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 65
TEST WRITE CHECK ERROR ON -B- PORT

```

2506 ;*****
2507 ;TEST 71 TEST WRITE CHECK ERROR ON -B- PORT
2508 ;*****
2509 013022 104400 TST71: SCOPE
2510
2511 ;WRITE A WORD OF 0 AND FLOAT A 1 THROUGH IT TO CAUSE WCE
2512 ;SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
2513 ;TO CAUSE WCE
2514
2515 013024 032737 020000 001160 WCETSB: BIT #BIT13,ONCEE ; -B- PORT?
2516 013032 001402 BEQ 1$ ; YES
2517 013034 000137 013400 JMP WCEDOS ; NO GET OUT
2518 013040 013737 001170 001102 1$: MOV BPORTT, OBUFSV ; GET -B- PORT BUFFER
2519 013046 104414 CLROK ; CLEAR ALL RS REG
2520 013050 005077 166114 CLR @BPORTT ; WRITE A WD OF 0
2521 013054 013777 001102 166030 WCETB: MOV @#OBUFSV, @RSBA ; SET UP CURRENT ADDRESS
2522 013062 012777 177777 166020 MOV #-1, @RSWC ; SET WORD COUNT TO -1
2523 013070 012777 002061 166006 MOV #2061, @RSCS1 ; GO WRITE
2524 013076 105777 166002 3$: TSTB @RSCS1 ; DONE YET?
2525 013102 100375 BPL 3$ ; NO WAIT
2526 013104 032737 040000 001160 BIT #BIT14,ONCEE ; WRITE A 1 OR 0?
2527 013112 001410 BEQ 2$ ; WRITE A 0
2528 013114 012777 177777 166046 MOV #-1, @BPORTT ; WRITE A 1
2529 013122 000241 CLC ; CLEAR CARRY
2530 013124 006137 027364 6$: ROL OUTBUF ; FLOAT A 0 THROUGH BAD WD
2531 013130 103123 BCC WCEDOS ; DONE GET OUT
2532 013132 000406 BR 5$ ; CHECK WCE
2533 013134 005077 166030 2$: CLR @BPORTT ; WRITE A 0
2534 013140 000261 SEC ; SET CARRY
2535 013142 006177 166022 1$: ROL @BPORTT ; FLOAT A 1
2536 013146 103503 BCS WCEDNB ; GET OUT WHEN DONE
2537 013150 013777 001102 165734 5$: MOV @#OBUFSV, @RSBA ; SET UP CURRENT ADDRESS
2538 013156 012777 177777 165724 MOV #-1, @RSWC ; SET WORD COUNT TO -1
2539 013164 005077 165724 CLR @RSDA
2540 013170 012777 002051 165706 4$: MOV #2051, @RSCS1 ; GO WRITE CHECK
2541 013176 105777 165702 TSTB @RSCS1 ; READY YET?
2542 013202 100375 BPL 4$ ; NO WAIT
2543 013204 017700 165676 MOV @RSCS2, BAD ; GET CS2
2544 013210 013701 001152 MOV UNNUM, GOOD ; SET UNIT #
2545 013214 052701 040300 BIS #40300, GOOD ; SET BITS
2546 013220 020100 CMP GOOD, BAD ; IS CS2 CORRECT?
2547 013222 001413 BEQ 7$ ; YES
2548 013224 104000 HLT ; CS2=BAD GOOD=CORRECT ANS
2549 013226 017700 165736 MOV @BPORTT, BAD ; GET BAD WD THAT SHOULD CAUSE WCE
2550 013232 005001 CLR GOOD ; GET GOOD WD IF WRITING 0
2551 013234 032737 040000 001160 BIT #BIT14,ONCEE ; ARE WE WRITING 1 OR 0
2552 013242 001402 BEQ 8$ ; 0
2553 013244 012701 177777 MOV #-1, GOOD ; GET GOOD WD FOR 1
2554 013250 104000 8$: HLT ; GOOD = CORRECT WD WRITTEN
2555 ; BAD = INCORRECT WD THAT WCE DID NOT CATCH

```

N05

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST71

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 66
TEST WRITE CHECK ERROR ON -B- PORT

2556	013252	022777	146250	165624	7\$:	CMP	#146250, @RSCS1	;DID TRE SET?
2557	013260	001401				BEQ	.+4	;YES
2558	013262	104043				HLT	!CS1!ER!DS	;TRE SHOULD SET IF WCE SETS
2559	013264	017700	165622			MOV	@RSBA, BAD	;FETCH CURRENT ADDRESS
2560	013270	013701	001102			MOV	@#OBUFSV, GOOD	;WHAT RSBA SHOULD EQUAL
2561	013274	062701	000002			ADD	#2, GOOD	;UPDATE IT
2562	013300	020001				CMP	BAD, GOOD	;IS RSBA CORRECT
2563	013302	001401				BEQ	.+4	;YES EXECUTE CONTINUE
2564	013304	104000				HLT		;RSBA FAILED TO INCREMENT
2565	013306	104414				CLRDK		;CLEAR ALL RS REG
2566	013310	013701	001152			MOV	UNNUM, GOOD	;PUT DRIVE IN GOOD
2567	013314	052701	000100			BIS	#100, GOOD	;SET IR BIT
2568	013320	017700	165562			MOV	@RSCS2, BAD	;GET CS2
2569	013324	020100				CMP	GOOD, BAD	;IS CS2 CORRECT
2570	013326	001401				BEQ	.+4	;YES
2571	013330	104000				HLT		;BAD =CS2 GOOD IS CORRECT ANS
2572	013332	022777	004200	165544		CMP	#4200, @RSCS1	;DID TRE CLEAR?
2573	013340	001401				BEQ	.+4	;YES
2574	013342	104001				HLT	!CS1	;TRE DID NOT CLEAR WITH CLEAR
2575	013344	032737	040000	001160		BIT	#BIT14, ONCEE	;FLOATION A 1 OR 0?
2576	013352	001673				BEQ	1\$;FLOAT 1
2577	013354	000663				BR	6\$;FLOAT 0
2578	013356	052737	040000	001160	WCEDNB:	BIS	#BIT14, ONCEE	;SET BIT14
2579	013364	104414				CLRDK		
2580	013366	012777	177777	165574		MOV	#-1, @BPORTT	
2581	013374	000137	013054			JMP	WCETB	;NOW WRITE -1 IN OUTBUF
2582	013400	012737	027364	001102	WCEDOS:	MOV	#OUTBUF, OBUFSV	;RESTORE OBUFSV
2583	013406	042737	040000	001160		BIC	#BIT14, ONCEE	;CLEAR TEST FLAG

```

*****
:TEST 72                TEST PROGRAM ERROR BIT IN RSC52
*****
TST72: SCOPE

PGETST: CLRDK          ;CLEAR ALL RS REG
MOV #177777,OUTBUF    ;DATA TO BE X-FERED
MOV @#0BUFSV,@RSBA   ;SET UP CURRENT ADDRESS
MOV #177000,@RSMC    ;SET WORD COUNT
MVC #61,@RSCS1       ;GO WRITE
25: TSTB @RSCS1       ;IS RDY CLEARED YET?
BMI 25                ;NO WAIT
MOV #71,@RSCS1       ;GO READ
JSR PC,WAITRY        ;WAIT FOR READY
HLT !CS1              ;RDY NEVER CAME UP
CMP #144260,@RSCS1  ;IS CS1 CORRECT?
BEQ .+4               ;YES
HLT !CS1              ;TRE SHOULD SET BY SETTING PGE
TST @RSAS             ;AS SHOULD = 0
BEQ .+4               ;YES
HLT !AS               ;RSAS SHOULD = 0
MOV UNUM,GOOD        ;GET UNIT #
BIS #2300,GOOD       ;SET PGE, IR, AND OR
MOV @RSCS2,BAD       ;GET CS2
CMP GOOD,BAD         ;IS IT CORRECT?
BEQ .+4               ;YES
HLT                   ;BAD = CS2
TST @RSMC             ;SHOULD NOT BE 0
BNE .+4              ;BECAUSE PGE SHOULD ABORT
HLT !WC!CS1           ;CURRENT OPERATION
TST @RSER             ;DID ANY ERRORS SET?
BEQ .+4               ;NO
HLT !DS               ;RMR SHOULD BE SET
BIS #TRE,@RSCS1      ;CLEAR ERRORS
BIC #PGE,GOOD        ;CLEAR PGE ERROR
MOV @RSCS2,BAD       ;GET CS2
CMP GOOD,BAD         ;IS CS2 CORRECT?
BEQ .+4               ;YES
HLT !CS2              ;PGE DID NOT CLEAR BY CLEARING TRE BAD = CS2
CMP #4260,@RSCS1    ;DID SC CLEAR
BEQ .+4               ;YES
HLT !DS               ;DID NOT CLEAR BY CLEARING TRE

```

C06

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST73

R411-RSC3LA-RSG3-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 68
TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSDA

```

2626
2627
2628
2629 013614 104400
2630
2631 013616 104414
2632 013620 013777 001102 165264
2633 013626 012777 177700 165254
2634 013634 012703 172060
2635 013640 011304
2636 013642 042704 000017
2637 013646 022704 000020
2638 013652 001372
2639 013654 012777 000031 165222
2640 013662 012777 007777 165224
2641 013670 004737 026600
2642 013674 104001
2643 013676 022777 000004 165214
2644 013704 001401
2645 013706 104002
2646 013710 022777 007777 165176
2647 013716 001001
2648 013720 104004
2649 013722 022777 150600 165166
2650 013730 001401
2651 013732 104042
2652 013734 022777 104230 165142
2653 013742 001401
2654 013744 104040
2655 013746 104414
2656 013750 005777 165144
2657 013754 001401
2658 013756 104002
2659 013760 022777 004200 165116
2660 013766 001401
2661 013770 104040

```

```

:*****
:TEST 73 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSDA
:*****
†TST73: 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,RSRCS1 ;SEARCH
MOV #7777,RSRDA ;CAUSE ERROR
JSR PC,WAITRY ;WAIT FOR READY
HLT !CSI ;RDY NEVER CAME UP
CMP #4,RSER ;DID RMR SET?
BEQ .+4 ;YES
HLT !ER ;ER SHOULD = 4
CMP #7777,RSRDA ;DID DA GET MODIFIED?
BNE .+4 ;NO
HLT !DA
CMP #150600,RSRDS ;DID ERR SET?
BEQ .+4 ;YES
HLT !DS!ER ;ER DID NOT SET BECAUSE OF RMR
CMP #104230,RSRCS1 ;IS CSI CORRECT?
BEQ .+4 ;YES
HLT !DS ;CSI 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,RSRCS1 ;IS CSI CORRECT?
BEQ .+4 ;YES
HLT !DS ;NO

```

```

2662
2663
2664
2665
2666 013772 104400
2667
2668 013774 104414
2669 013776 013777 001102 165106
2670 014004 012777 177700 165076
2671 014012 012777 000061 165064
2672 014020 105777 165060
2673 014024 100775
2674 014026 012777 177773 165064

```

```

:*****
:TEST 74 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSER
:*****
†TST74: SCOPE

RMRT2: CLRDK ;CLEAR ALL RS REG
MOV #0BUFSV,RSBA ;SET UP CURRENT ADDRESS
MOV #177700,RSWC ;SET WORD COUNT
MOV #61,RSRCS1 ;GO WRITE
2$: TST #RSRCS1 ;IS RDY SET?
BMI 2$ ;YES WAIT FOR IT TO CLEAR
MOV #177773,RSER ;CAUSE ERROR

```

```

2675 014034 004737 026600 JSR PC WAITRY
2676 014040 104001 HLT !CSI ;RDY NEVER CAME UP
2677 014042 022777 000004 165050 CMP #4,RSER ;DID RMR SET?
2678 014050 001401 BEQ +4 ;YES
2679 014052 104002 HLT !ER ;ER SHOULD = 4
2680 014054 022777 150600 165034 CMP #150600,RSDS ;DID ERR SET?
2681 014062 001401 BEQ +4 ;YES
2682 014064 104042 HLT !DS!ER ;ERR DID NOT SET BECAUSE OF RMR
2683 014066 022777 144260 165010 CMP #144260,RSCSI ;IS CSI CORRECT?
2684 014074 001401 BEQ +4 ;YES
2685 014076 104040 HLT !DS ;CSI SHOULD = 144260
2686 014100 104414 CLRDK ;CLEAR ALL RS REG
2687
2688 ;*****
2689 ;TEST 75 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSCSI
2690 ;*****
2691 014102 104400 TST75: SCOPE
2692
2693 014104 104414 RMRT3: CLRDK ;CLEAR ALL RS REG
2694 014106 013777 001102 164776 MOV #08UFSV,RSBA ;SET UP CURRENT ADDRESS
2695 014114 012777 177700 164766 MOV #177700,RSWC ;SET WORD COUNT
2696 014122 012777 000061 164754 MOV #61,RSCSI ;GO WRITE
2697 014130 105777 164750 2$: TSTB RSCSI ;IS RDY SET?
2698 014134 100775 BMI 2$ ;YES WAIT FOR IT TO CLEAR
2699 014136 012777 000030 164740 MOV #30,RSCSI ;CAUSE ERROR
2700 014144 004737 026600 JSR PC WAITRY ;WAIT FOR READY
2701 014150 104001 HLT !CSI ;RDY NEVER CAME UP
2702 014152 022777 000004 164740 CMP #4,RSER ;DID RMR SET?
2703 014160 001401 BEQ +4 ;YES
2704 014162 104002 HLT !ER ;ER SHOULD = 4
2705 014164 022777 150600 164724 CMP #150600,RSDS ;DID ERR SET?
2706 014172 001401 BEQ +4 ;YES
2707 014174 104042 HLT !DS!ER ;ERID NOT SET BECAUSE OF RMR
2708 014176 022777 144260 164700 CMP #144260,RSCSI ;IS CSI CORRECT?
2709 014204 001401 BEQ +4 ;YES
2710 014206 104040 HLT !DS ;CSI SHOULD = 144260
2711 014210 104414 CLRDK ;CLEAR ALL RS REG

```

E06

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST76

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 70
TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS

```

2712 ;*****
2713 ;TEST 76 TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS
2714 ;*****
2715 014212 104400 †TST76: SCOPE
2716
2717 014214 104414 RMRT4: CLRDK ;CLEAR ALL RS REG
2718 014216 013777 001102 164666 MOV #0BUFSV,RSBA ;SET UP CURRENT ADDRESS
2719 014224 012777 177700 164656 MOV #177700,RSWC ;SET WORD COUNT
2720 014232 012703 172060 MOV #172060,R3 ;GET RSLA REG
2721 014236 011304 1S: MOV #R3,R4 ;WAIT FOR
2722 014240 042704 000017 BIC #17,R4 ;THE MIDDLE
2723 014244 022704 000020 CMP #20,R4 ;OF SECTOR 0
2724 014250 001372 BNE 1S ;BEFORE DOING A SEARCH
2725 014252 012777 000031 164624 MOV #31,RS1 ;SEARCH
2726 014260 012777 000000 164634 MOV #0,RSAS ;TRY TO CAUSE ERROR
2727 014266 005037 001174 CLR WORK ;CLEAR COUNTER
2728 014272 032777 000200 164616 2S: BIT #BIT7,RS1 ;WAIT FOR DRY
2729 014300 001004 BNE 3S ;READY CONT
2730 014302 005237 001174 INC WORK ;COUNT
2731 014306 001371 BNE 2S ;RETRY
2732 014310 104001 HLT !CS1 ;RDY NEVER CAME UP
2733 014312 005777 164602 3S: TST #RSER ;DID RMR SET?
2734 014316 001401 BEQ +4 ;NO
2735 014320 104002 HLT !ER ;ER SHOULD = 0
2736 014322 022777 110600 164566 CMP #110600,RS1 ;DID ERR SET?
2737 014330 001401 BEQ +4 ;NO
2738 014332 104042 HLT !DS!ER ;DS SHOULD = 110600
2739 014334 022777 104230 164542 CMP #104230,RS1 ;IS CS1 CORRECT?
2740 014342 001401 BEQ +4 ;YES
2741 014344 104040 HLT !DS ;CS1 SHOULD = 144260
2742 014346 104414 CLRDK ;CLEAR ALL RS REG
2743 014350 022777 004200 164526 CMP #4200,RS1 ;IS CS1 CORRECT?
2744 014356 001401 BEQ +4 ;YES
2745 014360 104040 HLT !DS ;NO

```

F06

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST77

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 71
TEST DCK IN RSER

```

2746 ;*****
2747 ;TEST 77 TEST DCK IN RSER
2748 ;*****
2749 014362 104400 †ST77: SCOPE
2750 ;DO A WRITE AND THEN A CLEAR FUNCTION THAT SHOULD CAUSE DCK TO SET
2751
2752
2753 014364 104414 DCKTST: CLRDK ;CLEAR ALL RS REG
2754 014366 022737 000004 001162 CMP #4,RS04DT ;IS THIS A LA DISK?
2755 014374 001004 BNE 7$ ;NO
2756 014376 012737 177640 001200 MOV #177640,WORK2 ;GET WC FOR LA DISK
2757 014404 000411 BR 1$ ;CONTINUE
2758 014406 012737 177500 001200 7$: MOV #177500,WORK2 ;GET WC FOR RS04
2759 014414 005737 001162 TST RS04DT ;IS THIS A RS04?
2760 014420 001003 BNE 1$ ;YES
2761 014422 012737 177600 001200 MOV #177600,WORK2 ;NO
2762 014430 013777 001200 164452 1$: MOV WORK2,RSWC ;LOAD WC
2763 014436 012737 177777 027364 MOV #-1,OUTBUF ;WRITE -1
2764 014444 013777 001102 164440 MOV @#0BUFSV,RSBA ;SET UP CURRENT ADDRESS
2765 014452 052777 000010 164426 4$: BIS #10,RS0CS2 ;SET BAI BIT
2766 014460 012777 000061 164416 MOV #61,RS0CS1 ;GO WRITE
2767 014466 105777 164412 5$: TSTB RS0CS1 ;IS RDY SET?
2768 014472 100375 BPL 5$ ;WAIT FOR WRITE TO FINISH
2769 014474 005077 164414 2$: CLR RS0DA ;SET DSK ADDRESS TO 0
2770 014500 005037 027364 CLR OUTBUF ;WRITE 0
2771 014504 013777 001102 164400 MOV @#0BUFSV,RSBA ;SET UP CURRENT ADDRESS
2772 014512 012777 177777 164370 MOV #-1,RSWC ;LOAD WC
2773 014520 012702 172060 MOV #172060,R2 ;PUT RSLA ADDR INTO R2
2774 014524 011203 3$: MOV (R2),R3 ;GET LA AND WAIT FOR
2775 014526 042703 000077 BIC #77,R3 ;SECTOR 40
2776 014532 022703 004000 CMP #4000,R3 ;BEFORE
2777 014536 001372 BNE 3$ ;WRITING
2778 014540 012777 000061 164336 MOV #61,RS0CS1 ;GO WRITE
2779 014546 011203 6$: MOV (R2),R3 ;GET RSLA AND WAIT FOR
2780 014550 042703 000017 BIC #17,R3 ;MIDDLE OF SECTOR
2781 014554 022703 000020 CMP #20,R3 ;0 BEFORE EXECUTING
2782 014560 001372 BNE 6$ ;A CLEAR FUNCTION
2783 014562 012777 000040 164316 MOV #40,RS0CS2 ;CLEAR ALL REG. DO IT THIS WAY
2784 014570 013777 001152 164310 MOV UNNUM,RS0CS2 ;DO NOT USE TRAP

```

G06

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST77

R411-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC
TEST DCK IN R5ER

MACY11 27(732) 25-SEP-76 09:06 PAGE 72

2785	014576	105777	164302		INCW:	TSTB	RSRCS1		: IS BUSY CLEARED
2786	014602	100401				BMI	6\$: FLAG CLEARED
2787	014604	104001				HLT	:CS1		: RDY NEVER CAME UP
2788	014606	013777	001102	164276	6\$:	MOV	#0BUFSV,RSBA		: SET UP CURRENT ADDRESS
2789	014614	013777	001200	164266		MOV	WORK2,RSWC		: LOAD WC
2790	014622	012777	000071	164254		MOV	#71,RSRCS1		: GO READ
2791	014630	105777	164250		5\$:	TSTB	RSRCS1		: IS RDY SET?
2792	014634	100375				BPL	5\$: WAIT FOR READ TO FINISH
2793	014636	022777	100000	164254		CMP	#100000,RSER		: DID DCK SET?
2794	014644	001401				BEQ	.+4		: YES
2795	014646	104002				HLT	:ER		: DCK DID NOT SET
2796	014650	022777	150600	164240		CMP	#150600,RSOS		: DID ERR SET?
2797	014656	001401				BEQ	.+4		: YES
2798	014660	104040				HLT	:DS		: ER DID NOT SET BY DCK
2799	014662	022777	144270	164214		CMP	#144270,RSRCS1		: IS CS1 CORRECT?
2800	014670	001401				BEQ	.+4		: YES
2801	014672	104044				HLT	:DS!DA		
2802	014674	017700	164206			MOV	RSRCS2,BAD		: GET CS2
2803	014700	013701	001152			MOV	UNNUM,GOOD		: GET UNIT #
2804	014704	052701	000100			BIS	#100,GOOD		: SET IR
2805	014710	020100				CMP	GOOD,BAD		: IS CS2 CORRECT?
2806	014712	001401				BEQ	.+4		: YES
2807	014714	104000				HLT			
2808	014716	012701	177700			MOV	#177700,GOOD		: NO
2809	014722	017700	164162		1\$:	MOV	RSRWC,BAD		: DID TRANSFER STOP AT END OF SECTOR?
2810	014726	020100				CMP	GOOD,BAD		
2811	014730	001401				BEQ	.+4		: YES
2812	014732	104000				HLT			: NO
2813	014734	012701	027364			MOV	#OUTBUF,GOOD		: GET BA
2814	014740	022737	000004	001162		CMP	#4,RSO4DT		: LA DISK?
2815	014746	001003				BNE	7\$: NO
2816	014750	062701	000100			ADD	#100,GOOD		: YES
2817	014754	000410				BR	3\$: CONTINUE
2818	014756	005737	001162		7\$:	TST	RSO4DT		: RSO4?
2819	014762	001003				BNE	2\$: YES
2820	014764	062701	000200			ADD	#200,GOOD		: RSO3
2821	014770	000402				BR	3\$		
2822	014772	062701	000400		2\$:	ADD	#400,GOOD		: GET CORRECT ANS FOR BA
2823	014776	017700	164110		3\$:	MOV	RSRBA,BAD		: GET BA
2824	015002	020001				CMP	BAD,GOOD		: IS BA CORRECT?
2825	015004	001401				BEQ	.+4		: YES
2826	015006	104000				HLT			: NO
2827	015010	104414				CLDRK			: CLEAR ALL RS REG
2828	015012	005777	164102			TST	RSER		: DID DCK CLEAR?
2829	015016	001401				BEQ	.+4		: YES
2830	015020	104002				HLT	:ER		: DCK DID NOT CLEAR WITH CLEAR
2831	015022	012777	177500	164060		MOV	#177500,RSWC		: CLEAR DCK ON
2832	015030	013777	001102	164054		MOV	#0BUFSV,RSBA		: DRIVE BY WRITING
2833	015036	012777	000061	164040		MOV	#61,RSRCS1		: GOOD DATA
2834	015044	105777	164034		4\$:	TSTB	RSRCS1		: ON DRIVE
2835	015050	100375				BPL	4\$		

H06

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST77

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 73
TEST DCK IN RSER

```
2836 ;TEST THE ABILITY OF THE DISK CONTROL TO
2837 ;INCREMENT THE TRACK REGISTER.
2838
2839 ;A ONE WORD WRITE WILL BE EXECUTED
2840 ;RSDA=7777 RSWC = -1
2841 ;AT THE COMPLETION OF THE WRITE RSDA = 10000
2842 ;*****
2843 ;TEST 100 TEST DISK ADDRESS REGISTER
2844 ;*****
2845 015052 104400 †ST100: SCOPE
2846
2847 015054 104414 DKADR: CLRDK ;CLEAR ALL RS REG
2848 015056 012777 177777 164024 MOV #177777,@RSWC ;SET WORD COUNT TO -1
2849 015064 013777 001102 164020 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
2850 015072 012777 007777 164014 MOV #7777,@RSDA ;SET RSDA TO ALL ONES
2851 015100 012777 000061 163776 MOV #61,@RSCSI ;GO WRITE ONE WORD
2852 015106 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
2853 015112 104001 HLT !CSI ;RDY DID NOT COME UP
2854 015114 027727 163774 010000 5$: CMP @RSDA,#10000 ;DOES RSDA=0
2855 015122 001401 BEQ .+4 ;RSDA OK
2856 015124 104004 HLT !DA ;DA DID NOT INCREMENT
2857
2858 ;*****
2859 ;TEST 101 TEST IAE ERROR
2860 ;*****
2861 015126 104400 †ST101: SCOPE
2862
2863 ;IAE ERROR SHOULD SET ERR,ATA AND SC BITS
2864
2865 015130 104414 IAERR: CLRDK ;CLEAR ALL RS REG
2866 015132 012777 177777 163750 MOV #177777,@RSWC ;SET WC TO -1
2867 015140 013777 001102 163744 MOV @#0BUFSV,@RSBA ;SET UP BUS ADDRESS
2868 015146 012777 017777 163740 MOV #17777,@RSDA ;SET DA TO RECEIVE ERROR
2869 015154 012777 000061 163722 MOV #61,@RSCSI ;GO WRITE ONE WD
2870 015162 105777 163716 7$: TSTB @RSCSI ;TEST FOR ERR OR RDY
2871 015166 100401 BMI .+4 ;OK CONT.
2872 015170 000774 BR 7$ ;WAIT
2873 015172 022777 002000 163720 CMP #2000,@RSER ;DID IAE SET?
2874 015200 001401 BEQ .+4 ;YES
2875 015202 104002 HLT !ER ;IAE SHOULD BE SET
2876 015204 022777 150600 163704 CMP #150600,@RSDS ;DID ERR SET?
2877 015212 001401 BEQ .+4 ;YES
2878 015214 104140 HLT !DS!AS ;ERR SHOULD BE SET
2879 015216 022777 144260 163660 CMP #144260,@RSCSI ;DID SC SET?
2880 015224 001401 BEQ .+4 ;YES
2881 015226 104001 HLT !CSI ;SC SHOULD BE SET
2882 015230 104414 CLRDK ;CLEAR ALL RS REG
2883 015232 005777 163662 TST @RSER ;CLR ERRORS?
2884 015236 001401 BEQ .+4 ;YES
2885 015240 104002 HLT !ER ;ERR DID NOT CLR WITH 40 IN CS2
```

```

2886 ; IN THIS ROUTINE THE PROGRAM WILL GENERATE A
2887 ; NON-EXISTENT DISK ERROR
2888
2889 ; *****
2890 ; TEST 102 TEST FOR NON-EXISTENT DISK ERROR
2891 ; *****
2892 015242 104400 TST102: SCOPE
2893
2894 015244 104414 NEDTST: CLROK ; CLEAR ALL RS REG
2895 015246 012737 000401 001174 MOV #401,WORK ; SET UP FOR N.E.D. NUMBER
2896 015254 005001 CLR GOOD ; LOOK FOR NON EXISTENT DRIVES
2897 015256 033737 001174 001154 15: BIT WORK,UNITSV ; ON THE SYSTEM
2898 015264 001405 BEQ 35 ; FOUND NON EXISTENT DRIVE
2899 015266 005201 INC GOOD ; CONTAINS UNIT #
2900 015270 006137 001174 ROL WORK ; KEEP LOOKING
2901 015274 103452 BCS NEDDON ; COULD NOT FIND ANY NON EXISTENT DRIVES
2902 015276 000767 BR 15 ; LOOK FOR NED
2903 015300 010177 163602 35: MOV GOOD,RS2 ; LOAD NED IN CS2
2904 015304 005077 163604 CLR RSDA ; WRITE DRIVE REG
2905 015310 005777 163604 TST RSER ; DID ANY BITS SET IN RSER?
2906 015314 001401 BEQ +4 ; NO
2907 015316 104040 HLT DS ; WHY DID RSER CHANGE?
2908 015320 017700 163562 MOV RS2,BAD ; GET CS2
2909 015324 052701 010100 BIS #10100,GOOD ; SET NED AND IR
2910 015330 020100 CMP GOOD,BAD ; IS CS2 CORRECT?
2911 015332 001401 BEQ +4 ; YES
2912 015334 104000 HLT ; GOOD=CORRECT CS2 BAD=CS2
2913 015336 022777 160200 163540 CMP #160200,RS1 ; IS CS1 CORRECT?
2914 015344 001401 BEQ +4 ; YES
2915 015346 104200 HLT CS2 ; TRE SHOULD SET BY NED ERROR
2916 015350 005777 163546 TST RSAS ; DID ANY BITS SET?
2917 015354 001401 BEQ +4 ; NO
2918 015356 104100 HLT AS ; WHY DID AT BITS SET?
2919 015360 112777 000100 163552 MOVB #100,RS1B ; CLEAR TRE
2920 015366 032777 010000 163512 BIT #NED,RS2 ; DID NED CLEAR
2921 015374 001401 BEQ +4 ; YES
2922 015376 104200 HLT CS2 ; NED DID NOT CLEAR
2923 015400 017737 163510 001174 MOV RSDA,WORK ; READ DRIVE REG
2924 015406 032777 010000 163472 BIT #NED,RS2 ; DID NED SET?
2925 015414 001001 BNE +4
2926 015416 104040 HLT DS ; NED DID NOT SET
2927 015420 000431 BR NNDD ; GET OUT
2928 015422 032737 010000 001160 NEDDON: BIT #BIT12,ONCEE ; WAS THIS TYPED BEFORE?
2929 015430 001025 BNE NNDD ; YES
2930 015432 104402 015436 TYPE ; .ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
2931 015504 052737 010000 001160 NNDD: BIS #BIT12,ONCEE ; SET TYPED FLAG

```

JOB

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST103

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 75
TEST THAT DAO IN RSER AND LBT IN RSDS DO SET

```

2932 ;*****
2933 ;TEST 103 TEST THAT DAO IN RSER AND LBT IN RSDS DO SET
2934 ;*****
2935 015512 104400 TST103: SCOPE
2936
2937 015514 104414 DAOTST: CLRDK ;CLEAR ALL RS REG
2938 015516 022737 000004 001162 CMP #4,RS04DT ;RS03LA?
2939 015524 001004 BNE 3$ ;NO
2940 015526 012777 177737 163354 MOV #-41,ARSWC ;LOAD WORD COUNT
2941 015534 000411 BR 1$ ;CONT
2942 015536 012777 177577 163344 3$: MOV #-201,ARSWC ;LOAD WC FOR RS04
2943 015544 005737 001162 TST RS04DT ;IS THIS A RS04?
2944 015550 001003 BNE 1$ ;YES
2945 015552 012777 177677 163330 MOV #-101,ARSWC ;NO
2946 015560 012777 007777 163326 1$: MOV #7777,ARSDA ;SET RSDA=TO ALL ONES
2947 015566 013777 001102 163316 2$: MOV @#OBUFSV,ARSB A ;CURRENT ADDRESS=OUTBUF
2948 015574 012777 000061 163302 MOV #61,ARSCS1 ;WRITE
2949 015602 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
2950 015606 104001 HLT !CS1 ;ROY DID NOT SET
2951 015610 022777 001000 163302 CMP #1000,ARSER ;DID DAO SET?
2952 015616 001401 BEQ .+4 ;YES
2953 015620 104002 HLT !ER ;DAO DID NOT SET
2954 015622 022777 152600 163266 CMP #152600,ARSDS ;DID LBT SET?
2955 015630 001401 BEQ .+4 ;YES
2956 015632 104040 HLT !DS ;LBT DID NOT SET
2957 015634 005777 163244 TST ARSCS1 ;IS ERROR FLAG SET
2958 015640 100401 BMI .+4 ;ERROR IS SET
2959 015642 104001 HLT !CS1 ;SC DID NOT SET
2960 015644 104414 CLRDK ;CLEAR ALL RS REG
2961 015646 022777 010600 163242 CMP #10600,ARSDS ;DID ATA +LBT CLEAR
2962 015654 001401 BEQ .+4 ;YES
2963 015656 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
2964 015660 005777 163234 TST ARSER ;DID DAO CLEAR?
2965 015664 001401 BEQ .+4 ;YES
2966 015666 104002 HLT !ER ;DAO DID NOT CLEAR WITH CLEAR

```

K06

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST104

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 76
TEST THAT LBT DOES SET AND DAO DOES NOT

```

2967 ;*****
2968 ;TEST 104 TEST THAT LBT DOES SET AND DAO DOES NOT
2969 ;*****
2970 015670 104400 †TST104: SCOPE
2971
2972 015672 104414 DAOTT: CLRDK ;CLEAR ALL RS REG
2973 015674 022737 000004 001162 CMP #4,RS04DT ;RS03LA?
2974 015702 001004 BNE 3$ ;NO
2975 015704 012777 177741 163176 MOV #-37,@RSWC ;LOAD WORD COUNT
2976 015712 000411 BR 1$ ;CONT
2977 015714 012777 177601 163166 3$: MOV #-177,@RSWC ;LOAD WC FOR RS04
2978 015722 005737 001162 TST RS04DT ;IS THIS A RS04?
2979 015726 001003 BNE 1$ ;YES
2980 015730 012777 177701 163152 MOV #-77,@RSWC ;NO
2981 015736 012777 007777 163150 1$: MOV #7777,@RSDA ;SET RSDA=TO ALL ONES
2982 015744 013777 001102 163140 2$: MOV @#0BUFSV,@RSBA ;CURRENT ADDRESS=OUTBUF
2983 015752 012777 000061 163124 MOV #61,@RSCS1 ;WRITE
2984 015760 004737 026600 JSR PC,WAITRY ;WAIT FOR READY
2985 015764 104001 HLT !CS1 ;RDY DID NOT SET
2986 015766 005777 163126 TST @RSER ;ANY ERRORS?
2987 015772 001401 BEQ .+4 ;NO
2988 015774 104002 HLT !ER ;YES
2989 015776 022777 012600 163112 CMP #12600,@RSDS ;DID LBT SET?
2990 016004 001401 BEQ .+4 ;YES
2991 016006 104040 HLT !DS ;LBT DID NOT SET
2992 016010 005777 163070 TST @RSCS1 ;IS ERROR FLAG SET
2993 016014 100001 BPL .+4 ;NO
2994 016016 104001 HLT !CS1 ;ERROR
2995 016020 104414 CLRDK ;CLEAR ALL RS REG
2996 016022 022777 010600 163066 CMP #10600,@RSDS ;DID LBT CLEAR
2997 016030 001401 BEQ .+4 ;YES
2998 016032 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT

```

```

2999 ;*****
3000 ;TEST 105 EXECUTE FUNCTION WITH ERROR BITS SET
3001 ;*****
3002 016034 104400 TST105: SCOPE
3003
3004 016036 104414 ERTST: CLRDK ;CLEAR ALL RS REG
3005 016040 012777 177017 163052 MOV #177017,RSER ;LOAD ER
3006 016046 017700 163050 MOV @RSAS,BAD ;GET AS REG
3007 016052 013701 001156 MOV UNCMP,GOOD ;GET UNIT ATA BIT
3008 016056 042701 177400 BIC #177400,GOOD ;CLEAR JUNK
3009 016062 020100 CMP GOOD,BAD ;IS AS REG CORRECT?
3010 016064 001401 BEQ .+4 ;YES
3011 016066 104100 HLT !AS ;AS BIT SHOULD BE SET
3012 016070 022777 104200 163006 CMP #104200,@RSCS1 ;DID ERRS SET IN CS1?
3013 016076 001401 BEQ .+4 ;YES
3014 016100 104040 HLT !DS ;CS1 SHOULD =104200
3015 016102 013777 001156 163012 MOV UNCMP,@RSAS ;CLEAR ATA BIT
3016 016110 005777 163006 TST @RSAS ;DID IT CLEAR?
3017 016114 001401 BEQ .+4 ;YES
3018 016116 104100 HLT !AS ;COULD NOT CLEAR AS BIT
3019 ;BY LOADING A 1 INTO IT
3020 016120 022777 004200 162756 CMP #4200,@RSCS1 ;DID SC CLEAR BY
3021 016126 001401 BEQ .+4 ;CLEARING ATA
3022 016130 104002 HLT !ER ;NO
3023 016132 012737 177777 027364 MOV #177777,OUTBUF ;DATA TO BE XFERED
3024 016140 013777 001102 162744 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3025 016146 012777 177777 162734 MOV #-1,@RSWC ;LOAD WC WITH -1
3026 016154 012777 000071 162722 MOV #71,@RSCS1 ;DO READ FUNCTION
3027 016162 032777 000001 162714 BIT #1,@RSCS1 ;DID GO BIT CLEAR
3028 016170 001401 BEQ .+4 ;YES
3029 016172 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
3030 016174 105777 162704 1S: TSTB @RSCS1 ;WAIT FOR READY
3031 016200 100375 BPL 1S ;WAIT
3032 016202 022777 144270 162674 CMP #144270,@RSCS1 ;DID ERRS CLEAR BY SETTING GO BIT?
3033 016210 001401 BEQ .+4 ;YES
3034 016212 104002 HLT !ER ;NO

```

M06

3035	016214	017700	162666		MOV	@RSCS2,BAD	;GET CS2
3036	016220	012701	001100		MOV	#1100,GOOD	;GET CORRECT ANS
3037	016224	053701	001152		BIS	UNNUM,GOOD	;GET UNIT #
3038	016230	020100			CMP	GOOD,BAD	;IS CS2 CORRECT?
3039	016232	001401			BEQ	.+4	;YES
3040	016234	104000			HLT		;GOOD = WHAT CS2 SHOULD =
3041	016236	022777	150600	162652	CMP	#150600,@RSDS	;DID ERR BITS SET?
3042	016244	001401			BEQ	.+4	;NO
3043	016246	104040			HLT	!DS	;ERR BIT SHOULD BE 1
3044	016250	022777	177777	162632	CMP	#-1,@RSWC	;DID WC MOVE?
3045	016256	001401			BEQ	.+4	;NO
3046	016260	104010			HLT	!WC	;WC SHOULD = 1777777
3047	016262	005777	162626		TST	@RSDA	;DID DA MOV
3048	016266	001401			BEQ	.+4	;NO
3049	016270	104004			HLT	!DA	;DA SHOULD =0
3050	016272	023777	001102	162612	CMP	@#CBUFV,@RSBA	;DID BA MOVE
3051	016300	001401			BEQ	.+4	;NO
3052	016302	104020			HLT	!BA	;BA MOVED
3053	016304	033777	001156	162610	BIT	UNCMP,@RSAS	;AS SHOULD BE SET
3054	016312	001001			BNE	.+4	;IS IT?
3055	016314	104100			HLT	!AS	;NO
3056	016316	022777	177017	162574	CMP	#177017,@RSER	;DID ER CHANGE?
3057	016324	001401			BEQ	.+4	;NO
3058	016326	104002			HLT	!ER	;ER SHOULD NOT CHANGE


```

3096
3097
3098
3099 016502 104400
3100
3101 016504 104414
3102 016506 052777 000010 162372
3103 016514 012737 177777 027364
3104 016522 013777 001102 162362
3105 016530 012777 177000 162352
3106 016536 052777 000020 162342
3107 016544 012777 000071 162332
3108 016552 105777 162326
3109 016556 100375
3110 016560 022777 144270 162316
3111 016566 001401
3112 016570 104040
3113 016572 023777 001102 162312
3114 016600 001401
3115 016602 104020
3116 016604 022777 177000 162276
3117 016612 001401
3118 016614 104010
3119
3120
3121
3122
3123 016616 104400
3124 016620 104414
3125 016622 052777 000010 162256
3126 016630 012737 177777 027364
3127 016636 013777 001102 162246
3128 016644 012777 177000 162236
3129 016652 012777 000071 162224
3130 016660 100774
3131 016662 052777 000020 162216
3132 016670 105777 162210
3133 016674 100375
3134 016676 022777 144270 162200
3135 016704 001401
3136 016706 104002
3137 016710 017700 162172
3138 016714 012701 000730
3139 016720 053701 001152
3140 016724 020100
3141 016726 001401
3142 016730 104000

```

```

*****
:TEST 107 SET PAT BIT AND LOAD FUNCTION
*****
TST107: SCOPE
SETPAT: CLRDK ;CLEAR ALL REG
BIS #BA1,ARSCS2 ;SET BAI
MOV #177777,OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV,ARSB A ;SET UP CURRENT ADDRESS
MOV #-1000,ARSWC ;LOAD WC WITH -1
BIS #BIT4,ARSCS2 ;SET PAT BIT
MOV #71,ARSCS1 ;DO READ FUNCTION
15: TSTB ARSCS1 ;WAIT FOR READY
BPL 15 ;WAIT
CMP #144270,ARSCS1 ;DID CS1 GET LOADED?
BEQ .+4 ;NO
HLT ;IT SHOULD NOT
CMP @#OBUFSV,ARSB A ;DID BA MOVE?
BEQ .+4 ;NO
HLT ;BA
CMP #-1000,ARSWC ;DID WC MOVE?
BEQ .+4 ;NO
HLT ;WC ;YES WHY?

```

```

*****
:TEST 110 DO FUNCTION THEN SET PAT BIT
*****
TST110: SCOPE
FUNDO: CLRDK ;CLEAR ALL REG
BIS #BA1,ARSCS2 ;SET BAI
MOV #177777,OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV,ARSB A ;SET UP CURRENT ADDRESS
MOV #-1000,ARSWC ;LOAD WC WITH -1
35: MOV #71,ARSCS1 ;DO A READ
BMI 35 ;WAIT FOR BUSY
BIS #BIT4,ARSCS2 ;SET PAT
25: TSTB ARSCS1 ;WAIT FOR READY
BPL 25 ;
CMP #144270,ARSCS1 ;DID MCPE SET?
BEQ .+4 ;NO
HLT ;ER ;YES
MOV ARSCS2,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 =

```



```

3143 016732 022777 010600 162156      CMP      #10600, @RS05      ;DID ERR BITS SET?
3144 016740 001401                      BEQ      .+4              ;NO
3145 016742 104040                      HLT      !DS              ;ERR BIT SHOULD BE 1
3146 016744 005777 162150      TST      @RSER           ;IS ER CLEAR?
3147 016750 001401                      BEQ      .+4              ;YES
3148 016752 104044                      HLT      !DS!DA         ;NO ERRORS SHOULD BE SET
3149 016754 104414                      CLRDK                     ;CLEAR ALL RS REG
3150 016756 022777 004200 162120      CMP      #4200, @RSCS1   ;IS CS1 CORRECT?
3151 016764 001401                      BEQ      .+4              ;YES
3152 016766 104040                      HLT      !DS
3153                                     ;*****
3154                                     ;TEST 111 TEST PAR BY SETTING PAT
3155                                     ;*****
3156 016770 104400      TST111: SCOPE
3157
3158 016772 104414      PATTST: CLRDK           ;CLEAR ALL RS REG
3159 016774 052777 000010 162104      BIS      #BA1, @RSCS2   ;SET BAI
3160 017002 012737 177777 027364      MOV      #177777, OUTBUF ;DATA TO BE XFERED
3161 017010 013777 001102 162074      MOV      @#0BUFSV, @RSBA ;SET UP CURRENT ADDRESS
3162 017016 012777 177000 162064      MOV      #-1000, @RSMC   ;LOAD WC WITH -1
3163 017024 012777 000061 162052      MOV      #61, @RSCS1    ;DO WRITE FUNCTION
3164 017032 105777 162046      1$: TSTB @RSCS1         ;WAIT FOR READY
3165 017036 100775                      BMI      1$              ;WAIT
3166 017040 052777 000020 162040      BIS      #BIT4, @RSCS2  ;SET PAT
3167 017046 105777 162032      2$: TSTB @RSCS1         ;WAIT FOR READY
3168 017052 100375                      BPL      2$
3169 017054 022777 144260 162022      CMP      #144260, @RSCS1 ;DID MCPE SET?
3170 017062 001401                      BEQ      .+4              ;NO
3171 017064 104002                      HLT      !ER             ;YES
3172 017066 017700 162014      MOV      @RSCS2, BAD     ;GET CS2
3173 017072 012701 000230      MOV      #230, GOOD      ;GET CORRECT ANS-- DO NOT CHECK IR - REASON 50 CYCLE
3174 017076 053701 001152      BIS      UNNUM, GOOD     ;GET UNIT #
3175 017102 042700 000100      BIC      #BIT6, BAD      ;CLEAR IR BIT FOR CS2 COMPARE
3176 017106 020100      CMP      GOOD, BAD       ;IS CS2 CORRECT?
3177 017110 001401                      BEQ      .+4              ;YES
3178 017112 104000                      HLT                     ;GOOD = WHAT CS2 SHOULD =
3179 017114 022777 150600 161774      CMP      #150600, @RS05  ;DID ERR BITS SET?
3180 017122 001401                      BEQ      .+4              ;NO
3181 017124 104040                      HLT      !DS              ;ERR BIT SHOULD BE 1
3182 017126 022777 000010 161764      CMP      #10, @RSER     ;DID PAR SET?
3183 017134 001401                      BEQ      .+4              ;YES
3184 017136 104044                      HLT      !DS!DA         ;NO
3185 017140 104414                      CLRDK                     ;CLEAR ALL RS REG
3186 017142 022777 004200 161734      CMP      #4200, @RSCS1   ;IS CS1 CORRECT?
3187 017150 001401                      BEQ      .+4              ;YES
3188 017152 104040                      HLT      !DS
3189 017154 005777 161740      TST      @RSER           ;DID PAR CLEAR?
3190 017160 001401                      BEQ      .+4              ;YES
3191 017162 104040                      HLT      !DS

```

```

3192 ;*****
3193 ;TEST 112 TEST THE ABILITY TO FILL THE LAST SECTOR
3194 ;*****
3195 017164 104400 TST112: SCOPE
3196
3197 C17166 104414 LASTSC: CLRDK ;CLEAR ALL RS REG
3198 017170 012777 007777 161716 MOV #7777, @RSDA ;SET RSDA=TO ALL ONES
3199 017176 022737 000004 001162 CMP #4, @RSD4T ;LA DISK?
3200 017204 001004 BNE 2S ;NO
3201 017206 012777 177740 161674 MOV #40, @RSWC ;LOAD WORD COUNT
3202 017214 000411 BR 1S ;CONTINUE
3203 017216 012777 177700 161664 2S: MOV #100, @RSWC ;WORD COUNT=-100
3204 017224 005737 001162 TST @RSD4T ;IS THIS A RSD4?
3205 017230 001403 BEQ 1S ;NO
3206 017232 012777 177600 161650 MOV #200, @RSWC ;YES
3207 017240 013777 001102 161644 1S: MOV @#0BUFSV, @RSBA ;CURRENT ADDRESS=OUTBUF
3208 017246 012777 000061 161630 MOV #61, @RSCS1 ;WRITE
3209 017254 004737 026600 JSR PC, WAITRY ;WAIT FOR READY
3210 017260 104001 HLT !CS1 ;RDY DID NOT SET
3211 017262 005777 161632 TST @RSE4 ;DID ANY ERROR BITS SET?
3212 017266 001401 BEQ +4 ;NO
3213 017270 104002 HLT !ER ;GOT AN ERROR
3214 017272 022777 012600 161616 CMP #12600, @RSDS ;DID LBT SET?
3215 017300 001401 BEQ +4 ;YES
3216 017302 104040 HLT !DS ;LBT DID NOT SET
3217 017304 005777 161574 TST @RSCS1 ;IS ERROR FLAG SET
3218 017310 100001 BPL +4 ;ERROR IS SET
3219 017312 104001 HLT !CS1 ;SC DID NOT SET
3220 017314 104414 CLRDK ;CLEAR ALL RS REG
3221 017316 022777 010600 161572 CMP #10600, @RSDS ;DID ATA +LBT CLEAR
3222 017324 001401 BEQ +4 ;YES
3223 017326 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT

```

E07

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST112

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 83
TEST THE ABILITY TO FILL THE LAST SECTOR

```

3224
3225
3226
3227
3228
3229
3230
3231 017330 104400
3232 017332 104414
3233 017334 012737 177777 027364
3234 017342 013777 001102 161542
3235 017350 022737 000004 001162
3236 017356 001004
3237 017360 012777 177740 161522
3238 017366 000411
3239 017370 012777 177600 161512 4$:
3240 017376 005737 001162
3241 017402 001003
3242 017404 012777 177700 161476
3243 017412 052777 000010 161466 5$:
3244 017420 012777 000061 161456
3245 017426 105777 161452 3$:
3246 017432 100375
3247 017434 005077 161454
3248 017440 012737 177777 027364
3249 017446 013777 001102 161436
3250 017454 012777 177777 161426
3251 017462 052777 000010 161416
3252 017470 012777 000061 161406
3253 017476 105777 161402 1$:
3254 017502 100375
3255 017504 042777 000010 161374
3256 017512 005737 001162
3257 017516 001404
3258 017520 012737 000200 001174
3259 017526 000403
3260 017530 012737 000100 001174 7$:
3261 017536 013701 001102 8$:
3262 017542 012721 177777
3263 017546 005021 6$:
3264 017550 005337 001174
3265 017554 001374
3266 017556 005077 161332
3267 017562 013777 001102 161322
3268 017570 022737 000004 001162
3269 017576 001004
3270 017600 012777 177740 161302
3271 017606 000412
3272 017610 005737 001162 11$:
3273 017614 001404
3274 017616 012777 177600 161264
3275 017624 000403
3276 017626 012777 177700 161254 9$:
3277 017634 012777 000051 161242 10$:

```

```

;FILL SECTOR WITH ALL ONES.
;NOW WRITE 1ST WORD IN SECTOR
;TEST REMAINING 63 WORDS FOR 0
;*****
;TEST 113 TEST FOR ZERO'S IN PARTIAL FILLED SECTOR
;*****
TST113: SCOPE
SECT: CLRDK ;CLEAR ALL RS REG
MOV # -1,OUTBUF ;PUT -1 INTO OUTBUF
MOV @#0BUFSV,RSBA ;SET UP CURRENT ADDR
CMP #4,RS04DT ;RS03LA DISK?
BNE 4$ ;NO
MOV #-40,RSWC ;LOAD WORD COUNT
BR 5$ ;CONTINUE
4$: MOV #-200,RSWC ;LOAD WC FOR RS04
TST RS04DT ;RS04?
BNE 5$ ;YES
MOV #-100,RSWC ;SET WORD COUNT TO -100
BIS #BAI,RS0CS2 ;SET BAI BIT
MOV #61,RS0CS1 ;WRITE
3$: TSTB RS0CS1 ;IS RDY SET?
BPL 3$ ;NO
CLR @RSDA ;SET DSK ADDRESS TO 0
MOV #-1,OUTBUF ;PUT 177777 INTO OUTBUF
MOV @#0BUFSV,RSBA ;SET UP CURRENT ADDR
MOV #-1,RSWC ;SET WORD COUNT TO -1
BIS #10,RS0CS2 ;SET BAI BIT
MOV #61,RS0CS1 ;WRITE
1$: TSTB RS0CS1 ;IS RDY SET?
BPL 1$ ;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 @#0BUFSV,R1 ;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 @RSDA ;SET DSK ADDRESS TO 0
MOV @#0BUFSV,RSBA ;SET UP CURRENT ADDR
CMP #4,RS04DT ;RS03LA DISK?
BNE 11$ ;NO
MOV #-40,RSWC ;LOAD WORD COUNT
BR 10$ ;CONTINUE
11$: TST RS04DT ;RS04?
BEQ 9$ ;NO
MOV #-200,RSWC ;YES
BR 10$ ;CONT
9$: MOV #-100,RSWC ;SET WORD COUNT TO -100
10$: MOV #51,RS0CS1 ;WRITE CHECK

```

F07

```

3278 017642 032777 000200 161234 2S: BIT #200, @RSCS1 ; IS RDY SET?
3279 017650 001774 BEQ 2$ ; NO
3280 017652 013701 001152 MOV UNNUM, GOOD ; GET UNIT #
3281 017656 052701 000100 BIS #100, GOOD ; SET IR BIT
3282 017662 017700 161220 MOV @RSCS2, BAD ; GET CS2
3283 017666 020100 CMP GOOD, BAD ; IS CS2 CORRECT?
3284 017670 001401 BEQ +4 ; YES
3285 017672 104002 HLT !ER ; THERE WAS A WRITE CHECK ERROR

```

```

3286
3287 ;*****
3288 ;TEST 114 IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDR
3289 ;*****

```

```

3290 017674 104400 TST114: SCOPE
3291

```

```

3292 017676 104414 EXTTST: CLDK ; CLEAR ALL RS REG.
3293 017700 013737 022414 001164 MOV TIMES, TIMSV ; SAVE LOOP #
3294 017706 012737 000010 022414 MOV #10, TIMES ; LOOP 10 TIMES
3295 017714 012737 020420 000004 MOV #EXTRP, 4 ; SETUP TIMEOUT TRAP
3296 017722 012737 000340 000006 MOV #340, 6
3297 017730 005737 177572 TST @SR0 ; IF MEMORY MANAGEMENT IS NOT
3298 ; AVAILABLE THE PROGRAM WILL TRAP
3299 ; AND TRANSFER TO END OF THE TEST
3300 017734 012737 020412 000004 MOV #EXTRP, 4
3301 017742 012737 007600 172356 MOV #7600, @KIPAR7 ; OPEN I/O REGISTERS
3302 017750 005037 172340 CLR @KIPAR0 ; FREE FIRST 4K
3303 017754 012737 000200 172342 MOV #200, @KIPAR1 ; ENABLE SECOND 4K
3304 017762 012737 002000 172344 MOV #2000, @KIPAR2
3305 017770 012737 177406 172300 MOV #400*256.-400+UP+RW, @KIPDR0 ; SET KIPDR0=RW UP 400 BLOCKS
3306 017776 012737 177406 172302 MOV #400*256.-400+UP+RW, @KIPDR1 ; SET KIPDR1=RW UP 400 BLOCKS
3307 020004 012737 177406 172304 MOV #400*256.-400+UP+RW, @KIPDR2 ; SET KIPDR2=RW UP 400 BLOCKS
3308 020012 012737 177406 172316 MOV #400*256.-400+UP+RW, @KIPDR7 ; SET KIPDR7=RW UP 400 BLOCKS
3309 020020 012737 000001 177572 MOV #1, @SR0 ; TURN ON MEMORY MANAGEMENT
3310 020026 012702 040000 MOV #40000, R2 ; R2 EQUALS BASE ADDR

```

G07

3311	020032	012712	177777		7\$:	MOV	#177777,(R2)	; INSERT PATTERN INTO 200000
3312	020036	012777	177776	161044		MOV	#-2,ARSWC	; SETUP WORDCOUNT
3313	020044	012777	177777	161040		MOV	#177777,ARSB A	; SETUP BUS ADDR
3314	020052	012777	000061	161024		MOV	#61,ARSCS1	; WRITE TWO WORDS ON DISK. RSBA
3315								; STARTS AT 177777 TO FORCE CARRY
3316								; TO SET A16
3317	020060	105777	161020			TSTB	ARSCS1	; WAIT FOR READY
3318	020064	100375				BPL	.-4	
3319	020066	005777	161012			TST	ARSCS1	
3320	020072	100002				BPL	1\$	
3321	020074	104046				HLT	!ER!DA!DS	; STATUS ERROR AFTER 2 WORD WRITE
3322	020076	000447				BR	2\$; USING MEXD
3323	020100	022777	004660	160776	1\$:	CMP	#4660,ARSCS1	; IS CS1 CORRECT
3324	020106	001402				BEQ	3\$; YES
3325	020110	104002				HLT	!ER	; CS2 DID NOT COMPARE
3326	020112	000441				BR	2\$	
3327	020114	005012			3\$:	CLR	(R2)	; CLEAR LOCATION 200000
3328	020116	005077	160772			CLR	ARSDA	; SETUP DA
3329	020122	012777	177777	160762		MOV	#177777,ARSB A	; SETUP BA
3330	020130	012777	177776	160752		MOV	#-2,ARSWC	; SETUP WC
3331	020136	012777	000071	160740		MOV	#71,ARSCS1	; READ TWO WORDS INTO LOCATIONS
3332								; 177777 AND 200000.
3333	020144	105777	160734			TSTB	ARSCS1	; WAIT FOR READY
3334	020150	100375				BPL	.-4	
3335	020152	005777	160726			TST	ARSCS1	; ANY ERRORS?
3336	020156	100002				BPL	4\$; BRANCH IF NO
3337	020160	104002				HLT	!ER	; ERROR OFTER READING 2 WORDS
3338	020162	000415				BR	2\$	
3339	020164	022777	004670	160712	4\$:	CMP	#4670,ARSCS1	; IS CS1 CORRECT?
3340	020172	001402				BEQ	5\$; YES
3341	020174	104002				HLT	!ER	; CS1 DID NOT COMPARE
3342	020176	000407				BR	2\$; READ STARTING AT 177777
3343	020200	022712	177777		5\$:	CMP	#177777,(R2)	; WAS DATA READ INTO LOCATION
3344	020204	001404				BEQ	2\$; 200000 CORRECTLY? - BRANCH IF YES
3345	020206	012701	177777			MOV	#177777,GOOD	
3346	020212	011200				MOV	(R2),BAD	
3347	020214	104000				HLT		; DATA COMPARE ERROR AT 200000
3348	020216	000240			2\$:	NOP		

H07

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST114

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 86
IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS BITS

```

3349 020220 104414          EXTT1:  CLRDK          ;CLEAR ALL REG
3350 020222 012737 004000 172344    MOV     #4000, @#KIPAR2
3351 020230 012702 040000          MOV     #40000, R2       ;R2 EQUALS THE BASE ADDR
3352 020234 012712 177777          7$:   MOV     #177777, (R2)   ;INSERT PATTERN INTO 400000
3353 020240 012777 177777 160644    MOV     #177777, @RSBA   ;SETUP BUS ADDR
3354 020246 012777 177776 160634    MOV     #-2, @RSWC      ;LOAD WC
3355 020254 012777 000461 160622    MOV     #461, @RSCS1    ;SET BIT A16 AND WRITE
3356 020262 105777 160616          TSTB   @RSCS1          ;WAIT FOR READY
3357 020266 100375          BPL     -4
3358 020270 005777 160610          TST     @RSCS1          ;ANY ERRORS?
3359 020274 100001          BPL     4$             ;BRANCH IF NO
3360 020276 104002          HLT     !ER           ;ERROR AFTER READING 2 WORDS
3361 020300 022777 005260 160576  4$:   CMP     #5260, @RSCS1   ;IS CS1 CORRECT?
3362 020306 001401          BEQ     10$           ;BRANCH IF YES
3363 020310 104002          HLT     !ER           ;NO CS1 DID NOT COMPARE
3364                                   ;READ STARTING AT 377777
3365 020312 005012          10$:  CLR     (R2)        ;CLEAR LOCATION 400000
3366                                   ;READ TWO WORDS STARTING AT 377777
3367 020314 012777 177776 160566    MOV     #-2, @RSWC      ;SETUP WC
3368 020322 005077 160566          CLR     @RSDA          ;SETUP DA
3369 020326 012777 177777 160556    MOV     #177777, @RSBA
3370 020334 012777 009471 160542    MOV     #471, @RSCS1   ;CLEAR A 17 SET A16, READ
3371 020342 105777 160536          TSTB   @RSCS1          ;WAIT FOR READY
3372 020346 100375          BPL     -4
3373 020350 005777 160530          TST     @RSCS1          ;ANY ERRORS?
3374 020354 100002          BPL     11$           ;BRANCH IF NO
3375 020356 104002          HLT     !ER           ;ERROR WHILE READING TWO WORDS
3376 020360 000414          BR     EXTRP
3377 020362 022777 005270 160514  11$:  CMP     #5270, @RSCS1   ;IS CS1 CORRECT?
3378 020370 001401          BEQ     12$           ;BRANCH IF YES
3379 020372 104002          HLT     !ER           ;CS1 DID NOT COMPARE
3380                                   ;READ STARTING AT 377777
3381 020374 022712 177777          12$:  CMP     #177777, (R2)  ;WAS DATA READ INTO LOCATION 400000
3382 020400 001404          BEQ     EXTRP         ;CORRECTLY? - BRANCH IF YES
3383 020402 012701 177777          MOV     #177777, GOOD
3384 020406 011200          MOV     (R2), BAD
3385 020410 104000          HLT
3386                                   ;DATA COMPARE ERROR AT 400000 IF
3387 020412 005037 177572          EXTRP: CLR   @SRO      ;RECEIVED=0 - LOCATION WASN'T ACCESSED
3388 020416 000401          BR     EXT1           ;TURN OFF MEMORY MANAGEMENT
3389 020420 000240          EXTTRP: NOP
3390 020422 012706 000500          EXT1:  MOV     #500, SP ;UPDATE TEST NUMBERS
3391 020426 012737 000006 000004  MEMOUT: MOV    #6, 4      ;RESTORE STACK
3392 020434 005037 000006          CLR     6

```

```

3393 ;*****
3394 ;TEST 115 TEST PROGRAM INTERRUPT BY MOVING 300 INTO RSCS1
3395 ;*****
3396 020440 104400 †TST115: SCOPE
3397
3398 020442 104414 QES: CLRDK ;CLEAR ALL DRIVES
3399 020444 012706 000500 MOV #500,SP ;SETUP STACK
3400 020450 012777 020522 160456 MOV #PGTRAP,‡RSVEC ;SET UP VECTOR
3401 020456 012777 000340 160452 MOV #340,‡RSVCPS ;SET TRAP PS
3402 020464 012737 000200 177776 MOV #200,‡PS ;SET PS AT PRIORITY 4
3403 020472 012777 000300 160404 MOV #300,‡RSCS1 ;THIS SHOULD CAUSE A TRAP
3404 020500 012737 000500 001174 MOV #500,WORK ;SETUP LOOP
3405 020506 005337 001174 1$: DEC WORK ;DEC LOOP SHOULD
3406 020512 001375 BNE 1$ ;INTERRUPE BEFORE LOOP IS DONE
3407 020514 104001 HLT !CS1 ;SHOULD NEVER GET HERE
3408 020516 000137 020536 JMP QESDON ;GET OUT
3409
3410 020522 022626 PGTRAP: CMP (6)+,(6)+ ;TRAP OK
3411 020524 022777 004200 160352 CMP #4200,‡RSCS1 ;DID IE CLEAR?
3412 020532 001401 BEQ .+4 ;YES
3413 020534 104001 HLT !CS1 ;IE SHOULD BE CLEARED
3414 020536
3415 QESDON:
3416 ;*****
3417 ;TEST 116 TEST THAT DISK DOES NOT INTERRUPT WHEN PS IS AT 5
3418 ;*****
3419 020536 104400 †TST116: SCOPE
3420
3421 020540 012706 000500 INTR5: MOV #500,SP ;SETUP STACK
3422 020544 013737 001164 022414 MOV TIMSV,TIMES ;RESTORE LOOP COUNTER
3423 020552 104414 CLRDK ;CLEAR ALL RS REG
3424 020554 012777 020634 160352 MOV #INT112,‡RSVEC ;SET UP INTERRUPT VECTOR
3425 020562 012777 000340 160346 MOV #340,‡RSVCPS ;SET PRIO.
3426 020570 012737 000240 177776 MOV #240,‡PS ;LOCK OUT ALL INTERRUPTS ABOVE
3427 020576 013700 177776 MOV ‡PS,BAD ;GET PS
3428 020602 012777 177777 160300 MOV #177777,‡RSWC ;SET WORD COUNT TO -1
3429 020610 013777 001102 160274 MOV ‡#0BUFSV,‡RSBA ;LOAD CURRENT ADDRESS
3430 020616 012777 000161 160260 MOV #161,‡RSCS1 ;GO WRITE (INTERRUPT ENABLED)
3431 020624 004737 026600 JSR PC,‡WAITRY ;WAIT FOR READY
3432 020630 104001 HLT !CS1 ;NO RDY NEVER CAME UP
3433 020632 000403 BR INTDON ;RESTART ROUTINE
3434 ;PROCESSOR SHOULD NOT TRAP TO INT112
3435
3436 020634 012701 000240 INT112: MOV #240,GOOD ;WHAT PS SHOULD HAVE
3437 020640 104000 HLT ;GOOD = CORRECT ANS FOR PS
3438 020642 INTDON: ;DONE GET OUT

```

```

3439 ;*****
3440 ;TEST 117 TEST THAT DISK DOES INTERRUPT WHEN PS IS AT 4
3441 ;*****
3442 020642 104400 TST117: SCOPE
3443
3444 020644 012706 000500 INTR4: MOV #500, SP ;SETUP STACK
3445 020650 104414 CLRDK ;CLEAR ALL RS REG
3446 020652 012777 020746 160254 MOV #INT114, @RSVEC ;SET UP DISK TRAP VECTOR
3447 020660 012777 000340 160250 MOV #340, @RSVCPS ;SET PRIO.
3448 020666 012737 000200 177776 MOV #200, @#PS ;SET PROCESSOR TO PRIORITY 4
3449 020674 013700 177776 MOV @#PS, BAD ;GET PS
3450 020700 012701 000200 MOV #200, GOOD ;GET CORRECT PS
3451 020704 012777 177777 160176 MOV #177777, @RSWC ;SET WORD COUNT TO -1
3452 020712 013777 001102 160172 MOV @#OBUFSV, @RSBA ;LOAD CURRENT ADDRESS
3453 020720 012777 000161 160156 MOV #161, @RSCSI ;WRITE (INTERRUPT ENABLE
3454 020726 005037 001174 CLR WORK
3455 020732 005237 001174 INC WORK ;WAIT FOR INTERRUPT TO OCCUR
3456 020736 001375 BNE .-4
3457 020740 104000 HLT ;GOOD=CORRECT PS BAD=WRONG PS
3458 020742 104042 HLT !ER!DS
3459 020744 000405 BR DONINT ;CONT
3460 020746 022777 004260 160130 INT114: CMP #4260, @RSCSI ;DID IE CLEAR?
3461 020754 001401 BEQ .+4 ;YES
3462 020756 104001 HLT !CSI ;WHY DID NOT IE CLEAR
3463 020760 DONINT:
3464
3465 ;*****
3466 ;TEST 120 TEST INTERRUPT ON ERROR
3467 ;*****
3468 020760 104400 TST120: SCOPE
3469
3470 020762 012706 000500 ERINT: MOV #500, SP ;SETUP STACK
3471 020766 012737 000200 177776 MOV #200, @#PS ;SET PS AT PRI 4
3472 020774 012777 021052 160132 MOV #ERRINT, @RSVEC ;SET UP INTERRUPT ADD.
3473 021002 104414 CLRDK ;CLEAR ALL RS REG
3474 021004 012777 000340 160124 MOV #340, @RSVCPS ;SET PRIO.
3475 021012 012777 177777 160074 MOV #177777, @RSDA ;SET PSDA=TO ALL ONES
3476 021020 012777 177600 160062 MOV #177600, @RSWC ;WORD COUNT=-200
3477 021026 013777 001102 160056 MOV @#OBUFSV, @RSBA ;CURRENT ADDRESS=OUTBUF
3478 021034 012777 000161 160042 MOV #161, @RSCSI ;WRITE
3479 021042 004737 026600 JSR PC, WAITRY ;WAIT FOR READY
3480 021046 104042 IS: HLT !ER!DS ;Y DIDN'T PGM INTERRUPT IS RDY SET?
3481 021050 000406 BR FINTST ;GET OUT
3482 021052 022777 144260 160024 ERRINT: CMP #144260, @RSCSI ;IS CSI RIGHT?
3483 021060 001401 BEQ .+4 ;YES
3484 021062 104042 HLT !ER!DS
3485 021064 022626 CMP (6)+, (6)+ ;CLEAR STACK
3486 021066 FINTST:

```


K07

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST121

RH11-RS03LA-RSG3-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 03:06 PAGE 89
DYNAMIC FUNCTION TEST

```
3487 ;*****
3488 ;TEST 121 DYNAMIC FUNCTION TEST
3489 ;*****
3490 021066 104400 †TST121: SCOPE
3491 ;EXECUTE FUNCTION MODIFY UNIT # AND DO A DRIVE SEARCH
3492 ;DRIVE SEARCH WILL ONLY BE DONE IF THERE ARE AT LEAST 2 DRIVES
3493 ;2ND DRIVE MAY NOT BE TESTED YET SO IF THIS TEST FAILS CHECK 2ND DRIVE
3494 ;BEFORE TRYING TO DEBUG THIS TEST
3495
3496 021070 104414 MODNUM: CLRDK ;CLEAR ALL RS REG
3497 021072 013737 022414 001164 MOV TIMES,TIMSV ;SAVE LOOP COUNT
3498 021100 012737 000010 022414 MOV #10,TIMES ;LOOP ONLY 10 TIMES
3499 021106 005037 001176 CLR WORK1 ;CLEAR WORK LOC.
3500 021112 005003 CLR R3
3501 021114 005004 CLR R4
3502 021116 012702 022044 MOV #DVTAB,R2 ;SETUP TABLE
3503 021122 012737 000401 001174 MOV #401,WORK ;SETUP TO TEST FOR MORE DRIVES
3504 021130 033737 001174 001154 7$: BIT WORK,UNITSV ;IS DRIVE ON SYSTEM?
3505 021136 001403 BEQ 6$ ;NO
3506 021140 020437 001152 CMP R4,UNNUM ;IS THIS THE SAME DRIVE?
3507 021144 001017 BNE 8$ ;NO
3508 021146 005204 6$: INC R4 ;UPDATE 2ND UNIT #
3509 021150 000241 CLC
3510 021152 006137 001174 ROL WORK ;CHECK FOR NEXT DRIVE
3511 021156 103364 BCC 7$ ;NOT DONE YET
3512 021160 032737 000010 001176 BIT #BIT3,WORK1 ;MULTI DRIVE?
3513 021166 001016 BNE 12$ ;YES
3514 021170 013705 001152 MOV UNNUM,R5 ;LOAD UNIT NO
3515 021174 005205 INC R5 ;CHANGE IT
3516 021176 042705 177770 BIC #177770,R5 ;CLEAR JUNK
3517 021202 000410 BR 12$
3518 021204 052737 000010 001176 8$: BIS #BIT3,WORK1 ;SET FOUND MULTI DRIVE
3519 021212 010422 MOV R4,(R2)+ ;LOAD UNT # INTO TABLE
3520 021214 005203 INC R3 ;COUNT # OF DRIVES
3521 021216 010337 001172 MOV R3,SAVEE ;SAVE IT
3522 021222 000751 BR 6$
3523 021224 012777 021340 157702 12$: MOV #TSTVEC,RSVEC ;SETUP INT. TRAP
3524 021232 012777 000340 157676 MOV #340,RSVCPS ;SETUP PRIO.
3525 021240 005037 027364 CLR OUTBUF ;CLR TO READ INTO
3526 021244 013777 001102 157640 MOV #OBUFSV,RSBA ;SET UP CURRENT ADDRESS
3527 021252 012777 177000 157630 MOV #-1000,RSWC ;SET WORD COUNT
3528 021260 012777 000060 157626 1$: MOV #60,RSDA ;LOAD DA
3529 021266 012702 022044 MOV #DVTAB,R2 ;GET TABLE
3530 021272 012777 000161 157604 MOV #161,RSCSI ;GO WRITE
3531 021300 005703 TST R3 ;MORE THEN 1 DRIVE?
3532 021302 001003 BNE 13$ ;YES
3533 021304 010577 157576 MOV R5,RSWSC2 ;NO MODIFY UNIT #
3534 021310 000407 BR 14$
```

3535	021312	012277	157570		13\$:	MOV	(R2)+, @RSCS2	:LOAD UNIT#
3536	021316	012777	000131	157560		MOV	#131, @RSCS1	:DO SEARCH
3537	021324	005303				DEC	R3	:DONE ALL DRIVES YET?
3538	021326	001371				BNE	13\$:NO
3539	021330	012737	000200	177776	14\$:	MOV	#200, @#PS	:ENABLE INTERRUPTS
3540	021336	000001			WTDV:	WAIT		
3541	021340	013777	001152	157540	TSTVEC:	MOV	UNNUM, @RSCS2	:GET 1ST DRIVE
3542	021346	017700	157532			MOV	@RSCS1, BAD	:GET CS1
3543	021352	042700	100000			BIC	#BIT15, BAD	:CLEAR SC
3544	021356	012701	004260			MOV	#4260, GOOD	:GET CORRECT ANS
3545	021362	020100				CMP	GOOD, BAD	:IS CS1 CORRECT?
3546	021364	001402				BEQ	4\$:NO! X-FER OK
3547	021366	104000				HLT		:CS1 SHOULD = 14270 OR 4270
3548	021370	104140				HLT	!DS!AS	
3549	021372	005777	157512		4\$:	TST	@RSCS2	:TEST WC
3550	021376	001401				BEQ	.+4	:WORD COUNT DID OVERFLOW
3551	021400	104010				HLT	!WC	:SHOULD = 0
3552	021402	013701	001152			MOV	UNNUM, GOOD	:GET CORRECT
3553	021406	052701	000100			BIS	#100, GOOD	:ANS OF CS2
3554	021412	017700	157470			MOV	@RSCS2, BAD	:GET CS2
3555	021416	020100				CMP	GOOD, BAD	:IS CS2 CORRECT?
3556	021420	001401				BEQ	.+4	:YES
3557	021422	104000				HLT		:GOOD = CORRECT ANS FOR CS2
3558	021424	017700	157462			MOV	@RSBA, BAD	:FETCH CURRENT ADDRESS
3559	021430	013701	001102			MOV	@#OBUFSV, GOOD	:WHAT RSBA SHOULD EQUAL
3560	021434	062701	002000			ADD	#2000, GOOD	:UPDATE IT
3561	021440	020001				CMP	BAD, GOOD	:IS RSBA CORRECT
3562	021442	001401				BEQ	.+4	:YES EXECUTE CONTINUE
3563	021444	104000				HLT		:RSBA FAILED TO INCREMENT
3564	021446	022737	000004	001162		CMP	#4, RS04DT	:RS03LA?
3565	021454	001006				BNE	2\$:NO
3566	021456	022777	000100	157430		CMP	#100, @RSDA	:IS DA CORRECT?
3567	021464	001420				BEQ	3\$:OK
3568	021466	104104				HLT	!DA!AS	
3569	021470	000416				BR	3\$:CONTINUE
3570	021472	005737	001162		2\$:	TST	RS04DT	:IS THIS A RS04?
3571	021476	001006				BNE	5\$:YES
3572	021500	022777	000070	157406		CMP	#70, @RSDA	:IS DA CORRECT?
3573	021506	001407				BEQ	3\$:YES
3574	021510	104104				HLT	!DA!AS	:DA NOT CORRECT
3575	021512	000405				BR	3\$:CONTINUE
3576	021514	022777	000064	157372	5\$:	CMP	#64, @RSDA	:WAS RSDA INCREMENTED
3577	021522	001401				BEQ	.+4	:RSDA OK
3578	021524	104046				HLT	!DA!ER!DS	:RSDA SHOULD CONTAIN A 64
3579	021526	012777	040000	157350	3\$:	MOV	#TRE, @RSCS1	:CLEAR ALL ERRORS IF ANY
3580	021534	032737	000010	001176		BIT	#BIT3, WORK1	:MULTI DRIVE?
3581	021542	001461				BEQ	WTDV1	:NO
3582	021544	012777	021570	157362	1\$:	MOV	#TTVEC, @RSVEC	:SETUP INT FOR NEXT DRIVE
3583	021552	012716	021566			MOV	#WTDV2, (SP)	:GET WAIT
3584	021556	012777	000100	157320		MOV	#100, @RSCS1	:SET IE
3585	021564	000002				RTI		:RETURN
3586	021566	000001			WTDV2:	WAIT		

```

3587          ;SERVICE ROUTINE FOR SEARCH FUNCTIONS
3588 021570 005002          TTVEC: CLR R2          ;CLEAR UNIT #
3589 021572 000241          CLC
3590 021574 012737 000401 001174  MOV #401,WORK
3591 021602 033777 001174 157312 1S: BIT WORK,@RSAS          ;DID THIS DRIVE INT?
3592 021610 001006          BNE 2S          ;YES
3593 021612 005202          INC R2          ;UPDATE UNIT #
3594 021614 000241          CLC
3595 021616 006137 001174  ROL WORK
3596 021622 103367          BCC 1S
3597 021624 104100          HLT !AS          ;WHY DID WE INT WITH NO ATA??
3598 021626 010277 157254 2S: MOV R2,@RSCS2          ;GET DRIVE
3599 021632 022777 110600 157256  CMP #110600,@RSDS          ;DID PIP CLEAR?
3600 021640 001401          BEQ +4          ;YES
3601 021642 104140          HLT !DS!AS          ;PIP BIT DID NOT CLEAR
3602 021644 022777 104230 157232  CMP #104230,@RSCS1          ;DID SC SET?
3603 021652 001401          BEQ +4          ;YES
3604 021654 104140          HLT !AS!DS          ;SC DID NOT SET
3605 021656 005337 001172  DEC SAVEE          ;COUNT # OF INT
3606 021662 001411          BEQ WTDV1          ;DONE YET?
3607 021664 013777 001174 157230  MOV WORK,@RSAS          ;CLEAR AS
3608 021672 012777 000100 157204  MOV #100,@RSCS1          ;SET IE
3609 021700 012716 021566  MOV #WTDV2,(SP)          ;RETURN TO WAIT
3610 021704 000002          RTI
3611 021706 012737 000340 177776  WTDV1: MOV #340,@#PS
3612 021714 012706 000500  MOV #500,SP          ;CLEAR STACK
3613 021720 013777 001136 157206  MOV RSVCPS,@RSVEC          ;RESTORE INT VECTOR
3614 021726 005077 157204  CLR @RSVCPS
3615 021732 104400          MODDON: SCOPE          ;DONE
3616 021734 013737 001164 022414  MOV TIMSV,TIMES          ;RESTORE LOOP COUNT
3617 021742 012737 000340 177776  MOV #340,@#PS          ;RESTORE PS
3618 021750 012737 000001 001000  MOV #1,ICNT          ;FUGE TEST NUMBERS
3619 021756 000137 001634  OUT: JMP @#TRYNX          ;TEST NEXT DRIVE
3620          .SBTTL          $DONE - BELL AND SCOPE ROUTINE
3621
3622 021762 104400          DONE: SCOPE          ;TERMINATIONG SCOPE FOR LOOPING
3623 021764 062737 000001 001006  ADD #1,PCNT+2          ;ADD 1 TO THE PASS COUNT
3624 021772 005537 001004  ADC PCNT          ;MAKE IT DOUBLE PREC.
3625 021776 032777 002000 157022  BIT #SW10,@SWR          ;RING THE BELL?
3626 022004 001004          BNE 4S          ;NO!
3627 022006 104402 022012          TYPE +2          ;.ASCIZ <BELL><177>
3628 022016 013700 000042 4S: MOV @#42,RO          ;GET MONITOR ADDRESS
3629 022022 001405          BEQ $END1          ;IF NONE
3630 022024 000005          RESET
3631 022026 004710          $ENDAD: JSR 7,(0)          ;GO TO MONITOR
3632 022030 000240 000240 240,240,240  ;SAVE ROOM FOR ACT11
3633 022036 000137 001202  $END1: JMP @#BEGIN          ;RETURN
3634
3635 022042 000000          .TBIT: 0          ;T BIT FLAG
3636
3637 022044 000010          DVTAB: .2LKW 10
    
```

```

3638          .SBTTL          $TYPE - TTY TYPEOUT ROUTINE
3639
3640          ; THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
3641          ; CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ADR" - TYPES THE
3642          ; MESSAGE STARTING IN LOCATION "ADR:" 2) "TYPE CHAR" - TYPES
3643          ; THE ASCII "CHAR", AND 3) "PRINT (<15><12>"MESSAGE") - TYPES
3644          ; THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
3645          ; TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
3646          ; IS IN FILCHR+1.
3647
3648          .TYPE:  MOV      R4, -(6)          ; SAVE R4
3649                  MOV      R5, -(6)          ; SAVE R5
3650          022064 010446          000004      MOV      @4(6), R5      ; GET ADDRESS TO BE TYPED
3651          022074 032705          177400      BIT      #177400, R5    ; IS IT A TYPEN?
3652          022100 001002          000004      BNE      1$          ; NO
3653          022102 016605          000004      MOV      4(6), R5    ; GET ADDRESS OF CHARACTER
3654          022106 105715          1$:      TSTB     (R5)        ; TERMINATOR?
3655          022110 001423          000012      BEQ      2$          ; GET OUT IF SO
3656          022112 122715          000012      CMPB     #12, (R5)   ; IS THE CHAR A LINE FEED
3657          022116 001012          001015      BNE      4$          ; NO - GET OUT
3658          022120 113704          001014      MOVB     FILCHR+1, R4 ; GET THE FILL COUNT
3659          022124 113777          001014      MOVB     FILCHR, @TPB ; TYPE A FILLER
3660          022132 105777          156660      TSTB     @TPS        ; DONE YET?
3661          022136 100375          005304      BPL      -4          ; NO - WAIT
3662          022140 005304          001370      DEC      R4          ; DEC COUNT
3663          022142 001370          156654      BNE      5$          ; LOOP UNTIL 0
3664          022144 112577          156642      MOVB     (R5)+, @TPB ; LOAD AND TYPE THE CHARACTER
3665          022150 105777          000753      TSTB     @TPS        ; IS THE PRINTER READY
3666          022154 100375          017646      BPL      -4          ; WAIT UNTIL IT IS
3667          022156 000753          000004      BR       1$          ; GET THE NEXT CHARACTER
3668          022160 017646          000002      MOV      @4(6), -(6) ; GET ADDRESS TO BE TYPED
3669          022164 062766          000002      ADD      #2, 6(6)    ; ADD 2 TO THE ADDRESS
3670          022172 022666          000004      CMP      (6)+, 4(6)  ; IS IT .+2?
3671          022176 001006          000002      BNE      3$          ; NO
3672          022200 062705          000001      ADD      #2, R5      ; ADD 2 TO THE ADDRESS
3673          022204 042705          000004      BIC      #1, R5      ; BACK UP TO AN EVEN BYTE
3674          022210 010566          000004      MOV      R5, 4(6)    ; RESTORE ADDRESS
3675          022214 012605          3$:      MOV      (6)+, R5    ; RESTORE R5
3676          022216 012604          3$:      MOV      (6)+, R4    ; RESTORE R4
3677          022220 000002          RTI                          ; RETURN

```

```

3678          .SBTTL          $SCOPE - SCOPE LOOP HANDLER
3679
3680          ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
3681          ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
3682          ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
3683          ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
3684
3685          022222 104416          .SCOPE: KBDIN          ;GO CHECK FOR ↑G
3686          022224 032777 000400 156574          BIT          #SWB, @SWR          ;LOOP ON SPEC. TEST?
3687          022232 0C1404          BEQ          1$          ;NO LOOP ON SPEC. TEST
3688          022234 127737 156566 001000          CMPB         @SWR, ICNT          ;ON RIGHT TEST? *SW7-0*
3689          022242 001453          BEQ          .OVER          ;NOT RIGHT TEST
3690          022244 032777 040000 156554 1$:          BIT          #SW14, @SWR          ;LOOP ON TEST?
3691          022252 001045          BNE          .KIT          ;LOOP ON TEST IS SET
3692          022254 000416          BR          3$          ;SKIP - NOP FOR XOR TESTER
3693          022256 013746 000004          MOV          @#4, -(6)          ;PUSH @#4 ON STACK
3694          022262 012737 022302 000004          MOV          #4$,@#4          ;SET FOR TIMEOUT
3695          022270 005737 177060          TST          @#177060          ;ERROR ON XOR?
3696          022274 012637 000004          MOV          (6)+,@#4          ;POP STACK INTO @#4
3697          022300 000422          BR          .SVLAD          ;NO ERROR - GO TO NEXT TEST
3698          022302 022626          4$:          CMP          (6)+,(6)+          ;CLEAR STACK
3699          022304 012637 000004          MOV          (6)+,@#4          ;POP STACK INTO @#4
3700          022310 000426          BR          .KIT          ;ERROR - LOOP ON TEST
3701          022312 032777 004000 156506 3$:          BIT          #SW11, @SWR          ;KILL ITERATIONS
3702          022320 001012          BNE          .SVLAD          ;YES - KILL ITERATIONS
3703          022322 105737 001001          TSTB         ICNT+1          ;FIRST ONE?
3704          022326 0C1404          BEQ          2$          ;BRANCH IF FIRST
3705          022330 123737 022414 001001          CMPB         TIMES, ICNT+1          ;DONE?
3706          022336 003013          BGT          .KIT          ;BRANCH IF NOT
3707          022340 112737 000001 001001 2$:          MOVB         #1, ICNT+1          ;FIRST ITERATION
3708          022346 105237 001000          .SVLAD: INCB         ICNT          ;COUNT TEST NUMBERS
3709          022352 011637 001010          MOV          (6), LAD          ;SAVE LOOP ADDRESS
3710          022356 013777 001000 156444          MOV          ICNT, @DISPLAY          ;DISPLAY TEST NO. AND ITERATION COUNT
3711          022364 000002          RTI          ;RETURN
3712
3713          022366 105237 001001          .KIT:          INCB         ICNT+1          ;INC THE ITERATION COUNT
3714          022372 013777 001000 156430          .OVER:          MOV          ICNT, @DISPLAY          ;SET UP DISPLAY
3715          022400 005737 001010          TST          LAD          ;FIRST ONE?
3716          022404 001760          BEQ          .SVLAD          ;YES
3717          022406 013716 001010          MOV          LAD, (6)          ;FUDGE RETURN ADDRESS
3718          022412 000002          RTI          ;FIXES PS
3719
3720          022414 000100          TIMES:          100          ;RUN 100 TIMES

```

```

3721          .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)
3722
3723          ; THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
3724          ; ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
3725          ; AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
3726          ; "HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
3727          ; (HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADIIONAL TYPEOUTS.
3728
3729          .HLT:   KBDIN          ; GO CHECK FOR 1G
3730          BIT     #SW10, @SWR    ; BELL ON ERROR?
3731          BEQ     1$            ; NO - SKIP
3732          TYPE    , BELL        ; RING BELL
3733          INC     ERRORS        ; COUNT THE NUMBER OF ERRORS
3734          BIT     #SW13, @SWR    ; SKIP TYPEOUT IF SET
3735          BNE     2$            ; SKIP TYPEOUTS
3736          TYPE    , -           ; .ASCIZ '<15><12>'
3737          MOV     (6), HLTADR    ; PUT ADDRESS OF INSTRUCTION ON STACK
3738          SUB     #2, HLTADR    ; FUDGE ADDRESS
3739          MOV     @HLTADR, .HLTCT ; GET HLT ARGUMENT
3740          MOV     HLTADR, -(6)   ; PUT HLTADR ON STACK
3741          TYPE    , -           ; TYPE STACK IN OCTAL
3742          TYPE    , -           ; .ASCIZ " "
3743          JSR     PC, RSREG      ; GO TO USER ERROR ROUTINE
3744          TST     @SWR          ; HALT ON ERROR
3745          BPL     .+4           ; SKIP IF CONTINUE
3746          HALT                    ; HALT ON ERROR!
3747          BIT     #SW9, @SWR     ; CHECK FOR INHIBIT LOOP ON ERROR
3748          BNE     3$            ; SKIP IF LOOP ON ERROR
3749          CLRE   ICNT+1         ; CLEAR ITERATION COUNT
3750          RTI                    ; RETURN
3751          JMP     .KIT          ; LOOP ON TEST UNTIL NO ERRORS
3752
3753          .HLTCT: 0              ; HLT ARGUMENT

```

```

3754          .SBTTL          SOCTAL - OCTAL TYPEOUT ROUTINE
3755
3756          ; THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
3757          ; ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
3758          ; 16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.
3759
3760 022556 012737 170101 022744 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
3761 022564 000411              BR          .PTIT          ;NOW TYPE IT IN BIT FORM
3762 022566 112737 000001 022744 .TYPEO: MOVB   #1,.PR          ;SET ZERO FILL SWITCH
3763 022574 000402              BR          .+6           ;SKIP
3764 022576 005037 022744 .TYPES: CLR    .PR          ;SUPPRESS LEADING ZERO'S
3765 022602 112737 177772 022745 .TYPEO: MOVB   #-6,.PR+1 ;SET COUNT
3766 022610
3767 022610 010446              MOV     R4,-(6)      ;PUSH R4 ON STACK
3768 022612 010546              MOV     R5,-(6)      ;PUSH R5 ON STACK
3769 022614 016605 000010      MOV     10(6),R5    ;GET THE DATA
3770 022620 012704 022746      MOV     #.PR+2,R4   ;SET POINTER TO FIRST ASCII CHAR.
3771 022624 105014              CLRB   (4)          ;CLEAR FIRST BYTE
3772 022626 000411              BR     .PRF         ;ROTATE FIRST BIT
3773 022630 105014              .PRL:  CLRB   (4)          ;CLEAR BYTE OF CHARACTER
3774 022632 032737 000100 022744 .PRL:  BIT     #100,.PR ;BIT TYPING MODE?
3775 022640 001004              BNE    .PRF         ;YES - SKIP 2 ROTATES
3776 022642 006105              ROL    R5           ;ROTATE BIT INTO C
3777 022644 106114              ROLB  (4)          ;PACK IT
3778 022646 006105              ROL    R5           ;ROTATE BIT INTO C
3779 022650 106114              ROLB  (4)          ;PACK IT
3780 022652 006105              .PRF:  ROL    R5           ;ROTATE BIT INTO C
3781 022654 106114              ROLB  (4)          ;PACK IT
3782 022656 105714              TSTB  (4)          ;IS IT ZERO?
3783 022660 001402              BEQ    .+6         ;SKIP INC
3784 022662 105237 022744      INCB  .PR          ;SET FILL SWITCH
3785 022666 105737 022744      TSTB  .PR          ;CHECK FILL SWITCH
3786 022672 001402              BEQ    .+6         ;SKIP BITSET
3787 022674 152724 000060      BISB  #'0,(4)+    ;MAKE INTO ASCII CHAR
3788 022700 105237 022745      INCB  .PR+1       ;INC COUNT
3789 022704 001351              BNE    .PRL        ;REPEAT
3790 022706 022704 022746      CMP   #.PR+2,R4   ;EMPTY BUFFER?
3791 022712 001002              BNE    .+6         ;SKIP IF NOT
3792 022714 112724 000060      MOVB  #'0,(4)+    ;LOAD 1 ZERO
3793 022720 105014              CLRB  (4)          ;NULL TERMINATOR
3794 022722 104402 022746      TYPE  .PR+2       ;TYPE IT
3795 022726 012605              MOV   (6)+,R5     ;POP STACK INTO R5
3796 022730 012604              MOV   (6)+,R4     ;POP STACK INTO R4
3797 022732 016666 000002 000004 .TYPEO: MOV   2(6),4(6) ;GET RID OF
3798 022740 012616              MOV   (6)+,(6)   ;DATA WORD
3799 022742 000002              RTI
3800
3801 022744 000012          .PR:  .BLKW  12      ;COUNT, SWITCH, AND OUTPUT BUFFER

```

```

3802          .SBTTL          $POWER - POWER DOWN AND UP ROUTINES
3803
3804          ; THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
3805          ; THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
3806          ; WAIT FOR POWER TO GO DOWN AND BE RESTORED.
3807          ; IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
3808          ; THE PROGRAM WILL HALT AT '.ILLUP'.
3809
3810 022770 012777 023114 000124 .POWER: MOV      #.ILLUP,2.PUVEC ;SET FOR FAST UP
3811 022776 012777 000340 000120      MOV      #340,2.PUVEC$+2 ;PRIO:7
3812 023004 010046          MOV      R0,-(6) ;PUSH R0 ON STACK
3813 023006 010146          MOV      R1,-(6) ;PUSH R1 ON STACK
3814 023010 010246          MOV      R2,-(6) ;PUSH R2 ON STACK
3815 023012 010346          MOV      R3,-(6) ;PUSH R3 ON STACK
3816 023014 010446          MOV      R4,-(6) ;PUSH R4 ON STACK
3817 023016 010546          MOV      R5,-(6) ;PUSH R5 ON STACK
3818 023020 010637 023120          MOV      SP,SAVR6 ;SAVE SP
3819 023024 012777 023034 000070      MOV      #.POWUP,2.PUVEC ;SET UP VECTOR
3820 023032 000000          HALT      ;WAIT FOR PF
3821
3822 023034 013706 023120          .POWUP: MOV      .SAVR6,SP ;GET SP
3823 023040 005001          CLR      R1 ;WAIT LOOP FOR THE TTY
3824 023042 005201          IS:    INC      R1 ;WAIT FOR THE INC
3825 023044 001376          BNE     IS ;OF WORD
3826 023046 012605          MOV      (6)+,R5 ;POP STACK INTO R5
3827 023050 012604          MOV      (6)+,R4 ;POP STACK INTO R4
3828 023052 012603          MOV      (6)+,R3 ;POP STACK INTO R3
3829 023054 012602          MOV      (6)+,R2 ;POP STACK INTO R2
3830 023056 012601          MOV      (6)+,R1 ;POP STACK INTO R1
3831 023060 012600          MOV      (6)+,R0 ;POP STACK INTO R0
3832 023062 012737 022770 000024      MOV      #.POWER,2#24 ;SET UP THE POWER DOWN VECTOR
3833 023070 012737 000340 000026      MOV      #340,2#26 ;PRIO:7
3834 023076 104402 023102          TYPE   ,.+2 ;ASCIZ <15><12>"POWER"
3835 023112 000002          RTI     ;RETURN
3836
3837 023114 000000          .ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
3838 023116 000776          BR      .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
3839
3840 023120 000000          .SAVR6: 0 ;PUT THE SP HERE
3841 023122 000024 000026          .PUVEC: 24,26 ;POWER UP VECTOR

```


F08

MAINDEC-11-DJRSE -E
DZRSBE.P11

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 97
SRDOCT - OCTAL INPUT ROUTINE

```

3842          .SBTTL          SRDOCT - OCTAL INPUT ROUTINE
3843
3844          ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
3845          ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
3846
3847 023126 011646          .RDOCT: MOV      (6),-(6)          ;MOVE THE PC
3848 023130 016666 000004 000002 MOV      4(6),2(6)          ;MOVE THE PS
3849 023136 010146          MOV      R1,-(6)          ;PUSH R1 ON STACK
3850 023140 010246          MOV      R2,-(6)          ;PUSH R2 ON STACK
3851 023142 010346          MOV      R3,-(6)          ;PUSH R3 ON STACK
3852 023144 104412 4$: RDLIN          ;READ A LINE INTO INPUT
3853 023146 005001          CLR      R1          ;INIT DATA WORD
3854 023150 005037 027276          CLR      CTN          ;CLEAR COUNT WORD
3855 023154 012703 023402          MOV      #INPUT,R3          ;INIT POINTER
3856 023160 112302 1$: MOVB      (3)+,R2          ;GET A BYTE
3857 023162 122702 000015          CMPB     #15,R2          ;WAS IT A CR?
3858 023166 001421          BEQ     2$          ;GET OUT IF YES
3859 023170 122702 000060          CMPB     #'0,R2          ;CHECK FOR 0 OR GREATER
3860 023174 003024          BGT     3$          ;ERROR - LESS THAN 0
3861 023176 122702 000067          CMPB     #'7,R2          ;CHECK FOR 7 OR LESS
3862 023202 002421          BLT     3$          ;ERROR - GREATER THAN 7
3863 023204 006002          ROR     R2          ;GET
3864 023206 006002          ROR     R2          ;INTO
3865 023210 006002          ROR     R2          ;POSITION
3866 023212 006101          ROL     R1          ;FIRST BIT
3867 023214 006102          ROL     R2          ;GET
3868 023216 006101          ROL     R1          ;SECOND BIT
3869 023220 006102          ROL     R2          ;GET
3870 023222 006101          ROL     R1          ;THIRD BIT
3871 023224 005237 027276          INC     CTN          ;YES HE TYPED SOMETHING
3872 023230 000753          BR      1$          ;LOOP
3873 023232 010166 000012 2$: MOV      R1,12(6)          ;SAVE THE RESULT
3874 023236 012603          MOV     (6)+,R3          ;POP STACK INTO R3
3875 023240 012602          MOV     (6)+,R2          ;POP STACK INTO R2
3876 023242 012601          MOV     (6)+,R1          ;POP STACK INTO R1
3877 023244 000002          RTI
3878
3879 023246          3$:          ;ASCIZ "'"(15)<12>
3880 023246 104402 023252          BR      4$+2          ;TRY AGAIN
3881 023256 000732

```

```

3882
3883
3884
3885
3886
3887
3888
3889 023260 010546
3890 023262 012705 023402
3891 023266 022705 023422
3892 023272 001423
3893 023274 105737 177560
3894 023300 100375
3895 023302 113715 177562
3896 023306 142715 000200
3897 023312 122715 000025
3898 023316 001006
3899 023320 104402 023324
3900 023332 000753
3901 023334 122715 000177
3902 023340 001005
3903 023342
3904 023342 104402 023346
3905 023352 000743
3906 023354 111527 000000
3907 023360 104402 023356
3908 023364 122725 000015
3909 023370 001336
3910 023372 104402 000012
3911 023376 012605
3912 023400 000002
3913
3914 023402 000020
3915
3916
3917
3918
3919
3920
3921
3922 023422 011646
3923 023424 162716 000002
3924 023430 017616 000000
3925 023434 062716 117042
3926 023440 013607
3927
3928 023442 022222
3929 023444 022064
3930 023446 022566
3931 023450 022576
3932 023452 023126
3933 023454 023260
3934 023456 026050
3935 023460 027126
3936 023462 027006
3937 023464 027210

```

```

.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
1$:      MOV      #INPUT, R5      ;GET ADDRESS
2$:      CMP      #INPUT+16., R5   ;BUFFER FULL?
        BEQ      4$              ;YES - TYPE "?"
        TSTB     @#177560        ;WAIT FOR
        BPL      -4              ;A CHARACTER
        MOVB     @#177562, (5)    ;GET CHARACTER
        BICB     #200, (5)       ;GET RID OF JUNK
        CMPB     #25, (5)        ;IS IT A ↑U
        BNE     5$              ;BRANCH IF NOT
        TYPE     .+2             ;.ASCIZ "↑U"<15><12>
        BR       1$            ;START OVER
5$:      CMPB     #177, (5)       ;IS IT A RUBOUT
        BNE     3$              ;SKIP IF NOT
4$:      TYPE     .+2             ;.ASCIZ "?"<15><12>
        BR       1$            ;ZAP THE BUFFER AND LOOP
3$:      MOVB     (5), #0        ;SET UP FOR TYPING
        TYPE     .3$+2          ;ECHO IT
        CMPB     #15, (5)+       ;CHECK FOR RETURN
        BNE     2$              ;LOOP IF NOT RETURN
        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      #.TRP+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)
.CLRDK     = TRAP+14     (104414)
.KBDIN     = TRAP+16     (104416)
.SUSWR     = TRAP+20     (104420)
.CNTLU     = TRAP+22     (104422)

```

H08

M:INDEC-11-DZRSB-E
DZRSBE.P11

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 99
\$TRAP - TRAP HANDLER

```

3938 ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
3939 ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
3940
3941 ;WORD COUNT REGISTER
3942 023466 012706 0C0500 $RSWC: MOV #500,SP ;SET UP STACK FOR TRAP CALL
3943 023472 104416 $S: KBDIN ;CHECK THE WORLD
3944 023474 017777 155326 155406 MOV @SWR,@RSWC ;MOV SWR INTO WORD COUNT REG
3945 023502 017777 155402 155320 MOV @RSWC,@DISPLAY ;DISPLAY IN LIGHTS
3946 023510 000770 BR $S
3947
3948 ;CURRENT ADDRESS REGISTER
3949 023512 012706 000500 $RSBA: MOV #500,SP ;INIT THE STACK
3950 023516 104416 $S: KBDIN ;CTW
3951 023520 017777 155302 155364 MOV @SWR,@RSBA ;MOV SWR INTO CURRENT ADDR REG
3952 023526 017777 155360 155274 MOV @RSBA,@DISPLAY ;SHOW IN LIGHTS
3953 023534 000770 BR $S
3954
3955 ;DISK ADDRESS REGISTER
3956 023536 012706 000500 $RSDA: MOV #500,SP ;INIT THE STACK
3957 023542 104416 $S: KBDIN ;CTW
3958 023544 017777 155256 155342 MOV @SWR,@RSDA ;MOV SWR INTO DISK ADDR REG
3959 023552 017777 155336 155250 MOV @RSDA,@DISPLAY ;SHOW IN LIGHTS
3960 023560 000770 BR $S
3961
3962 ;DRIVE STATUS REGISTER
3963 023562 012706 000500 $RSDS: MOV #500,SP ;INIT THE STACK
3964 023566 104416 $S: KBDIN ;CTW
3965 023570 017777 155232 155320 MOV @SWR,@RSDS ;MOV SWR INTO DRIVE STATUS
3966 023576 017777 155314 155224 MOV @RSDS,@DISPLAY ;SHOW IN LIGHTS
3967 023604 000770 BR $S
3968
3969 ;DRIVE ERROR REGISTER
3970 023606 012706 000500 $RSER: MOV #500,SP ;INIT THE STACK
3971 023612 104416 $S: KBDIN ;CTW
3972 023614 017777 155206 155276 MOV @SWR,@RSER ;LOAD ER REG
3973 023622 017777 155272 155200 MOV @RSER,@DISPLAY ;DISPLAY IT IN LIGHTS
3974 023630 000770 BR $S ;LOOP
3975
3976 ;WATCH LOOK AHEAD REGISTER
3977 023632 017777 155266 155170 $RSLA: MOV @RSLA,@DISPLAY ;SHOW IN LIGHTS
3978 023640 000774 BR $RSLA

```

```

3979                                     :RSCS2 REGISTER
3980 023642 012706 000500 SRCS2:  MOV    #500,SP           ;INIT THE STACK
3981 023646 104416 1$:      KBDIN                    ;CTW
3982 023650 017777 155152 155230      MOV    @SWR,@RSCS2      ;LOAD CS2
3983 023656 017777 155224 155144      MOV    @RSCS2,@DISPLAY ;DISPLAY IT
3984 023664 000770 1$:      BR                      1$
3985
3986                                     :RSAS REGISTER
3987 023666 012706 000500 SRAS:  MOV    #500,SP           ;INIT THE STACK
3988 023672 104416 1$:      KBDIN                    ;CTW
3989 023674 017777 155126 155220      MOV    @SWR,@RSAS      ;LOAD RSAS
3990 023702 017777 155214 155120      MOV    @RSAS,@DISPLAY ;DISPLAY IT
3991 023710 000770 1$:      BR                      1$
3992
3993                                     :RSMR REGISTER
3994 023712 012706 000500 RSMR:  MOV    #500,SP           ;INIT THE STACK
3995 023716 104416 1$:      KBDIN                    ;CTW
3996 023720 017777 155102 155202      MOV    @SWR,@RSMR      ;LOAD RSMR
3997 023726 017777 155176 155074      MOV    @RSMR,@DISPLAY ;DISPLAY IT
3998 023734 000770 1$:      BR                      1$
3999
4000                                     :DISK CONTROL STATUS REGISTER
4001 023736 012706 000500 SRSCS1: MOV    #500,SP           ;INIT THE STACK
4002 023742 104416 1$:      KBDIN                    ;CTW
4003 023744 012737 000340 177776      MOV    #340,@#PS       ;LOCK UP INTERRUPTS
4004 023752 012777 177777 155130      MOV    #177777,@RSWC   ;SET WORD COUNT -1 WORD
4005 023760 013777 001102 155124      MOV    @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
4006 023766 017777 155034 155110      MOV    @SWR,@RSCS1     ;MOV SWR INTO CONTROL REG
4007 023774 032777 000001 155102      BIT    #BIT0,@RSCS1    ;IS FUNCTION BITS SET
4008 024002 001757 1$:      BEQ    1$           ;FUNCTION BITS NOT SET
4009 024004 105777 155074 2$:      TSTB  @RSCS1        ;TEST FOR DISK READY
4010 024010 100375 2$:      BPL   2$           ;DISK STILL NOT READY
4011 024012 000753 1$:      BR    1$           ;DISK NOT BUSY SECT NEW CR

```

JOB

MAINDEC-11-DZRSB-E
DZRSBE.P11

RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 101
STRAP - TRAP HANDLER

```

4012      ; THIS ROUTINE GIVES THE OPERATOR THE ABILITY TO
4013      ; SELECT DA, WC, UNIT # AND DESIRED PATTERN. PATTERN = NUMBER TYPED
4014      ; WITH SW12 SET THE PROGRAM WILL LOOP ON A READ WITH LOC OUTBUF+2 AS
4015      ; THE BA ADDR. WITH BIT12 0 IN THE SWR THE PROGRAM
4016      ; WILL WRITE WITH OUTBUF AS THE BA ADDR. BAI IS ALWAYS SET
4017      ; SWITCHES 0 TO 11 WILL DETERMINE THE DA
4018      ; EXAMPLE:
4019      ;           TYPE UNIT # 5
4020      ;           TYPE POSITIVE (OCTAL) WC 64
4021      ;           TYPE PATTERN DESIRED 1252525
4022      TKSEL:  MOV     #500,SP           ;SET STACK
4023      NOP
4024      TYPE    ,.+2                     ;.ASCIZ <15><12>"TYPE UNIT # "
4025      RDOCT
4026      MOV     (6)+,UNNUM
4027      TYPE    ,.+2                     ;.ASCIZ <15><12>"TYPE POSITIVE (OCTAL) WC "
4028      RDOCT
4029      MOV     (6)+,WORK
4030      COM     WORK
4031      TYPE    ,.+2                     ;.ASCIZ <15><12>"TYPE PATTERN DESIRED "
4032      RDOCT
4033      MOV     (6)+,OUTBUF
4034      BIC     #BIT0,SWI                 ;CLEAR THE BEENHEREBIT
4035      SUSWR
4036      MOV     @SWR,WORK2                ;INIT SWITCHLESS
4037      CLRDK
4038      BIS     #BIT3,@RSCS2              ;SAVE SWR
4039      MOV     @SWR,WORK1                ;CLEAR ALL RS REG
4040      BIC     #170000,WORK1             ;SET BAI
4041      MOV     WORK1,@RSDA                ;GET SWR FOR DSK ADDR
4042      MOV     WORK,@RSCS1               ;CLEAR UNIT #
4043      BIT     #BIT12,@SWR                ;LOAD THE DA
4044      BEQ     WTE                         ;LOAD WORD COUNT
4045      MOV     #OUTBUF+2,@RSBA           ;READ?
4046      MOV     #71,@RSCS1                ;NO
4047      TSTB   @RSCS1                      ;LOAD CURRENT ADDRESS
4048      BPL    -4                          ;GO AND READ
4049      BR     SWRCHG                       ;TEST FOR READY
4050      MOV     #OUTBUF,@RSBA
4051      MOV     #61,@RSCS1
4052      BR     WT
4053      MOV     @RSCS1,@DISPLAY           ;DISPLAY CS1
4054      TST     @RSCS1                     ;ANY ERRORS?
4055      BPL    $                           ;NO
4056      HLT    !DA!WC
4057      KBDIN
4058      CMP     @SWR,WORK2                ;CHECK FOR NEW VALUE
4059      BNE    TK1                          ;DID SWR CHANGE?
4060      BR     TK2                          ;YES
                                         ;NO

```

K08

MAINDEC-11-DZRSB-E
DZRSBE.P11

TST122

RH11-R503LA-R503-R504 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 102
WRITE LOCK TEST

```
4061 ;*****
4062 ;TEST 122 WRITE LOCK TEST
4063 ;*****
4064 024352 104400 †TST122: SCOPE
4065
4066 WRTLCK:
4067 024354 104402 024360 TYPE ;.ASCIZ <15><12>"LOAD SW WITH UNIT # AND CONT"
4068 024420 022737 000176 001026 CMP #SWREG,SWR ;CHECK IF SWITCHLESS CPU
4069 024426 001402 BEQ 7$
4070 024430 000000 HALT
4071 024432 000401 BR 8$ ;GO AROUND TRAP CALL
4072 024434 104422 7$: CNTLU ;GET SWREG VALUE
4073 024436 017737 154364 001152 8$: MOV @SWR,UNNUM ;GET UNIT #
4074 024444 104414 CLRDK ;CLEAR ALL REG
4075 024446 005037 027364 CLR OUTBUF ;PUT A 0 INTO DATA BUFFER
4076 024452 012777 027364 154432 MOV #OUTBUF,@RSBA ;SETUP REG TO
4077 024460 012777 007700 154426 MOV #7700,@RSDA ;TO A WRITE
4078 024466 012777 177777 154414 MOV #-1,@RSWC
4079 024474 012777 000061 154402 MOV #61,@RSCS1
4080 024502 105777 154376 6$: TSTB @RSCS1 ;WAIT FOR DONE
4081 024506 100375 BPL 6$
4082 024510 104402 024514 TYPE ;.ASCIZ <15><12>"SET WRITE LOCK ENABLE AND CONTINUE"
4083 024562 000000 HALT ;.ASCIZ <15><12>"SET WRT LOC SW 0 AND CONT"
4084 024564 012777 000000 154322 1$: MOV #0,@RSDA
4085 024572 022777 014600 154316 CMP #14600,@RSDS
4086 024600 001401 BEQ +4
4087 024602 104044 HLT !DS!DA ;DS SHOULD=14600
4088 024604 012777 000100 154302 MOV #100,@RSDA
4089 024612 022777 010600 154276 CMP #10600,@RSDS
4090 024620 001401 BEQ +4
4091 024622 104044 HLT !DS!DA ;DS SHOULD=10600
4092 024624 104402 024630 TYPE ;.ASCIZ <15><12>"SET WRT LOC SW 0 AND CONT"
4093 024664 000000 HALT
4094 024666 022777 014600 154222 2$: CMP #14600,@RSDS
4095 024674 001401 BEQ +4
4096 024676 104044 HLT !DS!DA ;DS SHOULD=14600
4097 024700 012777 000300 154206 MOV #300,@RSDA
4098 024706 022777 010600 154202 CMP #10600,@RSDS
4099 024714 001401 BEQ +4
4100 024716 104044 HLT !DS!DA ;DS SHOULD=10600
4101 024720 104402 024724 TYPE ;.ASCIZ <15><12>"SET WRT LOC SW 1 AND CONT"
4102 024760 000000 HALT
4103 024762 022777 014600 154126 3$: CMP #14600,@RSDS
4104 024770 001401 BEQ +4
4105 024772 104044 HLT !DS!DA ;DS SHOULD=14600
4106 024774 012777 000700 154112 MOV #700,@RSDA
4107 025002 022777 010600 154106 CMP #10600,@RSDS
4108 025010 001401 BEQ +4
4109 025012 104044 HLT !DS!DA ;DS SHOULD=10600
4110 025014 104402 025020 TYPE ;.ASCIZ <15><12>"SET WRT LCK SW 2 AND CONT"
4111 025054 000000 HALT
```

L08

4112	025056	022777	014600	154032	4\$:	CMP	#14600, @RSDS	
4113	025064	001401				BEQ	.+4	
4114	025066	104044				HLT	!DS!DA	;DS SHOULD=14600
4115	025070	012777	001700	154016		MOV	#1700, @RSDA	
4116	025076	022777	010600	154012		CMP	#10600, @RSDS	
4117	025104	001401				BEQ	.+4	
4118	025106	104044				HLT	!DS!DA	;DS SHOULD=10600
4119	025110	104402	025114			TYPE	,.+2	;.ASCIZ <15><12>"SET WRT LCK SW 3 AND CONT"
4120	025150	000000			HALT:	HALT		
4121	025152	022777	014600	153736		CMP	#14600, @RSDS	
4122	025160	001401				BEQ	.+4	
4123	025162	104044				HLT	!DS!DA	;DS SHOULD=14600
4124	025164	012777	003700	153722		MOV	#3700, @RSDA	
4125	025172	022777	010600	153716		CMP	#10600, @RSDS	
4126	025200	001401				BEQ	.+4	
4127	025202	104044				HLT	!DS!DA	;DS SHOULD=10600
4128	025204	104402	025210			TYPE	,.+2	;.ASCIZ <15><12>"SET WRT LCK SW 4 AND CONT"
4129	025244	000000				HALT		
4130	025246	022777	014600	153642		CMP	#14600, @RSDS	
4131	025254	001401				BEQ	.+4	
4132	025256	104044				HLT	!DS!DA	;DS SHOULD=14600
4133	025260	012777	006000	153626		MOV	#6000, @RSDA	
4134	025266	022777	010600	153622		CMP	#10600, @RSDS	
4135	025274	001401				BEQ	.+4	
4136	025276	104044				HLT	!DS!DA	;DS SHOULD=10600
4137	025300	104402	025304			TYPE	,.+2	;.ASCIZ <15><12>"SET WRT LCK SW 5 AND CONT"
4138	025340	000000				HALT		
4139	025342	022777	014600	153546		CMP	#14600, @RSDS	
4140	025350	001401				BEQ	.+4	
4141	025352	104044				HLT	!DS!DA	;DS SHOULD=14600
4142	025354	012777	007700	153532		MOV	#7700, @RSDA	
4143	025362	022777	014600	153526		CMP	#14600, @RSDS	
4144	025370	001401				BEQ	.+4	
4145	025372	104044				HLT	!DS!DA	;DS SHOULD=14600
4146	025374	012737	177777	027364		MOV	#-1, @OUTBUF	;PUT A 1 INTO DATA BUFFER
4147	025402	012777	027364	153502		MOV	@OUTBUF, @RSBA	;SETUP REG TO
4148	025410	012777	007700	153476		MOV	#7700, @RSDA	;TO A WRITE
4149	025416	012777	177777	153464		MOV	#-1, @RSWC	
4150	025424	012777	000061	153452		MOV	#61, @RSCS1	;TRY TO WRITE
4151	025432	105777	153446		7\$:	TSTB	@RSCS1	;WAIT FOR DONE
4152	025436	100375				BPL	7\$	
4153	025440	105777	153440		5\$:	TSTB	@RSCS1	;WAIT FOR READY
4154	025444	100375				BPL	5\$	
4155	025446	022777	154600	153442		CMP	#154600, @RSDS	
4156	025454	001401				BEQ	.+4	
4157	025456	104044				HLT	!DS!DA	;DS SHOULD=154600

MO8

4158	025460	022777	004000	153432		CMP	#4000, @RSER	
4159	025466	001401				BEQ	:+4	
4160	025470	104044				HLT	!DS!DA	;ER SHOULD=4000
4161	025472	104414				CLRDK		;CLEAR ALL REG
4162	025474	022777	014600	153414		CMP	#14600, @RSOS	;DID WLE CLEAR?
4163	025502	001401				BEQ	:+4	;NO
4164	025504	104040				HLT	!DS	;A CLEAR SHOULD NOT CLEAR WLE
4165	025506	005037	027364			CLR	OUTBUF	;PUT A 0 INTO DATA BUFFER
4166	025512	012777	027364	153372		MOV	#OUTBUF, @RSBA	;SETUP REG TO
4167	025520	012777	007700	153366		MOV	#7700, @RSDA	;TO A WRITE
4168	025528	012777	177777	153354		MOV	#-1, @RSWC	
4169	025534	012777	000071	153342		MOV	#71, @RSCS1	;DO A READ TO SEE IF DISK DID
4170	025542	105777	153336		8\$:	TSTB	@RSCS1	;ACTUALLY GET WRITTEN ON TO
4171	025546	100375				BPL	ES	;WAIT FOR DONE
4172	025550	005737	027364			TST	OUTBUF	;IS DATA STILL 0 ON THE DSK?
4173	025554	001401				BEQ	:+4	;YES
4174	025556	104040				HLT	!DS	;NO DSK DID GET WRITTEN ONTO WITH WLE SET
4175	025560	104402	025564			TYPE	..+2	;ASCIZ <15><12>"RESET ALL W/L SWITCHES"
4176	025616	000000				HALT		
4177	025620	022777	010600	153270		CMP	#10600, @RSDS	;DID WLE CLEAR
4178	025626	001401				BEQ	:+4	;YES
4179	025630	104040				HLT	!DS	;NO
4180								
4181								
4182	025632	012737	025716	000004		ROUTINE FOR FINDING MEMORY ON "B" PORT		
4183	025640	012737	000340	000006	FINDM:	MOV	#MAXREF, 4	;SET UP I/O BUS TRAP
4184	025646	032777	010000	153152		MOV	#340, 6	;SET PS
4185	025654	001003				BIT	#BIT12, @SWR	;INHIBIT OBUFSV FROM CHANGING?
4186	025656	062737	020000	001102		BNE	EXREF	;YES
4187	025664	005777	153212		EXREF:	ADD	#20000, OBUFSV	;ADD 4 K
4188	025670	012737	000006	000004		TST	@OBUFSV	;LEGAL LOC ? IF NO TRAPS
4189	025676	005037	000006			MOV	#6, 4	;RESTORE I/O BUS TRAP
4190	025702	022737	177446	001102		CLR	6	
4191	025710	102403				CMP	#177446, OBUFSV	;TEST FOR GREATER THEN 28K
4192	025712	000137	006736			BLO	MAXRF1	;YES
4193	025716	022626				JMP	WRTSTB	;RETRY WRITING
4194	025720	012737	000006	000004	MAXREF:	CMP	(6)+, (6)+	;CLEAR STACK
4195	025726	005037	000006		MAXRF1:	MOV	#6, 4	;RESTORE I/O BUS TRAP
4196	025732	012737	027364	001102		CLR	6	
4197	025740	032777	020000	153060		MOV	#OUTBUF, OBUFSV	;RESTORE ORIGINAL VALUE
4198	025746	001031				BIT	#BIT13, @SWR	;INHIBIT TYPEOUT?
4199	025750	032737	020000	001160		BNE	1\$;YES
4200	025756	001025				BIT	#BIT13, ONCEE	;DID WE TYPE THIS YET?
4201	025760	104402	025764			BNE	1\$;YES
4202						TYPE	..+2	;ASCIZ <15><12>"COULD NOT FIND MEMORY ON -B- PORT"<15><
4203	026032	052737	020000	001160	.EVEN			
4204	026040	104400			1\$:	BIS	#BIT13, ONCEE	;SET TYPE FLAG
4205	026042	104400				SCOPE		;UPDATE
4206	026044	000137	007526			SCOPE		;TEST NUMBERS
4207						JMP	NXM	;CONT TESTING
4208								
4209	026050	012777	000040	153L30		;CLEAR ALL DISK REGISTERS		
4210	026056	013777	001152	153022	.CLRDK:	MOV	#40, @RSCS2	;CLEAR ALL DSK REG
4211	026064	000002				MOV	UNNUM, @RSCS2	;GET UNIT NUMBER
						RTI		


```

;ERROR TYPTXTOUT ROUTINE
4212
4213
4214 026066 005737 022554 RSREG: TST .HLTCT ;SHOULD WE TYPTXT GOOD AND BAD
4215 026072 001021 BNE 8$ ;NO
4216 026074 104402 026100 TYPE .+2 ;ASCIZ "BAD="
4217 026106 010046 MOV BAD,-(6) ;PUT BAD ON STACK
4218 026110 104404 TYPEO ;TYPE STACK IN OCTAL
4219 026112 104402 026116 TYPE .+2 ;ASCIZ "GOOD="
4220 026126 010146 MOV GOOD,-(6) ;PUT GOOD ON STACK
4221 026130 104404 TYPEO ;TYPE STACK IN OCTAL
4222 026132 000137 026570 JMP PTDONE ;GET OUT
4223 026136
4224 026136 104402 026142 8$: TYPE .+2 ;ASCIZ "CS1="
4225 026150 017746 152730 MOV @RSCS1,-(6) ;PUT @RSCS1 ON STACK
4226 026154 104404 TYPEO ;TYPE STACK IN OCTAL
4227 026156
4228 026156 104402 026162 1$: TYPE .+2 ;ASCIZ "FR="
4229 026170 017746 152724 MOV @RSER,-(6) ;PUT @RSER ON STACK
4230 026174 104404 TYPEO ;TYPE STACK IN OCTAL
4231 026176
4232 026176 104402 026202 2$: TYPE .+2 ;ASCIZ "CS2="
4233 026210 017746 152672 MOV @RSCS2,-(6) ;PUT @RSCS2 ON STACK
4234 026214 104404 TYPEO ;TYPE STACK IN OCTAL
4235 026216 032737 000200 022554 BIT #200,.HLTCT ;TYPTXT SECOND SET ?
4236 026224 001076 BNE SEEC ;YES
4237 026226 032737 000100 022554 BIT #AS,.HLTCT ;TYPTXT ER ?
4238 026234 001410 BEQ 3$ ;NO
4239 026236 104402 026242 TYPE .+2 ;ASCIZ "AS="
4240 026250 017746 152646 MOV @RSAS,-(6) ;PUT @RSAS ON STACK
4241 026254 104404 TYPEO ;TYPE STACK IN OCTAL
4242 026256 032737 000020 022554 3$: BIT #BA,.HLTCT ;TYPTXT BUS ASSRESS
4243 026264 001410 BEQ 4$ ;NO
4244 026266 104402 026272 TYPE .+2 ;ASCIZ "BA="
4245 026300 017746 152606 MOV @RSBA,-(6) ;PUT @RSBA ON STACK
4246 026304 104404 TYPEO ;TYPE STACK IN OCTAL
4247 026306 032737 000004 022554 4$: BIT #DA,.HLTCT ;TYPTXT DA ?
4248 026314 001410 BEQ 5$ ;NO
4249 026316 104402 026322 TYPE .+2 ;ASCIZ "DA="
4250 026330 017746 152560 MOV @RSDA,-(6) ;PUT @RSDA ON STACK
4251 026334 104404 TYPEO ;TYPE STACK IN OCTAL
4252 026336 032737 000010 022554 5$: BIT #WC,.HLTCT ;TYPTXT WC ?
4253 026344 001410 BEQ 6$ ;NO
4254 026346 104402 026352 TYPE .+2 ;ASCIZ "WC="
4255 026360 017746 152524 MOV @RSWC,-(6) ;PUT @RSWC ON STACK
4256 026364 104404 TYPEO ;TYPE STACK IN OCTAL
    
```

4257	026366	032737	000040	022554	6\$:	BIT	#DS,.HLTCT	:DRIVE STATUS
4258	026374	001475				BEQ	PTDONE	:NO
4259	026376	104402	026402			TYPE	.+2	:.ASCIZ "DS="
4260	026410	017746	152502			MOV	ARSDS,-(6)	:PUT ARSDS ON STACK
4261	026414	104404				TYPE0		:TYPE STACK IN OCTAL
4262	026416	000137	026570			JMP	PTDONE	:GET OUT
4263	026422	042737	000200	022554	SEEC:	BIC	#200,.HLTCT	:CLEAR COMMON BIT
4264	026430	032737	000240	022554		BIT	#DT,.HLTCT	:TYPTXT DRIVE TYPE?
4265	026436	001410				BEQ	9\$:NO
4266	026440	104402	026444			TYPE	.+2	:.ASCIZ "DT="
4267	026452	017746	152454			MOV	ARSDT,-(6)	:PUT ARSDT ON STACK
4268	026456	104404				TYPE0		:TYPE STACK IN OCTAL
4269	026460	032737	000210	022554	9\$:	BIT	#DB,.HLTCT	:TYPTXT DATA BUFFER
4270	026466	001410				BEQ	10\$:NO
4271	026470	104402	026474			TYPE	.+2	:.ASCIZ "DB="
4272	026502	017746	152420			MOV	ARSDB,-(6)	:PUT ARSDB ON STACK
4273	026506	104404				TYPE0		:TYPE STACK IN OCTAL
4274	026510	032737	000220	022554	10\$:	BIT	#MR,.HLTCT	:TYPTXT MN?
4275	026516	001410				BEQ	11\$:NO
4276	026520	104402	026524			TYPE	.+2	:.ASCIZ "MR="
4277	026532	017746	152372			MOV	ARSMR,-(6)	:PUT ARSMR ON STACK
4278	026536	104404				TYPE0		:TYPE STACK IN OCTAL
4279	026540	032737	000204	022554	11\$:	BIT	#LA,.HLTCT	:TYPTXT LA?
4280	026546	001410				BEQ	PTDONE	:NO
4281	026550	104402	026554			TYPE	.+2	:.ASCIZ "LA="
4282	026562	017746	152336			MOV	ARSLA,-(6)	:PUT ARSLA ON STACK
4283	026566	104404				TYPE0		:TYPE STACK IN OCTAL
4284	026570	052737	100000	001160	PTDONE:	BIS	#BIT15,ONCE	:SET FORND ERROR FLAG
4285	026576	000207				RTS	PC	
4286								
4287	026600	005037	001174		WAITRY:	CLR	WORK	:CLEAR COUNTER
4288	026604	105777	152274		1\$:	TSTB	ARSCS1	:TEST READY
4289	026610	100406				BMI	2\$:OK CONT
4290	026612	005237	001174			INC	WORK	:UPDATE COUNTER
4291	026616	005737	001174			TST	WORK	:DONE YET?
4292	026622	001403				BEQ	3\$:READY DID NOT COME UP
4293	026624	000767				BR	1\$:CONTINUE WAITING
4294	026626	062716	000002		2\$:	ADD	#2,(SP)	:UPDATE RETURN PC
4295	026632	000207			3\$:	RTS	PC	:RETURN

```

;RANDOM DATA GENERATOR SUBROUTINE
4296
4297
4298 026634 013737 026776 027002 RANDOM: MOV LONUM,LOSAY
4299 026642 013737 027000 027004 MOV HINUM,HISAY
4300 026650 013700 026776 RAND1: MOV LONUM,R0
4301 026654 013704 027000 MOV HINUM,R4
4302 026660 012703 000007 MOV #7,R3
4303 026664 005002 CLR R2
4304 026666 006300 SHIFT: ASL R0
4305 026670 006104 ROL R4
4306 026672 006102 ROL R2
4307 026674 005303 DEC R3
4308 026676 001373 SNE SHIFT
4309 026700 063700 026776 ADD LONUM,R0
4310 026704 005504 ADC R4
4311 026706 063704 027000 ADD HINUM,R4
4312 026712 005502 ADC R2
4313 026714 062700 001057 ADD #1057,R0
4314 026720 005504 ADC R4
4315 026722 005502 ADC R2
4316 026724 062704 047401 ADD #47401,R4
4317 026730 005502 ADC R2
4318 026732 062702 000006 ADD #6,R2
4319 026736 062700 000002 ADD #2,R0
4320 026742 005504 ADC R4
4321 026744 010037 026776 MOV R0,LONUM
4322 026750 010021 MOV R0,(R1)+
4323 026752 005337 001174 DEC WORK
4324 026756 001406 BEQ EXGEN
4325 026760 010437 027000 MOV R4,HINUM
4326 026764 010421 MOV R4,(1)+
4327 026766 005337 001174 DEC WORK
4328 026772 001326 BNE RAND1
4329 026774 000205 EXGEN: RTS
4330 026776 000000 LONUM: 0
4331 027000 000000 HINUM: 0
4332 027002 000000 LOSAV: 0
4333 027004 000000 HISAV: 0
4334 027006 RANEND:
4335
4336 027006 032737 000001 027124 .SUSWR: BIT #BIT0,SWI
4337 027014 001037 BNE XXX
4338 027016 013746 000006 MOV 6,-(SP)
4339 027022 013746 000004 MOV 4,-(SP)
4340 027026 012737 027046 000004 MOV #1$,4
4341 027034 022777 177777 151764 CMP #-1,$SWR
4342 027042 001402 BEQ 2$
4343 027044 000407 BR 3$
4344 027046 022626 1$: CMP (SP)+,(SP)+
4345 027050 012737 000176 001026 2$: MOV #SWREG,SWR
4346 027056 012737 000174 001030 3$: MOV #DISPREG,DISPLAY
4347 027064 022737 000176 001026 3$: CMP #SWREG,SWR
4348 027072 001004 BNE 4$
4349 027074 005737 000042 TST 4$
4350 027100 001001 BNE 4$
4351 027102 !04422 CNTLU
;SET UP R0 WITH 5 DIGITS LOW
;SET UP R1 WITH 5 DIGITS HIGH
;SET UP SHIFT COUNT
;CLEAR R2
;SHIFT R0 LEFT AND
;ROTATE CARRY INTO LSB OF R1 INTO
;ROTATE CARRY OUT OF R1 INTO R2
;DECREMENT R3
;CONTINUE SHIFT LOOP
;ADDN IN NUMBER TO MAKE X 129
;PROPOGATE CARRY
;ADDN IN NUMBER TO MAKE X 129
;PROPOGATE CARRY
;ADDN LOW CONSTANT
;PROPOGATE CARRIES
;PROPOGATE AGAIN
;ADDN HIGH CONSTANT
;PROPOGATE CARRY
;ADDN HIGHEST CONSTANT
;REPRIME R0 WITH HIGH DIGIT
;PROPOGATE CARRY
;PUT R0 BACK IN LONUM
;LOAD WC
;PUT R1 BACK IN HINUM
;HOLD HINUM FOR PROGRAM
;RETURN TO PROGRAM
;SAVE 6 ON STACK
;SAVE 4 ON STACK
;SET UP TRAP ADDRESS
;TEST 177570
;FAKE OUT
;HARDWARE AVAILABLE
;ADJUST STACK
;SET UP SOFTWARE REGISTERS
;1ST TIME THRU?
;NO CHANGE STILL 177570
;ANY XXDP OR ACT
;SWR=000000
;GET INITIAL SETTINGS
    
```

```

4352 027104 012637 000004      4S:  MOV      (SP)+,4      ;REPLACE 4 FROM STACK
4353 027110 012637 000006      MOV      (SP)+,6      ;REPLACE 6 FROM STACK
4354 027114 052737 000001 027124  XXX:  BIS      #BIT0,SWI    ;SET THE BEENHEREBIT
4355 027122 000002      RTI                      ;ALL DONE
4356
4357 027124 000000      SWI:   0
4358
4359
4360
4361 027126 005737 000042      .KBDIN: TST      42      ;GOT XXDP OR ACT
4362 027132 001057      BNE OKT                ;YES,GET OUT
4363 027134 022737 000176 001026  CMP      #SWREG,SWR    ;GOT SWITCH-LESS MACHINE?
4364 027142 001053      BNE OKT                ;NO GET OUT
4365 027144 105777 151650      TSTB     @TKS          ;HAVE A CHARACTER
4366 027150 100050      BPL OKT                ;NO GET OUT
4367 027152 017737 151644 027274  MOV      @TKB,.MSG     ;
4368 027160 042737 177600 027274  BIC      #177600,.MSG   ;STRIP OFF GARBAGE
4369 027166 122737 000007 027274  CMPB     #7,.MSG       ;DO WE HAVE A 1G
4370 027174 001036      BNE OKT                ;NO,GET OUT
4371 027176 104402 027202      TYPE     ,.+2          ;.ASCIZ <15><12>"1G"
4372 027210
4373 027210 104402 027214      .CNTLU: TYPE     ,.+2          ;.ASCIZ <15><12>"SWR="
4374 027224 013746 000176      MOV      SWREG,-(6)    ;PUT SWREG ON STACK
4375 027230 104404      TYPE0    ;TYPE STACK IN OCTAL
4376 027232 104402 027236      TYPE     ,.+2          ;.ASCIZ " NEW="
4377 027250 104410      RDOCT
4378 027252 012637 027274      MOV      (SP)+,.MSG    ;GET NEW VALUE OFF STACK
4379 027256 005737 027276      TST      CTN          ;DID HE TYPE <CR> OF 000000?
4380 027262 001403      BEQ OKT                ;DONT CHANGE IF <CR>
4381 027264 013737 027274 000176  MOV      .MSG,SWREG    ;CHANGE VALUE OF SWREG
4382 027272 000002      OKT:  RTI              ;ALL DONE-EXIT
4383
4384 027274 000000      .MSG:   0
4385 027276 000000      CTN:   0
    
```

E09

MAINDEC-11-DZRSB-E
DZRSBE.P11 TST122

RH11-RS03LA-RSG3-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 109
WRITE LOCK TEST

```
4386  
4387 ;TABLES FOR ILLEGAL FUNCTION TESTS  
4388  
4389 027300 000003 ILLTAB: 3  
4390 027302 000005 5  
4391 027304 000007 7  
4392 027306 000013 13  
4393 027310 000015 15  
4394 027312 000017 17  
4395 027314 000023 23  
4396 027316 000025 25  
4397 027320 000027 27  
4398 027322 000033 33  
4399 027324 000035 35  
4400 027326 000037 37  
4401 027330 000041 41  
4402 027332 000043 43  
4403 027334 000045 45  
4404 027336 000047 47  
4405 027340 000000 0  
4406  
4407 027342 000053 ILFTB2: 53  
4408 027344 000055 55  
4409 027346 000057 57  
4410 027350 000063 63  
4411 027352 000065 65  
4412 027354 000073 73  
4413 027356 000075 75  
4414 027360 000077 77  
4415 027362 000000 0  
4416 027364 000300 OUTBUF: .BLKW 300  
4417 030164 000300 INBUF: .BLKW 300  
4418 000001 .END
```


H09

EXGEN	026774	4324	4329#											
EXREF	025664	4185	4187#											
EXTRP	020412	3300	3376	3382	3387#									
EXTTRP	020420	3295	3389#											
EXTTST	017676	3292#												
EXTT1	020220	3349#												
EXT1	020422	3388	3390#											
FILCHR	001014	680#	3658	3659										
FINDM	025632	1811	4182#											
FINTST	021066	3481	3486#											
FLOTBA	003052	1083#												
FLCTDA	003352	1158#												
FLOTWC	003212	1121#												
FUNDO	016620	3124#												
GOOD	=%000001	670#	1048*	1083*	1085	1087	1090*	1121*	1123	1125	1128*	1158*	1160	1162
		1165*	1258*	1272*	1287*	1314*	1315*	1317	1355*	1356*	1357	1368*	1359	1381*
		1382*	1384	1389*	1391	1394*	1396	1408*	1409*	1411	1414*	1415*	1416	1419
		1441*	1442	1450*	1451	1452*	1455*	1457	1458*	1464*	1466	1469*	1472*	1474
		1478*	1627*	1628*	1629	1687*	1688*	1690	1694*	1695*	1696	1728*	1729*	1731
		1735*	1736*	1737	1741*	1742	1763*	1764*	1766	1779*	1780*	1781	1824*	1825*
		1827	1831*	1832*	1833	1860*	1861*	1863	1867*	1868*	16.9	1874*	1875	1895*
		1896*	1898	1911*	1912*	1913	1954*	1955*	1957	1961*	1962*	1963	1982*	1984
		2006*	2007*	2009	2017*	2018*	2019	2047*	2048*	2059*	2061*	2066	2070	2106*
		2109*	2111	2121*	2122*	2123	2126*	2127*	2129	2174*	2177*	2179	2194*	2195*
		2196	2207*	2208*	2209	2243*	2244	2247*	2248*	2250	2267*	2269	2272*	2273*
		2275	2327*	2328*	2333	2348*	2351*	2374*	2375*	2376	2413*	2414*	2415	2467*
		2468*	2469	2473*	2476*	2483*	2484*	2485	2489*	2490*	2492	2544*	2545*	2546
		2550*	2553*	2560*	2561*	2562	2566*	2567*	2569	2605*	2606*	2608	2618*	2620
		2803*	2804*	2805	2808*	2810	2813*	2816*	2820*	2822*	2824	2896*	2899*	2903
		2909*	2910	3007*	3008*	3009	3036*	3037*	3038	3068*	3069*	3070	3078*	3079*
		3080	3091*	3092*	3093	3138*	3139*	3140	3173*	3174*	3176	3280*	3281*	3283
		3345*	3383*	3436*	3450*	3544*	3545	3552*	3553*	3555	3559*	3560*	3561	4220
HALT	025150	4120#												
HINUM	027000	4299	4301	4311	4325*	4331#								
HISAV	027004	4299*	4333#											
HLT	= 104000	642#	911	915	919	924	930	936	942	948	954	973	977	982
		988	994	999	1004	1017	1021	1025	1033	1043	1049	1053	1057	1062
		1070	1074	1078	1089	1100	1108	1112	1116	1127	1137	1145	1149	1153
		1164	1174	1182	1186	1193	1199	1206	1216	1223	1231	1239	1259	1273
		1288	1303	1307	1319	1327	1331	1339	1343	1359	1362	1366	1371	1388
		1393	1398	1413	1421	1424	1436	1444	1468	1476	1492	1495	1498	1501
		1504	1507	1517	1524	1527	1530	1533	1536	1539	1542	1555	1558	1561
		1564	1567	1570	1573	1583	1590	1593	1596	1599	1602	1605	1608	1611
		1631	1634	1637	1642	1645	1648	1651	1665	1668	1672	1674	1677	1680
		1683	1686	1692	1698	1701	1716	1718	1721	1724	1727	1733	1739	1744
		1760	1762	1768	1771	1774	1777	1783	1804	1806	1814	1817	1820	1823
		1829	1835	1848	1850	1853	1856	1859	1865	1871	1877	1892	1894	1900
		1903	1906	1909	1915	1941	1944	1947	1950	1953	1959	1965	1981	1986
		1989	1993	2005	2011	2014	2021	2037	2042	2046	2054	2057	2069	2073
		2076	2079	2082	2084	2113	2116	2119	2125	2131	2134	2137	2140	2144
		2147	2181	2184	2137	2190	2198	2201	2204	2211	2216	2219	2223	2226
		2246	2252	2255	2259	2271	2277	2280	2284	2287	2290	2302	2306	2309
		2321	2338	2355	2378	2381	2417	2420	2471	2477	2481	2487	2494	2497
		2548	2554	2558	2564	2571	2574	2598	2601	2604	2610	2613	2616	2622
		2625	2642	2645	2648	2651	2654	2658	2661	2676	2679	2682	2685	2701
		2704	2707	2710	2732	2735	2738	2741	2745	2787	2795	2798	2801	2807

		2812	2826	2830	2853	2856	2875	2878	2881	2885	2907	2912	2915	2918
		2922	2926	2950	2953	2956	2959	2963	2966	2985	2988	2991	2994	2998
		3011	3014	3018	3022	3029	3034	3040	3043	3046	3049	3052	3055	3058
		3072	3076	3082	3085	3089	3095	3112	3115	3118	3136	3142	3145	3148
		3152	3171	3178	3181	3184	3188	3191	3210	3213	3216	3219	3223	3285
		3321	3325	3337	3341	3347	3360	3363	3375	3379	3395	3407	3413	3432
		3437	3457	3458	3462	3480	3484	3547	3548	3551	3557	3563	3568	3574
		3578	3597	3601	3604	4056	4087	4091	4096	4100	4105	4109	4114	4118
		4123	4127	4132	4136	4141	4145	4157	4160	4164	4174	4179		
HLADR	001012	679#	3737*	3738*	3739	3740	3754							
IAERR	015130	2865#												
ICNT	001000	675#	805*	3618*	3688	3703	3705	3707*	3708*	3710	3713*	3714	3720	3749*
IE	= 000100	768#												
ILFDN	010762	2101	2149#											
ILFDNE	011332	2164	2228#											
ILFTB2	027342	2161	4407#											
ILF67	011334	2234#												
ILLFUN	010764	2160#												
ILLTAB	027300	2099	4389#											
ILLS1	010506	2096#												
INBUF	030164	4417#												
INCH	014576	2785#												
INPUT	023402	3855	3890	3891	3914#									
INTDON	020642	3433	3438#											
INTR4	020644	3444#												
INTRS	020540	3421#												
INT112	020634	3424	3436#											
INT114	020746	3446	3460#											
IR	= 000100	754#												
KBDIN	= 104416	3685	3729	3935#	3943	3950	3957	3964	3971	3981	3988	3995	4002	4057
KIPARO	= 172340	716#	3302*											
KIPAR1	= 172342	717#	3303*											
KIPAR2	= 172344	718#	3304*	3350*										
KIPAR7	= 172356	719#	3301*											
KIPDRO	= 172300	720#	3305*											
KIPDR1	= 172302	721#	3306*											
KIPDR2	= 172304	722#	3307*											
KIPDR7	= 172316	723#	3308*											
LA	= 000204	745#	2321	2338	4279									
LAD	001010	678#	806*	1249*	1263*	1277*	3709*	3715	3717	3720				
LASTSC	017166	3197#												
LATDON	012046	2335	2339#											
LATST	011724	2319#												
LBT	= 002000	762#												
LONUM	026776	4298	4300	4309	4321*	4330#								
LOP1	003772	1249	1250#											
LOP2	004042	1263	1264#											
LOP3	004112	1277	1278#											
LOSAY	027002	4298#	4332#											
MAXREF	025716	4182	4193#											
MAXRF1	025720	4191	4194#											
MEMOUT	020426	3391#												
MODDON	021732	3615#												
MODNUM	021070	3496#												
MPRO	= 172100	780#	2392											
MR	= 000220	747#	942	999	1216	1223	1231	1239	1642	2037	2355	2381	4274	

MULTI = 001316
N = 000123

819#													
641#	814	818#	892	896#	955	959#	1005	1009#	1026	1027#	1035	1039#	
1064	1065#	1080	1081#	1092	1093#	1101	1102#	1117	1118#	1131	1132#	1138	
1139#	1154	1155#	1168	1169#	1175	1176#	1188	1189#	1194	1195#	1200	1201#	
1207	1208#	1217	1218#	1224	1225#	1232	1233#	1240	1244#	1293	1297#	1309	
1310#	1321	1322#	1333	1334#	1345	1349#	1372	1376#	1399	1400#	1445	1446#	
1480	1484#	1508	1512#	1543	1547#	1574	1578#	1612	1616#	1653	1657#	1704	
1708#	1745	1749#	1791	1795#	1836	1840#	1878	1882#	1920	1924#	1970	1974#	
1994	1998#	2023	2027#	2086	2090#	2150	2154#	2229	2233#	2292	2296#	2314	
2318#	2322	2323#	2340	2344#	2384	2388#	2433	2437#	2506	2510#	2584	2588#	
2626	2630#	2663	2667#	2688	2692#	2712	2716#	2746	2750#	2842	2846#	2858	
2862#	2889	2893#	2932	2936#	2967	2971#	2999	3003#	3059	3063#	3096	3100#	
3120	3124#	3153	3157#	3192	3196#	3228	3232#	3287	3291#	3393	3397#	3416	
3420#	3439	3443#	3465	3469#	3487	3491#	4061	4065#					
757#	2920	2924											
2901	2928#												
2894#													
2927	2929	2931#											
1513#													
1579#													
1485#													
1548#													
2421	2427	2431#											
865	882	888#											
1916#	4206												
1999#													
690#	808#	888#	1487	1502	1519	1534	1550	1565	1585	1600	1660	1666	
1694	1711	1735	1755	1779	1797	1799	1831	1843	1867	1872	1885	1887	
1911	1916	1917#	1931	1961	1976	1982	2030	2077	2102	2132	2165	2199	
2212	2236	2243	2261	2267	2444	2460	2483	2518#	2521	2537	2560	2582*	
2591	2632	2669	2694	2718	2764	2771	2788	2832	2849	2867	2947	2982	
3024	3050	3104	3113	3127	3161	3207	3234	3249	3261	3267	3429	3452	
3477	3526	3559	4005	4186#	4187	4190	4196#						
4362	4364	4366	4370	4380	4382#								
617#	635#	775#	807#	809	832*	857	864#	871	889#	2449	2474	2498	
2501#	2505#	2515	2526	2551	2575	2578#	2583#	2928	2931#	4199	4203*	4284*	
2043	2083	2085#											
755#	1422												
3619#													
808	888	1246	1248	1262	1276	1486#	1518#	1549#	1584#	1659#	1710#	1740	
1754#	1917	1930*	2404*	2443*	2451*	2453*	2456*	2458*	2472	2503*	2530*	2582	
2590*	2763*	2770*	2813	3023*	3103*	3126*	3160*	3233*	3248*	3525*	4033*	4045	
4050	4075*	4076	4146*	4147	4165*	4166	4172	4196	4416#				
2389#													
2297#													
3064#													
3158#													
653#	1673*	1717*	1761*	1805*	1849*	1893*	1940*	1980*	2004*	2597*	2641*	2675*	
2700*	2852*	2949*	2984*	3209*	3431*	3479*	3743*	3926*	4285*	4295*			
677#	3623*	3624*	3635										
756#	2618												
2589#													
3400	3410#												
761#													
2390	2425#												
2401	2426#												

NED = 010000
 NEDDON 015422
 NEDTST 015244
 NND0 015504
 NNOOP 005244
 NNOP21 005562
 NOOP 005120
 NOOP21 005424
 NOPAR 012446
 NOWGO 002022
 NXM 007526
 NXMTSM 010060
 OBUFSV 001102

OKT 027272
 ONCEE 001160
 OOUT 010504
 OR = 000200
 OUT 021756
 OUTBUF 027364

PART 012240
 PARTST 011646
 PATST 016332
 PATTST 016772
 PC =%000007

PCNT 001004
 PGE = 002000
 PGETST 013416
 PGTRAP 020522
 PIP = 020000
 PRTP 012426
 PRTP1 012430

E10

MAINDEC-11-DZRSB-E RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 123
DZRSBE.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

.TYPEB	022556	3760#	
.TYPE0	022566	3762#	3930
.TYPES	022576	3764#	3931

G10

MAINDEC-11-DZRSB-E RH11-RS03LA-RS03-RS04 BASIC FUNCTION DIAGNOSTIC MACY11 27(732) 25-SEP-76 09:06 PAGE 126
DZRSBE.P11 CROSS REFERENCE TABLE -- MACRO NAMES

STYPED 18
SUMMR 18

:

	1543	1547	1574	1578	1612	1616	1653	1657	1704	1708	1745	1749	1791	1795	1836
	1840	1878	1882	1920	1924	1970	1974	1994	1998	2023	2027	2086	2090	2150	2154
	2229	2233	2292	2296	2314	2318	2322	2323	2338	2340	2344	2384	2388	2433	2437
	2506	2510	2584	2588	2626	2630	2663	2667	2688	2692	2712	2716	2746	2750	2842
	2846	2858	2862	2889	2893	2931	2932	2936	2967	2971	2999	3003	3059	3063	3096
	3100	3120	3124	3153	3157	3192	3196	3228	3232	3287	3291	3393	3397	3416	3420
	3439	3443	3465	3469	3487	3491	3628	3737	3743	3835	3881	3900	3905	3915	3928
	3929	3930	3931	3932	3933	3934	3935	3936	3937	3938	4025	4028	4032	4061	4065
	4068	4083	4093	4102	4111	4120	4129	4138	4176	4202	4217	4220	4225	4229	4233
	4240	4245	4250	4255	4260	4267	4272	4277	4282	4372	4374	4377			
.MACR	727														
.MACRO	1	3915													
.MCALL	584	670	814	892	955	1005	1026	1035	1064	1080	1092	1101	1117	1131	1138
	1154	1168	1175	1188	1194	1200	1207	1217	1224	1232	1240	1293	1309	1321	1333
	1345	1372	1399	1445	1480	1508	1543	1574	1612	1653	1704	1745	1791	1836	1878
	1920	1970	1994	2023	2086	2150	2229	2292	2314	2322	2340	2384	2433	2506	2584
	2626	2663	2688	2712	2746	2842	2858	2889	2932	2967	2999	3059	3096	3120	3153
.NLIST	3192	3228	3287	3393	3416	3439	3465	3487	4061						
1	584	601	670	814	818	832	860	881	892	896	955	959	1005	1009	
	1026	1027	1035	1039	1064	1065	1080	1081	1092	1093	1101	1102	1117	1118	1131
	1132	1138	1139	1154	1155	1168	1169	1175	1176	1188	1189	1194	1195	1200	1201
	1207	1208	1217	1218	1224	1225	1232	1233	1240	1244	1293	1297	1309	1310	1321
	1322	1333	1334	1345	1349	1372	1376	1399	1400	1445	1446	1480	1484	1508	1512
	1543	1547	1574	1578	1612	1616	1653	1657	1704	1708	1745	1749	1791	1795	1836
	1840	1878	1882	1920	1924	1970	1974	1994	1998	2023	2027	2086	2090	2150	2154
	2229	2233	2292	2296	2314	2318	2322	2323	2338	2340	2344	2384	2388	2433	2437
	2506	2510	2584	2588	2626	2630	2663	2667	2688	2692	2712	2716	2746	2750	2842
	2846	2858	2862	2889	2893	2931	2932	2936	2967	2971	2999	3003	3059	3063	3096
	3100	3120	3124	3153	3157	3192	3196	3228	3232	3287	3291	3393	3397	3416	3420
	3439	3443	3465	3469	3487	3491	3628	3737	3743	3835	3881	3900	3905	3915	3928
	3929	3930	3931	3932	3933	3934	3935	3936	3937	3938	4025	4028	4032	4061	4065
	4068	4083	4093	4102	4111	4120	4129	4138	4176	4202	4217	4220	4225	4229	4233
	4240	4245	4250	4255	4260	4267	4272	4277	4282	4372	4374	4377			
.PAGE	640	673	3678	3721	3754	3802	3842	3882							
.REM	1														
.REPT	601														
.SBTTL	814	892	955	1005	1035	1240	1293	1345	1372	1480	1508	1543	1574	1612	1653
	1704	1745	1791	1836	1878	1920	1970	1994	2023	2086	2150	2229	2292	2314	2340
	2384	2433	2506	2584	2626	2663	2688	2712	2746	2842	2858	2889	2932	2967	2999
	3059	3096	3120	3153	3192	3228	3287	3393	3416	3439	3465	3487	3620	3638	3678
.TITLE	584	3754	3802	3842	3882	3915	4061								

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DZRSBE.SEG/SOL/CRF/PAGNUM/NL:TOC=DZRSBE.SML,DZRSBE.P11
RUN-TIME: 28 45 8 SECONDS
RUN-TIME RATIO: 204/82=2.4
CORE USED: 23K (45 PAGES)

